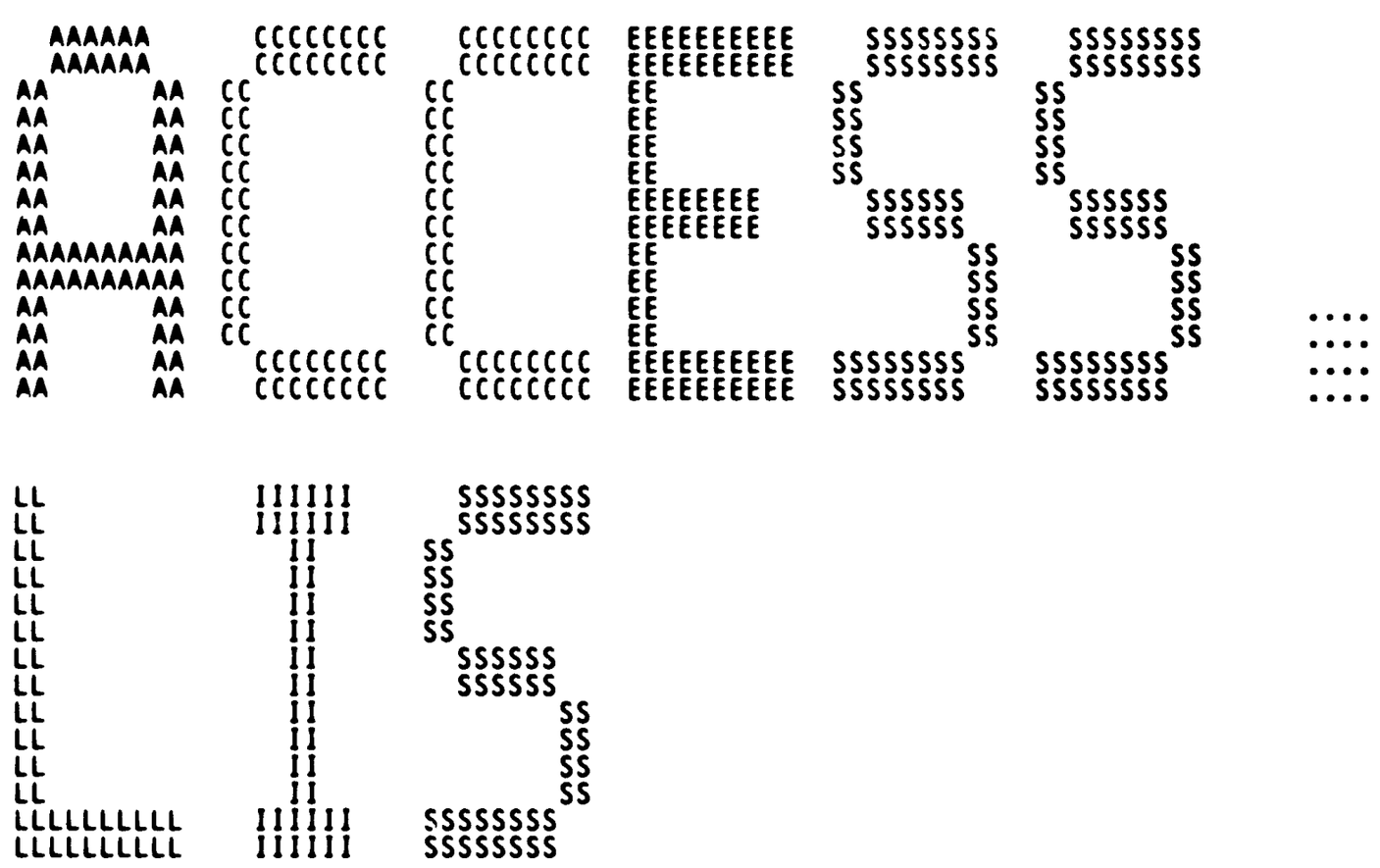

MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP

FILEID**ACCESS



```
;  1          0001  0
;  2          0002  0 MODULE ACCESS (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 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                            *
; 12          0012  1 !*                                                                    *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 18          0018  1 !*  TRANSFERRED.                                                       *
; 19          0019  1 !*                                                                    *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 22          0022  1 !*  CORPORATION.                                                       *
; 23          0023  1 !*                                                                    *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
; 26          0026  1 !*                                                                    *
; 27          0027  1 !*                                                                    *
; 28          0028  1 !**********************************************************************
; 29          0029  1
; 30          0030  1 !++
; 31          0031  1 !
