```
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
```

```
_$a
Va
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

FILEID**FALDAPCRC

```
FFFFFFFFFF    AAAAAA    LL          DDDDDDDD      AAAAAA    PPPPPPPP      CCCCCCCC  RRRRRRRR      CCCCCCCC
FFFFFFFFFF    AAAAAA    LL          DDDDDDDD      AAAAAA    PPPPPPPP      CCCCCCCC  RRRRRRRR      CCCCCCCC
FF           AA    AA   LL          DD      DD   AA    AA   PP      PP   CC        RR      RR  CC
FF           AA    AA   LL          DD      DD   AA    AA   PP      PP   CC        RR      RR  CC
FF           AA    AA   LL          DD      DD   AA    AA   PP      PP   CC        RR      RR  CC
FF           AA    AA   LL          DD      DD   AA    AA   PP      PP   CC        RR      RR  CC
FFFFFFF      AA    AA   LL          DD      DD   AA    AA   PPPPPPPP     CC        RRRRRRRR    CC
FFFFFFF      AA    AA   LL          DD      DD   AA    AA   PPPPPPPP     CC        RRRRRRRR    CC
FF           AAAAAAAAAA LL          DD      DD   AAAAAAAAAA PP           CC        RR  RR      CC
FF           AAAAAAAAAA LL          DD      DD   AAAAAAAAAA PP           CC        RR  RR      CC
FF           AA    AA   LL          DD      DD   AA    AA   PP           CC        RR    RR    CC                ....
FF           AA    AA   LL          DD      DD   AA    AA   PP           CC        RR    RR    CC                ....
FF           AA    AA   LLLLLLLLLL  DDDDDDDD    AA    AA   PP             CCCCCCCC  RR      RR    CCCCCCCC        ....
FF           AA    AA   LLLLLLLLLL  DDDDDDDD    AA    AA   PP             CCCCCCCC  RR      RR    CCCCCCCC        ....

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     1          .TITLE  FALDAPCRC - DAP LEVEL CRC
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:
0000    33  ;
0000    34  ;       This module contains the CRC table for DAP CRC checksum computation
0000    35  ;       plus routines related to CRC computation and error logging.
0000    36  ;
0000    37  ; Environment: VAX/VMS, user mode
0000    38  ;
0000    39  ; Author: James A. Krycka,      Creation Date:  16-JUN-1979
0000    40  ;
0000    41  ; Modified By:
0000    42  ;
0000    43  ;       V03-001 JAK0144         J A Krycka      11-APR-1984
0000    44  ;               Minor cleanup.
0000    45  ;
0000    46  ;--
```

```
                     0000     48          .SBTTL  DECLARATIONS
                 00000000     49          .PSECT  FAL$DATA                    SHR,NOEXE,RD,WRT,BYTE
                     0000     50 ;
                     0000     51 ; Include Files:
                     0000     52 ;
                     0000     53
                     0000     54          $DAPCRCDEF                          ; Define DAP CRC checksum symbols
                     0000     55          $EVCDEF                             ; Define event class symbols
                     0000     56          $FALWRKDEF                          ; Define FAL work area symbols
                     0000     57          $IODEF                              ; Define QIO function codes
                     0000     58          $NFBDEF                             ; Define Network Function Block symbols
                     0000     59          $NMADEF                             ; Define Network Management symbols
                     0000     60          $RAWDEF                             ; Define raw event record format
                     0000     61
                     0000     62 ;
                     0000     63 ; Macros:
                     0000     64 ;
                     0000     65 ;        None
                     0000     66 ;
                     0000     67 ; Equated Symbols:
                     0000     68 ;
                     0000     69 ;        None
                     0000     70 ;
                     0000     71 ; Own Storage:
                     0000     72 ;
                     0000     73
                     0000     74 FAL$GT_NODENAME::
                 00000007    0000     75          .BLKB   7                   ; FALMAIN will have stored nodename
                     0007     76                                              ;  of partner here
                     0007     77 NFB_DESC:                                    ;
                 00000005    0007     78          .LONG   5                   ; Size of NFB
                 0000000F'   000B     79          .LONG   NFB                 ; Address of NFB
                       1C    000F     80 NFB:     .BYTE   NFB$C_LOGEVENT      ; Network function block
                 00000000    0010     81          .LONG   0                   ;
                     0014     82 EVTBUF_DESC:                                 ;
                 00000028    0014     83          .LONG   40                  ; Length of event buffer
                 0000001C'   0018     84          .LONG   EVTBUF              ; Address of event buffer
                 00000044    001C     85 EVTBUF:  .BLKB   40                  ; Event buffer
```

