


```

FFFFFFFFF      AAAAAA      LL      DDDDDDD      EEEEEEEEE      FFFFFFFFF
FFFFFFFFF      AAAAAA      LL      DDDDDDD      EEEEEEEEE      FFFFFFFFF
FF            AA          AA      LL      DD          DD      EE            FF
FF            AA          AA      LL      DD          DD      EE            FF
FF            AA          AA      LL      DD          DD      EE            FF
FF            AA          AA      LL      DD          DD      EE            FF
FFFFFFFF      AA          AA      LL      DD          DD      EEEEEEEEE      FFFFFFFF
FFFFFFFF      AA          AA      LL      DD          DD      EEEEEEEEE      FFFFFFFF
FF            AAAAAAAAAA      LL      DD          DD      EE            FF
FF            AAAAAAAAAA      LL      DD          DD      EE            FF
FF            AA          AA      LL      DD          DD      EE            FF
FF            AA          AA      LL      DD          DD      EE            FF
FF            AA          AA      LLLLLLLLLL      DDDDDDD      EEEEEEEEE      FF
FF            AA          AA      LLLLLLLLLL      DDDDDDD      EEEEEEEEE      FF

```

```

MM          MM      DDDDDDD      LL
MM          MM      DDDDDDD      LL
MMMM      MMMM      DD          DD      LL
MMMM      MMMM      DD          DD      LL
MM      MM      MM      DD          DD      LL
MM      MM      MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DD          DD      LL
MM          MM      DDDDDDD      LLLLLLLLLL
MM          MM      DDDDDDD      LLLLLLLLLL

```

.TITLE \$FALDEF - FAL CONTROL BLOCK DEFINITIONS
 .IDENT 'V04-000'

 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
 * ALL RIGHTS RESERVED. *
 * * * * *

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
 * TRANSFERRED. *
 * * * * *

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
 * CORPORATION. *
 * * * * *

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
 * * * * *

♦♦ Facility: FAL (DECnet File Access Listener)

Abstract:

This module defines symbols for the FAL control blocks that are defined by the \$FALWRKDEF and \$FALSTBDEF macros.

Environment:

The MDL translator must be used to convert FALDEF.MDL into FALDEF.MAR (and FALDEF.B32).

Author: James A. Krycka, Creation Date: 16-JUN-1977

Modified By:

V03-007 JAK0145 J A Krycka 12-APR-1984
 Track changes in DAP message building algorithm: rename FALSQ_XMT to FALSQ_BLD, rename FALSQ_AUX to FALSQ_XMT, and remove FALSQ_AUX_PTR.

V03-006 JAK0137 J A Krycka 12-MAR-1984
 Add FALSQV_LOG_CNT logging option and FALSQV_DIS_PMR option.

V03-005 JAK0136 J A Krycka 07-MAR-1984
 Expand the FAL work area to 8K bytes.
 Extend size of Key Definition XAB storage area.

Replace FALSQ_NCB and FALST_NCBBUF with FALSQ_FALLOG,
FALST_FALLOG, FALSQ_SYSNET, and FALST_SYSNET.
Add FALST_PRTBUF1 and FALST_PRTBUF2 for FAL logging.
Replace FALS_LLOGFLG with four subfields that identify FALSLOG
options by class.
Also add several new fields to store FALSLOG option values.
General cleanup and reorganization.

- V03-004 JAK0107 J A Krycka 06-MAY-1983
Extend size of Protection XAB storage area.
- V03-003 JAK0104 J A Krycka 17-MAR-1983
Add FALSQ_RMS and FALS_L_RMS_PTR.
Add FALS_V_DIS_RBK logging option.
Add FALS_V_TERMINATE flag.
- V03-002 KRM0098 K Malik 06-Apr-1983
Add FALS_V_NEWNAM flag for use during a rename operation to
tell FALS_ENCODE_NAM and FALS_LOG_RESNAM which NAM block to use.
- V03-001 KRM0070 K Malik 23-Nov-1982
Add FALS_L_FAB2, FALS_L_NAM2, FALST_NAMESPEC, FALST_EXPAND2,
and FALST_RESULT2 for use by \$RENAME.
Increase size of arrays to support 32 Allocation and Key
Definition XABs.
Change FALS_B_LOGFLG to FALS_L_LOGFLG.
Rename some symbols for consistency.

```

:++
: SFALWRKDEF defines FAL work area offsets and symbols.
:--