; 32          0032  1 ! FACILITY:  MTAACP
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !       This module performs the access function for mtaacp.
; 36          0036  1 !
; 37          0037  1 ! ENVIRONMENT:
; 38          0038  1 !
; 39          0039  1 !       Starlet operating system, including privileged system services
; 40          0040  1 !       and internal exec routines.
; 41          0041  1 !
; 42          0042  1 !--
; 43          0043  1
; 44          0044  1 !
; 45          0045  1 !
; 46          0046  1 ! AUTHOR:  D. H. Gillespie,      CREATION DATE: 13-MAY-77 16:20
; 47          0047  1 !
; 48          0048  1 ! MODIFIED BY:
; 49          0049  1 !
; 50          0050  1 !       V03-002 MMD0275         Meg Dumont,     23-Mar-1984  9:49
; 51          0051  1 !               Change order of calling the routines CHECK_ACCESS and
; 52          0052  1 !               CHECK_FILE_ACC
; 53          0053  1 !
; 54          0054  1 !       V03-001 MMD0001         Meg Dumont,     5-Nov-1982  15:53
; 55          0055  1 !               Added support for setting user handling of EOT on a file access
; 56          0056  1 !
; 57          0057  1 !       V02-005 DMW00050        David Michael Walp      10-Nov-1981
```

```
;    58          0058  1 !               Return "NO SUCH FILE" error for when a "NO MORE FILES" error
;    59          0059  1 !               is returned and no files have been found.
;    60          0060  1 !
;    61          0061  1 !       V02-004 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    62          0062  1 !
;    63          0063  1 !       V02-003 SPR27361        Maria del C. Nasr       10-Jun-1980
;    64          0064  1 !               Fix design problem in which the IO_PACKET was being returned
;    65          0065  1 !               before IO was really completed. Now,a file will not be accessed
;    66          0066  1 !               until the tape mark is successfully skipped.  Also, the user
;    67          0067  1 !               labels code was eliminated, since it was never used.
;    68          0068  1 !
;    69          0069  1 !       A0002   MCN0001         Maria del C. Nasr       11-Sep-79   17:02
;    70          0070  1 !               Added argument to CHECK_FILE_ACC call to fix bug in
;    71          0071  1 !               "create if" function.
;    72          0072  1 !
;    73          0073  1 !**
;    74          0074  1
;    75          0075  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    76          0076  1
;    77          0077  1 REQUIRE 'SRC$:MTADEF.B32';
;    78          0461  1
;    79          0462  1 FORWARD ROUTINE
;    80          0463  1     MTA_ACCESS  : NOPRES NOVALUE,       ! main  control for access function
;    81          0464  1     CHECK_FIND  : COMMON_CALL,                  ! conditional directory search
;    82          0465  1     HANDLER;                    ! conditional handler to catch error exit
;    83          0466  1
```

```
;    85          0467  1 GLOBAL ROUTINE MTA_ACCESS : NOPRES NOVALUE =
;    86          0468  1
;    87          0469  1 !++
;    88          0470  1 !
;    89          0471  1 ! FUNCTIONAL DESCRIPTION:
;    90          0472  1 !       This is the main processing routine for mtaacp access function
;    91          0473  1 !
;    92          0474  1 ! CALLING SEQUENCE:
;    93          0475  1 !       MTA_ACCESS()
;    94          0476  1 !
;    95          0477  1 ! INPUT PARAMETERS:
;    96          0478  1 !       None
;    97          0479  1 !
;    98          0480  1 ! IMPLICIT INPUTS:
;    99          0481  1 !       CURRENT_UCB - address of current ucb
;   100          0482  1 !       CURRENT_WCB - address of current wcb
;   101          0483  1 !       CURRENT_VCB - address of current vcb
;   102          0484  1 !       IO_PACKET   - address of current io request packet
;   103          0485  1 !
;   104          0486  1 ! OUTPUT PARAMETERS:
;   105          0487  1 !       None
;   106          0488  1 !
