



```

FFFFFFFFF      AAAAAA      LL      MM      MM      AAAAAA      CCCCCCCC      RRRRRRRR      000000      SSSSSSSS
FFFFFFFFF      AAAAAA      LL      MM      MM      AAAAAA      CCCCCCCC      RRRRRRRR      000000      SSSSSSSS
FF           AA      AA      LL      MMMM      MMMM      AA      AA      CC      RR      RR      00      00      SS
FF           AA      AA      LL      MMMM      MMMM      AA      AA      CC      RR      RR      00      00      SS
FF           AA      AA      LL      MM      MM      MM      AA      AA      CC      RR      RR      00      00      SS
FF           AA      AA      LL      MM      MM      MM      AA      AA      CC      RR      RR      00      00      SS
FFFFFFFFF      AA      AA      LL      MM      MM      MM      AA      AA      CC      RRRRRRRR      00      00      SSSSSS
FFFFFFFFF      AA      AA      LL      MM      MM      MM      AA      AA      CC      RRRRRRRR      00      00      SSSSSS
FF           AAAAAAAAAA      LL      MM      MM      AAAAAAAAAA      CC      RR      RR      00      00      SS
FF           AAAAAAAAAA      LL      MM      MM      AAAAAAAAAA      CC      RR      RR      00      00      SS
FF           AA      AA      LL      MM      MM      AA      AA      CC      RR      RR      00      00      SS
FF           AA      AA      LL      MM      MM      AA      AA      CC      RR      RR      00      00      SS
FF           AA      AA      LLLLLLLLLL      MM      MM      AA      AA      CCCCCCCC      RR      RR      000000      SSSSSSSS
FF           AA      AA      LLLLLLLLLL      MM      MM      AA      AA      CCCCCCCC      RR      RR      000000      SSSSSSSS

```

```

MM      MM      AAAAAA      RRRRRRRR
MM      MM      AAAAAA      RRRRRRRR
MMMM      MMMM      AA      AA      RR      RR
MMMM      MMMM      AA      AA      RR      RR
MM      MM      MM      AA      AA      RR      RR
MM      MM      MM      AA      AA      RR      RR
MM      MM      AA      AA      RRRRRRRR
MM      MM      AA      AA      RRRRRRRR
MM      MM      AAAAAAAAAA      RR      RR
MM      MM      AAAAAAAAAA      RR      RR
MM      MM      AA      AA      RR      RR
MM      MM      AA      AA      RR      RR
MM      MM      AA      AA      RR      RR
MM      MM      AA      AA      RR      RR

```

.TITLE FALMACROS - MACRO DEFINITIONS FOR FAL  
.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 contains MACRO definitions that are placed in FAL.MLB for
:   use by FAL source modules.
:
: Environment: VAX/VMS, user mode
:
: Author: James A. Krycka,      Creation Date: 16-JUN-1977
:
: Modified By:
:
:   V03-001 JAK0136      J A Krycka      07-MAR-1984
:   Change CHECK_SS to $CHECK_SS.
:   Chnage CHECK_RMS to $CHECK_STATUS.
:   Change QBLOCK to $QBLOCK.
:
--

```

FAL  
VO

```
.SBTTL CODE GENERATION MACROS
```

```
:++  
: $SETBIT sets a single bit in a field.  
:--
```

```
DISPL: .MACRO $SETBIT POS,BASE,?DISPL  
BBSS POS,BASE,DISPL  
.ENDM $SETBIT
```

```
:++  
: $CLRBIT clears a single bit in a field.  
:--
```

```
DISPL: .MACRO $CLRBIT POS,BASE,?DISPL  
BBCC POS,BASE,DISPL  
.ENDM $CLRBIT
```

```
:++  
: $MAPBIT maps the designated bit from R1 into the designated bit in R2.  
: The bit is set in R2 only if the corresponding bit is set in R1.  
:--
```

```
LABEL: .MACRO $MAPBIT SRCBIT,DSTBIT,?LABEL  
BBC #SRCBIT,R1,LABEL  
BBCS #DSTBIT,R2,LABEL  
.ENDM $MAPBIT
```

```
:++  
: $ZERO_FILL writes zeroes into the specified buffer. On completion R0-R5 are  
: destroyed (with R3 containing the address of one byte beyond the buffer).  
: The default is to zero 512 bytes (one page) at the specified address.  
:--
```

