```
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
```

```
FFFFFFFFFF    AAAAAA    LL          SSSSSSSS  TTTTTTTTTT    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
FFFFFFFFFF    AAAAAA    LL          SSSSSSSS  TTTTTTTTTT    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
FF          AA      AA  LL        SS              TT      AA      AA      TT      EE
FF          AA      AA  LL        SS              TT      AA      AA      TT      EE
FF          AA      AA  LL        SS              TT      AA      AA      TT      EE
FF          AA      AA  LL        SS              TT      AA      AA      TT      EE
FFFFFFFF    AA      AA  LL          SSSSSS        TT      AA      AA      TT      EEEEEEEE
FFFFFFFF    AA      AA  LL          SSSSSS        TT      AA      AA      TT      EEEEEEEE
FF          AAAAAAAAAA  LL                SS      TT      AAAAAAAAAA      TT      EE
FF          AAAAAAAAAA  LL                SS      TT      AAAAAAAAAA      TT      EE
FF          AA      AA  LL                SS      TT      AA      AA      TT      EE              ....
FF          AA      AA  LL                SS      TT      AA      AA      TT      EE              ....
FF          AA      AA  LLLLLLLLLL  SSSSSSSS      TT      AA      AA      TT      EEEEEEEEEE      ....
FF          AA      AA  LLLLLLLLLL  SSSSSSSS      TT      AA      AA      TT      EEEEEEEEEE      ....

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
```

Table of contents

```
0000          .TITLE  FALSTATE - STATE TRANSITION PROCESSING
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;*****************************************************************************
0000    ;*                                                                           *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    ;*  ALL RIGHTS RESERVED.                                                     *
0000    ;*                                                                           *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    ;*  TRANSFERRED.                                                             *
0000    ;*                                                                           *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    ;*  CORPORATION.                                                             *
0000    ;*                                                                           *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    ;*                                                                           *
0000    ;*                                                                           *
0000    ;*****************************************************************************
0000    ;
0000
0000    ;++
0000    ; Facility: FAL (DECnet File Access Listener)
0000    ;
0000    ; Abstract:
0000    ;
0000    ;       This module contains the state table manager and the state transition
0000    ;       table for FAL.
0000    ;
0000    ; Environment: VAX/VMS, user mode
0000    ;
0000    ; Author: James A. Krycka,      Creation Date:  17-JUN-1977
0000    ;
0000    ; Modified By:
0000    ;
0000    ;       V03-005 JAK0135         J A Krycka      05-MAR-1984
0000    ;               Correct occurrence of duplicate label in state table ST13 for
0000    ;               wildcard rename.
0000    ;
0000    ;       V03-004 JAK0128         J A Krycka      16-SEP-1983
0000    ;               CRC check data that is being discarded during file transfer
0000    ;               error recovery.
0000    ;
0000    ;       V03-003 JAK0120         J A Krycka      29-JUL-1983
0000    ;               Provide better error recovery in the file transfer mode storage
0000    ;               state (ST7) and minor cleanup in other file transfer mode
0000    ;               states (ST4, ST5, and ST6).
0000    ;
0000    ;       V03-002 JAK0104         J A Krycka      19-APR-1982
0000    ;               Changes to support optimized file transfer mode storage and
```

```
0000    58 ;                    retrieval.
0000    59 ;
0000    60 ;        V03-001 KRM0073         K Malik         23-Nov-1982
0000    61 ;                    Support the DAP RENAME operation.
0000    62 ;
0000    63 ;--
```

```
0000          .SBTTL  DECLARATIONS

;
; Include Files:
;

        $DAPHDRDEF                      ; Define DAP message header
        $DAPACCDEF                      ; Define DAP Access message
        $DAPCTLDEF                      ; Define DAP Control message
        $DAPCONDEF                      ; Define DAP Continue Transfer message
        $DAPCMPDEF                      ; Define DAP Access Complete message
        $FALWRKDEF                      ; Define FAL Work Area symbols

;
; Macros:
;
;++
; STATE_INIT initializes the state transition table, where:
;
;       LABEL   = the global symbol to denote the start of the table.
;--

        .MACRO  STATE_INIT      LABEL
        .ALIGN  LONG
LABEL::
COUNT...=0
        .ENDM   STATE_INIT

;++
; STATE defines a transfer of control point in the state transition table
; (i.e., the state of a new state), where:
;
;       NAME    = the name of the state.
;--

        .MACRO  STATE   NAME
NAME=COUNT...
        .ENDM   STATE

;++
; TRAN defines a state transition table entry, where:
;
;       VALUE   = the value to compare against the state transition value.
;                 (Null denotes that state value is a match.)
;       ACTION  = address of optional action routine to execute.
;                 (Null denotes no action routine to execute.)
;       SUCCESS = name of the state to goto on action rouitne success.
;                 (If no action routine is specified, then this path is taken.)
;                 (Null denotes goto next transition entry.)
;       FAILURE = name of the state to goto on action routine failure.
;                 (Null denotes goto next transition entry.)
;--

        .MACRO  TRAN    VALUE,ACTION,SUCCESS,FAILURE
        .IF B   VALUE
        .WORD   M_MATCH
        .IFF
```

```
                0000          .WORD   VALUE@8
                0000          .ENDC
                0000          .IF B   SUCCESS
                0000          .BYTE   COUNT...+1
                0000          .IFF
                0000          .BYTE   SUCCESS
                0000          .ENDC
                0000          .IF B   FAILURE
                0000          .BYTE   COUNT...+1
                0000          .IFF
                0000          .BYTE   FAILURE
                0000          .ENDC
                0000          .IF B   ACTION
                0000          .ADDRESS 0
                0000          .IFF
                0000          .ADDRESS ACTION
                0000          .ENDC
                0000          COUNT...=COUNT...+1
                0000          .ENDM   TRAN
                0000
                0000  ;++
                0000  ; RETURN defines a state transition table entry that directs the transition
                0000  ; table manager (FAL$STATE) to exit to its caller.
