```
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
```

FILEID**CLENUP   H 9

```
   CCCCCCCC  LL          EEEEEEEEEE  NN        NN  UU        UU  PPPPPPPP
   CCCCCCCC  LL          EEEEEEEEEE  NN        NN  UU        UU  PPPPPPPP
 CC          LL          EE          NN        NN  UU        UU  PP        PP
 CC          LL          EE          NN        NN  UU        UU  PP        PP
 CC          LL          EE          NNNN      NN  UU        UU  PP        PP
 CC          LL          EE          NNNN      NN  UU        UU  PP        PP
 CC          LL          EEEEEEEE    NN  NN    NN  UU        UU  PPPPPPPP
 CC          LL          EEEEEEEE    NN  NN    NN  UU        UU  PPPPPPPP
 CC          LL          EE          NN    NNNN    UU        UU  PP
 CC          LL          EE          NN    NNNN    UU        UU  PP
 CC          LL          EE          NN        NN  UU        UU  PP                ....
 CC          LL          EE          NN        NN  UU        UU  PP                ....
   CCCCCCCC  LLLLLLLLLL  EEEEEEEEEE  NN        NN  UUUUUUUUUU    PP                ....
   CCCCCCCC  LLLLLLLLLL  EEEEEEEEEE  NN        NN  UUUUUUUUUU    PP                ....

 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
```