;   107          0489  1 ! IMPLICIT OUTPUTS:
;   108          0490  1 !       None
;   109          0491  1 !
;   110          0492  1 ! ROUTINE VALUE:
;   111          0493  1 !       None
;   112          0494  1 !
;   113          0495  1 ! SIDE EFFECTS:
;   114          0496  1 !       Access request processed
;   115          0497  1 ! USER ERRORS
;   116          0498  1 !       SS$_FILALRACC - file already accessed
;   117          0499  1 !       SS$_BADPARAM - bad input parameters
;   118          0500  1 !
;   119          0501  1 !--
;   120          0502  1
;   121          0503  2     BEGIN
;   122          0504  2
;   123          0505  2     EXTERNAL REGISTER
;   124          0506  2         COMMON_REG;
;   125          0507  2
;   126          0508  2     LOCAL
;   127          0509  2         FUNCTION        : BLOCK [1],                    ! function and qualifiers
;   128          0510  2         PACKET  : REF BBLOCK,                   ! address of io packet
;   129          0511  2
;   130          0512  2         ! address of buffer descriptors
;   131          0513  2         !
;   132          0514  2         ABD     : REF BBLOCKVECTOR [, ABD$C_LENGTH],
;   133          0515  2         FIB     : REF BBLOCK;           ! address of file information block
;   134          0516  2
;   135          0517  2     EXTERNAL
;   136          0518  2         CURRENT_UCB     : REF BBLOCK,           ! address of current ucb
;   137          0519  2         CURRENT_WCB     : REF BBLOCK,   ! address of window control block
;   138          0520  2         IO_PACKET       : REF BBLOCK,                   ! address of io request packet
;   139          0521  2         USER_STATUS     : VECTOR;                       ! status to return to user
;   140          0522  2
;   141          0523  2     EXTERNAL ROUTINE
```

```
;  142          0524  2
;  143          0525  2          ! Make data base changes nec for access
;  144          0526  2          !
;  145          0527  2          ACCESS_FILE     : COMMON_CALL,
;  146          0528  2          ACCESS_NEW_FILE : COMMON_CALL,          ! access newly created file
;  147          0529  2
;  148          0530  2          ! Check volume protection on each file access
;  149          0531  2          !
;  150          0532  2          CHECK_ACCESS    : COMMON_CALL,
;  151          0533  2          CHECK_FILE_ACC  : COMMON_CALL,          ! check access to the file
;  152          0534  2          CLOSE_FILE      : L$CLOSE_FILE,         ! close out file
;  153          0535  2          GET_FIB : COMMON_CALL,                  ! get file information block
;  154          0536  2          MTA_CREATE,                             ! Main control for create function
;  155          0537  2          POSITION_BY_FID : COMMON_CALL,          ! position to file by fid
;  156          0538  2          READ_ATTRIBUTE  : COMMON_CALL,          ! read file attributes
;  157          0539  2          SET_USER_EOT    : COMMON_CALL,          ! call to set user eot handling
;  158          0540  2          SPACE_TM        : COMMON_CALL NOVALUE,  ! space tape mark
;  159          0541  2          START_VIO       : COMMON_CALL;          ! start up virtual io
;  160          0542  2
;  161          0543  2      ! Setup pointers
;  162          0544  2      !
;  163          0545  2      PACKET = .IO_PACKET;                        ! address of io packet
;  164          0546  2
;  165          0547  2      ! Address of buffer descriptors
;  166          0548  2      !
;  167          0549  2      ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  168          0550  2      FIB = GET_FIB(.ABD);                        ! get file information block
;  169          0551  2
;  170          0552  2      ! Get the function to be performed
;  171          0553  2      !
;  172          0554  2      FUNCTION = .PACKET[IRP$W_FUNC];
;  173          0555  2
;  174          0556  2      ! If anyone has a file accessed then another file can not be access nor can
;  175          0557  2      ! the tape be searched
;  176          0558  2      !
