D 16

```
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFF                 111111           111111         AAA         AAA
FFF                 111111           111111         AAA         AAA
FFF                 111111           111111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                 111111111        111111111      AAA         AAA
FFF                 111111111        111111111      AAA         AAA
FFF                 111111111        111111111      AAA         AAA
```

```
**FILE**ID**DELFIL                                            C  2                                                              DIRA
                                                                                                                                V04
```

```
DDDDDDDD    EEEEEEEEEE  LL          FFFFFFFFFF    IIIIII    LL
DDDDDDDD    EEEEEEEEEE  LL          FFFFFFFFFF    IIIIII    LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EEEEEEEE    LL          FFFFFFFF        II      LL
DD      DD  EEEEEEEE    LL          FFFFFFFF        II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL
DD      DD  EE          LL          FF              II      LL              ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  FF            IIIIII    LLLLLLLLLL      ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  FF            IIIIII    LLLLLLLLLL      ....


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
```

```
0 MODULE DELFIL (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 1
1 !
1 ! ABSTRACT:
1 !
1 !       This module deletes a file, returning its blocks to the storage map
1 !       and releasing the file header.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  4-Apr-1977  15:50
1 !
1 ! MODIFIED BY:
1 !
1 !       A0101   ACG26369        Andrew C. Goldstein,    28-Dec-1979  15:38
1 !               Fix multi-header file interlock bug
1 !
1 !       A0100   ACG00001        Andrew C. Goldstein, 10-Oct-1978  20:02
1 !       Previous revision history moved to F11A.REV
1 !**
```

```
;    58          0058  1
;    59          0059  1
;    60          0060  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    61          0061  1 REQUIRE 'SRC$:FCPDEF.B32';
;    62          0376  1
;    63          0377  1
;    64          0378  1 FORWARD ROUTINE
;    65          0379  1         DELETE_FILE     : NOVALUE,      ! complete file deletion
;    66          0380  1         DELETE_FID      : NOVALUE;      ! just release file header
```

```
1 GLOBAL ROUTINE DELETE_FILE (FIB, FILEHEADER) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine deletes a file by releasing its blocks to the storage
1 !       bitmap and then releasing the header.
1 !
1 ! CALLING SEQUENCE:
1 !       DELETE_FILE (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: FIB of operation
1 !       ARG2: address of file header buffer
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       File deleted, storage map and index file bitmap modified, VCB modified
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FIB             : REF BBLOCK,   ! address of user FIB
2       FILEHEADER      : REF BBLOCK;   ! address of file header
2
2 LOCAL
2       HEADER          : REF BBLOCK,   ! local address of file header
2       FCB             : REF BBLOCK,   ! FCB of header in process
2       FILE_NUMBER,                    ! file number of header being deleted
2       MAP_AREA        : REF BBLOCK,   ! address of file header map area
2       EXT_FID         : BBLOCK [FID$C_LENGTH], ! extension file ID
2       EX_SEGNUM;                      ! header extension segment number
2
2 EXTERNAL
2       CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
2       FILE_HEADER     : REF BBLOCK;   ! global file header address
2
2 EXTERNAL ROUTINE
2       SEARCH_FCB,                     ! search FCB list for FCB
2       MARK_DIRTY,                     ! mark buffer for write-back
2       MARKDEL_FCB,                    ! mark FCB for deletion
2       CHECKSUM,                       ! compute file header checksum
2       FLUSH_FID,                      ! flush file from buffer pool
```

```
2               WRITE_BLOCK,                    ! write block to disk
2               INVALIDATE,                     ! invalidate block buffer
2               TRUNCATE_HEADER,                ! truncate file header
2               NEXT_HEADER;                    ! read next file extension header
2
2
2 ! If the file looks like a directory file flush it from the buffer pool
2 ! to avoid retaining stale directory data.
2 !
2
2
2 HEADER = .FILEHEADER;
2 IF .BBLOCK [HEADER[FH1$W_RECATTR], FAT$B_RTYPE] EQL FAT$C_FIXED
2 AND .BBLOCK [HEADER[FH1$W_RECATTR], FAT$W_RSIZE] EQL NMB$C_DIRENTRY
2 THEN FLUSH_FID (HEADER[FH1$W_FID]);
2
2 ! Loop for all headers, releasing the blocks mapped and the headers.
2 ! If this is an extension header, search the FCB list for the off chance
2 ! that this header is open as a file. If so, mark it for delete and get out.
2 ! First write out the deleted file header. Thus, if the system bombs during
2 ! the delete, we will not have a valid header on the disk mapping blocks
2 ! that may have been returned to the storage map.
2 !
2
2 WHILE 1 DO
3     BEGIN
3     MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
3     IF .MAP_AREA[FM1$B_EX_SEGNUM] NEQ 0
3     THEN
4         BEGIN
4         FCB = SEARCH_FCB (HEADER[FH1$W_FID]);
4         IF .FCB NEQ 0
4         THEN