CL
V0

```
0 MODULE CLENUP (
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 performs the necessary cleanup after an operation.
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:  6-Jan-1977  23:53
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-002 ACG0346         Andrew C. Goldstein,    2-Aug-1983  15:16
1 !               Interface change to SEND_SYMBIONT routine
1 !
1 !       V03-001 LJK0199         Lawrence J. Kenah       27-Apr-1983
1 !               Do not credit FILCNT when giving back shared window
1 !
1 !       V02-008 LMP0005         L. Mark Pilant,         29-Dec-1981  14:00
```

```
1 !                     Properly cleanup any Cathedral windows.
1 !
1 !             V02-007 ACG0246         Andrew C. Goldstein,    22-Dec-1981  20:44
1 !                     Make /NOCACHE flush all caches
1 !
1 !             V02-006 ACG0245         Andrew C. Goldstein,    18-Dec-1981  18:29
1 !                     Send spool file to print during cleanup
1 !
1 !             V02-005 ACG0244         Andrew C. Goldstein,    18-Dec-1981  17:42
1 !                     Do buffer flush before deallocating control blocks
1 !
1 !             V02-004 ACG0167         Andrew C. Goldstein,    7-May-1980  18:47
1 !                     Previous revision history moved to F11A.REV
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 REQUIRE 'SRC$:FCPDEF.B32';
1
1
1 FORWARD ROUTINE
1         CLEANUP,                              ! normal cleanup
1         ZERO_WINDOWS,                         ! invalidate all windows of file
1         ERR_CLEANUP,                          ! cleanup after error
1         MAKE_DEACCESS,                        ! deaccess the file
1         DEL_EXTFCB,                           ! deallocate extension FCB's
1         ZERO_CHANNEL;                         ! zero user channel pointer
```

```
1 GLOBAL ROUTINE CLEANUP =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine performs the cleanup needed after a successfully
1 !       completed file operation.
1 !
1 ! CALLING SEQUENCE:
1 !       CLEANUP ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CLEANUP_FLAGS: indicate specific actions to do
1 !       PRIMARY_FCB: FCB of file
1 !       CURRENT_WINDOW: window of file
1 !       DIR_FCB: FCB of directory
1 !       DIR_WINDOW: window of directory
1 !       CURRENT_VCB: VCB of volume in process
1 !       IO_PACKET: I/O packet of request
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       FCB's and windows deleted when appropriate
1 !       header written
1 !       FCB updated
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       HEADER          : REF BBLOCK;   ! file header
2
2 EXTERNAL
2       CONTEXT_START,                  ! start of cleanup context area
2       CONTEXT_SAVE,                   ! start of context save area
2       CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
2       FILE_HEADER     : REF BBLOCK,   ! address of last file header read
2       CURRENT_FIB     : REF BBLOCK,   ! address of current FIB in use
2       PRIMARY_FCB     : REF BBLOCK,   ! FCB of file
2       CURRENT_WINDOW  : REF BBLOCK,   ! window of file
2       DIR_FCB         : REF BBLOCK,   ! FCB of directory
2       DIR_WINDOW      : REF BBLOCK,   ! window of directory
2       CURRENT_VCB     : REF BBLOCK,   ! VCB of volume
2       IO_PACKET       : REF BBLOCK;   ! I/O packet in process
```

```
2
2 EXTERNAL LITERAL
2         CONTEXT_SIZE;                     ! length of context area
2
2 EXTERNAL ROUTINE
2         FLUSH_BUFFERS,                    ! write all dirty buffers
2         FLUSH_FID,                        ! flush a file from buffer pool
2         READ_HEADER,                      ! read file header
2         INIT_FCB,                         ! initialize FCB
2         DEALLOCATE;                       ! deallocate dynamic memory
2
2
2 ! Switch back to the primary context area if necessary (no normal cleanup
2 ! is ever necessary on secondary context).
2 !
2
2 IF .CONTEXT_SAVE NEQ 0
2 THEN
3     BEGIN
3     CH$MOVE (CONTEXT_SIZE, CONTEXT_SAVE, CONTEXT_START);
3     CONTEXT_SAVE = 0;
2     END;
2
2 ! ***** Note: The primary header of the current file is not necessarily
2 ! resident at this point.
2 !
2 ! If the index file or storage map is write accessed, flush the buffer pool
2 ! of any of their blocks.
2 !
2
2 IF .CURRENT_VCB[VCB$V_WRITE_IF]
2 THEN FLUSH_FID (UPLIT WORD (1, 1, 0));
2 IF .CURRENT_VCB[VCB$V_WRITE_SM]
2 THEN FLUSH_FID (UPLIT WORD (2, 2, 0));
2
2 ! If the volume is mounted /NOCACHE, flush all buffers from the buffer
2 ! pool.
2 !
2
2 IF .CURRENT_VCB[VCB$V_NOCACHE]
2 THEN FLUSH_FID (0);
2
2 ! Flush all dirty buffers.
2 !
2
2 FLUSH_BUFFERS ();
2
2 ! If a directory FCB and window are left about with no use, dispose of them.
2 ! If the directory file is write accessed, flush the buffer pool of any
2 ! blocks that might be resident.
2 !
2
2 IF .DIR_FCB NEQ 0
2 THEN
3     BEGIN
3     IF .DIR_FCB[FCB$W_ACNT] EQL 0
3     THEN
```

```
;  200          0513 4          BEGIN
;  201          0514 4          IF NOT .DIR_FCB[FCB$V_DIR]
;  202          0515 4          THEN
;  203          0516 5              BEGIN
;  204          0517 5              KERNEL_CALL (DEALLOCATE, .DIR_FCB);
;  205          0518 5              DIR_FCB = 0;
;  206          0519 4              END;
;  207          0520 4          END
;  208          0521 3      ELSE
;  209          0522 4          BEGIN
;  210          0523 4          IF .DIR_FCB[FCB$W_WCNT] NEQ 0
;  211          0524 4          THEN FLUSH_FID (DIR_FCB[FCB$W_FID]);
;  212          0525 3          END;
;  213          0526 2      END;
;  214          0527 2
;  215          0528 2  IF .DIR_WINDOW NEQ 0
;  216          0529 2  THEN
;  217          0530 3      BEGIN
;  218          0531 3      KERNEL_CALL (DEALLOCATE, .DIR_WINDOW);
;  219          0532 3      DIR_WINDOW = 0;
;  220          0533 2      END;
;  221          0534 2
;  222          0535 2  ! If an FCB is left about with no use, dispose of it.
;  223          0536 2  !
;  224          0537 2
;  225          0538 2  IF .PRIMARY_FCB NEQ 0
;  226          0539 2  THEN
;  227          0540 3      BEGIN
;  228          0541 3      IF .PRIMARY_FCB[FCB$W_ACNT] EQL 0
;  229          0542 3      AND NOT .PRIMARY_FCB[FCB$V_DIR]
;  230          0543 3      THEN
;  231          0544 4          BEGIN
;  232          0545 4          KERNEL_CALL (DEALLOCATE, .PRIMARY_FCB);
;  233          0546 4          PRIMARY_FCB = 0;
;  234          0547 3          END;
;  235          0548 2      END;
;  236          0549 2
;  237          0550 2  ! Invalidate any windows on the file, if requested.
;  238          0551 2  !
;  239          0552 2
;  240          0553 2  IF TESTBITSC (CLEANUP_FLAGS[CLF_INVWINDOW])
;  241          0554 2  AND .PRIMARY_FCB NEQ 0
;  242          0555 2  THEN KERNEL_CALL (ZERO_WINDOWS, .PRIMARY_FCB);
;  243          0556 2
;  244          0557 2  RETURN 1;
;  245          0558 2
;  246          0559 1  END;                                    ! end of routine CLEANUP


                                                                    .TITLE  CLENUP
                                                                    .IDENT  \V04-000\

                                                                    .PSECT  $CODE$,NOWRT,2

                                        0000  0001  0001  00000 P.AAA:  .WORD   1, 1, 0                                   :
                                        0000  0002  0002  00006 P.AAB:  .WORD   2, 2, 0                                   :
```

```
                                                                  .EXTRN   CONTEXT_START, CONTEXT_SAVE
                                                                  .EXTRN   CLEANUP_FLAGS, FILE_HEADER
                                                                  .EXTRN   CURRENT_FIB, PRIMARY_FCB
                                                                  .EXTRN   CURRENT_WINDOW, DIR_FCB
                                                                  .EXTRN   DIR_WINDOW, CURRENT_VCB
                                                                  .EXTRN   IO_PACKET, CONTEXT_SIZE
                                                                  .EXTRN   FLUSH_BUFFERS, FLUSH_FID
                                                                  .EXTRN   READ_HEADER, INIT_FCB
                                                                  .EXTRN   DEALLOCATE, SYS$CMKRNL

                                                 0FFC 00000       .ENTRY   CLEANUP, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0399
                                                                           R11                                          ;
                           5B       0000G   CF   9E 00002         MOVAB    CONTEXT_SAVE, R11                            ;
                           5A       0000G   CF   9E 00007         MOVAB    DEALLOCATE, R10                              ;
                           59       0000G   CF   9E 0000C         MOVAB    CURRENT_VCB, R9                              ;
                           58       0000G   CF   9E 00011         MOVAB    PRIMARY_FCB, R8                              ;
                           57       0000G   CF   9E 00016         MOVAB    FLUSH_FID, R7                                ;
                           56 00000000G     9F   9E 0001B         MOVAB    @#SYS$CMKRNL, R6                             ;
                                    6B      D5 00022              TSTL     CONTEXT_SAVE                                 ; 0472
                                    0A      13 00024              BEQL     1$                                           ;
      0000G   CF           6B       0000G   8F   28 00026         MOVC3    #CONTEXT_SIZE, CONTEXT_SAVE, CONTEXT_START   ; 0475
                                    6B      D4 0002E              CLRL     CONTEXT_SAVE                                 ; 0476