                0000  ;--
                0000
                0000          .MACRO  RETURN
                0000          .QUAD   M_EXIT
                0000          .ENDM   RETURN
                0000
                0000  ;
                0000  ; Equated Symbols:
                0000  ;
                0000  ;       State table entry offsets:
                0000  ;
                0000
00000000        0000  V_MATCH=0
00000001        0000  M_MATCH=<1@V_MATCH>
00000001        0000  V_EXIT=1
00000002        0000  M_EXIT=<1@V_EXIT>
00000000        0000  FLAGS=0
00000001        0000  VALUE=1
00000002        0000  SUCCESS=2
00000003        0000  FAILURE=3
00000004        0000  ACTION=4
                0000
                0000          ASSUME  FAL$Q_FLG EQ 0
                0000
                0000  ;
                0000  ; Own Storage:
                0000  ;
                0000  ;       None
```

```
0000          .SBTTL  FAL$STATE - STATE TABLE MANAGER
00000000      .PSECT  FAL$CODE        NOSHR,EXE,RD,NOWRT,BYTE

0000  ;++
0000  ; Functional Description:
0000  ;
0000  ;       FAL$STATE controls the execution of FAL by maintaining its state context
0000  ;       and executing action routines as directed by the state transition table.
0000  ;
0000  ; Calling Sequence:
0000  ;
0000  ;       BSBW    FAL$STATE
0000  ;
0000  ; Input Parameters:
0000  ;
0000  ;       R2      Address of state transition table
0000  ;       R8      Address of FAL work area
0000  ;       R9      Address of DAP control block
0000  ;       R10     Address of FAB
0000  ;       R11     Address of RAB
0000  ;
0000  ; Implicit Inputs:
0000  ;
0000  ;       FAL$Q_STATE_CTX
0000  ;       FAL$B_VALUE
0000  ;
0000  ; Output Parameters:
0000  ;
0000  ;       R0-R7   Destroyed
0000  ;
0000  ; Implicit Outputs:
0000  ;
0000  ;       FAL$Q_STATE_CTX
0000  ;       FAL$B_VALUE
0000  ;
0000  ; Completion Codes:
0000  ;
0000  ;       Each action routine is required to return a success/failure (or a
0000  ;       true/false) code in R0 which is used by FAL$STATE to determine the
0000  ;       next state to enter.
0000  ;
0000  ; Side Effects:
0000  ;
0000  ;       None
0000  ;
0000  ;--
```

```
                        0000   FAL$STATE::                                  ; Entry point
          53    52  D0  0000          MOVL    R2,R3                        ; Initialize state transition
                        0003                                               ;  entry address
       37 63    01  E0  0003   10$:    BBS     #V_EXIT,FLAGS(R3),50$        ; Branch if end of table
       0C 63    00  E0  0007          BBS     #V_MATCH,FLAGS(R3),20$       ; Branch if any value is a match
    10 A8    01  A3  91  000B          CMPB    VALUE(R3),FAL$B_VALUE(R8)    ; Is state value a match?
                 05  13  0010          BEQL    20$                          ; Branch if yes
          53    08  C0  0012          ADDL2   #8,R3                        ; No, form address of next table entry
                 EC  11  0015          BRB     10$                          ; Try again
       54    04  A3  D0  0017   20$:   MOVL    ACTION(R3),R4                ; Get address of action routine
                 15  13  001B          BEQL    30$                          ; Branch if no routine specified
       08 A8    52  7D  001D          MOVQ    R2,FAL$Q_STATE_CTX(R8)       ; Save state table context <R2,R3>
                 64  16  0021          JSB     (R4)                         ; Execute action routine
       52    08  A8  7D  0023          MOVQ    FAL$Q_STATE_CTX(R8),R2       ; Restore state table context <R2,R3>
             08  50  E8  0027          BLBS    R0,30$                       ; Branch on success
54   03 A3    08  00  EF  002A          EXTZV   #0,#8,FAILURE(R3),R4         ; Get failure table entry index
                 06  11  0030          BRB     40$                          ;  (as an unsigned integer value)
54   02 A3    08  00  EF  0032   30$:   EXTZV   #0,#8,SUCCESS(R3),R4         ; Get success table entry index
                         0038                                               ;  (as an unsigned integer value)
       53   6244  7E  0038   40$:   MOVAQ   (R2)[R4],R3                     ; Compute address of new table entry
                         003C                                               ;  taking advantage of the fact that
                         003C                                               ;  each entry is 8 bytes long
                 C5  11  003C          BRB     10$                          ; Start again
                     05  003E   50$:   RSB                                  ; Exit
```

```
     003F    .SBTTL  FAL$STATE_TABLE - FAL STATE TRANSITION TABLE
     003F    .SBTTL  STO -- INITIALIZATION
 00000000    .PSECT  FAL$STATE_TABLE SHR,NOEXE,RD,NOWRT,QUAD
     0000
     0000 ;++
     0000 ; Define beginning (entry point) of state transition table.
     0000 ;--
     0000
     0000 STATE_INIT      FAL$STATE_TABLE
     0000
     0000 ;++
     0000 ; Initialization state (also called setup state in DAP specification).
     0000 ;
     0000 ; Note: Success from FAL$INIT means that a Configuration message has been
     0000 ;       received; failure denotes that one has not been received.
     0000 ;--
     0000 ;       State   Value           Action-Routine  Success Failure Comments
     0000
     0000 STATE   STO
     0000         TRAN    ,               FAL$INIT,       ST1,            ; See note above
     0008         TRAN    ,               FAL$NEXT_MSG,   ,       STO
     0010         TRAN    DAP$K_CNF_MSG,  FAL$DECODE_CNF, ST1,    STO
     0018         TRAN    ,               FAL$OUT_OF_SEQ, STO,    STO
```

```
0020          .SBTTL  ST1 -- FILE ACCESS

;++
; Generalized file access state.
;
; Note: The check for a Continue Transfer message and response of an Access
;       Complete message in state ST1 is to help solve an error recovery problem
;       for wildcard file deletion and wildcard file execution.
