```
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**EXTEND

N 7

```
EEEEEEEEEE  XX      XX  TTTTTTTTTT  EEEEEEEEEE  NN      NN  DDDDDDDD
EEEEEEEEEE  XX      XX  TTTTTTTTTT  EEEEEEEEEE  NN      NN  DDDDDDDD
EE          XX      XX      TT      EE          NN      NN  DD      DD
EE          XX      XX      TT      EE          NN      NN  DD      DD
EE            XX  XX        TT      EE          NNNN    NN  DD      DD
EE            XX  XX        TT      EE          NNNN    NN  DD      DD
EEEEEEEE        XX          TT      EEEEEEEE    NN  NN  NN  DD      DD
EEEEEEEE        XX          TT      EEEEEEEE    NN  NN  NN  DD      DD
EE            XX  XX        TT      EE          NN    NNNN  DD      DD
EE            XX  XX        TT      EE          NN    NNNN  DD      DD
EE          XX      XX      TT      EE          NN      NN  DD      DD    ....
EE          XX      XX      TT      EE          NN      NN  DD      DD    ....
EEEEEEEEEE  XX      XX      TT      EEEEEEEEEE  NN      NN  DDDDDDDD    ....
EEEEEEEEEE  XX      XX      TT      EEEEEEEEEE  NN      NN  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
```

```
;     1          0001  0 MODULE EXTEND (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**********************************************************************
;     9          0009  1 !*                                                                    *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                             *
;    13          0013  1 !*                                                                    *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    19          0019  1 !*  TRANSFERRED.                                                       *
;    20          0020  1 !*                                                                    *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    23          0023  1 !*  CORPORATION.                                                       *
;    24          0024  1 !*                                                                    *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    27          0027  1 !*                                                                    *
;    28          0028  1 !*                                                                    *
;    29          0029  1 !**********************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine extends a file by the requested number of blocks.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       STARLET operating system, including privileged system services
;    42          0042  1 !       and internal exec routines.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  24-Feb-1977  15:42
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V02-002 LMP0005         L. Mark Pilant,         29-Dec-1981  14:50
;    52          0052  1 !               Added support for extending a file with Cathedral windows.
;    53          0053  1 !
;    54          0054  1 !       A0101   ACG26369        Andrew C. Goldstein,    28-Dec-1979  15:46
;    55          0055  1 !               Fix multi-header interlock bug
;    56          0056  1 !
;    57          0057  1 !       A0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:02
```

```
;    58          0058  1 !       Previous revision history moved to F11A.REV
;    59          0059  1 !**
;    60          0060  1
;    61          0061  1
;    62          0062  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    63          0063  1 REQUIRE 'SRC$:FCPDEF.B32';
;    64          0378  1
;    65          0379  1 FORWARD ROUTINE
;    66          0380  1         EXTEND              : NOVALUE,      ! extend a file
;    67          0381  1         UPDATE_FILESIZE     : NOVALUE;      ! update file size in FCB
```

```
;    69          0382  1 GLOBAL ROUTINE EXTEND (USER_FIB, FILEHEADER) : NOVALUE =
;    70          0383  1
;    71          0384  1 !++
;    72          0385  1 !
;    73          0386  1 ! FUNCTIONAL DESCRIPTION:
;    74          0387  1 !
;    75          0388  1 !       This routine extends the given file by the amount and in the
;    76          0389  1 !       mode given in the FIB. The blocks are allocated from the storage
;    77          0390  1 !       bitmap and retrieval pointers are constructed in the header.
;    78          0391  1 !
;    79          0392  1 ! CALLING SEQUENCE:
;    80          0393  1 !       EXTEND (ARG1, ARG2)
;    81          0394  1 !
;    82          0395  1 ! INPUT PARAMETERS:
;    83          0396  1 !       ARG1: address of FIB for operation
;    84          0397  1 !       ARG2: addres of file header
;    85          0398  1 !
;    86          0399  1 ! IMPLICIT INPUTS:
;    87          0400  1 !       CURRENT_WINDOW: window of file, if open
;    88          0401  1 !
;    89          0402  1 ! OUTPUT PARAMETERS:
;    90          0403  1 !       NONE
;    91          0404  1 !
;    92          0405  1 ! IMPLICIT OUTPUTS:
;    93          0406  1 !       NONE
;    94          0407  1 !
;    95          0408  1 ! ROUTINE VALUE:
;    96          0409  1 !       NONE
;    97          0410  1 !
;    98          0411  1 ! SIDE EFFECTS:
;    99          0412  1 !       blocks allocated, file header altered
;   100          0413  1 !
;   101          0414  1 !--
;   102          0415  1
;   103          0416  2 BEGIN
;   104          0417  2
;   105          0418  2 MAP
;   106          0419  2         USER_FIB        : REF BBLOCK,   ! FIB of operation
;   107          0420  2         FILEHEADER      : REF BBLOCK;   ! file header to extend
;   108          0421  2
;   109          0422  2 LABEL
;   110          0423  2         ALLOC_LOOP;                     ! Block allocation and recording loop
;   111          0424  2 LOCAL
;   112          0425  2         FIB             : REF BBLOCK,   ! address of FIB
;   113          0426  2         HEADER          : REF BBLOCK,   ! address of current file header
;   114          0427  2         FCB             : REF BBLOCK,   ! FCB of header being extended
;   115          0428  2         NEW_HEADER      : REF BBLOCK,   ! next extension file header
;   116          0429  2         WINDOW_SEGMENT  : REF BBLOCK,   ! address of the next window segment
;   117          0430  2         BLOCKS_NEEDED,                  ! number of blocks to be allocated
;   118          0431  2         MAP_AREA        : REF BBLOCK,   ! address of header map area
;   119          0432  2         MAP_POINTER     : REF BBLOCK,   ! pointer to current retrieval pointer
;   120          0433  2         CBT_COUNT,                      ! count of bitmap scans
;   121          0434  2         EXTEND_VBN,                     ! starting VBN of extend
;   122          0435  2         LBN,                            ! LBN of blocks allocated
;   123          0436  2         ALLOC_COUNT,                    ! number of blocks allocated
;   124          0437  2         COUNT,                          ! count of blocks for map pointers
;   125          0438  2         OLD_LBN,                        ! LBN of previous retrieval pointer
```