                           50               69   D0 00030 1$:     MOVL     CURRENT_VCB, R0                              ; 0486
                           06       0B      A0   E9 00033         BLBC     11(R0), 2$                                   ;
                                    BA      AF   9F 00037         PUSHAB   P.AAA                                        ; 0487
                           67               01   FB 0003A         CALLS    #1, FLUSH_FID                                ;
                           50               69   D0 0003D 2$:     MOVL     CURRENT_VCB, R0                              ; 0488
              06    0B     A0               01   E1 00040         BBC      #1, 11(R0), 3$                               ;
                                    B2      AF   9F 00045         PUSHAB   P.AAB                                        ; 0489
                           67               01   FB 00048         CALLS    #1, FLUSH_FID                                ;
                           50               69   D0 0004B 3$:     MOVL     CURRENT_VCB, R0                              ; 0495
              05    53     A0               01   E1 0004E         BBC      #1, 83(R0), 4$                               ;
                                            7E   D4 00053         CLRL     -(SP)                                        ; 0496
                           67               01   FB 00055         CALLS    #1, FLUSH_FID                                ;
                    0000G  CF               00   FB 00058 4$:     CALLS    #0, FLUSH_BUFFERS                            ; 0501
                           50       0000G   CF   D0 0005D         MOVL     DIR_FCB, R0                                  ; 0508
                                            25   13 00062         BEQL     6$                                           ;
                                    1A      A0   B5 00064         TSTW     26(R0)                                       ; 0511
                                            15   12 00067         BNEQ     5$                                           ;
                           1C       22      A0   E8 00069         BLBS     34(R0), 6$                                   ; 0514
                                            50   DD 0006D         PUSHL    R0                                           ; 0517
                                            01   DD 0006F         PUSHL    #1                                           ;
                                    4400    8F   BB 00071         PUSHR    #^M<R10,SP>                                  ;
                           66               04   FB 00075         CALLS    #4, SYS$CMKRNL                               ;
                                    0000G   CF   D4 00078         CLRL     DIR_FCB                                      ; 0518
                                            0B   11 0007C         BRB      6$                                           ; 0511
                                    1C      A0   B5 0007E 5$:     TSTW     28(R0)                                       ; 0523
                                            06   13 00081         BEQL     6$                                           ;
                                    24      A0   9F 00083         PUSHAB   36(R0)                                       ; 0524
                           67               01   FB 00086         CALLS    #1, FLUSH_FID                                ;
                           50       0000G   CF   D0 00089 6$:     MOVL     DIR_WINDOW, R0                               ; 0528
                                            0F   13 0008E         BEQL     7$                                           ;
                                            50   DD 00090         PUSHL    R0                                           ; 0531
                                            01   DD 00092         PUSHL    #1                                           ;
                                    4400    8F   BB 00094         PUSHR    #^M<R10,SP>                                  ;
                           66               04   FB 00098         CALLS    #4, SYS$CMKRNL                               ;
                                    0000G   CF   D4 0009B         CLRL     DIR_WINDOW                                   ; 0532
```

```
                        50            68  D0 0009F 7$:     MOVL    PRIMARY_FCB, R0                          ; 0538
                                      16  13 000A2         BEQL    8$                                       ;
                                1A    A0  B5 000A4         TSTW    26(R0)                                   ; 0541
                                      11  12 000A7         BNEQ    8$                                       ;
                        0D      22    A0  E8 000A9         BLBS    34(R0), 8$                               ; 0542
                                      50  DD 000AD         PUSHL   R0                                       ; 0545
                                      01  DD 000AF         PUSHL   #1                                       ;
                              4400    8F  BB 000B1         PUSHR   #^M<R10,SP>                              ;
                        66            04  FB 000B5         CALLS   #4, SYS$CMKRNL                           ;
                                      68  D4 000B8         CLRL    PRIMARY_FCB                              ; 0546
          11     0000G  CF            04  E5 000BA 8$:     BBCC    #4, CLEANUP_FLAGS, 9$                    ; 0553
                                      68  D5 000C0         TSTL    PRIMARY_FCB                              ; 0554
                                      0D  13 000C2         BEQL    9$                                       ;
                                      68  DD 000C4         PUSHL   PRIMARY_FCB                              ; 0555
                                      01  DD 000C6         PUSHL   #1                                       ;
                                      5E  DD 000C8         PUSHL   SP                                       ;
                              0000V   CF  9F 000CA         PUSHAB  ZERO_WINDOWS                             ;
                        66            04  FB 000CE         CALLS   #4, SYS$CMKRNL                           ;
                        50            01  D0 000D1 9$:     MOVL    #1, R0                                   ; 0557
                                          04 000D4         RET                                              ; 0559

; Routine Size:  213 bytes,    Routine Base:  $CODE$ + 000C
```

```
1 GLOBAL ROUTINE ZERO_WINDOWS (FCB) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine invalidates all windows currently in use on the
1 !       indicated FCB. This routine must be executed in kernel mode.
1 !
1 ! CALLING SEQUENCE:
1 !       ZERO_WINDOWS (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of FCB
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_WINDOW: address of caller's window, if any
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       all windows marked empty, caller's turned
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FCB             : REF BBLOCK;
2
2 LOCAL
2       P               : REF BBLOCK,   ! window pointer
2       DUMMY,                          ! dummy sturage for REMQUE return
2       WINDOW_SEGMENT  : REF BBLOCK,   ! pointer to window segment
2       NEXT_SEGMENT    : REF BBLOCK;   ! pointer to window after next one
2
2
2 EXTERNAL ROUTINE
2       DEALLOCATE;                     ! deallocate dynamic memory
2
2 ! Loop through the window list off the FCB, zeroing all the retrieval pointer
2 ! counts. Then turn the user's window to VBN 1 if it exists.
2 !
2
2 P = .FCB[FCB$L_WLFL];
2
2 UNTIL .P EQL FCB[FCB$L_WLFL] DO
3     BEGIN
3     P[WCB$W_NMAP] = 0;
3     WINDOW_SEGMENT = .P[WCB$L_LINK];
```

```
;   305         0617 3      UNTIL .WINDOW_SEGMENT EQL 0
;   306         0618 3      DO
;   307         0619 4          BEGIN
;   308         0620 4          NEXT_SEGMENT = .WINDOW_SEGMENT[WCB$L_LINK];
;   309         0621 4          REMQUE (.WINDOW_SEGMENT, DUMMY);
;   310         0622 4          DEALLOCATE (.WINDOW_SEGMENT);
;   311         0623 4          WINDOW_SEGMENT = .NEXT_SEGMENT;
;   312         0624 3          END;
;   313         0625 3      P[WCB$L_LINK] = 0;
;   314         0626 3      P[WCB$V_COMPLETE] = 0;
;   315         0627 3      P = .P[WCB$L_WLFL];
;   316         0628 2      END;
;   317         0629 2
;   318         0630 2 ! ***** Note: When handling of window misses goes into its final form,
;   319         0631 2 ! this routine must also scan the I/O queue on the UCB and look for I/O
;   320         0632 2 ! into the blocks just deallocated. All such requests must be yanked out
;   321         0633 2 ! of the queue and routed to the ACP for error processing.
;   322         0634 2
;   323         0635 2 RETURN 1;
;   324         0636 2
;   325         0637 1 END;                                     ! end of routine ZERO_WINDOWS



                                            003C 00000          .ENTRY  ZERO_WINDOWS, Save R2,R3,R4,R5              ; 0560
                          50        04  AC  D0 00002          MOVL    FCB, R0                                     ; 0611
                          52        10  A0  D0 00006          MOVL    16(R0), P
              50     04   AC            10  C1 0000A 1$:      ADDL3   #16, FCB, R0                                ; 0613
                          50            52  D1 0000F          CMPL    P, R0
                                        28  13 00012          BEQL    4$
                                    16  A2  B4 00014          CLRW    22(P)                                       ; 0615
                          53        20  A2  D0 00017          MOVL    32(P), WINDOW_SEGMENT                       ; 0616