```
.MACRO $ZERO_FILL DST=,SIZE=#512  
MOVCS #0,DST,#0,SIZE,DST  
.ENDM $ZERO_FILL
```

```
:++  
: $CHECK_SS calls a subroutine that checks the status code in R0 and takes  
: appropriate action. This MACRO is intended to check the results of a call  
: to a System Service.  
:--
```

```
.MACRO $CHECK_SS  
BSBW FAL$CHECK_SS  
.ENDM $CHECK_SS
```

```
:++  
: $CHECK_STATUS calls a subroutine that checks the status code in R0 and takes  
: appropriate action. This MACRO is intended to check the results of a FAL  
: logging operation where a failure should not result in an image exit.  
:--
```

```
.MACRO $CHECK_STATUS
BSBW  FAL$CHECK_STATUS
.ENDM  $CHECK_STATUS
```

```
:++
: $QBLOCK generates a quadword descriptor block followed by the character string
: itself and/or allocated space.
:--
```

```
.MACRO $QBLOCK TEXT,SPACE=0,BUFADR,?LABEL1,?LABEL2
.LONG LABEL2-LABEL1
.LONG LABEL1
.IF NB BUFADR
BUFADR==.
.ENDC
LABEL1:
.IRP STR,<TEXT>
.ASCII \STR\
.ENDR
.IF NE SPACE
.BLKB SPACE
.ENDC
LABEL2:
.FNDM $QBLOCK
```

```
:++
: $CASEB, $CASEW, and $CASEL generate a CASEB, CASEW, CASEL instruction,
: respectively, followed by the case displacement table. The parameters for
: each MACRO are:
```

```
:
: SELECTOR= the selector operand
: BASE     = the base operand
: DISPL    = the case displacement list
```

```
: Note: There is no LIMIT operand because the limit value is calculated from
: the number of entries specified in the case displacement list.
```

```
: Note: These MACRO definitions place BASE after SELECTOR and DISPL so that
: BASE can be omitted when keywords are not used in the MACRO invocation.
:--
```

```
.MACRO $CASEB SELECTOR,DISPL,BASE=#0
$CASE SELECTOR,<DISPL>,BASE,TYPE=B
.ENDM $CASEB
```

```
.MACRO $CASEW SELECTOR,DISPL,BASE=#0
$CASE SELECTOR,<DISPL>,BASE,TYPE=W
.ENDM $CASEW
```

```
.MACRO $CASEL SELECTOR,DISPL,BASE=#0
$CASE SELECTOR,<DISPL>,BASE,TYPE=L
.ENDM $CASEL
```

```
:++
: $CASE is a support MACRO used by $CASEB, $CASEW, and $CASEL.
```

: \$CASE generates a CASE[B/W/L] instruction followed by the case displacement  
 : table. The parameters for the MACRO are:

```

: TYPE      = operand datatype of B, W, or L
: SELECTOR= the selector operand
: BASE      = the base operand
: DISPL     = the case displacement list

```

: Note: There is no LIMIT operand because the limit value is calculated from  
 : the number of entries specified in the case displacement list.

: Note: This MACRO definition places SELECTOR and DISPL ahead of BASE and TYPE  
 : so that the latter two can be omitted when keywords are not used in the  
 : MACRO invocation.

:--

```

.MACRO $CASE  SELECTOR,DISPL,BASE=#0,TYPE=B,?TABLE
$$COUNT=0
.IRP  EP,<DISPL>
$$COUNT=$$COUNT+1
.ENDR
.IF  EQ,$$COUNT
.ERROR ; ***** case displacement list is null ***** ;
.MEXIT
.ENDC
CASE'TYPE      SELECTOR,BASE,#<$$COUNT-1>

```

TABLE:

```

.IRP  EP,<DISPL>
.WORD EP-TABLE
.ENDR
.ENDM  $CASE

```

```

.END ; End of module

```

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

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 grid. Each window shows a different VAX/VMS command and its output. The windows are arranged in a 10x10 grid. Some windows have larger, bolded titles like 'FALACTION LIS', 'FALMACROS MAR', 'FALDEF MDL', 'FALACTINT LIS', 'FALACTMSG LIS', 'FALDAP10 LIS', 'FALBLDAB LIS', 'FALBLDATT LIS', and 'FALBLDSTS LIS'. The output in each window consists of text-based data, including lists, tables, and status information.