5             BEGIN
5             HEADER[FH1$V_MARKDEL] = 1;
5             CHECKSUM (.HEADER);
5             MARK_DIRTY (.HEADER);
5             KERNEL_CALL (MARKDEL_FCB, .FCB);
5             RETURN;
4             END;
3         END;
3
3     FILE_NUMBER = .HEADER[FH1$W_FID_NUM];
3     HEADER[FH1$W_FID_NUM] = 0;          ! deleted header has zero file number
3     HEADER[FH1$W_CHECKSUM] = 0;         ! and zero checksum
3     FILE_HEADER = 0;
3     WRITE_BLOCK (.HEADER);
3     INVALIDATE (.HEADER);
3
3 ! Now return the blocks mapped by the header to the storage map.
3 ! Then extract the extension header data.
3 !
3
3     TRUNCATE_HEADER (.FIB, .HEADER, DEALLOC_BLOCKS);
3
3     EX_SEGNUM = .MAP_AREA[FM1$B_EX_SEGNUM] + 1;
3     EXT_FID[FID$W_NUM] = .MAP_AREA[FM1$W_EX_FILNUM];
```

```
:  182          0495  3      EXT_FID[FID$W_SEQ] = .MAP_AREA[FM1$W_EX_FILSEQ];
:  183          0496  3      EXT_FID[FID$W_RVN] = 0;
:  184          0497  3
:  185          0498  3 ! Now free the header in the index file bitmap. Then chain to the next header,
:  186          0499  3 ! if any, and repeat.
:  187          0500  3 !
:  188          0501  3
:  189          0502  3      DELETE_FID (.FILE_NUMBER);
:  190          0503  3
:  191          0504  3      IF .EXT_FID[FID$W_NUM] EQL 0 THEN EXITLOOP;
:  192          0505  3      HEADER = NEXT_HEADER (0, 0, EXT_FID, .EX_SEGNUM);
:  193          0506  2      END;
:  194          0507  2
:  195          0508  1 END;                                        ! end of routine DELETE_FILE


                                                        .TITLE   DELFIL
                                                        .IDENT   \V04-000\

                                                        .EXTRN   CLEANUP_FLAGS, FILE_HEADER
                                                        .EXTRN   SEARCH_FCB, MARK_DIRTY
                                                        .EXTRN   MARKDEL_FCB, CHECKSUM
                                                        .EXTRN   FLUSH_FID, WRITE_BLOCK
                                                        .EXTRN   INVALIDATE, TRUNCATE_HEADER
                                                        .EXTRN   NEXT_HEADER, SYS$CMKRNL

                                                        .PSECT   $CODE$,NOWRT,2

                                   007C 00000           .ENTRY   DELETE_FILE, Save R2,R3,R4,R5,R6          ; 0381
                   5E           08   C2 00002           SUBL2    #8, SP                                      ;
                   52      08   AC   D0 00005           MOVL     FILEHEADER, HEADER                          ; 0449
                   50      0E   A2   9E 00009           MOVAB    14(HEADER), R0                              ; 0450
                   01           60   91 0000D           CMPB     (R0), #1                                    ;
                                0E   12 00010           BNEQ     1$                                          ;
                   10      02   A0   B1 00012           CMPW     2(R0), #16                                  ; 0451
                                08   12 00016           BNEQ     1$                                          ;
                           02   A2   9F 00018           PUSHAB   2(HEADER)                                   ; 0452
            0000G  CF           01   FB 0001B           CALLS    #1, FLUSH_FID                               ;
                   50      01   A2   9A 00020   1$:     MOVZBL   1(HEADER), R0                               ; 0464
                   53         6240   3E 00024           MOVAW    (HEADER)[R0], MAP_AREA                      ;
                                63   95 00028           TSTB     (MAP_AREA)                                  ; 0465
                                32   13 0002A           BEQL     2$                                          ;
                           02   A2   9F 0002C           PUSHAB   2(HEADER)                                   ; 0468
            0000G  CF           01   FB 0002F           CALLS    #1, SEARCH_FCB                              ;
                   56           50   D0 00034           MOVL     R0, FCB                                     ;
                                25   13 00037           BEQL     2$                                          ; 0469
              0D   A2      80   8F   88 00039           BISB2    #128, 13(HEADER)                            ; 0472
                                52   DD 0003E           PUSHL    HEADER                                      ; 0473
            0000G  CF           01   FB 00040           CALLS    #1, CHECKSUM                                ;
                                52   DD 00045           PUSHL    HEADER                                      ; 0474
            0000G  CF           01   FB 00047           CALLS    #1, MARK_DIRTY                              ;
                                56   DD 0004C           PUSHL    FCB                                         ; 0475
                                01   DD 0004E           PUSHL    #1                                          ;
                                5E   DD 00050           PUSHL    SP                                          ;
                        0000G   CF   9F 00052           PUSHAB   MARKDEL_FCB                                 ;
        00000000G  9F           04   FB 00056           CALLS    #4, @#SYS$CMKRNL                            ;