                                        13  13 0001B 2$:      BEQL    3$                                          ; 0617
                          54        20  A3  D0 0001D          MOVL    32(WINDOW_SEGMENT), NEXT_SEGMENT            ; 0620
                          55            63  0F 00021          REMQUE  (WINDOW_SEGMENT), DUMMY                     ; 0621
                                        53  DD 00024          PUSHL   WINDOW_SEGMENT                              ; 0622
                    00000G  CF          01  FB 00026          CALLS   #1, DEALLOCATE
                          53            54  D0 0002B          MOVL    NEXT_SEGMENT, WINDOW_SEGMENT                ; 0623
                                        EB  11 0002E          BRB     2$                                          ; 0617
                                    20  A2  D4 00030 3$:      CLRL    32(P)                                       ; 0625
                     0B   A2            20  8A 00033          BICB2   #32, 11(P)                                  ; 0626
                          52            62  D0 00037          MOVL    (P), P                                      ; 0627
                                        CE  11 0003A          BRB     1$                                          ; 0613
                          50            01  D0 0003C 4$:      MOVL    #1, R0                                      ; 0635
                                            04 0003F          RET                                                 ; 0637

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 00E1
;       1 GLOBAL ROUTINE ERR_CLEANUP =
;       1
;       1 !++
;       1 !
;       1 ! FUNCTIONAL DESCRIPTION:
;       1 !
;       1 !       This routine performs the cleanup needed after a file
;       1 !       operation that has terminated in an error.
;       1 !
;       1 ! CALLING SEQUENCE:
;       1 !       ERR_CLEANUP ()
;       1 !
;       1 ! INPUT PARAMETERS:
;       1 !       NONE
;       1 !
;       1 ! IMPLICIT INPUTS:
;       1 !       CLEANUP_FLAGS: indicate specific actions to do
;       1 !
;       1 ! OUTPUT PARAMETERS:
;       1 !       NONE
;       1 !
;       1 ! IMPLICIT OUTPUTS:
;       1 !       NONE
;       1 !
;       1 ! ROUTINE VALUE:
;       1 !       NONE
;       1 !
;       1 ! SIDE EFFECTS:
;       1 !       file deaccessed if necessary
;       1 !       channel window pointer cleared
;       1 !
;       1 !--
;       1
;       2 BEGIN
;       2
;       2 EXTERNAL
;       2       PMS_SUB_NEST,                   ! depth count on subfunction metering
;       2       CONTEXT_START,                  ! start of active context area
;       2       CONTEXT_SAVE,                   ! start of context save area
;       2       CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
;       2       UNREC_COUNT,                    ! count of unrecorded but allocated blocks
;       2       UNREC_LBN,                      ! LBN of above
;       2       NEW_FID,                        ! file number of unrecorded file ID
;       2       USER_STATUS     : VECTOR,       ! user I/O status block
;       2       SUPER_FID       : BBLOCK,       ! file ID of superseded file
;       2       SECOND_FIB      : BBLOCK,       ! FIB for secondary file operation
;       2       CURRENT_FIB     : REF BBLOCK,   ! pointer to FIB currently in use
;       2       FILE_HEADER     : REF BBLOCK,   ! current file header
;       2       PRIMARY_FCB     : REF BBLOCK,   ! FCB of this file
;       2       CURRENT_WINDOW  : REF BBLOCK,   ! window for this file
;       2       IO_PACKET       : REF BBLOCK,   ! I/O packet for this operation
;       2       DIR_RECORD;                     ! record number of directory entry
;       2
;       2 EXTERNAL LITERAL
;       2       CONTEXT_SIZE;                   ! length of context area
;       2
;       2 EXTERNAL ROUTINE
```

```
2               PMS_END_SUB,                    ! end metering of current subfunction
2               DEALLOCATE,                     ! deallocate dynamic memory
2               SEND_SYMBIONT,                  ! send file to job controller
2               DIRGET,                         ! read directory record
2               DIRPUT,                         ! write directory record
2               DELETE_FILE,                    ! delete a file
2               DELETE_FID,                     ! delete a file number
2               RETURN_BLOCKS,                  ! return blocks to storage map
2               TRUNCATE,                       ! file truncate routine
2               INVALIDATE,                     ! invalidate a buffer
2               READ_HEADER,                    ! read file header
2               INIT_FCB,                       ! initialize FCB
2               UPDATE_FCB,                     ! update FCB contents
2               NEXT_HEADER,                    ! read next extension file header
2               MAKE_EXTFCB,                    ! create extension FCB
2               CHECKSUM,                       ! checksum file header
2               REMAP_FILE;                     ! rebuild the windows for a file
2
2
2 ! If a subfunction was being executed, turn off metering now.
2 !
2
2 IF .PMS_SUB_NEST NEQ 0
2 THEN
3     BEGIN
3     PMS_SUB_NEST = 1;
3     PMS_END_SUB ();
2     END;
2
2 ! We repeat the entire procedure twice if a secondary file operation was
2 ! in progress (indicated by non-zero saved context).
2 !
2
2 WHILE 1 DO
3 BEGIN
3
3 ! Locals are declared here to prevent their scope from extending around the
3 ! entire main loop and raising havoc with register assignment.
3 !
3
3 LOCAL
3               HEADER          : REF BBLOCK,   ! address of file header
3               FCB             : REF BBLOCK,   ! FCB pointer
3               WINDOW_SEGMENT  : REF BBLOCK,   ! address of the next window segment
3               NEXT_SEGMENT    : REF BBLOCK,   ! address of one beyond the next window
3               RECADDR         : REF BBLOCK,   ! address of directory record
3               T1,                             ! random temps
3               T2,
3               T3;
3
3 ! Deaccess the file if requested.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_DEACCESS])
3 THEN KERNEL_CALL (MAKE_DEACCESS);
3
3 ! Deallocate the window block if called for.
```

```
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_DELWINDOW])
3 THEN
3     IF .CURRENT_WINDOW NEQ 0
3     THEN
4         BEGIN
4         WINDOW_SEGMENT = .CURRENT_WINDOW;
4         DO
5             BEGIN
5             NEXT_SEGMENT = .WINDOW_SEGMENT[WCB$L_LINK];
5             KERNEL_CALL (DEALLOCATE, .WINDOW_SEGMENT);
5             WINDOW_SEGMENT = .NEXT_SEGMENT;
5             END
4         UNTIL .WINDOW_SEGMENT EQL 0;
4         CURRENT_WINDOW = 0;
3         END;
3
3 ! Clean out the window pointer in the user's channel if necessary.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_ZCHANNEL])
3 THEN KERNEL_CALL (ZERO_CHANNEL);
3
3 ! If there is a file header resident, it probably needs to be checksummed,
3 ! except in the case of a failed truncate, where we discard the header.
3 ! Then read back the primary file header.
3 !
3
3 HEADER = .FILE_HEADER;
3 IF .HEADER NEQ 0
3 THEN
4     BEGIN
4     IF .CLEANUP_FLAGS[CLF_CLEANTRUNC]
4     THEN INVALIDATE (.FILE_HEADER)
4     ELSE CHECKSUM (.FILE_HEADER);
5     HEADER = READ_HEADER ((IF .CURRENT_FIB NEQ 0
5                            THEN CURRENT_FIB[FIB$W_FID]
4                            ELSE 0),
4                           .PRIMARY_FCB);
3     END;
3
3 ! Send the file to the job controller if it is to be spooled.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_DOSPOOL])
3 THEN SEND_SYMBIONT (.HEADER, .PRIMARY_FCB);
3
3 ! If a directory entry needs to be re-entered, do so.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_REENTER])
3 THEN
4     BEGIN
4     RECADDR = DIRGET (.DIR_RECORD, 0);
4     CH$MOVE (FIB$S_FID, SUPER_FID, RECADDR[NMB$W_FID]);
4     DIRPUT (.RECADDR);
```

```
4      CLEANUP_FLAGS[CLF_REMOVE] = 0;
3      END;
3
3  ! If a directory entry needs to be removed, do so.
3  !
3
3  IF TESTBITSC (CLEANUP_FLAGS[CLF_REMOVE])
3  THEN
4      BEGIN
4      RECADDR = DIRGET (.DIR_RECORD, 0);
4      RECADDR[NMB$W_FID_NUM] = 0;
4      DIRPUT (.RECADDR);
3      END;
3
3  ! If there are unrecorded blocks allocated from the storage map, return them.
3  !
3
3  IF .UNREC_COUNT NEQ 0
3  THEN
4      BEGIN
4      RETURN_BLOCKS (.UNREC_LBN, .UNREC_COUNT);
4      UNREC_COUNT = 0;
3      END;
3
3  ! If a file deletion is called for, do it. This is either a create that
3  ! failed later on, or a real delete.
3  !
3
3  IF TESTBITSC (CLEANUP_FLAGS[CLF_DELFILE])
3  THEN
4      BEGIN
4      CLEANUP_FLAGS[CLF_TRUNCATE] = 0;    ! no truncate necessary after a delete
4      CLEANUP_FLAGS[CLF_DELFID] = 0;      ! leave header behind if failure
4      DELETE_FILE (.CURRENT_FIB, .HEADER);
3      END;
3
3  ! If an extend operation failed, truncate the file.
3  !
3
3  IF TESTBITSC (CLEANUP_FLAGS[CLF_TRUNCATE])
3  THEN
4      BEGIN
4      T1 = .CURRENT_FIB[FIB$L_EXSZ];      ! save the data returned by EXTEND
4      T2 = .CURRENT_FIB[FIB$L_EXVBN];     ! so it won't be smashed by TRUNCATE
4      T3 = .USER_STATUS[1];
4      CURRENT_FIB[FIB$L_EXSZ] = 0;
4      TRUNCATE (.CURRENT_FIB, .HEADER, DEALLOC_BLOCKS);
4      HEADER = .FILE_HEADER;                          ! follow buffer shuffling
4      CURRENT_FIB[FIB$L_EXSZ] = .T1;
4      CURRENT_FIB[FIB$L_EXVBN] = .T2;
4      USER_STATUS[1] = .T3;
4      CLEANUP_FLAGS[CLF_INVWINDOW] = 0;   ! windows were never extended, so no need
4      CLEANUP_FLAGS[CLF_CLEANTRUNC] = 0;
4      CHECKSUM (.HEADER);
3      END;
3
3  ! If a truncate has failed, redo the operation to produce a correct file
```

```
3 ! header, but don't return blocks to the storage map. We assume the header
3 ! was nfg and contained bogus retrieval pointers.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_CLEANTRUNC])
3 THEN
4     BEGIN
4     CURRENT_FIB[FIB$L_EXSZ] = 0;