;--
;       State   Value           Action-Routine  Success Failure Comments

STATE   ST1
        TRAN    ,               FAL$NEXT_MSG,   ,       ST1
0028    TRAN    DAP$K_CNF_MSG,  FAL$DECODE_CNF, ST0,    ST1
0030    TRAN    DAP$K_ATT_MSG,  FAL$DECODE_ATT, ST1,    ST1
0038    TRAN    DAP$K_ACC_MSG,  FAL$DECODE_ACC, ST1_1,  ST1
0040    TRAN    DAP$K_TIM_MSG,  FAL$DECODE_TIM, ST1,    ST1
0048    TRAN    DAP$K_PRO_MSG,  FAL$DECODE_PRO, ST1,    ST1
0050    TRAN    DAP$K_ALL_MSG,  FAL$DECODE_ALL, ST1,    ST1
0058    TRAN    DAP$K_KEY_MSG,  FAL$DECODE_KEY, ST1,    ST1
0060    TRAN    DAP$K_CON_MSG,  FAL$DIR_END,    ST0,    ST0     ; See note above
0068    TRAN    ,               FAL$OUT_OF_SEQ, ST1,    ST1

STATE   ST1_1
        TRAN    DAP$K_OPEN,     FAL$FIL_PARSE,  ST1_1A, ST0
0078    TRAN    DAP$K_CREATE,   FAL$CREATE,     ST2,    ST0
0080    TRAN    DAP$K_DIR_LIST, FAL$DIR_PARSE,  ST8,    ST0
0088    TRAN    DAP$K_ERASE,    FAL$FIL_PARSE,  ST1_1B, ST0
0090    TRAN    DAP$K_SUBMIT,   FAL$SUBMIT,     ST2,    ST0
0098    TRAN    DAP$K_EXECUTE,  FAL$FIL_PARSE,  ST1_1C, ST0
00A0    TRAN    DAP$K_LOAD,     FAL$OPEN,       ST1_1D, ST0
00A8    TRAN    DAP$K_RENAME,   FAL$FIL_PARSE,  ST1_1E, ST0
00B0    TRAN    ,               FAL$INV_ACCFUNC,ST1,    ST1

STATE   ST1_1A
        TRAN    ,               FAL$CHECK_WILD, ST9,            ; True/False
00C0    TRAN    ,               FAL$OPEN,       ST2,    ST0

STATE   ST1_1B
        TRAN    ,               FAL$CHECK_WILD, ST10,           ; True/False
00D0    TRAN    ,               FAL$ERASE,      ST0,    ST0

STATE   ST1_1C
        TRAN    ,               FAL$CHECK_WILD, ST11,           ; True/False
00E0    TRAN    ,               FAL$EXECUTE,    ST0,    ST0

STATE   ST1_1D
        TRAN    ,               FAL$LOAD_IMAGE, EXIT,   EXIT    ; VMS specific

STATE   ST1_1E
        TRAN    ,               FAL$INIT_RENAME,        ST0
00F8    TRAN    ,               FAL$CHECK_WILD, ST13,   ST12    ; True/False
```

```
0100          .SBTTL  ST2 -- RECORD ACCESS

0100  ;++
0100  ; Generalized record access state (including record stream establishment and
0100  ; termination.)
0100  ;--
0100  ;       State   Value           Action-Routine  Success Failure Comments

0100  STATE   ST2
0100          TRAN    ,               FAL$NEXT_MSG,   ,       ST2
0108          TRAN    DAP$K_CTL_MSG,  FAL$DECODE_CTL, ST2_1,  ST2
0110          TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST2
0118          TRAN    ,               FAL$OUT_OF_SEQ, ST2,    ST2

0120  STATE   ST2_1
0120          TRAN    DAP$K_CONNECT,  FAL$CONNECT,    ST2,    ST2     ; One stream max
0128          TRAN    DAP$K_GET_READ, FAL$CHECK_FTM,  ST6,    ST4     ; True/False
0130          TRAN    DAP$K_PUT_WRITE,FAL$CHECK_FTM,  ST7,    ST5     ; True/False
0138          TRAN    DAP$K_UPDATE,   ,               ST2A,   EXIT
0140          TRAN    DAP$K_DELETE,   FAL$DELETE,     ST2,    ST2
0148          TRAN    DAP$K_FIND,     FAL$FIND,       ST2,    ST2
0150          TRAN    DAP$K_REWIND,   FAL$REWIND,     ST2,    ST2
0158          TRAN    DAP$K_DISPLAY,  FAL$DISPLAY,    ST2,    ST2
0160          TRAN    DAP$K_EXTEND_B, FAL$INIT_XABCHN,ST2B,   ST2
0168          TRAN    DAP$K_FREE,     FAL$FREE,       ST2,    ST2
0170          TRAN    DAP$K_RELEASE,  FAL$RELEASE,    ST2,    ST2
0178          TRAN    DAP$K_TRUNCATE, FAL$TRUNCATE,   ST2,    ST2
0180          TRAN    DAP$K_FLUSH,    FAL$FLUSH,      ST2,    ST2
0188          TRAN    DAP$K_SPACE_FW, FAL$SPACE_FW,   ST2,    ST2
0190          TRAN    DAP$K_SPACE_BW, FAL$SPACE_BW,   ST2,    ST2
0198          TRAN    ,               FAL$INV_CTLFUNC,ST2,    ST2

01A0  STATE   ST2_2                                                   ; Differentiate
01A0          TRAN    DAP$K_DISCONN,  FAL$DISCONNECT, ST2,    ST2     ;  disconnect
01A8          TRAN    ,               ,               ST3_1,  EXIT    ;  request from
01B0                                                                  ;  close request

01B0  STATE   ST2A
01B0          TRAN    ,               FAL$NEXT_MSG,   ,       ST2
01B8          TRAN    DAP$K_DAT_MSG,  FAL$UPDATE      ST2,    ST2
01C0          TRAN    ,               FAL$OUT_OF_SEQ, ST2,    ST2

01C8  STATE   ST2B
01C8          TRAN    ,               FAL$NEXT_MSG,   ,       ST2
01D0          TRAN    DAP$K_ALL_MSG,  FAL$DECODE_ALL, ST2B,   ST2
01D8          TRAN    DAP$K_CTL_MSG,  FAL$DECODE_CTL, ST2B_1, ST2
01E0          TRAN    ,               FAL$OUT_OF_SEQ, ST2,    ST2

01E8  STATE   ST2B_1
01E8          TRAN    DAP$K_EXTEND_E, FAL$EXTEND,     ST2,    ST2
01F0          TRAN    ,               FAL$INV_CTLFUNC,ST2,    ST2
```

```
01F8          .SBTTL  ST3 -- FILE CLOSE

;++
; Generalized file close state (for non-wildcard operations).