                                04   04 0005D           RET                                                  ; 0471
```

```
DIRA
V04-

; Si
; Ru
; El
; Li
; Le
; Me
; Co
```

```
                   55        02    A2  3C 0005E 2$:     MOVZWL  2(HEADER), FILE_NUMBER                      : 0480
                             02    A2  B4 00062         CLRW    2(HEADER)                                   : 0481
                           01FE    C2  B4 00065         CLRW    510(HEADER)                                 : 0482
                          0000G    CF  D4 00069         CLRL    FILE_HEADER                                 : 0483
                                   52  DD 0006D         PUSHL   HEADER                                      : 0484
          0000G    CF              01  FB 0006F         CALLS   #1, WRITE_BLOCK                             :
                                   52  DD 00074         PUSHL   HEADER                                      : 0485
          0000G    CF              01  FB 00076         CALLS   #1, INVALIDATE                              :
                                   01  DD 0007B         PUSHL   #1                                          : 0491
                                   52  DD 0007D         PUSHL   HEADER                                      :
                             04    AC  DD 0007F         PUSHL   FIB                                         :
          0000G    CF              03  FB 00082         CALLS   #3, TRUNCATE_HEADER                         :
                   54              63  9A 00087         MOVZBL  (MAP_AREA), EX_SEGNUM                       : 0493
                                   54  D6 0008A         INCL    EX_SEGNUM                                   :
                   6E        02    A3  D0 0008C         MOVL    2(MAP_AREA), EXT_FID                        : 0494
                             04    AE  B4 00090         CLRW    EXT_FID+4                                   : 0496
                                   55  DD 00093         PUSHL   FILE_NUMBER                                 : 0502
          0000V    CF              01  FB 00095         CALLS   #1, DELETE_FID                              :
                                   6E  B5 0009A         TSTW    EXT_FID                                     : 0504
                                   12  13 0009C         BEQL    3$                                          :
                                   54  DD 0009E         PUSHL   EX_SEGNUM                                   : 0505
                             04    AE  9F 000A0         PUSHAB  EXT_FID                                     :
                                   7E  7C 000A3         CLRQ    -(SP)                                       :
          0000G    CF              04  FB 000A5         CALLS   #4, NEXT_HEADER                             :
                   52              50  D0 000AA         MOVL    R0, HEADER                                  :
                                 FF70  31 000AD         BRW     1$                                          : 0462
                                       04 000B0 3$:     RET                                                 : 0508

; Routine Size:  177 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE DELETE_FID (FILENUM) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine marks the indicated file header free in the index
1 !       file bitmap.
1 !
1 ! CALLING SEQUENCE:
1 !       DELETE_HEADER (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: file number of header
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       Header deleted - index file bitmap & VCB altered
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       FILE_NUMBER,                    ! file number - 1 of header
2       VBN,                            ! relative block in bitmap
2       BITPOS,                         ! bit number in bitmap
2       BUFFER          : REF BITVECTOR; ! bitmap buffer
2
2 EXTERNAL
2       CURRENT_VCB     : REF BBLOCK;   ! VCB of operation
2
2 EXTERNAL ROUTINE
2       READ_BLOCK,                     ! read a block from the disk
2       WRITE_BLOCK,                    ! write it back
2       UPDATE_IBVBN;                   ! update index file VBN in VCB
2
2
2 ! Deleting a file header consists of simply reading in the appropriate block
2 ! of the index file bitmap, zeroing the bit representing that file number,
2 ! and writing the block back out.
2 !
2
2 FILE_NUMBER = .FILENUM - 1;
2 VBN = .FILE_NUMBER<12,20>;
2 BITPOS = .FILE_NUMBER<0,12>;
```

```
DELFIL                                                         16-Sep-1984 00:57:10    VAX-11 Bliss-32 V4.0-742              Page   8
V04-000                                                        14-Sep-1984 12:29:28    DISK$VMSMASTER:[F11A.SRC]DELFIL.B32;1       (3)

;  254          0566  2
;  255          0567  2 IF .VBN GEQU .CURRENT_VCB[VCB$B_IBMAPSIZE]
;  256          0568  2 THEN BUG_CHECK (BADFID, FATAL, 'ACP file number out of range for this volume');
;  257          0569  2
;  258          0570  2 BUFFER = READ_BLOCK (.VBN + .CURRENT_VCB[VCB$L_IBMAPLBN], 1, INDEX_TYPE);
;  259          0571  2 BUFFER[.BITPOS] = 0;
;  260          0572  2 WRITE_BLOCK (.BUFFER);
;  261          0573  2
;  262          0574  2 ! If the bitmap block just written precedes the current start point for
