```
FFFFFFFFFFFFFFF        111               111        XXX         XXX
FFFFFFFFFFFFFFF        111               111        XXX         XXX
FFFFFFFFFFFFFFF        111               111        XXX         XXX
FFF                 111111            111111        XXX         XXX
FFF                 111111            111111        XXX         XXX
FFF                 111111            111111        XXX         XXX
FFF                    111               111           XXX   XXX
FFF                    111               111           XXX   XXX
FFF                    111               111           XXX   XXX
FFFFFFFF.FFF           111               111              XXX
FFFFFFFFFFFF           111               111              XXX
FFFFFFFFFFFF           111               111              XXX
FFF                    111               111           XXX   XXX
FFF                    111               111           XXX   XXX
FFF                    111               111           XXX   XXX
FFF                    111               111        XXX         XXX
FFF                    111               111        XXX         XXX
FFF                    111               111        XXX         XXX
FFF                 111111111         111111111     XXX         XXX
FFF                 111111111         111111111     XXX         XXX
FFF                 111111111         111111111     XXX         XXX
```

```
_$25

Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CC
L_CC
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP_
MARK
MARK
MARK
MARK
MARK
```

```
**FILE**ID**DELBAD                                  H 7                                                        DEL
                                                                                                               V04

DDDDDDDD    EEEEEEEEEE  LL          BBBBBBBB      AAAAAA    DDDDDDDD
DDDDDDDD    EEEEEEEEEE  LL          BBBBBBBB      AAAAAA    DDDDDDDD
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD
DD      DD  EEEEEEEE    LL          BBBBBBBB    AA      AA  DD      DD
DD      DD  EEEEEEEE    LL          BBBBBBBB    AA      AA  DD      DD
DD      DD  EE          LL          BB      BB  AAAAAAAAAA  DD      DD
DD      DD  EE          LL          BB      BB  AAAAAAAAAA  DD      DD
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD      ....
DD      DD  EE          LL          BB      BB  AA      AA  DD      DD      ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  BBBBBBBB    AA      AA  DDDDDDDD        ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  BBBBBBBB    AA      AA  DDDDDDDD        ....

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 DELBAD (
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 2
1 !
1 ! ABSTRACT:
1 !
1 !       This routine removes the indicated blocks from the given file header
1 !       and appends them to the bad block file.
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:  29-May-1978  22:43
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-006 CDS0004         Christian D. Saether    14-Aug-1984
1 !               Remove obsolete reference to update_filesize routine.
1 !
1 !       V03-005 CDS0003         Christian D. Saether    31-July-1984
1 !               Remove local definition of get_map_pointer linkage.
1 !
```

```
1 !       V03-004 CDS0002          Christian D. Saether     2-May-1984
1 !               Perform deallocation to bad block file in secondary
1 !               context.  Add appropriate serialization.
1 !
1 !       V03-003 CDS0001          Christian D. Saether    29-Dec-1983
1 !               Use L_NORM linkage and BIND_COMMON macro.
1 !
1 !       V03-002 ACG0367          Andrew C. Goldstein,    26-Oct-1983  19:49
1 !               Update BADBLK.SYS file highwater mark
1 !
1 !       V03-001 LMP0037          L. Mark Pilant,         28-Jun-1982  15:10
1 !               Remove the addressing mode module switch.
1 !
1 !       V02-003 ACG0230          Andrew C. Goldstein,    24-Dec-1981  0:16
1 !               Go to longword external addressing
1 !
1 !       V02-002 ACG0167          Andrew C. Goldstein,    16-Apr-1980  19:25
1 !               Previous revision history moved to F11B.REV
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
1 GLOBAL ROUTINE DEALLOCATE_BAD (FIB, FILE_HDR, POINTER, LAST_COUNT) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine removes the indicated blocks from the given file header