;
; Note: Access Complete (disconnect) is logically a record processing operation,
;       not a file close operation. Consequently, it is handled by state ST2.
;
; Note: State ST3_1 is nomally entered from state ST2_2.
;--
;       State   Value           Action-Routine  Success Failure Comments

STATE   ST3
        TRAN    ,               FAL$NEXT_MSG,   ,       ST3
        TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST3_1,  ST3
        TRAN    ,               FAL$OUT_OF_SEQ, ST3,    ST3

STATE   ST3_1
        TRAN    DAP$K_CLOSE,    FAL$CLOSE,      ST0,    ST3
        TRAN    DAP$K_CHANGE_B, FAL$INIT_XABCHN,ST3A,   ST3
        TRAN    DAP$K_RESET,    FAL$RESET,      ST0,    ST3
        TRAN    DAP$K_TERMINATE,FAL$RESET,      ST0,    ST3     ; Inappropriate
        TRAN    ,               FAL$INV_CMPFUNC,ST3,    ST3

STATE   ST3A
        TRAN    ,               FAL$NEXT_MSG,   ,       ST3
        TRAN    DAP$K_TIM_MSG,  FAL$DECODE_TIM, ST3A,   ST3
        TRAN    DAP$K_PRO_MSG,  FAL$DECODE_PRO, ST3A,   ST3
        TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST3A_1, ST3
        TRAN    ,               FAL$OUT_OF_SEQ, ST3,    ST3

STATE   ST3A_1
        TRAN    DAP$K_CHANGE_E, FAL$CHANGE,     ST0,    ST3
        TRAN    ,               FAL$INV_CMPFUNC,ST3,    ST3
```

```
0270          .SBTTL  ST4 -- RAM RETRIEVAL

0270  ;++
0270  ; Record access mode retrieval state (for $GET/$READ).
0270  ;
0270  ; Send the specified record/block to the partner process. Engage in error
0270  ; recovery dialogue on encountering an end-of-file condition or on an error
0270  ; in accessing the data.
0270  ;--
0270  ;       State   Value           Action-Routine  Success Failure Comments

0270  STATE   ST4
0270          TRAN    ,               FAL$RETRV_RAM,  ST2,    ST4A

0278  STATE   ST4A
0278          TRAN    ,               FAL$NEXT_MSG,   ,       ST4A
0280          TRAN    DAP$K_CTL_MSG,  FAL$DECODE_CTL, ST2_1,  ST4A
0288          TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST4A
0290          TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST4A_1, ST4A
0298          TRAN    ,               FAL$OUT_OF_SEQ, ST4A,   ST4A

02A0  STATE   ST4A_1
02A0          TRAN    DAP$K_ABORT,    ,               ST4A,   EXIT    ; Ok to allow it
02A8          TRAN    DAP$K_RESUME,   ,               ST4A,   EXIT    ; Inappropriate
02B0          TRAN    DAP$K_RETRY,    ,               ST4,    EXIT
02B8          TRAN    DAP$K_SKIP_REC, FAL$UNS_CONFUNC,ST4A,   ST4A
02C0          TRAN    ,               FAL$INV_CONFUNC,ST4A,   ST4A
```

```
02C8          .SBTTL  ST5 -- RAM STORAGE

;++
; Record access mode storage state (for $PUT/$WRITE).
;
; Store each record/block received from the partner process until directed
; to perform another function. Engage in error recovery dialogue on an error
; in storing the data.
;--
;       State   Value           Action-Routine  Success Failure Comments

STATE   ST5
        TRAN    ,               FAL$NEXT_MSG,   ,       ST5
        TRAN    DAP$K_DAT_MSG,  FAL$STORE_RAM,  ST5,    ST5A
        TRAN    DAP$K_CTL_MSG,  FAL$DECODE_CTL, ST2_1,  ST5
        TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST5
        TRAN    ,               FAL$OUT_OF_SEQ, ST5,    ST5

STATE   ST5A
        TRAN    ,               FAL$INTE_MSG,   ,       ST5A
        TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST5A_1, ST5A
        TRAN    ,               FAL$OUT_OF_SEQ, ST5A,   ST5A

STATE   ST5A_1
        TRAN    DAP$K_ABORT,    ,               ST5B,   EXIT
        TRAN    DAP$K_RESUME,   FAL$UNS_CONFUNC,ST5A,   ST5A
        TRAN    DAP$K_RETRY,    FAL$UNS_CONFUNC,ST5A,   ST5A
        TRAN    DAP$K_SKIP_REC, ,               ST5,    EXIT
        TRAN    ,               FAL$INV_CONFUNC,ST5A,   ST5A

STATE   ST5B
        TRAN    ,               FAL$NEXT_MSG,   ,       ST5B
        TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST5B
        TRAN    DAP$K_DAT_MSG,  FAL$DISCARD_DAT,ST5B,   EXIT
        TRAN    ,               FAL$OUT_OF_SEQ, ST5B,   ST5B
```

```
0350          .SBTTL  ST6 -- FTM RETRIEVAL

0350  ;++
0350  ; File transfer mode retrieval state (for $GET/$READ).
0350  ;
0350  ; Send each record/block remaining in the file to the partner process. Engage
0350  ; in error recovery dialogue on encountering an end-of-file condition, on an
0350  ; error in accessing the data, or if directed by partner to stop before the
0350  ; end-of-file is reached.