4     TRUNCATE (.CURRENT_FIB, .HEADER, 0);
4     HEADER = .FILE_HEADER;                      ! follow buffer shuffling
3     END;
3
3 ! Various errors leave the file control block screwed up. If needed,
3 ! rebuild it and its extensions from scratch.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_FIXFCB])
3 AND .PRIMARY_FCB NEQ 0
3 AND .HEADER NEQ 0
3 THEN
3     IF .PRIMARY_FCB[FCB$V_DIR]
3     OR .PRIMARY_FCB[FCB$W_ACNT] NEQ 0
3     THEN
4         BEGIN
4         FCB = .PRIMARY_FCB;
4         KERNEL_CALL (DEL_EXTFCB, .FCB);
4         KERNEL_CALL (INIT_FCB, .FCB, .HEADER);
4         WHILE 1 DO
5             BEGIN
5             HEADER = NEXT_HEADER (.HEADER, .FCB);
5             IF .HEADER EQL 0 THEN EXITLOOP;
5             FCB = KERNEL_CALL (MAKE_EXTFCB, .HEADER, .FCB, 1);
4             END;
4         IF .FCB NEQ .PRIMARY_FCB
4         THEN
5             BEGIN
5             HEADER = READ_HEADER (0, .PRIMARY_FCB);
5             KERNEL_CALL (UPDATE_FCB, .HEADER);
4             END;
3         END;
3
3 ! Clean up any Cathedral windows which have broken.
3 !
3
3 IF TESTBITSC (CLEANUP_FLAGS[CLF_REMAP]) THEN REMAP_FILE ();
3
3 ! If there is a dangling file ID (from a partial create or header extension),
3 ! dispose of it.
3 !
3
3 IF .NEW_FID NEQ 0
3 THEN DELETE_FID (.NEW_FID);
3
3 ! Copy the saved context, if any back into the primary context and repeat
3 ! the cleanup.
3 !
3
```

```
;  612          0923  3 IF .CONTEXT_SAVE EQL 0 THEN EXITLOOP;
;  613          0924  3 CH$MOVE (CONTEXT_SIZE, CONTEXT_SAVE, CONTEXT_START);
;  614          0925  3 CONTEXT_SAVE = 0;
;  615          0926  3
;  616          0927  2 END;                                    ! end of major loop
;  617          0928  2
;  618          0929  2 RETURN 1;
;  619          0930  2
;  620          0931  1 END;                                    ! end of routine ERR_CLEANUP


                                                           .EXTRN  PMS_SUB_NEST, UNREC_COUNT
                                                           .EXTRN  UNREC_LBN, NEW_FID
                                                           .EXTRN  USER_STATUS, SUPER_FID
                                                           .EXTRN  SECOND_FIB, DIR_RECORD
                                                           .EXTRN  PMS_END_SUB, SEND_SYMBIONT
                                                           .EXTRN  DIRGET, DIRPUT, DELETE_FILE
                                                           .EXTRN  DELETE_FID, RETURN_BLOCKS
                                                           .EXTRN  TRUNCATE, INVALIDATE
                                                           .EXTRN  UPDATE_FCB, NEXT_HEADER
                                                           .EXTRN  MAKE_EXTFCB, CHECKSUM
                                                           .EXTRN  REMAP_FILE

                                           0FFC 00000      .ENTRY  ERR_CLEANUP, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0638
                                                                   R10,R11                                     ;
                          5B      0000G  CF   9E 00002     MOVAB   CURRENT_FIB, R11                            ;
                          5A      0000G  CF   9E 00007     MOVAB   PRIMARY_FCB, R10                            ;
                          59 00000000G  9F   9E 0000C     MOVAB   @#SYS$CMKRNL, R9                            ;
                          58      0000G  CF   9E 00013     MOVAB   CLEANUP_FLAGS, R8                           ;
                                  0000G  CF   D5 00018     TSTL    PMS_SUB_NEST                                ; 0717
                                         0A   13 0001C     BEQL    1$                                          ;
                 0000G   CF              01   D0 0001E     MOVL    #1, PMS_SUB_NEST                            ; 0720
                 0000G   CF              00   FB 00023     CALLS   #0, PMS_END_SUB                             ; 0721
            0B            68             10   E5 00028 1$: BBCC    #16, CLEANUP_FLAGS, 2$                      ; 0748
                                         7E   D4 0002C     CLRL    -(SP)                                       ; 0749
                                         5E   DD 0002E     PUSHL   SP                                          ;
                                  0000V  CF   9F 00030     PUSHAB  MAKE_DEACCESS                               ;
                          69             03   FB 00034     CALLS   #3, SYS$CMKRNL                              ;
            24            68             1A   E5 00037 2$: BBCC    #26, CLEANUP_FLAGS, 4$                      ; 0754
                          50      0000G  CF   D0 0003B     MOVL    CURRENT_WINDOW, R0                          ; 0756
                                         1D   13 00040     BEQL    4$                                          ;
                          52             50   D0 00042     MOVL    R0, WINDOW_SEGMENT                          ; 0759
                          53        20   A2   D0 00045 3$: MOVL    32(WINDOW_SEGMENT), NEXT_SEGMENT            ; 0762
                                         52   DD 00049     PUSHL   WINDOW_SEGMENT                              ; 0763
                                         01   DD 0004B     PUSHL   #1                                          ;
                                         5E   DD 0004D     PUSHL   SP                                          ;
                                  0000G  CF   9F 0004F     PUSHAB  DEALLOCATE                                  ;
                          69             04   FB 00053     CALLS   #4, SYS$CMKRNL                              ;
                          52             53   D0 00056     MOVL    NEXT_SEGMENT, WINDOW_SEGMENT                ; 0764
                                         EA   12 00059     BNEQ    3$                                          ; 0766
                                  0000G  CF   D4 0005B     CLRL    CURRENT_WINDOW                              ; 0767
            0B            68             11   E5 0005F 4$: BBCC    #17, CLEANUP_FLAGS, 5$                      ; 0773
                                         7E   D4 00063     CLRL    -(SP)                                       ; 0774
                                         5E   DD 00065     PUSHL   SP                                          ;
                                  0000V  CF   9F 00067     PUSHAB  ZERO_CHANNEL                                ;
                          69             03   FB 0006B     CALLS   #3, SYS$CMKRNL                              ;
```

```
                50    0000G  CF  D0 0006E  5$:    MOVL    FILE_HEADER, R0                          ; 0781
                57           50  D0 00073         MOVL    R0, HEADER                               ;
                             2D  13 00076         BEQL    10$                                      ; 0782
09       02     A8           03  E1 00078         BBC     #3, CLEANUP_FLAGS+2, 6$                  ; 0785
                             50  DD 0007D         PUSHL   R0                                       ; 0786
       0000G    CF           01  FB 0007F         CALLS   #1, INVALIDATE                           ;
                             07  11 00084         BRB     7$                                       ;
                             50  DD 00086  6$:    PUSHL   R0                                       ; 0787
       0000G    CF           01  FB 00088         CALLS   #1, CHECKSUM                             ;
                             6A  DD 0008D  7$:    PUSHL   PRIMARY_FCB                              ; 0791
                50           6B  D0 0008F         MOVL    CURRENT_FIB, R0                          ; 0788
                             07  13 00092         BEQL    8$                                       ;
                50           04  C0 00094         ADDL2   #4, R0                                   ; 0789
                             50  DD 00097         PUSHL   R0                                       ;
                             02  11 00099         BRB     9$                                       ;
                             7E  D4 0009B  8$:    CLRL    -(SP)                                    ;
       0000G    CF           02  FB 0009D  9$:    CALLS   #2, READ_HEADER                          ; 0788
                57           50  D0 000A2         MOVL    R0, HEADER                               ;
09              68           02  E5 000A5  10$:   BBCC    #2, CLEANUP_FLAGS, 11$                   ; 0797
                             6A  DD 000A9         PUSHL   PRIMARY_FCB                              ; 0798
                             57  DD 000AB         PUSHL   HEADER                                   ;
       0000G    CF           02  FB 000AD         CALLS   #2, SEND_SYMBIONT                        ;
20              68           17  E5 000B2  11$:   BBCC    #23, CLEANUP_FLAGS, 12$                  ; 0803
                             7E  D4 000B6         CLRL    -(SP)                                    ; 0806
                      0000G  CF  DD 000B8         PUSHL   DIR_RECORD                               ;
       0000G    CF           02  FB 000BC         CALLS   #2, DIRGET                               ;
                56           50  D0 000C1         MOVL    R0, RECADDR                              ;
66     0000G    CF           06  28 000C4         MOVC3   #6, SUPER_FID, (RECADDR)                 ; 0807
                             56  DD 000CA         PUSHL   RECADDR                                  ; 0808
       0000G    CF           01  FB 000CC         CALLS   #1, DIRPUT                               ;
         02     A8       40  8F  8A 000D1         BICB2   #64, CLEANUP_FLAGS+2                     ; 0809
17              68           16  E5 000D6  12$:   BBCC    #22, CLEANUP_FLAGS, 13$                  ; 0815