```
;  126     0439  2          OLD_COUNT;                      ! count of previous retrieval pointer
;  127     0440  2
;  128     0441  2      EXTERNAL
;  129     0442  2              USER_STATUS     : VECTOR,   ! I/O status block of user
;  130     0443  2              PRIMARY_FCB     : REF BBLOCK,   ! FCB of file
;  131     0444  2              UNREC_LBN,                  ! LBN of unrecorded blocks
;  132     0445  2              UNREC_COUNT,                ! count of unrecorded blocks
;  133     0446  2              CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
;  134     0447  2              CURRENT_VCB     : REF BBLOCK,   ! VCB of volume
;  135     0448  2              CURRENT_WINDOW  : REF BBLOCK;   ! window of file if open
;  136     0449  2
;  137     0450  2      EXTERNAL ROUTINE
;  138     0451  2              PMS_START_SUB,              ! start subfunction metering
;  139     0452  2              PMS_END_SUB,                ! end subfunction metering
;  140     0453  2              SEARCH_FCB,                 ! search FCB list for FCB
;  141     0454  2              NEXT_HEADER,                ! read next extension header
;  142     0455  2              MARK_DIRTY,                 ! mark buffer for write-back
;  143     0456  2              ALLOC_BLOCKS,               ! allocate blocks from storage map
;  144     0457  2              EXTEND_HEADER,              ! create extension header
;  145     0458  2              RETURN_BLOCKS,              ! return blocks to storage map
;  146     0459  2              CHECKSUM,                   ! compute file header checksum
;  147     0460  2              TURN_WINDOW,                ! update file window
;  148     0461  2              INIT_FCB,                   ! initialize FCB
;  149     0462  2              WRITE_HEADER,               ! write file header
;  150     0463  2              READ_HEADER,                ! read file header
;  151     0464  2              MARK_INCOMPLETE;            ! flag all windows as incomplete
;  152     0465  2
;  153     0466  2
;  154     0467  2      ! Start metering for this subfunction.
;  155     0468  2      !
;  156     0469  2
;  157     0470  2      PMS_START_SUB (PMS_ALLOC);
;  158     0471  2
;  159     0472  2      ! Check the allocation control bits for validity. Then get the block count
;  160     0473  2      ! and set up pointers.
;  161     0474  2      !
;  162     0475  2
;  163     0476  2      FIB = .USER_FIB;
;  164     0477  3      IF (NOT .FIB[FIB$V_ALCON] AND .FIB[FIB$V_FILCON])
;  165     0478  2      OR .FIB[FIB$L_EXSZ] LSS 0
;  166     0479  2      THEN ERR_EXIT (SS$_BADPARAM);
;  167     0480  2
;  168     0481  3      BLOCKS_NEEDED = (
;  169     0482  3          IF .FIB[FIB$V_ALDEF]
;  170     0483  3          THEN MAXU (.CURRENT_VCB[VCB$W_EXTEND], .FIB[FIB$L_EXSZ])
;  171     0484  3          ELSE .FIB[FIB$L_EXSZ]
;  172     0485  2          );
;  173     0486  2
;  174     0487  2      HEADER = .FILEHEADER;
;  175     0488  2      FCB = .PRIMARY_FCB;
;  176     0489  2      EXTEND_VBN = 1;
;  177     0490  2
;  178     0491  2      ! If the file is marked contiguous best effort, make the extend so.
;  179     0492  2      !
;  180     0493  2
;  181     0494  2      IF .HEADER[FH1$V_CONTIGB]
;  182     0495  2      THEN
```