1 !       and appends them to the bad block file.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       DEALLOCATE_BAD (ARG1, ARG2, ARG3, ARG4)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of user FIB
1 !       ARG2: address of file header
1 !       ARG3: address of map pointer at which to start
1 !       ARG4: new value for last pointer block count
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 header updated, bad block log file updated, bad block file extended
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FIB             : REF BBLOCK,   ! user FIB argument
2       FILE_HDR        : REF BBLOCK;   ! address of file header
2
2 LINKAGE
2       L_MAKE_POINTER  = CALL :
2                         GLOBAL (BUILD_POINTER = 9);
2
2 GLOBAL REGISTER
2       COUNT           = 6,            ! count of blocks returned
2       LBN             = 7,            ! LBN of map entry
2       MAP_POINTER     = 8 : REF BBLOCK, ! pointer to scan map
2       BUILD_POINTER   = 9 : REF BBLOCK; ! pointer to build new map entry
2
2 LOCAL
2       HEADER          : REF BBLOCK,   ! local address of file header
2       EXT_HEADER      : REF BBLOCK;   ! address of extension header
2
2 BIND
```

```
        BB_FID = UPLIT WORD (BADBLK_FID, BADBLK_FID, 0);

BIND_COMMON;

EXTERNAL ROUTINE
        SAVE_CONTEXT     : L_NORM,          ! save primary context
        RESTORE_CONTEXT  : L_NORM,          ! restore primary context
        SERIAL_FILE      : L_NORM,          ! file serialization lock.
        RELEASE_SERIAL_LOCK : L_NORM,       ! relinquish file serialization
        WRITE_DIRTY      : L_NORM,          ! write modified buffers
        GET_MAP_POINTER  : L_MAP_POINTER,   ! get value of next map entry
        MAKE_POINTER     : L_MAKE_POINTER,  ! build new map entry
        NEXT_HEADER      : L_NORM,          ! read next extension header
        MARK_DIRTY       : L_NORM,          ! mark buffer for rewrite
        ZERO_WINDOWS     : L_NORM,          ! invalidate windows of file
        CHECKSUM         : L_NORM,          ! compute file header checksum
        READ_HEADER      : L_NORM,          ! read file header
        EXTEND_HEADER    : L_NORM,          ! create extention header
        SCAN_BADLOG      : L_NORM;          ! scan pending bad block log file


! Get into secondary context.
!

SAVE_CONTEXT ();

! Construct pointers into the file header and get the current contents of the
! last map pointer.
!

HEADER = .FILE_HDR;
MAP_POINTER = .POINTER;

GET_MAP_POINTER ();

! Now append the blocks to the bad block file.
!

LBN = .LBN + .LAST_COUNT;                   ! compute LBN of bad cluster
COUNT = .COUNT - .LAST_COUNT;

! Serialize on the bad block file.
!

PRIM_LCKINDX = SERIAL_FILE (BB_FID);

HEADER = READ_HEADER (BB_FID, 0);
WHILE 1 DO
    BEGIN
    EXT_HEADER = NEXT_HEADER (.HEADER, 0);
    IF .EXT_HEADER EQL 0 THEN EXITLOOP;
    HEADER = .EXT_HEADER;
    END;
MARK_DIRTY (.HEADER);
BUILD_POINTER = .HEADER + (.HEADER[FH2$B_MPOFFSET] + .HEADER[FH2$B_MAP_INUSE]) * 2;

IF NOT MAKE_POINTER (.COUNT, .LBN, .HEADER)
```

```
2 THEN
3     BEGIN
3     HEADER = EXTEND_HEADER (UPLIT BYTE (REP FIB$C_LENGTH OF (0)), .HEADER, 0);
3     BUILD_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET] * 2;
3     MAKE_POINTER (.COUNT, .LBN, .HEADER);
3     END;
2
2 BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK] =
2         ROT (ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + .COUNT, 16);
2
2 ! If this file header supports it, stuff the high water field to
2 ! be the allocated size.
2 !
2
2 IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
2 THEN
2     HEADER [FH2$L_HIGHWATER] =  ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + 1;
2
2 CHECKSUM (.HEADER);
2
2 ! Write the modified header(s), release the serialization lock, and return to
2 ! primary context.
2 !
2
2 WRITE_DIRTY (.LB_BASIS [.PRIM_LCKINDX]);
2
2 RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
2
2 RESTORE_CONTEXT ();
2
2 ! Finally, remove the bad block cluster from the volume pending bad block log
2 ! file, if it was there.
2 !
2
2 SCAN_BADLOG (0, 0, .LBN, REMOVE_BADBLOCK, .COUNT);
2
1 END;                                    ! end of routine DEALLOCATE_BAD


                                                        .TITLE  DELBAD
                                                        .IDENT  \V04-000\

                                                        .PSECT  $CODE$,NOWRT,2