```
                     0044    87          .SBTTL  FAL$CRC_TABLE - CRC POLYNOMIAL TABLE
                  00000000   88          .PSECT  FAL$CRC_TABLE   SHR,NOEXE,RD,NOWRT,LONG
                     0000    89
                     0000    90 ;++
                     0000    91 ; This is the CRC table for use in DAP CRC checksum computation.
                     0000    92 ; The CRC polynomial function (order 16) used is:
                     0000    93 ;
                     0000    94 ;       X**16 + X**15 + X**13 + X**7 + X**4 + X**2 + X**1 + 1
                     0000    95 ;--
                     0000    96
                     0000    97 FAL$CRC_TABLE::                             ; CRC polynomial table
00000000             0000    98          .LONG   DAP$K_CRC_TBL0             ;  Table entry 0
000053E3             0004    99          .LONG   DAP$K_CRC_TBL1             ;  Table entry 1
0000A7C6             0008   100          .LONG   DAP$K_CRC_TBL2             ;  Table entry 2
0000F425             000C   101          .LONG   DAP$K_CRC_TBL3             ;  Table entry 3
00009D87             0010   102          .LONG   DAP$K_CRC_TBL4             ;  Table entry 4
0000CE64             0014   103          .LONG   DAP$K_CRC_TBL5             ;  Table entry 5
00003A41             0018   104          .LONG   DAP$K_CRC_TBL6             ;  Table entry 6
000069A2             001C   105          .LONG   DAP$K_CRC_TBL7             ;  Table entry 7
0000E905             0020   106          .LONG   DAP$K_CRC_TBL8             ;  Table entry 8
0000BAE6             0024   107          .LONG   DAP$K_CRC_TBL9             ;  Table entry 9
00004EC3             0028   108          .LONG   DAP$K_CRC_TBLA             ;  Table entry 10
00001D20             002C   109          .LONG   DAP$K_CRC_TBLB             ;  Table entry 11
00007482             0030   110          .LONG   DAP$K_CRC_TBLC             ;  Table entry 12
00002761             0034   111          .LONG   DAP$K_CRC_TBLD             ;  Table entry 13
0000D344             0038   112          .LONG   DAP$K_CRC_TBLE             ;  Table entry 14
000080A7             003C   113          .LONG   DAP$K_CRC_TBLF             ;  Table entry 15
                     0040   114
```

```
                     0040   116          .SBTTL  FAL$CRC_LOGERR - LOG DAP CRC ERROR
                00000000    117          .PSECT  FAL$CODE                NOSHR,EXE,RD,NOWRT,BYTE
                     0000   118
                     0000   119 ;++
                     0000   120 ;       FAL$CRC_LOGERR logs DAP CRC errors to the DECnet Event Logger.
                     0000   121 ;
                     0000   122 ; Calling Sequence:
                     0000   123 ;
                     0000   124 ;       BSBW    FAL$CRC_LOGERR
                     0000   125 ;
                     0000   126 ; Input Parameters:
                     0000   127 ;
                     0000   128 ;       R8      FAL work area address
                     0000   129 ;
                     0000   130 ; Implicit Inputs:
                     0000   131 ;
                     0000   132 ;       FAL$W_LNKCHN
                     0000   133 ;
                     0000   134 ; Output Parameters:
                     0000   135 ;
                     0000   136 ;       None
                     0000   137 ;
                     0000   138 ; Implicit Outputs:
                     0000   139 ;
                     0000   140 ;       None
                     0000   141 ;
                     0000   142 ; Completion Codes:
                     0000   143 ;
                     0000   144 ;       None
                     0000   145 ;
                     0000   146 ; Side Effects:
                     0000   147 ;
                     0000   148 ;       An attempt is made to log a DAP CRC error to the DEC net Event Logger.
                     0000   149 ;       Return status of the request is neither checked nor returned.
                     0000   150 ;
                     0000   151 ;--
                     0000   152
                     0000   153 FAL$CRC_LOGERR::                         ; Entry point
                     0000   154
            3F   BB  0000   155         PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save registers
                     0002   156
                     0002   157 ;
                     0002   158 ; Build message in the event buffer.
                     0002   159 ;
                     0002   160
55   001C'CF   9E    0002   161         MOVAB   W^EVTBUF,R5             ; Get address of event buffer
          22   81    0007   162         ADDB3   #RAWST_DATA+4,-         ; Byte count for event
50   0000'CF         0009   163                 W^FAL$GT_NODENAME,R0     ;
     85   50   9B    000D   164         MOVZBW  R0,(R5)+                ; Put event count in event buffer
0014'CF   50   9A    0010   165         MOVZBL  R0,W^EVTBUF_DESC        ; Update the event buffer descriptor
                     0015   166         $GETTIM_S -                     ; Get the system time
                     0015   167                 TIMADR=(R5)             ; Put time in the event buffer
     55   08   C0    001E   168         ADDL2   #8,R5                   ; Bump the event buffer pointer
85   2001 8F   B0    0021   169         MOVW    #EVC$C_VMS_DPC,(R5)+    ; Put in the event code
  85   FF 8F   90    0026   170         MOVB    #EVC$C_SRC_NON,(R5)+    ; Put in the source type
     55   11   C0    002A   171         ADDL2   #17,R5                  ; Skip source ID field
     85   00   B0    002D   172         MOVW    #EVC$C_VMS_PNOD,(R5)+   ; Remote node name
```