0350  ;--
0350  ;       State   Value           Action-Routine  Success Failure Comments

0350  STATE   ST6
0350          TRAN    ,               FAL$TEST_MSG,   ST6A,
0358          TRAN    ,               FAL$RETRV_FTM,  ST6,    ST6A
0360
0360  STATE   ST6A
0360          TRAN    ,               FAL$NEXT_MSG,   ,       ST6A
0368          TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST6A
0370          TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST6A_1, ST6A
0378          TRAN    ,               FAL$OUT_OF_SEQ, ST6A,   ST6A
0380
0380  STATE   ST6A_1
0380          TRAN    DAP$K_ABORT,    ,               ST6A,   EXIT    ; Ok to allow it
0388          TRAN    DAP$K_RESUME,   ,               ST6,    EXIT    ; Same as retry
0390          TRAN    DAP$K_RETRY,    ,               ST6,    EXIT
0398          TRAN    DAP$K_SKIP_REC, FAL$UNS_CONFUNC,ST6A,   ST6A
03A0          TRAN    ,               FAL$INV_CONFUNC,ST6A,   ST6A
```

```
03A8          .SBTTL  ST7 -- FTM STORAGE

;++
; File transfer mode storage/append state (for $PUT/$WRITE).
;
; Store each record/block received from partner process until directed to stop.
; Engage in error recovery dialogue on an error in accessing the file.
;--
;       State   Value           Action-Routine  Success Failure Comments

STATE   ST7
        TRAN    ,               FAL$NEXT_MSG,   ,       ST7
        TRAN    DAP$K_DAT_MSG,  FAL$STORE_FTM,  ST7,    ST7A
        TRAN    DAP$K_CMP_MSG,  FAL$STORE_END,  ST7_1,  ST7_2
        TRAN    ,               FAL$OUT_OF_SEQ, ST7,    ST7

STATE   ST7_1
        TRAN    ,               FAL$DECODE_CMP, ST2_2,  ST7

STATE   ST7_2
        TRAN    ,               FAL$SAVE_MSG,   ST7A,   EXIT

STATE   ST7A
        TRAN    ,               FAL$INTE_MSG,   ,       ST7A
        TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST7A_1, ST7A
        TRAN    ,               FAL$OUT_OF_SEQ, ST7A,   ST7A

STATE   ST7A_1
        TRAN    DAP$K_ABORT,    FAL$BIT_BUCKET, ST7B,   EXIT
        TRAN    DAP$K_RESUME,   FAL$UNS_CONFUNC,ST7A,   ST7A
        TRAN    DAP$K_RETRY,    FAL$UNS_CONFUNC,ST7A,   ST7A
        TRAN    DAP$K_SKIP_REC, FAL$BIT_BUCKET, ST7,    EXIT
        TRAN    ,               FAL$INV_CONFUNC,ST7A,   ST7A

STATE   ST7B
        TRAN    ,               FAL$NEXT_MSG,   ,       ST7B    ; Discard DAT
        TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST2_2,  ST7B    ;  msgs in pipe
        TRAN    DAP$K_DAT_MSG,  FAL$DISCARD_DAT,ST7B,   EXIT    ;  until CMP msg
        TRAN    ,               FAL$OUT_OF_SEQ, ST7B,   ST7B    ;  is received
```

```
0438          .SBTTL  ST8 -- DIRECTORY LIST

0438  ;++
0438  ; Directory list retrieval state.
0438  ;
0438  ; Send a file specification and requested file attributes for each directory
0438  ; found to partner process. Engage in error recovery dialogue on an error in
0438  ; accessing directory information or if directed by partner to stop before
0438  ; the function is completed.
0438  ;--
0438  ;       State   Value           Action-Routine  Success Failure Comments

0438  STATE   ST8
0438          TRAN    ,               FAL$TEST_MSG,   ST8A,
0440          TRAN    ,               FAL$DIR_SEARCH, ,       ST8A
0448          TRAN    ,               FAL$CHECK_NMF,  ST0,    ST8     ; True/False
0450
0450  STATE   ST8A
0450          TRAN    ,               FAL$NEXT_MSG,   ,       ST8A
0458          TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST8A_1, ST8A
0460          TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST8A_2, ST8A
0468          TRAN    ,               FAL$OUT_OF_SEQ, ST8A,   ST8A
0470
0470  STATE   ST8A_1
0470          TRAN    DAP$K_CLOSE,    FAL$DIR_END,    ST0,    EXIT
0478          TRAN    DAP$K_TERMINATE,FAL$DIR_END,    ST0,    ST3     ; Inappropriate
0480          TRAN    ,               FAL$OUT_OF_SEQ, ST8A,   ST8A
0488
0488  STATE   ST8A_2
0488          TRAN    DAP$K_SKIP_REC, ,               ST8,    EXIT    ; Goto next file
0490          TRAN    DAP$K_RETRY,    FAL$UNS_CONFUNC,ST8A,   ST8A
0498          TRAN    ,               FAL$INV_CONFUNC,ST8A,   ST8A
```

```
04A0          .SBTTL  ST9 -- WILDCARD FILE RETRIEVAL
04A0
04A0    ;++
04A0    ; Wildcard file retrieval state.