;  177          0559  2
;  178          0560  2      IF .CURRENT_WCB NEQ 0
;  179          0561  2          AND
;  180          0562  3          (.FUNCTION[IO$V_ACCESS]
;  181          0563  3          OR
;  182          0564  3          .FIB[FIB$W_DID_NUM] NEQ 0)
;  183          0565  2      THEN
;  184          0566  2          ERR_EXIT(SS$_FILALRACC);
;  185          0567  2
;  186          0568  2      IF .FIB[FIB$V_TRUNC]
;  187          0569  2          OR
;  188          0570  2          .FUNCTION[IO$V_DELETE]
;  189          0571  2      THEN
;  190          0572  2          ERR_EXIT(SS$_BADPARAM);
;  191          0573  2
;  192          0574  2      ! It is possible that a create was issued without an access in which case
;  193          0575  2      ! there is a partial file on this tape.  if the access is to a different
;  194          0576  2      ! file then close out the partial file before preceding
;  195          0577  2      !
;  196          0578  2
;  197          0579  2      IF .CURRENT_VCB[VCB$V_PARTFILE]             ! if there is a partial file
;  198          0580  2      THEN
```

```
;  199          0581  3          BEGIN
;  200          0582  3
;  201          0583  3          ! Is this a find or a request for a different file
;  202          0584  3          !
;  203          0585  3
;  204          0586  3          IF .FIB[FIB$W_DID_NUM] NEQ 0
;  205          0587  3              OR
;  206          0588  3              .(FIB[FIB$W_FID])<0, 32> NEQU .CURRENT_VCB[VCB$L_CUR_FID]
;  207          0589  3          THEN
;  208          0590  3              CLOSE_FILE()                ! close partial file before preceding
;  209          0591  3          ELSE
;  210          0592  4              BEGIN
;  211          0593  4
;  212          0594  4              ! if any attributes requested,
;  213          0595  4              !
;  214          0596  4
;  215          0597  4              IF .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
;  216          0598  4              THEN
;  217          0599  4                  READ_ATTRIBUTE(.ABD);           ! then read them
;  218          0600  4
;  219          0601  4              IF .FUNCTION[IO$V_ACCESS]
;  220          0602  4              THEN
;  221          0603  5                  BEGIN
;  222          0604  5
;  223          0605  5                  IF NOT .FIB[FIB$V_WRITE]
;  224          0606  5                  THEN
;  225          0607  5                      ERR_EXIT(SS$_BADPARAM);     ! must write new file
;  226          0608  5
;  227          0609  5                  ACCESS_NEW_FILE(.FIB, .PACKET, .ABD);   ! access partial file
;  228          0610  4                  END;
;  229          0611  4
;  230          0612  4              RETURN;
;  231          0613  4
;  232          0614  3              END;
;  233          0615  3
;  234          0616  2          END;
;  235          0617  2
;  236          0618  2          ! If file is accessed, the only function that can be performed is read
;  237          0619  2          ! attributes
;  238          0620  2          !
;  239          0621  2
;  240          0622  2          IF .CURRENT_WCB NEQ 0
;  241          0623  2          THEN
;  242          0624  3              BEGIN
;  243          0625  3
;  244          0626  3              IF .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
;  245          0627  3              THEN
;  246          0628  3                  READ_ATTRIBUTE(.ABD);
;  247          0629  3
;  248          0630  3              RETURN;
;  249          0631  3
;  250          0632  2              END;
;  251          0633  2
;  252          0634  2          ! If directory id given, find file by file name string
;  253          0635  2          !
;  254          0636  2
;  255          0637  2          IF .FIB[FIB$W_DID_NUM] NEQ 0
```

```
;  256          0638  2      THEN
;  257          0639  2
;  258          0640  2          IF NOT CHECK_FIND()
;  259          0641  2          THEN
;  260          0642  3              BEGIN
;  261          0643  3
;  262          0644  3              IF NOT .FUNCTION[IO$V_CREATE]
;  263          0645  3              THEN
;  264          0646  3                  ERR_EXIT(SS$_NOSUCHFILE);
;  265          0647  3
;  266          0648  3              USER_STATUS[0] = SS$_CREATED;
;  267          0649  3              RETURN MTA_CREATE();                    ! execute create function
;  268          0650  3
;  269          0651  2              END;
;  270          0652  2
;  271          0653  2      ! position volume set by fid
;  272          0654  2      !