```
          85     40 8F     90  0030          MOVB    #NMA$M_PTY_ASC,(R5)+     ; Data type
       54   0000'CF        9E  0034          MOVAB   W^FAL$GT_NODENAME,R4     ; Get the address of the nodename
               85    64    90  0039          MOVB    (R4),(R5)+               ; Put in the nodename count
               50    64    9A  003C          MOVZBL  (R4),R0                  ; Get nodename count as a longword
    65    01 A4    50      28  003F          MOVC3   R0,1(R4),(R5)            ; Put in the nodename
                               0044          $QIO_S -                         ;
                               0044                  CHAN=FAL$W_LNKCHN(R8)-   ; Use already assigned channel
                               0044                  FUNC=#IO$_ACPCONTROL-    ; Function code
                               0044                  P1=NFB_DESC-             ; Address of descriptor of NFB
                               0044                  P2=#EVTBUF_DESC          ; Address of descriptor of event buffer
                               0068
                               0068    ;
                               0068    ; Do not bother to check the return status since we wish to preserve the
                               0068    ; original DAP CRC error status in R0 (and not the success or failure of our
                               0068    ; attempt to log it).
                               0068    ;
                               0068
                     3F    BA  0068          POPR    #^M<R0,R1,R2,R3,R4,R5>   ; Restore registers
                           05  006A          RSB                              ; Return with status on entry
                               006B
                               006B          .END                             ; End of module
```

```
$$T1                    = 00000001
DAP$K_CRC_TBL0          = 00000000
DAP$K_CRC_TBL1          = 000053E3
DAP$K_CRC_TBL2          = 0000A7C6
DAP$K_CRC_TBL3          = 0000F425
DAP$K_CRC_TBL4          = 00009D87
DAP$K_CRC_TBL5          = 0000CE64
DAP$K_CRC_TBL6          = 00003A41
DAP$K_CRC_TBL7          = 000069A2
DAP$K_CRC_TBL8          = 0000E905
DAP$K_CRC_TBL9          = 0000BAE6
DAP$K_CRC_TBLA          = 00004EC3
DAP$K_CRC_TBLB          = 00001D20
DAP$K_CRC_TBLC          = 00007482
DAP$K_CRC_TBLD          = 00002761
DAP$K_CRC_TBLE          = 0000D344
DAP$K_CRC_TBLF          = 000080A7
EVC$C_SRC_NON           = 000000FF
EVC$C_VMS_DPC           = 00002001
EVC$C_VMS_PNOD          = 00000000
EVTBUF                    0000001C R      01
EVTBUF_DESC               00000014 R      01
FAL$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$CRC_LOGERR          00000000 RG     04
FAL$CRC_TABLE           00000000 RG     03
FAL$C_WRKBLN            00002000
FAL$GT_NODENAME         00000000 RG     01
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$W_DAPBUFSIZ         0000001A
FAL$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
IO$_ACPCONTROL          =  00000038
NFB                       0000000F R      01
NFB$C_LOGEVENT          =  0000001C
NFB_DESC                  00000007 R      C1
NMA$M_PTY_ASC           =  00000040
RAW$B_SRCTYP              0000000C
RAW$C_SIZE                0000001F
RAW$K_SIZE                0000001F
RAW$T_DATA                0000001E
RAW$T_SRCID               0000000D
RAW$T_SYSTIM              00000002
RAW$W_BYTES               00000000
RAW$W_EVTCODE             0000000A
```

```
SYS$GETTIM                      ******** GX   04
SYS$QIO                         ******** GX   04
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
FAL$DATA	00000044 (68.)	01 (1.)	NOPIC USR CON REL LCL SHR NOEXE RD WRT NOVEC BYTE
ABS	00002000 (8192.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
FAL$CRC_TABLE	00000040 (64.)	03 (3.)	NOPIC USR CON REL LCL SHR NOEXE RD NOWRT NOVEC LONG
FAL$CODE	0000006B (107.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:02.41
Command processing	141	00:00:00.68	00:00:06.21
Pass 1	523	00:00:13.17	00:00:52.56
Symbol table sort	0	00:00:01.99	00:00:05.37
Pass 2	48	00:00:01.91	00:00:03.19
Symbol table output	14	00:00:00.07	00:00:00.12
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	759	00:00:17.88	00:01:09.89

The working set limit was 1950 pages.
103965 bytes (204 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2215 non-local and 0 local symbols.
193 source lines were read in Pass 1, producing 17 object records in Pass 2.
19 pages of virtual memory were used to define 18 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	2
_255DUA28:[SHRLIB]NMALIBRY.MLB;1	0
_255DUA28:[SHRLIB]EVCDEF.MLB;1	2
_255DUA28:[FAL.OBJ]FAL.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	15

2394 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:FALDAPCRC/OBJ=OBJ$:FALDAPCRC MSRC$:FALDAPCRC/UPDATE=(ENH$:FALDAPCRC)+LIB$:FAL/LIB+SHRLIB$:EVCDEF/LIB+SHRLIB$:NMALIBRY
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FALACTION
LIS

FALDAPCRC
LIS

FALMACROS
MAR

FALBLDXAB
LIS

FALBLDATT
LIS

FALBLDSTS
LIS

FALDEF
MDL

FALDAPIO
LIS

FALACTINI
LIS

FALACTMSG
LIS