04A0    ;--
04A0    ;       State   Value           Action-Routine  Success Failure Comments
04A0
04A0    STATE   ST9
04A0            TRAN    ,               FAL$TEST_MSG,   ST9A,
04A8            TRAN    ,               FAL$FIL_SEARCH, ,       ST9A
04B0            TRAN    ,               FAL$CHECK_NMF,  ST0,            ; True/False
04B8            TRAN    ,               FAL$OPEN,       ST9A,   ST9A
04C0
04C0    STATE   ST9A
04C0            TRAN    ,               FAL$NEXT_MSG,   ,       ST9A
04C8            TRAN    DAP$K_CTL_MSG,  FAL$DECODE_CTL, ST9A_1, ST9A
04D0            TRAN    DAP$K_CMP_MSG,  FAL$DECODE_CMP, ST9A_2, ST9A
04D8            TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST9A_3, ST9A
04E0            TRAN    ,               FAL$OUT_OF_SEQ, ST9A,   ST9A
04E8
04E8    STATE   ST9A_1
04E8            TRAN    DAP$K_CONNECT,  FAL$CONNECT,    ST9A,   ST9A
04F0            TRAN    DAP$K_GET_READ, FAL$CHECK_FTM,  ST9B,   EXIT
04F8            TRAN    ,               FAL$INV_CTLFUNC,ST9A,   ST9A
0500
0500    STATE   ST9A_2
0500            TRAN    DAP$K_DISCONN,  FAL$DISCONNECT, ST9A,   ST9A
0508            TRAN    DAP$K_CLOSE,    FAL$CLOSE,      ST9,    ST9A    ; Goto next file
0510            TRAN    DAP$K_SKIP_FILE,,               ST9,    EXIT    ; Goto next file
0518            TRAN    DAP$K_TERMINATE,FAL$CLOSE,      ST0,    ST9A    ; Stop operation
0520            TRAN    ,               FAL$INV_CMPFUNC,ST9A,   ST9A
0528
0528    STATE   ST9A_3
0528            TRAN    DAP$K_RESUME,   ,               ST9B,   EXIT    ; Same as retry
0530            TRAN    DAP$K_RETRY,    ,               ST9B,   EXIT
0538            TRAN    DAP$K_SKIP_REC, FAL$UNS_CONFUNC,ST9A,   ST9A
0540            TRAN    ,               FAL$INV_CONFUNC,ST9A,   ST9A
0548
0548    STATE   ST9B
0548            TRAN    ,               FAL$TEST_MSG,   ST9A,
0550            TRAN    ,               FAL$RETRY_FTM,  ST9B,   ST9A
```

```
0558   618          .SBTTL  ST10 -- WILDCARD FILE DELETION
0558   619
0558   620 ;++
0558   621 ; Wildcard file deletion state.
0558   622 ;--
0558   623 ;        State   Value           Action-Routine  Success Failure Comments
0558   624
0558   625 STATE    ST10
0558   626          TRAN    ,               FAL$FIL_SEARCH, ,       ST0
0560   627          TRAN    ,               FAL$CHECK_NMF,  ST0,            ; True/False
0568   628          TRAN    ,               FAL$ERASE,      ST10,   ST10A
0570   629
0570   630 STATE    ST10A
0570   631          TRAN    ,               FAL$NEXT_MSG,   ,       ST10A
0578   632          TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST10A_1,ST10A
0580   633          TRAN    ,               FAL$OUT_OF_SEQ, ST10A,  ST10A
0588   634
0588   635 STATE    ST10A_1
0588   636          TRAN    DAP$K_SKIP_REC, ,               ST10,   EXIT
0590   637          TRAN    DAP$K_RETRY,    FAL$UNS_CONFUNC,ST10A,  ST10A
0598   638          TRAN    ,               FAL$INV_CONFUNC,ST10A,  ST10A
```

```
05A0          .SBTTL  ST11 -- WILDCARD FILE EXECUTION
05A0
05A0    ;++
05A0    ; Wildcard file execution state.
05A0    ;--
05A0    ;       State   Value           Action-Routine  Success Failure Comments
05A0
05A0    STATE   ST11
05A0            TRAN    ,               FAL$FIL_SEARCH, ,       ST0
05A8            TRAN    ,               FAL$CHECK_NMF,  ST0,            ; True/False
05B0            TRAN    ,               FAL$EXECUTE,    ST11,   ST11A
05B8
05B8    STATE   ST11A
05B8            TRAN    ,               FAL$NEXT_MSG,   ,       ST11A
05C0            TRAN    DAP$K_CON_MSG,  FAL$DECODE_CON, ST11A_1,ST11A
05C8            TRAN    ,               FAL$OUT_OF_SEQ, ST11A,  ST11A
05D0
05D0    STATE   ST11A_1
05D0            TRAN    DAP$K_SKIP_REC, ,               ST11,   EXIT
05D8            TRAN    DAP$K_RETRY,    FAL$UNS_CONFUNC,ST11A,  ST11A
05E0            TRAN    ,               FAL$INV_CONFUNC,ST11A,  ST11A
```

```
05E8          .SBTTL  ST12 -- FILE RENAME

;++
; Non-wildcard file rename state
;--
;       State   Value           Action-Routine  Success Failure Comments

STATE   ST12
        TRAN                    FAL$NEXT_MSG,   ,       ST12
        TRAN    DAP$K_NAM_MSG,  FAL$DECODE_NAM, ST12_1, ST12
        TRAN    ,               FAL$OUT_OF_SEQ, ST12,   ST12

STATE   ST12_1
        TRAN    ,               FAL$RENAME,     ST0,    ST0
```

```
0608    677          .SBTTL  ST13 -- WILDCARD RENAME
0608    678
0608    679 ;++
0608    680 ; Wildcard file rename state.
0608    681 ;--
0608    682 ;        State   Value            Action-Routine  Success Failure Comments
0608    683
0608    684 STATE    ST13
0608    685          TRAN    ,                FAL$NEXT_MSG,   ,       ST13
0610    686          TRAN    DAP$K_NAM_MSG,   FAL$DECODE_NAM, ST13A,  ST13
0618    687          TRAN    ,                FAL$OUT_OF_SEQ, ST13,   ST13
0620    688
0620    689 STATE    ST13A
0620    690          TRAN    ,                FAL$FIL_SEARCH, ,       ST0
0628    691          TRAN    ,                FAL$CHECK_NMF,  ST0,            ; True/False
0630    692          TRAN    ,                FAL$RENAME,     ST13A,  ST13B
0638    693
0638    694 STATE    ST13B
0638    695          TRAN    ,                FAL$NEXT_MSG,   ,       ST13B
0640    696          TRAN    DAP$K_CON_MSG,   FAL$DECODE_CON, ST13B_1,ST13B
0648    697          TRAN    ,                FAL$OUT_OF_SEQ, ST13B,  ST13B
0650    698
0650    699 STATE    ST13B_1
0650    700          TRAN    DAP$K_SKIP_REC,  ,               ST13A,  EXIT
0658    701          TRAN    DAP$K_RETRY,     FAL$UNS_CONFUNC,ST13B,  ST13B
0660    702          TRAN    ,                FAL$INV_CONFUNC,ST13B,  ST13B
```

```
0668   704 ;++
0668   705 ; Termination state.