```

```

$STRUCT FAL,WRKDEF

```

```

:-----
: Miscellaneous constants
:-----
K RCVEFN,1      : Link receive QIO event flag
K XMTEFN,2      : Link transmit QIO event flag
K MBXEFN,3      : Mailbox receive QIO event flag
K MAXBUFSIZ,<64*512-1> : Maximum DAP buffer size supported
K MIN_RBK,1     : Minimum RMS multi-block cache size in blocks
K MAX_RBK,127   : Maximum RMS multi-block cache size in blocks
K DFLT_RBK,64   : Default RMS multi-block cache size in blocks
K DFLT_BPM,20   : Default number of bytes per DAP message to
                  : display when DAP messages are logged
K DFLT_BPL,20   : Default number of bytes per line to display
                  : when DAP messages are logged
:-----
: ***** offset = ^X0000 = 0 *****
:-----
F FLG,0         : FAL status flags follow (bits 00-63):
S LOGGING,4,B   : Define bits 32-39 as FAL logging options
S ENABLE,1,B    : Define bits 40-47 as FAL enable options
S DISABLE,6,B  : Define bits 48-55 as FAL disable options
S MISCOPT,7,B  : Define bits 56-63 as FAL miscellaneous options
V <M

CNF_MSG        : DAP message state information:
ATT_MSG        : Processed Configuration message
.6             : Processed Attributes message

FTM            : Miscellaneous state information:
BLK_IO         : File transfer mode (vs record access mode)
WILD           : Block I/O mode (vs record I/O mode)
               : This is a wildcard operation (determined
               : via parse of filespec string)
NEWNAM         : Use NAM2 block (used during rename operation)
TERMINATE     : FAL has entered termination sequence
.3

RCVQIO        : QIO related state information:
RCVAST        : Link receive QIO outstanding flag
XMTQIO        : Link receive AST delivered flag
XMTAST        : Link transmit QIO outstanding flag
MBXQIO        : Link transmit AST delivered flag
MBXAST        : Mailbox receive QIO outstanding flag
.2            : Mailbox receive AST delivered flag

LAST_MSG      : Routine call function modifiers:
               : (The next bit is input to FALSTRANSMIT)
RET_RFA       : Last DAP message in sequence to be blocked
RET_RECNUM   : (The next 3 bits are input to FAL%ENCODE_STS)
RET_STV      : Include RFA field in Status message
.4            : Include RECNUM field in Status message
               : Include STV field in Status message

```

```

-----
LOG_NAM      : These flags specify FAL logging options:
LOG_STA      : Enable logging of filename/function requests
LOG_MSG      : Enable logging of data thrupt statistics
              : Enable logging of individual DAP messages
              : as they are encoded and decoded
LOG_AST      : Enable logging of DAP message packet AST
              : completions
LOG_QIO      : Enable logging of DAP message packet QIO
              : requests
LOG_CNT      : Enable logging of internal counters
LOG_BIT0     : Spare
.1
ENA_BIT0     : These flags enable FAL features:
.7           : Spare

DIS_CRC      : These flags disable FAL features:
DIS_MBK      : Disable file level CRC checksum computation
DIS_RBK      : Disable DAP message blocking
              : Disable RMS multi-block caching to/from
              : disk during block I/O file transfer mode
DIS_PMR      : Disable poor-man's routing (i.e., reject any
              : filespec received that contains a nodename)
DIS_BIT4     : Spare
.3

PARSE_ERR    : These flags signal a parsing failure or denote
USE_DBS      : qualifiers which require additional processing
USE_SYS      : Error in parse of FAL$LOG equivalence string
              : Use specified DAP buffer size
              : Use specified 2-byte operating system and
              : file system type identification
USE_VER      : Use specified 4-byte DAP version number value
USE_SC1      : Use specified 32-bit value as first part of
              : system capabilities mask to send
USE_SC2      : Use specified 32-bit value as second part of
              : system capabilities mask to send
.2
>
-----
F STATE_CTX,Q : State transition table context:
S TBL_ADDR,0,L : State table address
S CUR_ADDR,4,L : Current table entry address
F VALUE,B     : State transition value
F RCVBUFIDX,B : Receive buffer index (into address table)
F RBK_CACHE,B : Number of blocks in RMS multi-blocking cache
F ,B,5       : Unused locations
F QIOBUFSIZ,W : Largest QIO request that can be supported by
              : FAL (i.e., buffer size available)
F DAPBUFSIZ,W : Maximum DAP message size (determined via
              : exchange of configuration messages)
F LNKCHN,W    : Link channel #
F MBXCHN,W    : Associated mailbox channel #
F RCVIOSB,Q   : Link receive I/O status block
F XMTIOSB,Q   : Link transmit I/O status block
F MBXIOSB,Q   : Mailbox receive I/O status block
F MBX,Q       : Mailbox message descriptor block