;  263          0575  2 ! the bitmap scan, update the start point.
;  264          0576  2 !
;  265          0577  2
;  266          0578  2 IF .VBN LSSU .CURRENT_VCB[VCB$B_IBMAPVBN]
;  267          0579  2 THEN KERNEL_CALL (UPDATE_IBVBN, .VBN);
;  268          0580  2
;  269          0581  1 END;                                          ! end of routine DELETE_HEADER


                                                                         .EXTRN  CURRENT_VCB, READ_BLOCK
                                                                         .EXTRN  UPDATE_IBVBN, BUG$_BADFID

                                                       001C 00000        .ENTRY  DELETE_FID, Save R2,R3,R4                  ; 0509
                                 54      0000G  CF    9E 00002           MOVAB   CURRENT_VCB, R4                            ;
                  50       04    AC             01    C3 00007           SUBL3   #1, FILENUM, FILE_NUMBER                   ; 0563
        52        50             14             0C    EF 0000C           EXTZV   #12, #20, FILE_NUMBER, VBN                 ; 0564
        53        50             0C             00    EF 00011           EXTZV   #0, #12, FILE_NUMBER, BITPOS               ; 0565
                                 50             64    D0 00016           MOVL    CURRENT_VCB, R0                            ; 0567
        52   38   A0             08             00    ED 00019           CMPZV   #0, #8, 56(R0), VBN                        ;
                                                04    1A 0001F           BGTRU   1$                                         ;
                                                  FEFF 00021             BUGW                                               ; 0568
                                                 0000* 00023             .WORD   <BUG$_BADFID!4>                            ;
                                                03    DD 00025 1$:       PUSHL   #3                                         ; 0570
                                                01    DD 00027           PUSHL   #1                                         ;
                                 50             64    D0 00029           MOVL    CURRENT_VCB, R0                            ;
                                          30 B042     9F 0002C           PUSHAB  @48(R0)[VBN]                               ;
                           0000G CF             03    FB 00030           CALLS   #3, READ_BLOCK                             ;
                  00             60             53    E5 00035           BBCC    BITPOS, (BUFFER), 2$                       ; 0571
                                                50    DD 00039 2$:       PUSHL   BUFFER                                     ; 0572
                           0000G CF             01    FB 0003B           CALLS   #1, WRITE_BLOCK                            ;
                                 50             64    D0 00040           MOVL    CURRENT_VCB, R0                            ; 0578
        52   3A   A0             08             00    ED 00043           CMPZV   #0, #8, 58(R0), VBN                        ;
                                                11    1B 00049           BLEQU   3$                                         ;
                                                52    DD 0004B           PUSHL   VBN                                        ; 0579
                                                01    DD 0004D           PUSHL   #1                                         ;
                                                5E    DD 0004F           PUSHL   SP                                         ;
                                         0000G  CF    9F 00051           PUSHAB  UPDATE_IBVBN                               ;
                       00000000G  9F            04    FB 00055           CALLS   #4, @#SYS$CMKRNL                           ;
                                                      04 0005C 3$:       RET                                                ; 0581

; Routine Size:  93 bytes,    Routine Base:  $CODE$ + 00B1


;  270          0582  1
;  271          0583  1 END
;  272          0584  0 ELUDOM
;                                  PSECT SUMMARY
;
;      Name                         Bytes                        Attributes
;
;   $CODE$                               270  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                           Library Statistics
;
;                                               -------- Symbols --------      Pages        Processing
;      File                                     Total    Loaded   Percent      Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1               18619        21         0      1000           00:02.0




;                                  COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DELFIL/OBJ=OBJ$:DELFIL MSRC$:DELFIL/UPDATE=(ENH$:DELFIL)

; Size:          270 code + 0 data bytes
; Run Time:         00:09.6
; Elapsed Time:     00:25.7
; Lines/CPU Min:     3642
; Lexemes/CPU-Min: 13677
; Memory Used:  113 pages
; Compilation Complete
```

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTFCB
LIS

DELFIL
LIS

DIRGET
LIS

EXTIDX
LIS

IODONE
LIS

LOCKDN
LIS

ENTER
LIS

GETREQ
LIS

GETTIM
LIS

DISPAT
LIS

INIFCP
LIS

DIRFCB
LIS

EXTHDR
LIS

DIRSCN
LIS

LOGDEL
LIS

LOCKDB
LIS

FIND
LIS

GETFIB
LIS

DIRACC
LIS

INIFCB
LIS

EXTDIR
LIS

EXTEND
LIS