0668   706 ;--
0668   707
0668   708 STATE   EXIT
0668   709         RETURN
0670   710
0670   711         .END                                ; End of module
```

```
ACTION          = 00000004
COUNT...        = 000000CD
DAP$B_ACCFUNC     00000040
DAP$B_ACCOPT      00000041
DAP$B_BITCNT      00000035
DAP$B_BLKCNT      00000056
DAP$B_CMPFUNC     00000040
DAP$B_CONFUNC     00000040
DAP$B_CTLFUNC     00000040
DAP$B_FAC         00000042
DAP$B_FLAGS       00000031
DAP$B_KRF         00000047
DAP$B_LEN256      00000034
DAP$B_LENGTH      00000033
DAP$B_RAC         00000046
DAP$B_SHR         00000043
DAP$B_STREAMID    00000032
DAP$B_TYPE        00000030
DAP$K_ABORT     = 00000003
DAP$K_ACC_MSG   = 00000003
DAP$K_ALL_MSG   = 0000000B
DAP$K_ATT_MSG   = 00000002
DAP$K_CHANGE_B  = 00000006
DAP$K_CHANGE_E  = 00000007
DAP$K_CLOSE     = 00000001
DAP$K_CMP_MSG   = 00000007
DAP$K_CNF_MSG   = 00000001
DAP$K_CONNECT   = 00000002
DAP$K_CON_MSG   = 00000005
DAP$K_CREATE    = 00000002
DAP$K_CTL_MSG   = 00000004
DAP$K_DAT_MSG   = 00000008
DAP$K_DELETE    = 00000005
DAP$K_DIR_LIST  = 00000006
DAP$K_DISCONN   = 00000004
DAP$K_DISPLAY   = 00000010
DAP$K_ERASE     = 00000004
DAP$K_EXECUTE   = 00000008
DAP$K_EXTEND_B  = 0000000B
DAP$K_EXTEND_E  = 0000000F
DAP$K_FIND      = 0000000E
DAP$K_FLUSH     = 0000000C
DAP$K_FREE      = 0000000A
DAP$K_GET_READ  = 00000001
DAP$K_KEY_MSG   = 0000000A
DAP$K_LOAD      = 000000FF
DAP$K_NAM_MSG   = 0000000F
DAP$K_OPEN      = 00000001
DAP$K_PRO_MSG   = 0000000E
DAP$K_PUT_WRITE= 00000004
DAP$K_RELEASE   = 00000009
DAP$K_RENAME    = 00000003
DAP$K_RESET     = 00000003
DAP$K_RESUME    = 00000004
DAP$K_RETRY     = 00000001
DAP$K_REWIND    = 00000006
DAP$K_SEQ_ACC   = 00000000
DAP$K_SKIP_FILE= 00000005
DAP$K_SKIP_REC  = 00000002
DAP$K_SPACE_BW  = 00000012
DAP$K_SPACE_FW  = 00000011
DAP$K_SUBMIT    = 00000007
DAP$K_TERMINATE= 00000008
DAP$K_TIM_MSG   = 0000000D
DAP$K_TRUNCATE  = 00000007
DAP$K_UPDATE    = 00000003
DAP$L_FOP2        00000044
DAP$L_ROP         00000050
DAP$M_BITCNT    = 00000008
DAP$M_BLKCNT    = 00000040
DAP$M_DSP_3NAM  = 00000200
DAP$M_GET       = 00000002
DAP$M_GO_NOGO   = 00000010
DAP$M_MSE       = 00000010
DAP$M_SEGMENT   = 00000040
DAP$M_TMP1$     = 00000008
DAP$M_TMP2$     = FFF80000
DAP$Q_FILESPEC    00000044
DAP$Q_KEY         00000048
DAP$Q_PASSWORD    00000050
DAP$Q_SYSPEC      00000038
DAP$W_CHECK       00000042
DAP$W_CTLMENU     00000044
DAP$W_DISPLAY1    0000004C
DAP$W_DISPLAY2    00000054
EXIT            = 000000CD
FAILURE         = 00000003
FAL$BIT_BUCKET    ********   X   03
FAL$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$CHANGE        ********   X   03
FAL$CHECK_FTM     ********   X   03
FAL$CHECK_NMF     ********   X   03
FAL$CHECK_WILD    ********   X   03
FAL$CLOSE         ********   X   03
FAL$CONNECT       ********   X   03
FAL$CREATE        ********   X   03
FAL$C_WRKBLN      00002000
FAL$DECODE_ACC    ********   X   03
FAL$DECODE_ALL    ********   X   03
FAL$DECODE_ATT    ********   X   03
FAL$DECODE_CMP    ********   X   03
FAL$DECODE_CNF    ********   X   03
FAL$DECODE_CON    ********   X   03
FAL$DECODE_CTL    ********   X   03
FAL$DECODE_KEY    ********   X   03
FAL$DECODE_NAM    ********   X   03
FAL$DECODE_PRO    ********   X   03
FAL$DECODE_TIM    ********   X   03
FAL$DELETE        ********   X   03
FAL$DIR_END       ********   X   03
FAL$DIR_PARSE     ********   X   03
FAL$DIR_SEARCH    ********   X   03
FAL$DISCARD_DAT   ********   X   03
FAL$DISCONNECT    ********   X   03
FAL$DISPLAY       ********   X   03
FAL$ERASE         ********   X   03
FAL$EXECUTE       ********   X   03
FAL$EXTEND        ********   X   03
FAL$FIL_PARSE     ********   X   03
FAL$FIL_SEARCH    ********   X   03
FAL$FIND          ********   X   03
FAL$FLUSH         ********   X   03
FAL$FREE          ********   X   03
FAL$INIT          ********   X   03
FAL$INIT_RENAME   ********   X   03
FAL$INIT_XABCHN   ********   X   03
FAL$INTE_MSG      ********   X   03
FAL$INV_ACCFUNC   ********   X   03
FAL$INV_CMPFUNC   ********   X   03
FAL$INV_CONFUNC   ********   X   03
FAL$INV_CTLFUNC   ********   X   03
FAL$K_WRKBLN      00002000
FAL$LOAD_IMAGE    ********   X   03
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$L_USE_VER     000000A4
FAL$NEXT_MSG      ********   X   03
FAL$OPEN          ********   X   03
FAL$OUT_OF_SEQ    ********   X   03
```

```
FAL$Q_BLD          00000050              FLAGS          = 00000000          ST9A_3         = 000000A5
FAL$Q_DIRNAME      00000088              M_EXIT         = 00000002          ST9B           = 000000A9
FAL$Q_FALLOG       00000090              M_MATCH        = 00000001          SUCCESS        = 00000002
FAL$Q_FLG          00000000              ST0            = 00000000          VALUE          = 00000001
FAL$Q_MBX          00000038              ST1            = 00000004          V_EXIT         = 00000001
FAL$Q_MBXIOSB      00000030              ST10           = 000000AB          V_MATCH        = 00000000
FAL$Q_RCV          00000040              ST10A          = 000000AE
FAL$Q_RCVIOSB      00000020              ST10A_1        = 000000B1
FAL$Q_RMS          00000064              ST11           = 000000B4
FAL$Q_STATE_CTX    00000008              ST11A          = 000000B7