                             7E  D4 000DA         CLRL    -(SP)                                    ; 0818
                      0000G  CF  DD 000DC         PUSHL   DIR_RECORD                               ;
       0000G    CF           02  FB 000E0         CALLS   #2, DIRGET                               ;
                56           50  D0 000E5         MOVL    R0, RECADDR                              ;
                             66  B4 000E8         CLRW    (RECADDR)                                ; 0819
                             56  DD 000EA         PUSHL   RECADDR                                  ; 0820
       0000G    CF           01  FB 000EC         CALLS   #1, DIRPUT                               ;
                50    0000G  CF  D0 000F1  13$:   MOVL    UNREC_COUNT, R0                          ; 0826
                             0F  13 000F6         BEQL    14$                                      ;
                             50  DD 000F8         PUSHL   R0                                       ; 0829
                      0000G  CF  DD 000FA         PUSHL   UNREC_LBN                                ;
       0000G    CF           02  FB 000FE         CALLS   #2, RETURN_BLOCKS                        ;
                      0000G  CF  D4 00103         CLRL    UNREC_COUNT                              ; 0830
0D              68           15  E5 00107  14$:   BBCC    #21, CLEANUP_FLAGS, 15$                  ; 0837
         02     A8           14  8A 0010B         BICB2   #20, CLEANUP_FLAGS+2                     ; 0841
                             57  DD 0010F         PUSHL   HEADER                                   ; 0842
                             6B  DD 00111         PUSHL   CURRENT_FIB                              ;
       0000G    CF           02  FB 00113         CALLS   #2, DELETE_FILE                          ;
41              68           12  E5 00118  15$:   BBCC    #18, CLEANUP_FLAGS, 16$                  ; 0848
                50           6B  D0 0011C         MOVL    CURRENT_FIB, R0                          ; 0851
                54       18  A0  D0 0011F         MOVL    24(R0), T1                               ;
                53       1C  A0  D0 00123         MOVL    28(R0), T2                               ; 0852
                52    0000G  CF  D0 00127         MOVL    USER_STATUS+4, T3                        ; 0853
                         18  A0  D4 0012C         CLRL    24(R0)                                   ; 0854
                             01  DD 0012F         PUSHL   #1                                       ; 0855
```

```
                           0081    8F   BB 00131        PUSHR   #^M<R0,R7>                                ;      :
               0000G  CF           03   FB 00135        CALLS   #3, TRUNCATE                              ;      :
                      57   0000G   CF   D0 0013A        MOVL    FILE_HEADER, HEADER                       ; 0856 :
                      50           6B   D0 0013F        MOVL    CURRENT_FIB, R0                           ; 0857 :
                  18  A0           54   D0 00142        MOVL    T1, 24(R0)                                ;      :
                  1C  A0           53   D0 00146        MOVL    T2, 28(R0)                                ; 0858 :
               0000G  CF           52   D0 0014A        MOVL    T3, USER_STATUS+4                         ; 0859 :
                      68 00080010  8F   CA 0014F        BICL2   #524304, CLEANUP_FLAGS                    ; 0861 :
                                   57   DD 00156        PUSHL   HEADER                                    ; 0862 :
               0000G  CF           01   FB 00158        CALLS   #1, CHECKSUM                              ;      :
   16                 68           13   E5 0015D 16$:   BBCC    #19, CLEANUP_FLAGS, 17$                   ; 0870 :
                      50           6B   D0 00161        MOVL    CURRENT_FIB, R0                           ; 0873 :
                             18    A0   D4 00164        CLRL    24(R0)                                    ;      :
                                   7E   D4 00167        CLRL    -(SP)                                     ; 0874 :
                           0081    8F   BB 00169        PUSHR   #^M<R0,R7>                                ;      :
               0000G  CF           03   FB 0016D        CALLS   #3, TRUNCATE                              ;      :
                      57   0000G   CF   D0 00172        MOVL    FILE_HEADER, HEADER                       ; 0875 :
   75                 68           01   E5 00177 17$:   BBCC    #1, CLEANUP_FLAGS, 21$                    ; 0882 :
                                   6A   D5 0017B        TSTL    PRIMARY_FCB                               ; 0883 :
                                   71   13 0017D        BEQL    21$                                       ;      :
                                   57   D5 0017F        TSTL    HEADER                                    ; 0884 :
                                   6D   13 00181        BEQL    21$                                       ;      :
                      50           6A   D0 00183        MOVL    PRIMARY_FCB, R0                           ; 0886 :
                      05     22    A0   E8 00186        BLBS    34(R0), 18$                               ;      :
                             1A    A0   B5 0018A        TSTW    26(R0)                                    ; 0887 :
                                   61   13 0018D        BEQL    21$                                       ;      :
                      52           50   D0 0018F 18$:   MOVL    R0, FCB                                   ; 0890 :
                                   52   DD 00192        PUSHL   FCB                                       ; 0891 :
                                   01   DD 00194        PUSHL   #1                                        ;      :
                                   5E   DD 00196        PUSHL   SP                                        ;      :
                           0000V   CF   9F 00198        PUSHAB  DEL_EXTFCB                                ;      :
                      69           04   FB 0019C        CALLS   #4, SYS$CMKRNL                            ;      :
                           0084    8F   BB 0019F        PUSHR   #^M<R2,R7>                                ; 0892 :
                                   02   DD 001A3        PUSHL   #2                                        ;      :
                                   5E   DD 001A5        PUSHL   SP                                        ;      :
                           0000G   CF   9F 001A7        PUSHAB  INIT_FCB                                  ;      :
                      69           05   FB 001AB        CALLS   #5, SYS$CMKRNL                            ;      :
                                   52   DD 001AE 19$:   PUSHL   FCB                                       ; 0895 :
                                   57   DD 001B0        PUSHL   HEADER                                    ;      :
               0000G  CF           02   FB 001B2        CALLS   #2, NEXT_HEADER                           ;      :
                      57           50   D0 001B7        MOVL    R0, HEADER                                ;      :
                                   16   13 001BA        BEQL    20$                                       ; 0896 :
                                   01   DD 001BC        PUSHL   #1                                        ; 0897 :
                                   52   DD 001BE        PUSHL   FCB                                       ;      :
                                   57   DD 001C0        PUSHL   HEADER                                    ;      :
                                   03   DD 001C2        PUSHL   #3                                        ;      :
                                   5E   DD 001C4        PUSHL   SP                                        ;      :
                           0000G   CF   9F 001C6        PUSHAB  MAKE_EXTFCB                               ;      :
                      69           06   FB 001CA        CALLS   #6, SYS$CMKRNL                            ;      :
                      52           50   D0 001CD        MOVL    R0, FCB                                   ;      :
                                   DC   11 001D0        BRB     19$                                       ; 0893 :
                      6A           52   D1 001D2 20$:   CMPL    FCB, PRIMARY_FCB                          ; 0899 :
                                   19   13 001D5        BEQL    21$                                       ;      :
                                   6A   DD 001D7        PUSHL   PRIMARY_FCB                               ; 0902 :
                                   7E   D4 001D9        CLRL    -(SP)                                     ;      :
               0000G  CF           02   FB 001DB        CALLS   #2, READ_HEADER                           ;      :
                      57           50   D0 001E0        MOVL    R0, HEADER                                ;      :
```

```
                                        57  DD 001E3          PUSHL   HEADER                                         ; 0903
                                        01  DD 001E5          PUSHL   #1                                             ;
                                        5E  DD 001E7          PUSHL   SP                                             ;
                               0000G    CF  9F 001E9          PUSHAB  UPDATE_FCB                                     ;
                          69            04  FB 001ED          CALLS   #4, SYS$CMKRNL                                 ;
          05              68            1D  E5 001F0 21$:     BBCC    #29, CLEANUP_FLAGS, 22$                        ; 0910