                    0000  0003  0003  00000 P.AAA:      .WORD   3, 3, 0                                  ;
                                      00# 00006 P.AAB:  .BYTE   0[64]                                    ;

                                            BB_FID=             P.AAA
                                                        .EXTRN  SAVE_CONTEXT, RESTORE_CONTEXT
                                                        .EXTRN  SERIAL_FILE, RELEASE_SERIAL_LOCK
                                                        .EXTRN  WRITE_DIRTY, GET_MAP_POINTER
                                                        .EXTRN  MAKE_POINTER, NEXT_HEADER
                                                        .EXTRN  MARK_DIRTY, ZERO_WINDOWS
                                                        .EXTRN  CHECKSUM, READ_HEADER
                                                        .EXTRN  EXTEND_HEADER, SCAN_BADLOG

                                03C4 00000              .ENTRY  DEALLOCATE_BAD, Save R2,R6,R7,R8,R9           ; 1071
```

```
                 0000G  CF          00  FB 00002          CALLS   #0, SAVE_CONTEXT                          ; 1152
                        52     08   AC  D0 00007          MOVL    FILE_HDR, HEADER                          ; 1158
                        58     0C   AC  D0 0000B          MOVL    POINTER, MAP_POINTER                      ; 1159
                                 0000G  30 0000F          BSBW    GET_MAP_POINTER                           ; 1161
                        57     10   AC  C0 00012          ADDL2   LAST_COUNT, LBN                           ; 1166
                        56     10   AC  C2 00016          SUBL2   LAST_COUNT, COUNT                         ; 1167
                               9D   AF  9F 0001A          PUSHAB  BB_FID                                    ; 1172
                 0000G  CF          01  FB 0001D          CALLS   #1, SERIAL_FILE                           ;
                    18  AA          50  D0 00022          MOVL    R0, 24(BASE)                              ;
                                    7E  D4 00026          CLRL    -(SP)                                     ; 1174
                               8F   AF  9F 00028          PUSHAB  BB_FID                                    ;
                 0000G  CF          02  FB 0002B          CALLS   #2, READ_HEADER                           ;
                        52          50  D0 00030 1$:      MOVL    R0, HEADER                                ;
                                    7E  D4 00033          CLRL    -(SP)                                     ; 1177
                                    52  DD 00035          PUSHL   HEADER                                    ;
                 0000G  CF          02  FB 00037          CALLS   #2, NEXT_HEADER                           ;
                                    50  D5 0003C          TSTL    EXT_HEADER                                ; 1178
                                    F0  12 0003E          BNEQ    1$                                        ;
                                    52  DD 00040          PUSHL   HEADER                                    ; 1181
                 0000G  CF          01  FB 00042          CALLS   #1, MARK_DIRTY                            ;
                        50     01   A2  9A 00047          MOVZBL  1(HEADER), R0                             ; 1182
                        51     3A   A2  9A 0004B          MOVZBL  58(HEADER), R1                            ;
                        50          51  C0 0004F          ADDL2   R1, R0                                    ;
                        59        6240  3E 00052          MOVAW   (HEADER)[R0], BUILD_POINTER               ;
                                    52  DD 00056          PUSHL   HEADER                                    ; 1184
                        7E          56  7D 00058          MOVQ    COUNT, -(SP)                              ;
                 0000G  CF          03  FB 0005B          CALLS   #3, MAKE_POINTER                          ;
                        22          50  E8 00060          BLBS    R0, 2$                                    ;
                                    7E  D4 00063          CLRL    -(SP)                                     ; 1187
                                    52  DD 00065          PUSHL   HEADER                                    ;