FAL$Q_SYSNET       00000098              ST11A_1        = 000000BA
FAL$Q_TEMP         000003F8              ST12           = 000000BD
FAL$Q_VOLNAME      00000080              ST12_1         = 000000C0
FAL$Q_XMT          00000048              ST13           = 000000C1
FAL$Q_XMTIOSB      00000028              ST13A          = 000000C4
FAL$RELEASE        ********   X   03     ST13B          = 000000C7
FAL$RENAME         ********   X   03     ST13B_1        = 000000CA
FAL$RESET          ********   X   03     ST1_1          = 0000000E
FAL$RETRV_FTM      ********   X   03     ST1_1A         = 00000017
FAL$RETRV_RAM      ********   X   03     ST1_1B         = 00000019
FAL$REWIND         ********   X   03     ST1_1C         = 0000001B
FAL$SAVE_MSG       ********   X   03     ST1_1D         = 0000001D
FAL$SPACE_BW       ********   X   03     ST1_1E         = 0000001E
FAL$SPACE_FW       ********   X   03     ST2            = 00000020
FAL$STATE          00000000 RG    02     ST2A           = 00000036
FAL$STATE_TABLE    00000000 RG    03     ST2B           = 00000039
FAL$STORE_END      ********   X   03     ST2B_1         = 0000003D
FAL$STORE_FTM      ********   X   03     ST2_T          = 00000024
FAL$STORE_RAM      ********   X   03     ST2_2          = 00000034
FAL$SUBMIT         ********   X   03     ST3            = 0000003F
FAL$TEST_MSG       ********   X   03     ST3A           = 00000047
FAL$TRUNCATE       ********   X   03     ST3A_1         = 0000004C
FAL$T_DAP          00000100              ST3_T          = 00000042
FAL$T_DIRNAME      00001F00              ST4            = 0000004E
FAL$T_EXPAND       00000500              ST4A           = 0000004F
FAL$T_EXPAND2      00000A00              ST4A_1         = 00000054
FAL$T_FALLOG       00001C00              ST5            = 00000059
FAL$T_FILESPEC     00000400              ST5A           = 0000005E
FAL$T_FILESPEC2    00000900              ST5A_1         = 00000061
FAL$T_KEYBUF       00000700              ST5B           = 00000066
FAL$T_MBXBUF       00001980              ST6            = 0000006A
FAL$T_PRTBUF1      00001A00              ST6A           = 0000006C
FAL$T_PRTBUF2      00001B00              ST6A_1         = 00000070
FAL$T_RESULT       00000600              ST7            = 00000075
FAL$T_RESULT2      00000B00              ST7A           = 0000007B
FAL$T_SYSNET       00001D00              ST7A_1         = 0000007E
FAL$T_VOLNAME      00001E00              ST7B           = 00000083
FAL$UNS_CONFUNC    ********   X   03     ST7_1          = 00000079
FAL$UPDATE         ********   X   03     ST7_2          = 0000007A
FAL$W_DAPBUFSIZ    0000001A              ST8            = 00000087
FAL$W_DISPLAY      00000070              ST8A           = 0000008A
FAL$W_LNKCHN       0000001C              ST8A_1         = 0000008E
FAL$W_MBXCHN       0000001E              ST8A_2         = 00000091
FAL$W_QIOBUFSIZ    00000018              ST9            = 00000094
FAL$W_RECEIVED     00000072              ST9A           = 00000098
FAL$W_USE_DBS      000000A0              ST9A_1         = 0000009D
FAL$W_USE_SYS      000000A2              ST9A_2         = 000000A0
```

```
                                    +-----------------+
                                    ! 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$CODE                        0000003F  (   63.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE
FAL$STATE_TABLE                 00000670  ( 1648.)  03 (  3.)  NOPIC   USR   CON   REL   LCL   SHR NOEXE   RD  NOWRT NOVEC QUAD

                                  +-------------------------+
                                  ! Performance indicators !
                                  +-------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     29    00:00:00.06    00:00:00.68
Command processing                105    00:00:00.40    00:00:03.97
Pass 1                            308    00:00:09.70    00:00:34.94
Symbol table sort                   0    00:00:00.54    00:00:01.93
Pass 2                            141    00:00:02.14    00:00:07.30
Symbol table output                23    00:00:00.12    00:00:00.44
Psect synopsis output               3    00:00:00.01    00:00:00.02
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              611    00:00:12.97    00:00:49.28

The working set limit was 1350 pages.
70250 bytes (138 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 597 non-local and 5 local symbols.
711 source lines were read in Pass 1, producing 26 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

                                    +---------------------------+
                                    ! Macro library statistics !
                                    +---------------------------+

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[FAL.OBJ]FAL.MLB;1                                  6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                               4
TOTALS (all libraries)                                         10

666 GETS were required to define 10 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:FALSTATE/OBJ=OBJ$:FALSTATE MSRC$:FALSTATE/UPDATE=(ENH$:FALSTATE)+LIB$:FAL/LIB
```