```
;  183          0496  2      IF NOT .FIB[FIB$V_ALCON] THEN FIB[FIB$V_ALCONB] = 1;
;  184          0497  2
;  185          0498  2  ! Scan through this header's map area and through the map area of all
;  186          0499  2  ! extension headers to compute the current file size and find the end of file
;  187          0500  2  ! to start extension.
;  188          0501  2  !
;  189          0502  2
;  190          0503  2  WHILE 1 DO
;  191          0504  3      BEGIN
;  192          0505  3      MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  193          0506  3
;  194          0507  3      IF .MAP_AREA[FM1$B_INUSE] NEQ 0 AND .FIB[FIB$V_FILCON]
;  195          0508  3      THEN ERR_EXIT (SS$_BADPARAM);
;  196          0509  3
;  197          0510  3      MAP_POINTER = .MAP_AREA + FM1$C_POINTERS;
;  198          0511  3
;  199          0512  3      DECR MAPCOUNT FROM .MAP_AREA[FM1$B_INUSE]/2 TO 1 DO
;  200          0513  4          BEGIN
;  201          0514  4          EXTEND_VBN = .EXTEND_VBN + .MAP_POINTER[FM1$B_COUNT] + 1;
;  202          0515  4          MAP_POINTER = .MAP_POINTER + 4;
;  203          0516  3          END;
;  204          0517  3
;  205          0518  3      NEW_HEADER = NEXT_HEADER (.HEADER, .FCB);
;  206          0519  3      IF .NEW_HEADER EQL 0 THEN EXITLOOP;
;  207          0520  3      HEADER = .NEW_HEADER;
;  208          0521  3
;  209          0522  3      IF .FCB NEQ 0
;  210          0523  3      THEN FCB = .FCB[FCB$L_EXFCB]
;  211          0524  3      ELSE IF SEARCH_FCB (HEADER[FH1$W_FID]) NEQ 0
;  212          0525  3      THEN ERR_EXIT (SS$_ACCONFLICT);
;  213          0526  2      END;
;  214          0527  2
;  215          0528  2  ! Check the remaining parameters and set the relevant cleanup action flags.
;  216          0529  2  !
;  217          0530  2
;  218          0531  2  IF .FIB[FIB$L_EXVBN] NEQ 0 AND .FIB[FIB$L_EXVBN] NEQ .EXTEND_VBN
;  219          0532  2  THEN ERR_EXIT (SS$_BADPARAM);
;  220          0533  2
;  221          0534  2  MARK_DIRTY (.HEADER);
;  222          0535  2  CLEANUP_FLAGS[CLF_TRUNCATE] = 1;
;  223          0536  2  CLEANUP_FLAGS[CLF_FIXFCB] = 1;
;  224          0537  2
;  225          0538  2  CBT_COUNT = 0;                              ! init count of bitmap scans
;  226          0539  2  FIB[FIB$L_EXSZ] = 0;
;  227          0540  2  FIB[FIB$L_EXVBN] = .EXTEND_VBN;
;  228          0541  2
;  229          0542  2  ! Now loop, allocating blocks fom the storage map and building retrieval
;  230          0543  2  ! pointers in the header. Accumulate blocks allocated in the I/O
;  231          0544  2  ! status block.
;  232          0545  2  !
;  233          0546  2
;  234          0547  2  ALLOC_LOOP:
;  235          0548  3  BEGIN
;  236          0549  3  UNTIL .BLOCKS_NEEDED EQL 0 DO
;  237          0550  4      BEGIN
;  238          0551  4
;  239          0552  4      ALLOC_BLOCKS (.FIB, .BLOCKS_NEEDED, LBN, ALLOC_COUNT);
```

```
;  240          0553  4     COUNT = .ALLOC_COUNT;
;  241          0554  4     FIB[FIB$L_EXSZ] = .FIB[FIB$L_EXSZ] + .COUNT;
;  242          0555  4     BLOCKS_NEEDED = .BLOCKS_NEEDED - MINU (.BLOCKS_NEEDED, .COUNT);
;  243          0556  4
;  244          0557  4 ! Build the map pointers. If the new area allocated is contiguous with
;  245          0558  4 ! the last pointer in the header, merge the pointers.
;  246          0559  4 !
;  247          0560  4
;  248          0561  4     IF .MAP_AREA[FM1$B_INUSE] NEQ 0
;  249          0562  4     THEN
;  250          0563  5         BEGIN
;  251          0564  5         OLD_LBN = .MAP_POINTER[FM1$W_PREVLLBN];
;  252          0565  5         OLD_LBN<16,8> = .MAP_POINTER[FM1$B_PREVHLBN];
;  253          0566  5         OLD_COUNT = .MAP_POINTER[FM1$B_PREVCOUNT] + 1;
;  254          0567  5
;  255          0568  5         IF .OLD_LBN + .OLD_COUNT EQL .LBN
;  256          0569  5         THEN
;  257          0570  6             BEGIN
;  258          0571  6             MAP_POINTER = .MAP_POINTER - 4;
;  259          0572  6             MAP_AREA[FM1$B_INUSE] = .MAP_AREA[FM1$B_INUSE] - 2;
;  260          0573  6             COUNT = .COUNT + .OLD_COUNT;
;  261          0574  6             LBN = .OLD_LBN;
;  262          0575  5             END;
;  263          0576  4         END;
;  264          0577  4
;  265          0578  4 ! Now build retrieval pointers to map the allocated blocks. If the map fills
;  266          0579  4 ! up, store the unrecorded blocks in common so they can be returned
;  267          0580  4 ! by the extend cleanup, and create an extension header. If header
;  268          0581  4 ! extension is inhibited, return the unrecorded blocks and get out quietly.
;  269          0582  4 ! We return header full status only if no new blocks were recorded.
;  270          0583  4 !
;  271          0584  4
;  272          0585  4     UNTIL .COUNT EQL 0 DO
;  273          0586  5         BEGIN
;  274          0587  5         IF .MAP_AREA[FM1$B_INUSE] + 2 GTR .MAP_AREA[FM1$B_AVAIL]
;  275          0588  5         THEN
;  276          0589  6             BEGIN
;  277          0590  6             IF .FIB[FIB$V_NOHDREXT]
;  278          0591  6             THEN
;  279          0592  7                 BEGIN
;  280          0593  7                 RETURN_BLOCKS (.LBN, .COUNT);
;  281          0594  7                 FIB[FIB$L_EXSZ] = .FIB[FIB$L_EXSZ] - .COUNT;
;  282          0595  7                 IF .FIB[FIB$L_EXSZ] EQL 0
;  283          0596  7                 THEN ERR_EXIT (SS$_HEADERFULL);
;  284          0597  7                 LEAVE ALLOC_LOOP;
;  285          0598  7                 END
;  286          0599  6             ELSE
;  287          0600  7                 BEGIN
;  288          0601  7                 UNREC_LBN = .LBN;
;  289          0602  7                 UNREC_COUNT = .COUNT;
;  290          0603  7                 HEADER = EXTEND_HEADER (.HEADER, .FCB);
;  291          0604  7                 IF .FCB NEQ 0 THEN FCB = .FCB[FCB$L_EXFCB];
;  292          0605  7                 MAP_AREA = .HEADER + .HEADER[FM1$B_MPOFFSET]*2;
;  293          0606  7                 MAP_POINTER = .MAP_AREA + FM1$C_POINTERS;
;  294          0607  6                 END;
;  295          0608  5             END;
;  296          0609  5
```