```

```

F RCV,Q      : Receive DAP message descriptor
F XMT,Q      : Transmit DAP message descriptor
              (used for blocking DAP messages)
F BLD,Q      : Build DAP message descriptor
              (used in building a new DAP message)
              (note that BLD buffer overlays XMT buffer)

F .L        : Spare
F RCVBUF,L,2 : Receive buffer address table
K RCVBUFCNT,2 : Number of receive buffers
F RMS,Q      : RMS block buffer descriptor
F RMS_PTR,L  : Next byte pointer for RMS block buffer
F DISPLAY,W  : Mask of optional DAP messages to return
              (copy of DAPSW_DISPLAY1 or DAPSW_DISPLAY2)
F RECEIVED,W : Mask of received XABs to link to XAB chain
              (i.e., XABs generated as a result of
              receiving optional DAP messages)

V <M
  KEYXAB     : Key Definition XAB(s)
  ALLXAB     : Allocation XAB(s)
  DATXAB     : Date and Time XAB
  PROXAB     : Protection XAB
  RDTXAB     : Revision Date and Time XAB
>

F ALLXABINI,L : Bit vector of Allocation XABs initialized
F KEYXABINI,L : Bit vector of Key Definition XABs initialized
F CHAIN_NXT,L : Address of last pointer cell in XAB chain
              -----
              ***** offset = ^X0080 = 128 *****
              -----
              (VOLNAME and DIRNAME must be contiguous)
F VOLNAME,Q   : Volume name descriptor (for 3-part name)
F DIRNAME,Q   : Directory name descriptor (for 3-part name)
F FALLOG,Q    : FALSLOG equivalence string descriptor
F SYSNET,Q    : SYSSNET equivalence string descriptor (which
              will be made into an NCB descriptor)
F USE_DBS,W   : User entered DAP buffer size
F USE_SYS,W   : User entered 2-byte operating system and file
              system type identification field
F USE_VER,L   : User entered 4-byte DAP version number field
F USE_SC1,L   : User entered system capabilities mask (00-31)
F USE_SC2,L   : User entered system capabilities mask (31-63)
F .L,4        : Unused locations
F STB,L,16    : Statistics block
K STB,64      : Statistics block size
F DAP,T,192   : DAP control block storage area
K DAP,192     : DAP control block size
F .L,8        : Reserved for future expansion of DAP structure
F .L,5        : Unused locations
F DATATYPE,B  : Copy of DAP$B_DATATYPE
F ACCOPT,B    : Copy of DAP$B_ACCOPT
F ACCFUNC,B   : Copy of DAP$B_ACCFUNC
F RAC,B       : Copy of DAP$B_RAC
F FOP,L       : Copy of DAP$B_FOP
F NUMBER,L    : Requested random record number
              -----
              ***** offset = ^X0200 = 512 *****

```

```

-----
: (FAB, RAB, NAM, and FHCXAB must be contiguous)
: FAB storage area
: FAB size
: RAB storage area
: RAB size
: NAM block storage area
: NAM control block size
: File Header Characteristics XAB storage area
: File Header Characteristics XAB size
: Date and Time XAB storage area
: Date and Time XAB size
: Protection XAB storage area
: Protection XAB size
: Summary XAB storage area
: Summary XAB size
: Revision Date and Time XAB storage area
: Revision Date and Time XAB size
: Unused locations
: Temporary longword work area
: Temporary quadword work area
-----
: ***** offset = ^X0400 = 1024 *****
-----
: File specification string buffer
: File specification string buffer size
: Expanded name string buffer
: Expanded name string buffer size
: Resultant name string buffer
: Resultant name string buffer size
: Key buffer
: Key buffer size
-----
: ***** offset = ^X0800 = 2048 *****
-----
: RMS structures for use by RENAME operation
: (FAB2 and NAM2 must be contiguous)
: FAB storage area
: FAB size
: NAM storage area
: NAM size
: Unsued locations
: File specification string buffer
: File specification string buffer size
: Expanded name string buffer
: Expanded name string buffer size
: Resultant name string buffer
: Resultant name string buffer size
-----
: ***** offset = ^X0C00 = 3072 *****
-----
: Allocation XAB storage area
: Allocation XAB size
: Largest area ID value supported by FAL
: Space for 31 other Allocation XABs in array
-----

```

F FAB,L,20
K FAB,80
F RAB,L,17
K RAB,68
F NAM,L,24
K NAM,96
F FHCXAB,L,11
K FHCXAB,44
F DATXAB,L,11
K DATXAB,44
F PROXAB,L,22
K PROXAB,88
F SUMXAB,L,3
K SUMXAB,12
F RDTXAB,L,5
K RDTXAB,20
F .L,12
F TEMP,L
F TEMP,Q