                             FF55   CF  9F 00067          PUSHAB  P.AAB                                     ;
                 0000G  CF          03  FB 0006B          CALLS   #3, EXTEND_HEADER                         ;
                        52          50  D0 00070          MOVL    R0, HEADER                                ;
                        50     01   A2  9A 00073          MOVZBL  1(HEADER), R0                             ; 1188
                        59        6240  3E 00077          MOVAW   (HEADER)[R0], BUILD_POINTER               ;
                                    52  DD 0007B          PUSHL   HEADER                                    ; 1189
                        7E          56  7D 0007D          MOVQ    COUNT, -(SP)                              ;
                 0000G  CF          03  FB 00080          CALLS   #3, MAKE_POINTER                          ;
            50      18  A2          10  9C 00085 2$:      ROTL    #16, 24(HEADER), R0                       ; 1193
                        50          56  C0 0008A          ADDL2   COUNT, R0                                 ;
    18      A2          50          10  9C 0008D          ROTL    #16, R0, 24(HEADER)                       ;
                        28          62  91 00092          CMPB    (HEADER), #40                             ; 1199
                                    0A  1F 00095          BLSSU   3$                                        ;
            50      18  A2          10  9C 00097          ROTL    #16, 24(HEADER), R0                       ; 1201
                    4C  A2     01   A0  9E 0009C          MOVAB   1(R0), 76(HEADER)                         ;
                                    52  DD 000A1 3$:      PUSHL   HEADER                                    ; 1203
                 0000G  CF          01  FB 000A3          CALLS   #1, CHECKSUM                              ;
                        50     18   AA  D0 000A8          MOVL    24(BASE), R0                              ; 1209
                           0080   CA40  DD 000AC          PUSHL   @28(BASE)[R0]                             ;
                 0000G  CF          01  FB 000B1          CALLS   #1, WRITE_DIRTY                           ;
                               18   AA  DD 000B6          PUSHL   24(BASE)                                  ; 1211
                 0000G  CF          01  FB 000B9          CALLS   #1, RELEASE_SERIAL_LOCK                   ;
                 0000G  CF          00  FB 000BE          CALLS   #0, RESTORE_CONTEXT                       ; 1213
                                    56  DD 000C3          PUSHL   COUNT                                     ; 1219
                                    7E  D4 000C5          CLRL    -(SP)                                     ;
                                    57  DD 000C7          PUSHL   LBN                                       ;
                                    7E  7C 000C9          CLRQ    -(SP)                                     ;
```

```
                              0000G  CF          05  FB 000CB          CALLS   #5, SCAN_BADLOG                           ;
                                                     04 000D0          RET                                               ; 1221

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 0046


;  233          1222  1
;  234          1223  1 END
;  235          1224  0 ELUDOM



;                                             PSECT SUMMARY
;
;       Name                        Bytes                          Attributes
;
;  $CODE$                                 279  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                      Library Statistics
;
;                                                  -------- Symbols --------       Pages        Processing
;       File                                      Total    Loaded   Percent        Mapped       Time
;
;  _$255$DUA26:[SYSLIB]LIB.L32;1                  18619        24         0        1000          00:02.0




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

; Size:          209 code + 70 data bytes
; Run Time:         00:17.5
; Elapsed Time:     00:39.6
; Lines/CPU Min:      4206
; Lexemes/CPU-Min: 49494
; Memory Used:  222 pages
; Compilation Complete
```

0169 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY

DEACCS
LIS

DELETE
LIS

DIRSCN
LIS

DISPAT
LIS

CREHDR
LIS

DIRACC
LIS

DELBAD
LIS

CREFCB
LIS

CREWIN
LIS

DISPATCH
LIS

ENTER
LIS

DELFIL
LIS