```
;   297          0610  5 ! Finally build the next retrieval pointer.
;   298          0611  5 !
;   299          0612  5
;   300          0613  5         MAP_AREA[FM1$B_INUSE] = .MAP_AREA[FM1$B_INUSE] + 2;
;   301          0614  5
;   302          0615  5         MAP_POINTER[FM1$B_HIGHLBN] = .LBN<16,8>;
;   303          0616  5         MAP_POINTER[FM1$B_COUNT] = MIN (.COUNT, 256) - 1;
;   304          0617  5         MAP_POINTER[FM1$W_LOWLBN] = .LBN<0,16>;
;   305          0618  5         MAP_POINTER = .MAP_POINTER + 4;
;   306          0619  5
;   307          0620  5         LBN = .LBN + MIN (.COUNT, 256);
;   308          0621  5         COUNT = .COUNT - MIN (.COUNT, 256);
;   309          0622  4         END;
;   310          0623  4
;   311          0624  4     UNREC_COUNT = 0;                        ! all blocks are now recorded
;   312          0625  4
;   313          0626  4 ! If this was a contiguous allocation, we are done. Else count the pass
;   314          0627  4 ! through the allocator. After 3 passes, shut off the contiguous best try
;   315          0628  4 ! bit to avoid taking forever (since each CBT try is a full sweep of the map).
;   316          0629  4 !
;   317          0630  4
;   318          0631  4     IF .FIB[FIB$V_ALCON] THEN EXITLOOP;
;   319          0632  4     CBT_COUNT = .CBT_COUNT + 1;
;   320          0633  4     IF .CBT_COUNT GEQU 3
;   321          0634  4     THEN FIB[FIB$V_ALCONB] = 0;
;   322          0635  3     END;
;   323          0636  2 END;                                        ! end of allocation loop
;   324          0637  2
;   325          0638  2 ! If the file is open by the caller, turn the window to the last VBN
;   326          0639  2 ! that previously existed as a friendly gesture. Then, if the current header
;   327          0640  2 ! is an extension header, write it and read back the primary header. Also
;   328          0641  2 ! set the contiguous bit in the header appropriately and return the extend
;   329          0642  2 ! data in the FIB. Update the file size in the primary FCB.
;   330          0643  2 !
;   331          0644  2
;   332          0645  2 IF .CURRENT_WINDOW NEQ 0
;   333          0646  2 THEN
;   334          0647  3     BEGIN
;   335          0648  3     IF NOT .CURRENT_WINDOW[WCB$V_CATHEDRAL]
;   336          0649  4     THEN KERNEL_CALL (TURN_WINDOW, .CURRENT_WINDOW, .HEADER, .FIB[FIB$L_EXVBN]-1, .FCB[FCB$L_STVBN])
;   337          0650  3     ELSE KERNEL_CALL (MARK_INCOMPLETE, .PRIMARY_FCB);
;   338          0651  2     END;
;   339          0652  2
;   340          0653  2 IF .HEADER[FH1$W_FID_NUM] NEQ .FIB[FIB$W_FID_NUM]
;   341          0654  2 THEN
;   342          0655  3     BEGIN
;   343          0656  3     CHECKSUM (.HEADER);
;   344          0657  3     WRITE_HEADER ();
;   345          0658  3     IF .FCB NEQ 0 THEN KERNEL_CALL (INIT_FCB, .FCB, .HEADER);
;   346          0659  3     HEADER = READ_HEADER (FIB[FIB$W_FID], .PRIMARY_FCB);
;   347          0660  3     END;
;   348          0661  2
;   349          0662  2 ! Update the HIBLK field in the record attributes to reflect the new file
;   350          0663  2 ! size.
;   351          0664  2 !
;   352          0665  2
;   353          0666  2 MARK_DIRTY (.HEADER);
```

```
;  354          0667  2 BBLOCK [HEADER[FH1$W_RECATTR], FAT$L_HIBLK] = ROT (.FIB[FIB$L_EXVBN] + .FIB[FIB$L_EXSZ] - 1, 16);
;  355          0668  2 HEADER[FH1$V_CONTIG] = .FIB[FIB$V_FICON];
;  356          0669  2 HEADER[FH1$V_CONTIGB] = .FIB[FIB$V_ALCONB];
;  357          0670  2 USER_STATUS[T] = .FIB[FIB$L_EXSZ];
;  358          0671  2 KERNEL_CALL (UPDATE_FILESIZE, .FIB[FIB$L_EXVBN] + .FIB[FIB$L_EXSZ] - 1);
;  359          0672  2
;  360          0673  2 ! Stop metering of this subfunction
;  361          0674  2 !
;  362          0675  2
;  363          0676  2 PMS_END_SUB ();
;  364          0677  2
;  365          0678  1 END;                                    ! end of routine EXTEND