F FILESPEC,T,256
K FILESPEC,255
F EXPAND,T,256
K EXPAND,255
F RESULT,T,256
K RESULT,255
F KEYBUF,T,256
K KEYBUF,255

F FAB2,L,20
K FAB2,80
F NAM2,L,24
K NAM2,96
F .L,20
F FILESPEC2,T,256
K FILESPEC2,255
F EXPAND2,T,256
K EXPAND2,255
F RESULT2,T,256
K RESULT2,255

F ALLXAB,L,8
K ALLXAB,32
K MAX AID,31
F .L,248


```

: ***** offset = ^X1000 = 4096 *****
: -----
F KEYXAB,L,19      : Key Definition XAB storage area
K KEYXAB,76       : Key Definition XAB size
K MAX REF,31     : Largest key of reference value supported here
F ,L,589         : Space for 31 alternate Key Definition XABs
F MBXBUF,T,64    : Mailbox buffer
K MBXBUF,64      : Mailbox buffer size
K MBXQUOTA,<64*2> : Mailbox quota
F ,L,16         : Unused locations
: -----
: ***** offset = ^X1A00 = 6656 *****
: -----
F PRTBUF1,T,256  : Primary (non-AST-level) print buffer
F PRTBUF2,T,256  : Secondary (AST-level) print buffer
K PRTBUF,256     : Print buffer size (for FAL logging)
: -----
M 1              : ***** offset = ^X1C00 = 7168 *****
: -----
F KEYNAM,L,8     : Key Name buffer
K KEYNAM,32     : Key Name buffer size
F ,L,248        : Space for 31 alternate Key Name buffers
: -----
P 1              : ***** offset = ^X1C00 = 7168 *****
: -----
F FALLOG,T,256   : Logical name translation buffer for FAL$LOG
K FALLOG,255    : Logical name translation buffer size
F SYSNET,T,256   : Logical name translation buffer for SYSSNET
K SYSNET,255    : Logical name translation buffer size
F VOLNAME,T,256  : Volume name string buffer (to store
: (concatenated node specs plus device name)
K VOLNAME,255   : Volume name string buffer size
F DIRNAME,T,256  : Directory name string buffer
K DIRNAME,255   : Directory name string buffer size
: -----
: ***** offset = ^X2000 = 8192 *****
: -----
L WRKBLN        : Define length of this control block
E

```

```
:+  
: SFALSTBDEF defines FAL statistics block offsets and symbols.  
:--
```

```
$STRUCT FAL,STBDEF  
F RCV_PKT,L      : Total # DAP message packets received  
F RCV_MSG,L      : Total # DAP messages received  
F RCV_DAT,L      : Total # records/blocks received  
                  : (i.e., # DAP Data messages)  
F RCV_USR,L      : Total # bytes of user data received  
                  : (i.e., # bytes in all records/blocks)  
F RCV_LNK,L      : Total # bytes of link data received  
                  : (i.e., # bytes in all DAP messages)  
F XMT_PKT,L      : Total # DAP message packets transmitted  
F XMT_MSG,L      : Total # DAP messages transmitted  
F XMT_DAT,L      : Total # records/blocks transmitted  
                  : (i.e., # DAP Data messages)  
F XMT_USR,L      : Total # bytes of user data transmitted  
                  : (i.e., # bytes in all records/blocks)  
F XMT_LNK,L      : Total # bytes of link data transmitted  
                  : (i.e., # bytes in all DAP messages)  
F .L,6           : Unused locations  
L STBBLN         : Define length of this control block  
E  
  
: End of module
```


0174 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a dense grid of approximately 100 overlapping terminal window screenshots. Each window shows a monospaced text-based interface, typical of early computer systems. The text within the windows is mostly illegible due to the small size and overlap, but several larger, more prominent windows are clearly visible, each with a title bar:

- FALACTION LIS**: Located in the upper left quadrant.
- FALDAP10 LIS**: Located in the upper right quadrant.
- FALMACROS MAR**: Located in the middle left quadrant.
- FALBLDXAB LIS**: Located in the middle right quadrant.
- FALBLDATT LIS**: Located in the lower middle quadrant.
- FALBLDSTS LIS**: Located in the lower middle quadrant, below FALBLDATT.
- FALDEF MDL**: Located in the lower left quadrant.
- FALACTINT LIS**: Located in the lower left quadrant, below FALDEF MDL.
- FALACTMSG LIS**: Located in the lower center quadrant.

The overall appearance is that of a complex, multi-tasking operating environment where numerous processes or users are running simultaneously, each with its own dedicated terminal window.