                 0000G    CF            00  FB 001F4          CALLS   #0, REMAP_FILE                                 ;
                          50   0000G    CF  D0 001F9 22$:     MOVL    NEW_FID, R0                                    ; 0916
                                        07  13 001FE          BEQL    23$                                            ;
                                        50  DD 00200          PUSHL   R0                                             ; 0917
                 0000G    CF            01  FB 00202          CALLS   #1, DELETE_FID                                 ;
                               0000G    CF  D5 00207 23$:     TSTL    CONTEXT_SAVE                                   ; 0923
                                        11  13 0020B          BEQL    24$                                            ;
   0000G    CF   0000G    CF   0000G    8F  28 0020D          MOVC3   #CONTEXT_SIZE, CONTEXT_SAVE, CONTEXT_START     ; 0924
                               0000G    CF  D4 00217          CLRL    CONTEXT_SAVE                                   ; 0925
                                      FE0A  31 0021B          BRW     1$                                             ; 0728
                          50            01  D0 0021E 24$:     MOVL    #1, R0                                         ; 0929
                                            04 00221          RET                                                    ; 0931

; Routine Size:  546 bytes,    Routine Base:  $CODE$ + 0121
```

```
1 ROUTINE MAKE_DEACCESS =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine performs the machinery for deaccessing a file.
1 !
1 ! CALLING SEQUENCE:
1 !       MAKE_DEACCESS ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       PRIMARY_FCB: FCB of file
1 !       CURRENT_WINDOW: window of file
1 !       CURRENT_VCB: VCB of volume in process
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       file deaccessed
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       WINDOW_SEGMENT  : REF BBLOCK,   ! address of the next window segment
2       DUMMY;                          ! dummy local to receive REMQUE
2
2 EXTERNAL
2       PRIMARY_FCB     : REF BBLOCK,   ! FCB of file
2       CURRENT_WINDOW  : REF BBLOCK,   ! window of file
2       CURRENT_VCB     : REF BBLOCK,   ! VCB of volume
2       PMS$GL_OPEN     : ADDRESSING_MODE (ABSOLUTE);
2                                       ! system count of currently open files
2
2
2 ! Unlink the window from the FCB. Clear the applicable access conditions
2 ! in the FCB.
2 !
2
2 WINDOW_SEGMENT = .CURRENT_WINDOW;
2 DO
3     BEGIN
3     IF .WINDOW_SEGMENT[WCB$L_WLFL] NEQ 0 THEN REMQUE (.WINDOW_SEGMENT, DUMMY);
3     WINDOW_SEGMENT = .WINDOW_SEGMENT[WCB$L_LINK];
3     END
```

```
;  679           0989 2 UNTIL .WINDOW_SEGMENT EQL 0;
;  680           0990 2
;  681           0991 2 IF .CURRENT_WINDOW[WCB$V_NOREAD]
;  682           0992 2 THEN PRIMARY_FCB[FCB$V_EXCL] = 0;
;  683           0993 2
;  684           0994 2 IF .CURRENT_WINDOW[WCB$V_NOWRITE]
;  685           0995 2 THEN PRIMARY_FCB[FCB$W_LCNT] = .PRIMARY_FCB[FCB$W_LCNT] - 1;
;  686           0996 2
;  687           0997 2 IF .CURRENT_WINDOW[WCB$V_NOTRUNC]
;  688           0998 2 THEN PRIMARY_FCB[FCB$W_TCNT] = .PRIMARY_FCB[FCB$W_TCNT] - 1;
;  689           0999 2
;  690           1000 2 ! For a write access, bump down the writer count. If this is the
;  691           1001 2 ! last write, and the file is the index file or the storage map, clear
;  692           1002 2 ! the appropriate flag in the VCB.
;  693           1003 2 !
;  694           1004 2
;  695           1005 2 IF .CURRENT_WINDOW[WCB$V_WRITE]
;  696           1006 2 THEN
;  697           1007 3     BEGIN
;  698           1008 3     PRIMARY_FCB[FCB$W_WCNT] = .PRIMARY_FCB[FCB$W_WCNT] - 1;
;  699           1009 3     IF .PRIMARY_FCB[FCB$W_WCNT] EQL 0
;  700           1010 3     THEN
;  701           1011 4         BEGIN
;  702           1012 4         IF .PRIMARY_FCB[FCB$W_FID_NUM] EQL 1
;  703           1013 4             THEN CURRENT_VCB[VCB$V_WRITE_IF] = 0;
;  704           1014 4         IF .PRIMARY_FCB[FCB$W_FID_NUM] EQL 2
;  705           1015 4             THEN CURRENT_VCB[VCB$V_WRITE_SM] = 0;
;  706           1016 3         END;
;  707           1017 2     END;
;  708           1018 2
;  709           1019 2 PRIMARY_FCB[FCB$W_ACNT] = .PRIMARY_FCB[FCB$W_ACNT] - 1;
;  710           1020 2
;  711           1021 2 ! If this was the last access, yank the FCB out of the FCB list and dump its
;  712           1022 2 ! extensions, if any.
;  713           1023 2 !
;  714           1024 2
;  715           1025 2 IF .PRIMARY_FCB[FCB$W_ACNT] EQL 0
;  716           1026 2 THEN
;  717           1027 3     BEGIN
;  718           1028 3     REMQUE (.PRIMARY_FCB, DUMMY);
;  719           1029 3     DEL_EXTFCB (.PRIMARY_FCB);
;  720           1030 2     END;
;  721           1031 2
;  722           1032 2 PMS$GL_OPEN = .PMS$GL_OPEN - 1;          ! bump down count of open files
;  723           1033 2 CURRENT_VCB[VCB$W_TRANS] = .CURRENT_VCB[VCB$W_TRANS] - 1;
;  724           1034 2
;  725           1035 2 RETURN 1;
;  726           1036 2
;  727           1037 1 END;                                     ! end of routine MAKE_DEACCESS


                                                                 .EXTRN  PMS$GL_OPEN

                                           001C 00000 MAKE_DEACCESS:
                                                                 .WORD   Save R2,R3,R4                              ; 0932
                               54     0000G  CF   9E 00002       MOVAB   CURRENT_VCB, R4                            ;
```

```
                        53      0000G  CF  9E 00007         MOVAB   PRIMARY_FCB, R3                                     ;
                        50      0000G  CF  D0 0000C         MOVL    CURRENT_WINDOW, WINDOW_SEGMENT                      ; 0983
                                       60  D5 00011 1$:     TSTL    (WINDOW_SEGMENT)                                    ; 0986
                                       03  13 00013         BEQL    2$                                                  ;
                        52             60  0F 00015         REMQUE  (WINDOW_SEGMENT), DUMMY                             ;
                        50         20  A0  D0 00018 2$:     MOVL    32(WINDOW_SEGMENT), WINDOW_SEGMENT                  ; 0987
                                       F3  12 0001C         BNEQ    1$                                                  ; 0989
                        51      0000G  CF  D0 0001E         MOVL    CURRENT_WINDOW, R1                                  ; 0991
           07      15   A1             02  E1 00023         BBC     #2, 21(R1), 3$                                      ;
                        50             63  D0 00028         MOVL    PRIMARY_FCB, R0                                     ; 0992
                   22   A0             08  8A 0002B         BICB2   #8, 34(R0)                                          ;
                        06         14  A1  E9 0002F 3$:     BLBC    20(R1), 4$                                          ; 0994
                        50             63  D0 00033         MOVL    PRIMARY_FCB, R0                                     ; 0995
                                   1E  A0  B7 00036         DECW    30(R0)                                              ;
           06      15   A1             03  E1 00039 4$:     BBC     #3, 21(R1), 5$                                      ; 0997
                        50             63  D0 0003E         MOVL    PRIMARY_FCB, R0                                     ; 0998
                                   20  A0  B7 00041         DECW    32(R0)                                              ;
           22      0B   A1             01  E1 00044 5$:     BBC     #1, 11(R1), 7$                                      ; 1005
                        50             63  D0 00049         MOVL    PRIMARY_FCB, R0                                     ; 1008
                                   1C  A0  B7 0004C         DECW    28(R0)                                              ;
                                       1A  12 0004F         BNEQ    7$                                                  ; 1009
                        01         24  A0  B1 00051         CMPW    36(R0), #1                                          ; 1012
                                       07  12 00055         BNEQ    6$                                                  ;
                        51             64  D0 00057         MOVL    CURRENT_VCB, R1                                     ; 1013
                   0B   A1             01  8A 0005A         BICB2   #1, 11(R1)                                          ;
                        02         24  A0  B1 0005E 6$:     CMPW    36(R0), #2                                          ; 1014
                                       07  12 00062         BNEQ    7$                                                  ;
                        50             64  D0 00064         MOVL    CURRENT_VCB, R0                                     ; 1015
                   0B   A0             02  8A 00067         BICB2   #2, 11(R0)                                          ;
                        50             63  D0 0006B 7$:     MOVL    PRIMARY_FCB, R0                                     ; 1019
                                   1A  A0  B7 0006E         DECW    26(R0)                                              ;
                                       0A  12 00071         BNEQ    8$                                                  ; 1025
                        52             60  0F 00073         REMQUE  (R0), DUMMY                                         ; 1028
                                       63  DD 00076         PUSHL   PRIMARY_FCB                                         ; 1029