                                                        .TITLE  EXTEND
                                                        .IDENT  \V04-000\

                                                        .EXTRN  USER_STATUS, PRIMARY_FCB
                                                        .EXTRN  UNREC_LBN, UNREC_COUNT
                                                        .EXTRN  CLEANUP_FLAGS, CURRENT_VCB
                                                        .EXTRN  CURRENT_WINDOW, PMS_START_SUB
                                                        .EXTRN  PMS_END_SUB, SEARCH_FCB
                                                        .EXTRN  NEXT_HEADER, MARK_DIRTY
                                                        .EXTRN  ALLOC_BLOCKS, EXTEND_HEADER
                                                        .EXTRN  RETURN_BLOCKS, CHECKSUM
                                                        .EXTRN  TURN_WINDOW, INIT_FCB
                                                        .EXTRN  WRITE_HEADER, READ_HEADER
                                                        .EXTRN  MARK_INCOMPLETE
                                                        .EXTRN  SYS$CMKRNL

                                                        .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000          .ENTRY  EXTEND, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 0382
                                                                R11                                             ;
                          5E            0C   C2 00002         SUBL2   #12, SP                                   ;
                                        08   DD 00005         PUSHL   #8                                        ; 0470
                    0000G CF            01   FB 00007         CALLS   #1, PMS_START_SUB                         ;
                          52      04    AC   D0 0000C         MOVL    USER_FIB, FIB                             ; 0476
                          05      16    A2   E8 00010         BLBS    22(FIB), 1$                               ; 0477
                05        16    A2          02   E0 00014    BBS     #2, 22(FIB), 2$                           ;
                                      18    A2   D5 00019 1$:  TSTL    24(FIB)                                   ; 0478
                                            03   18 0001C     BGEQ    3$                                        ;
                                        00A1 31   0001E 2$:   BRW     13$                                       ;
                18        16    A2          03   E1 00021 3$:  BBC     #3, 22(FIB), 5$                           ; 0482
                          50    0000G CF    D0   00026         MOVL    CURRENT_VCB, R0                           ; 0483
                          50      3E  A0    3C   0002B         MOVZWL  62(R0), R0                                ;
                                  18  A2    50   D1 0002F      CMPL    R0, 24(FIB)                               ;
                                            04   1E 00033     BGEQU   4$                                        ;
                                50  18  A2    D0   00035         MOVL    24(FIB), R0                           ;
                                57          50   D0 00039 4$:   MOVL    R0, BLOCKS_NEEDED                     ;
                                            04   11 0003C     BRB     6$                                        ;
                                57  18  A2    D0   0003E 5$:   MOVL    24(FIB), BLOCKS_NEEDED                ; 0484
                                58  08  AC    D0   00042 6$:   MOVL    FILEHEADER, HEADER                    ; 0487
                                5A  0000G CF    D0   00046         MOVL    PRIMARY_FCB, FCB                      ; 0488
                                55          01   D0 0004B         MOVL    #1, EXTEND_VBN                        ; 0489
                08        0C    A8          05   E1 0004E         BBC     #5, 12(HEADER), 7$                    ; 0494
                          04    16  A2    E8   00053         BLBS    22(FIB), 7$                               ; 0496
```

```
                16  A2           02  88 00057       BISB2   #2, 22(FIB)                        ;
                    50       01  A8  9A 0005B 7$:   MOVZBL  1(HEADER), R0                      ; 0505
                    54     6840      3E 0005F       MOVAW   (HEADER)[R0], MAP_AREA             ;
                             08  A4  95 00063       TSTB    8(MAP_AREA)                        ; 0507
                                 05  13 00066       BEQL    8$                                 ;
          55    16  A2           02  E0 00068       BBS     #2, 22(FIB), 13$                   ;
                    53       0A  A4  9E 0006D 8$:   MOVAB   10(R4), MAP_POINTER                ; 0510
                    51       08  A4  9A 00071       MOVZBL  8(MAP_AREA), R1                    ; 0512
                    51           02  C6 00075       DIVL2   #2, R1                             ;
                                 51  D6 00078       INCL    MAPCOUNT                           ;
                                 0C  11 0007A       BRB     10$                                ;
                    50       01  A3  9A 0007C 9$:   MOVZBL  1(MAP_POINTER), R0                 ; 0514
                    55    01 A045    9E 00080       MOVAB   1(R0)[EXTEND_VBN], EXTEND_VBN      ;
                    53           04  C0 00085       ADDL2   #4, MAP_POINTER                    ; 0515
                    F1           51  F5 00088 10$:  SOBGTR  MAPCOUNT, 9$                       ; 0512
                           0500  8F  BB 0008B       PUSHR   #^M<R8,R10>                        ; 0518
             0000G  CF           02  FB 0008F       CALLS   #2, NEXT_HEADER                    ;
                    56           50  D0 00094       MOVL    R0, NEW_HEADER                     ;
                                 1E  13 00097       BEQL    12$                                ; 0519
                    58           56  D0 00099       MOVL    NEW_HEADER, HEADER                 ; 0520
                                 5A  D5 0009C       TSTL    FCB                                ; 0522
                                 06  13 0009E       BEQL    11$                                ;
                    5A       0C  AA  D0 000A0       MOVL    12(FCB), FCB                       ; 0523
                                 B5  11 000A4       BRB     7$                                 ;
                             02  A8  9F 000A6 11$:  PUSHAB  2(HEADER)                          ; 0524
             0000G  CF           01  FB 000A9       CALLS   #1, SEARCH_FCB                     ;
                                 50  D5 000AE       TSTL    R0                                 ;
                                 A9  13 000B0       BEQL    7$                                 ;
                           0800  8F  BF 000B2       CHMU    #2048                              ; 0525
                                     04 000B6       RET                                        ;
                             1C  A2  D5 000B7 12$:  TSTL    28(FIB)                            ; 0531
                                 09  13 000BA       BEQL    14$                                ;
                    55       1C  A2  D1 000BC       CMPL    28(FIB), EXTEND_VBN                ;
                                 03  13 000C0       BEQL    14$                                ;
                                 14  BF 000C2 13$:  CHMU    #20                                ; 0532
                                     04 000C4       RET                                        ;
                                 58  DD 000C5 14$:  PUSHL   HEADER                             ; 0534
             0000G  CF           01  FB 000C7       CALLS   #1, MARK_DIRTY                     ;
             0000G  CF 00040002  8F  C8 000CC       BISL2   #262146, CLEANUP_FLAGS             ; 0535
                                 6E  D4 000D5       CLRL    CBT_COUNT                          ; 0538
                             18  A2  D4 000D7       CLRL    24(FIB)                            ; 0539
                1C  A2           55  D0 000DA       MOVL    EXTEND_VBN, 28(FIB)                ; 0540
                                 57  D5 000DE 15$:  TSTL    BLOCKS_NEEDED                      ; 0549
                                 03  12 000E0       BNEQ    16$                                ;
                               00F9  31 000E2       BRW     27$                                ;
                             04  AE  9F 000E5 16$:  PUSHAB  ALLOC_COUNT                        ; 0552
                             0C  AE  9F 000E8       PUSHAB  LBN                                ;
                           0084  8F  BB 000EB       PUSHR   #^M<R2,R7>                         ;
             0000G  CF           04  FB 000EF       CALLS   #4, ALLOC_BLOCKS                   ;
                    55       04  AE  D0 000F4       MOVL    ALLOC_COUNT, COUNT                 ; 0553
                18  A2           55  C0 000F8       ADDL2   COUNT, 24(FIB)                     ; 0554
                    50           57  D0 000FC       MOVL    BLOCKS_NEEDED, R0                  ; 0555
                    55           50  D1 000FF       CMPL    R0, COUNT                          ;
                                 03  1B 00102       BLEQU   17$                                ;
                    50           55  D0 00104       MOVL    COUNT, R0                          ;
                    57           50  C2 00107 17$:  SUBL2   R0, BLOCKS_NEEDED                  ;
                             08  A4  95 0010A       TSTB    8(MAP_AREA)                        ; 0561
```