                 0000V  CF             01  FB 00078         CALLS   #1, DEL_EXTFCB                                      ;
                    00000000G          9F  D7 0007D 8$:     DECL    @#PMS$GL_OPEN                                       ; 1032
                        50             64  D0 00083         MOVL    CURRENT_VCB, R0                                     ; 1033
                                   0C  A0  B7 00086         DECW    12(R0)                                              ;
                        50             01  D0 00089         MOVL    #1, R0                                              ; 1035
                                           04 0008C         RET                                                         ; 1037

; Routine Size:  141 bytes,    Routine Base:  $CODE$ + 0343
```

```
GLOBAL ROUTINE DEL_EXTFCB (START_FCB) =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine removes and deallocates all extension FCB's, if any,
!       linked to the indicated FCB.
!
! CALLING SEQUENCE:
!       DEL_EXTFCB (ARG1)
!
! INPUT PARAMETERS:
!       ARG1: address of primary FCB or 0
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       FCB's deallocated
!
!--

BEGIN

MAP
        START_FCB       : REF BBLOCK;   ! FCB argument

LOCAL
        FCB             : REF BBLOCK,   ! running FCB pointer
        NEXT_FCB        : REF BBLOCK,   ! next extension FCB
        DUMMY;                          ! dummy local to receive REMQUE

EXTERNAL ROUTINE
        DEALLOCATE;                     ! deallocate dynamic memory

! Checking for null pointers, find the first extension FCB. Follow the extension
! list and remove and deallocate the extension FCB's, cleaning out the pointers
! on the way.
!

IF .START_FCB EQL 0 THEN RETURN 1;
FCB = .START_FCB[FCB$L_EXFCB];
START_FCB[FCB$L_EXFCB] = 0;
UNTIL .FCB EQL 0 DO
    BEGIN
    NEXT_FCB = .FCB[FCB$L_EXFCB];
    FCB[FCB$L_EXFCB] = 0;
```

```
;  786          1095  3        REMQUE (.FCB, DUMMY);
;  787          1096  3        DEALLOCATE (.FCB);
;  788          1097  3        FCB = .NEXT_FCB;
;  789          1098  2        END;
;  790          1099  2
;  791          1100  2  RETURN 1;
;  792          1101  2
;  793          1102  1  END;                                   . end of routine DEL_EXTFCB



                                    001C 00000          .ENTRY  DEL_EXTFCB, Save R2,R3,R4                          ; 1038
                       50      04  AC  D0 00002          MOVL    START_FCB, R0                                      ; 1088
                                   21  13 00006          BEQL    2$                                                 ;
                       52      0C  A0  D0 00008          MOVL    12(R0), FCB                                        ; 1089
                               0C  A0  D4 0000C          CLRL    12(R0)                                             ; 1090
                                   52  D5 0000F  1$:     TSTL    FCB                                                ; 1091
                                   16  13 00011          BEQL    2$                                                 ;
                       53      0C  A2  D0 00013          MOVL    12(FCB), NEXT_FCB                                  ; 1093      ; R
                               0C  A2  D4 00017          CLRL    12(FCB)                                            ; 1094
                       54          62  0F 0001A          REMQUE  (FCB), DUMMY                                       ; 1095
                                   52  DD 0001D          PUSHL   FCB                                                ; 1096
               0000G   CF          01  FB 0001F          CALLS   #1, DEALLOCATE                                     ;
                       52          53  D0 00024          MOVL    NEXT_FCB, FCB                                      ; 1097
                                   E6  11 00027          BRB     1$                                                 ; 1091
                       50          01  D0 00029  2$:     MOVL    #1, R0                                             ; 1100
                                       04 0002C          RET                                                        ; 1102

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 03D0
```

```
1 ROUTINE ZERO_CHANNEL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine zeroes out the window pointer being returned to
1 !       the user for his channel control block. It also credits one to the
1 !       user's open file quota, except for the case of a shared window.
1 !       This routine must be executed in kernel mode.
1 !
1 ! CALLING SEQUENCE:
1 !       ZERO_CHANNEL ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       IO_PACKET: I/O packet of request
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       channel window pointer cleared, file quota bumped unless shared window
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
2                                       ! buffer descriptors
2       JIB             : REF BBLOCK,   ! Job information block
2       PCB             : REF BBLOCK;   ! address of user process control block
2
2 EXTERNAL
2       CURRENT_WINDOW  : REF BBLOCK,   ! window address of file
2       IO_PACKET       : REF BBLOCK,   ! I/O packet in process
2       SCH$GL_PCBVEC   : REF VECTOR ADDRESSING_MODE (ABSOLUTE);
2                                       ! system PCB vector
2
2
2                                       ! pointer to buffer descriptors
2 ABD = .BBLOCK [.IO_PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
2 ABD[ABD$C_WINDOW, ABD$W_COUNT] = 4;
2 .ABD[ABD$C_WINDOW, ABD$W_TEXT] + ABD[ABD$C_WINDOW, ABD$W_TEXT] + 1 = 0;
2
2 IF
3    BEGIN
3
```

```
;  852          1160  3        ! The FILCNT quota is credited if a WCB has not yet been allocated or
;  853          1161  3        ! if the SHRWCB bit is not set in the WCB.
;  854          1162  3
;  855          1163  3        IF .CURRENT_WINDOW EQL 0
;  856          1164  3        THEN 1
;  857          1165  3        ELSE NOT .CURRENT_WINDOW[WCB$V_SHRWCB]
;  858          1166  3        END
;  859          1167  2    THEN
;  860          1168  3        BEGIN
;  861          1169  3        PCB = .SCH$GL_PCBVEC[.(IO_PACKET[IRP$L_PID])<0,16>];
;  862          1170  3        JIB = .PCB[PCB$L_JIB];
;  863          1171  3        JIB[JIB$W_FILCNT] = .JIB[JIB$W_FILCNT] + 1;
;  864          1172  2        END;
;  865          1173  2
;  866          1174  2    RETURN 1;
;  867          1175  2
;  868          1176  1    END;                                     ! end of routine ZERO_CHANNEL


                                                                      .EXTRN   SCH$GL_PCBVEC

                                                     0004 00000 ZERO_CHANNEL:
                                                                      .WORD    Save R2                                    ; 1103
                         52      0000G   CF   D0 00002                MOVL     IO_PACKET, R2                              ; 1153
                         51         2C   B2   D0 00007                MOVL     @44(R2), ABD                               ;
                 02      A1              04   B0 0000B                MOVW     #4, 2(ABD)                                 ; 1154
                         50              61   3C 0000F                MOVZWL   (ABD), R0                                  ; 1155
                                  01 A140     9F 00012                PUSHAB   1(ABD)[R0]                                 ;
                                              9E D4 00016             CLRL     @(SP)+                                     ;
                         51      0000G   CF   D0 00018                MOVL     CURRENT_WINDOW, R1                         ; 1163
                                         05   13 0001D                BEQL     1$                                         ;
          17     0B      A1              03   E0 0001F                BBS      #3, 11(R1), 2$                             ; 1165
                         51  00000000G   9F   D0 00024 1$:            MOVL     @#SCH$GL_PCBVEC, R1                        ; 1169
                         50         0C   A2   3C 0002B                MOVZWL   12(R2), R0                                 ;
                         50           6140    D0 0002F                MOVL     (R1)[R0], PCB                              ;
                         50       0080   C0   D0 00033                MOVL     128(PCB), JIB                              ; 1170
                                    30   A0   B6 00038                INCW     48(JIB)                                    ; 1171
                         50              01   D0 0003B 2$:            MOVL     #1, R0                                     ; 1174
                                              04 0003E                RET                                                 ; 1176

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 03FD


;  869          1177  1
;  870          1178  1    END
;  871          1179  0    ELUDOM



;                                   PSECT SUMMARY
;
;        Name                     Bytes                        Attributes
;
```

```
;  $CODE$                          1084  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         39         0      1000           00:02.0



;                                          COMMAND QUALIFIERS

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

; Size:          1072 code + 12 data bytes
; Run Time:         00:24.4
; Elapsed Time:     00:54.8
; Lines/CPU Min:     2901
; Lexemes/CPU-Min: 14286
; Memory Used:  203 pages
; Compilation Complete
```

0164 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CHKSUM
LIS

ACPCNTRL
LIS

CHKPRO
LIS

FCPDEF
B32

DEACCS
LIS

BADSCN
LIS

CLENUP
LIS

CPYNAM
LIS

CHKHDR
LIS

COMMON
LIS

CREHDR
LIS

CREWIN
LIS

ALLOCB
LIS

ACCESS
LIS

CHKDMO
LIS

DELETE
LIS

CREATE
LIS

CREFCB
LIS