```
                                            28    13 0010D          BEQL    18$                                     ;
                              56      FE    A3    3C 0010F          MOVZWL  -2(MAP_POINTER), OLD_LBN                ; 0564
 56          08               10      FC    A3    F0 00113          INSV    -4(MAP_POINTER), #16, #8, OLD_LBN       ; 0565
                              5B      FD    A3    9A 00119          MOVZBL  -3(MAP_POINTER), OLD_COUNT              ; 0566
                                            5B    D6 0011D          INCL    OLD_COUNT                               ;
             50               56            5B    C1 0011F          ADDL3   OLD_COUNT, OLD_LBN, R0                  ; 0568
                       08     AE            50    D1 00123          CMPL    R0, LBN                                 ;
                                            0E    12 00127          BNEQ    18$                                     ;
                              53            04    C2 00129          SUBL2   #4, MAP_POINTER                         ; 0571
                       08     A4            02    82 0012C          SUBB2   #2, 8(MAP_AREA)                         ; 0572
                              55            5B    C0 00130          ADDL2   OLD_COUNT, COUNT                        ; 0573
                       08     AE            56    D0 00133          MOVL    OLD_LBN, LBN                            ; 0574
                              59      08    A4    9E 00137  18$:    MOVAB   8(R4), R9                               ; 0587
                                            55    D5 0013B  19$:    TSTL    COUNT                                   ; 0585
                                            03    12 0013D          BNEQ    20$                                     ;
                                          0086    31 0013F          BRW     25$                                     ;
                              50            69    9A 00142  20$:    MOVZBL  (R9), R0                                ; 0587
                              50            02    C0 00145          ADDL2   #2, R0                                  ;
 50     09   A4               08            00    ED 00148          CMPZV   #0, #8, 9(MAP_AREA), R0                 ;
                                            45    18 0014E          BGEQ    23$                                     ;
             15        17     A2            01    E1 00150          BBC     #1, 23(FIB), 21$                        ; 0590
                                            55    DD 00155          PUSHL   COUNT                                   ; 0593
                              0C            AE    DD 00157          PUSHL   LBN                                     ;
                    0000G     CF            02    FB 0015A          CALLS   #2, RETURN_BLOCKS                       ;
                       18     A2            55    C2 0015F          SUBL2   COUNT, 24(FIB)                          ; 0594
                                            79    12 00163          BNEQ    27$                                     ; 0595
                                    08C8    8F    BF 00165          CHMU    #2248                                   ; 0596
                                                  04 00169          RET                                             ;
                    0000G     CF      08    AE    D0 0016A  21$:    MOVL    LBN, UNREC_LBN                          ; 0601
                    0000G     CF            55    D0 00170          MOVL    COUNT, UNREC_COUNT                      ; 0602
                                    0500    8F    BB 00175          PUSHR   #^M<R8,R10>                             ; 0603
                    0000G     CF            02    FB 00179          CALLS   #2, EXTEND_HEADER                       ;
                              58            50    D0 0017E          MOVL    R0, HEADER                              ;
                                            5A    D5 00181          TSTL    FCB                                     ; 0604
                                            04    13 00183          BEQL    22$                                     ;
                              5A      0C    AA    D0 00185          MOVL    12(FCB), FCB                            ;
                              50      01    A8    9A 00189  22$:    MOVZBL  1(HEADER), R0                           ; 0605
                              54          6840    3E 0018D          MOVAW   (HEADER)[R0], MAP_AREA                  ;
                              53      0A    A4    9E 00191          MOVAB   10(R4), MAP_POINTER                     ; 0606
                              59      08    A4    9E 00195  23$:    MOVAB   8(R4), R9                               ; 0613
                              69            02    80 00199          ADDB2   #2, (R9)                                ;
                              63      0A    AE    90 0019C          MOVB    LBN+2, (MAP_POINTER)                    ; 0615
                              50            55    D0 001A0          MOVL    COUNT, R0                               ; 0616
                 00000100     8F            50    D1 001A3          CMPL    R0, #256                                ;
                                            05    15 001AA          BLEQ    24$                                     ;
                              50    0100    8F    3C 001AC          MOVZWL  #256, R0                                ;
        01   A3               50            01    83 001B1  24$:    SUBB3   #1, R0, 1(MAP_POINTER)                  ;
                       02     A3      08    AE    B0 001B6          MOVW    LBN, 2(MAP_POINTER)                     ; 0617
                              53            04    C0 001BB          ADDL2   #4, MAP_POINTER                         ; 0618
                       08     AE            50    C0 001BE          ADDL2   R0, LBN                                 ; 0620
                              55            50    C2 001C2          SUBL2   R0, COUNT                               ; 0621
                                          FF73    31 001C5          BRW     19$                                     ; 0585
                                  0000G     CF    D4 001C8  25$:    CLRL    UNREC_COUNT                             ; 0624
                              0E      16    A2    E8 001CC          BLBS    22(FIB), 27$                            ; 0631
                                            6E    D6 001D0          INCL    CBT_COUNT                               ; 0632
                              03            6E    D1 001D2          CMPL    CBT_COUNT, #3                           ; 0633
                                            04    1F 001D5          BLSSU   26$                                     ;
```

```
                    16   A2           02  8A 001D7          BICB2   #2, 22(FIB)                       ; 0634
                                  FF00  31 001DB 26$:       BRW     15$                               ; 0549
                         50   0000G   CF  D0 001DE 27$:     MOVL    CURRENT_WINDOW, R0                ; 0645
                                      35  13 001E3          BEQL    29$                               ;
               1D        0B   A0      06  E0 001E5          BBS     #6, 11(R0), 28$                   ; 0648
                                 2C   AA  DD 001EA          PUSHL   44(FCB)                           ; 0649
               7E        1C   A2      01  C3 001ED          SUBL3   #1, 28(FIB), -(SP)                ;
                              0101    8F  BB 001F2          PUSHR   #^M<R0,R8>                        ;
                                      04  DD 001F6          PUSHL   #4                                ;
                                      5E  DD 001F8          PUSHL   SP                                ;
                              0000G   CF  9F 001FA          PUSHAB  TURN_WINDOW                       ;
                   00000000G  9F      07  FB 001FE          CALLS   #7, @#SYS$CMKRNL                  ;
                                      13  11 00205          BRB     29$                               ;
                              0000G   CF  DD 00207 28$:     PUSHL   PRIMARY_FCB                       ; 0650
                                      01  DD 0020B          PUSHL   #1                                ;
                                      5E  DD 0020D          PUSHL   SP                                ;
                              0000G   CF  9F 0020F          PUSHAB  MARK_INCOMPLETE                   ;
                   00000000G  9F      04  FB 00213          CALLS   #4, @#SYS$CMKRNL                  ;
                          04  A2  02  A8  B1 0021A 29$:     CMPW    2(HEADER), 4(FIB)                 ; 0653
                                      32  13 0021F          BEQL    31$                               ;
                                      58  DD 00221          PUSHL   HEADER                            ; 0656
                       0000G  CF      01  FB 00223          CALLS   #1, CHECKSUM                      ;
                       0000G  CF      00  FB 00228          CALLS   #0, WRITE_HEADER                  ; 0657
                                      5A  D5 0022D          TSTL    FCB                               ; 0658
                                      13  13 0022F          BEQL    30$                               ;
                                      58  DD 00231          PUSHL   HEADER                            ;
                                      5A  DD 00233          PUSHL   FCB                               ;
                                      02  DD 00235          PUSHL   #2                                ;
                                      5E  DD 00237          PUSHL   SP                                ;
                              0000G   CF  9F 00239          PUSHAB  INIT_FCB                          ;
                   00000000G  9F      05  FB 0023D          CALLS   #5, @#SYS$CMKRNL                  ;
                              0000G   CF  DD 00244 30$:     PUSHL   PRIMARY_FCB                       ; 0659
                                 04   A2  9F 00248          PUSHAB  4(FIB)                            ;
                       0000G  CF      02  FB 0024B          CALLS   #2, READ_HEADER                   ;
                              58      50  D0 00250          MOVL    R0, HEADER                        ;
                                      58  DD 00253 31$:     PUSHL   HEADER                            ; 0666
                       0000G  CF      01  FB 00255          CALLS   #1, MARK_DIRTY                    ;
               50        1C   A2  18  A2  C1 0025A          ADDL3   24(FIB), 28(FIB), R0              ; 0667
                                      50  D7 00260          DECL    R0                                ;
               12   A8        50      10  9C 00262          ROTL    #16, R0, 18(HEADER)               ;
       51      16   A2        01      02  EF 00267          EXTZV   #2, #1, 22(FIB), R1               ; 0668
   0C  A8           01        07      51  F0 0026D          INSV    R1, #7, #1, 12(HEADER)            ;
       51      16   A2        01      01  EF 00273          EXTZV   #1, #1, 22(FIB), R1               ; 0669
   0C  A8           01        05      51  F0 00279          INSV    R1, #5, #1, 12(HEADER)            ;
                       0000G  CF  18  A2  D0 0027F          MOVL    24(FIB), USER_STATUS+4            ; 0670
                                      50  DD 00285          PUSHL   R0                                ; 0671
                                      01  DD 00287          PUSHL   #1                                ;
                                      5E  DD 00289          PUSHL   SP                                ;
                              0000V   CF  9F 0028B          PUSHAB  UPDATE_FILESIZE                   ;
                   00000000G  9F      04  FB 0028F          CALLS   #4, @#SYS$CMKRNL                  ;
                       0000G  CF      00  FB 00296          CALLS   #0, PMS_END_SUB                   ; 0676
                                          04 0029B          RET                                       ; 0678

; Routine Size:  668 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  367          0679  1 GLOBAL ROUTINE UPDATE_FILESIZE (SIZE) : NOVALUE =
;  368          0680  1
;  369          0681  1 !++
;  370          0682  1 !
;  371          0683  1 ! FUNCTIONAL DESCRIPTION:
;  372          0684  1 !
;  373          0685  1 !       This routine updates the file size recorded in the primary FCB of
;  374          0686  1 !       the file extended.
;  375          0687  1 !
;  376          0688  1 ! CALLING SEQUENCE:
;  377          0689  1 !       UPDATE_FILESIZE (ARG1)
;  378          0690  1 !
;  379          0691  1 ! INPUT PARAMETERS:
;  380          0692  1 !       ARG1: amount to increase size to
;  381          0693  1 !
;  382          0694  1 ! IMPLICIT INPUTS:
;  383          0695  1 !       PRIMARY_FCB: address of file FCB or zero
;  384          0696  1 !
;  385          0697  1 ! OUTPUT PARAMETERS:
;  386          0698  1 !       NONE
;  387          0699  1 !
;  388          0700  1 ! IMPLICIT OUTPUTS:
;  389          0701  1 !       NONE
;  390          0702  1 !
;  391          0703  1 ! ROUTINE VALUE:
;  392          0704  1 !       NONE
;  393          0705  1 !
;  394          0706  1 ! SIDE EFFECTS:
;  395          0707  1 !       FCB updated
;  396          0708  1 !
;  397          0709  1 !--
;  398          0710  1
;  399          0711  2 BEGIN
;  400          0712  2
;  401          0713  2 EXTERNAL
;  402          0714  2         PRIMARY_FCB     : REF BBLOCK;   ! FCB of file
;  403          0715  2
;  404          0716  2 IF .PRIMARY_FCB NEQ 0
;  405          0717  2 THEN PRIMARY_FCB[FCB$L_FILESIZE] = .SIZE;
;  406          0718  2
;  407          0719  1 END;                                     ! end of routine UPDATE_FILESIZE



                                        0000 00000          .ENTRY  UPDATE_FILESIZE, Save nothing                   ; 0679
                          50    0000G   CF   D0 00002       MOVL    PRIMARY_FCB, R0                                 ; 0716
                                        05   13 00007       BEQL    1$                                              ;
                    38    A0      04    AC   D0 00009       MOVL    SIZE, 56(R0)                                    ; 0717
                                        04 0000E 1$:        RET                                                     ; 0719

; Routine Size:  15 bytes,    Routine Base:  $CODE$ + 029C

;  408          0720  1
;  409          0721  1 END
```

```
;  410          0722  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                         683  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         34         0        1000        00:02.0



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

; Size:          683 code + 0 data bytes
; Run Time:          00:17.4
; Elapsed Time:      00:45.0
; Lines/CPU Min:      2489
; Lexemes/CPU-Min: 15493
; Memory Used:  252 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTFCB.LIS

DELFIL.LIS

DIRGET.LIS

EXTIDX.LIS

IODONE.LIS

LOCKDN.LIS

ENTER.LIS

GETREQ.LIS

GETTIM.LIS

DISPAT.LIS

INIFCP.LIS

DIRFCB.LIS

EXTHDR.LIS

DIRSCN.LIS

LOGDEL.LIS

LOCKDB.LIS

FIND.LIS

GETFIB.LIS

DIRACC.LIS

INIFCB.LIS

EXTDIR.LIS

EXTEND.LIS