;  273          0655  2      POSITION_BY_FID(.(FIB[FIB$W_FID])<0, 32>, .FIB[FIB$W_FID_RVN]);
;  274          0656  2
;  275          0657  2      IF .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
;  276          0658  2      THEN
;  277          0659  2          READ_ATTRIBUTE(.ABD);
;  278          0660  2
;  279          0661  2      ! if access requested
;  280          0662  2      !
;  281          0663  2
;  282          0664  2      IF .FUNCTION[IO$V_ACCESS]
;  283          0665  2      THEN
;  284          0666  3          BEGIN
;  285          0667  3          CHECK_FILE_ACC(1);              ! check access to file, (1)=mta_access
;  286          0668  3          CHECK_ACCESS(.FIB[FIB$V_WRITE]);        ! check volume access
;  287          0669  3
;  288          0670  3          IF .CURRENT_VCB[VCB$B_TM] EQLU 0
;  289          0671  3          THEN
;  290          0672  3              SPACE_TM(1);                        ! position to data
;  291          0673  3
;  292          0674  3          KERNEL_CALL(SET_USER_EOT, .FIB);        ! call to set user EOT handling
;  293        P 0675  3          KERNEL_CALL(ACCESS_FILE, .FIB[FIB$L_ACCTL], .PACKET[IRP$L_PID], 1,
;  294          0676  3              .FIB[FIB$V_WRITE], .ABD);
;  295          0677  3          KERNEL_CALL(START_VIO);                 ! start virtual io
;  296          0678  3          RETURN;
;  297          0679  3
;  298          0680  2          END;
;  299          0681  2
;  300          0682  1      END;


                                .TITLE  ACCESS
                                .IDENT  \V04-000\

                                .EXTRN  CURRENT_UCB, CURRENT_WCB
                                .EXTRN  IO_PACKET, USER_STATUS
                                .EXTRN  ACCESS_FILE, ACCESS_NEW_FILE
                                .EXTRN  CHECK_ACCESS, CHECK_FILE_ACC
                                .EXTRN  CLOSE_FILE, GET_FIB
                                .EXTRN  MTA_CREATE, POSITION_BY_FID
                                .EXTRN  READ_ATTRIBUTE, SET_USER_EOT
```

```
                                        .EXTRN  SPACE_TM, START_VIO
                                        .EXTRN  SYS$CMKRNL

                                        .PSECT  $CODE$,NOWRT,2

                          0000 00000    .ENTRY  MTA_ACCESS, Save nothing                ; 0467
                53        0000G CF DO 00002     MOVL    IO_PACKET, PACKET               ; 0545
                55          2C  B3 DO 00007     MOVL    @44(PACKET), ABD                ; 0549
                                55 DD 0000B     PUSHL   ABD                             ; 0550
          0000G CF                  01 FB 0000D     CALLS   #1, GET_FIB
                52                  50 DO 00012     MOVL    R0, FIB
                54          20      A3 3C 00015     MOVZWL  32(PACKET), FUNCTION            ; 0554
                          0000G CF  D5 00019     TSTL    CURRENT_WCB                     ; 0560
                                    0D 13 0001D     BEQL    2$
          05                    54  06 E0 0001F     BBS     #6, FUNCTION, 1$                ; 0562
                              0A    A2 B5 00023     TSTW    10(FIB)                         ; 0564
                                    04 13 00026     BEQL    2$
                            00A4    8F BF 00028 1$: CHMU    #164                            ; 0566
                        04    17    A2 E8 0002C 2$: BLBS    23(FIB), 3$                     ; 0568
          02                    54  08 E1 00030     BBC     #8, FUNCTION, 4$                ; 0570
                                    14 BF 00034 3$: CHMU    #20                             ; 0572
                        30      0B  AB E9 00036 4$: BLBC    11(CURRENT_VCB), 9$             ; 0579
                                0A  A2 B5 0003A     TSTW    10(FIB)                         ; 0586
                                    07 12 0003D     BNEQ    5$
                    24  AB      04  A2 D1 0003F     CMPL    4(FIB), 36(CURRENT_VCB)         ; 0588
                                    05 13 00044     BEQL    6$
                              0000G 30 00046 5$:    BSBW    CLOSE_FILE                      ; 0590
                                    1F 11 00049     BRB     9$
                        05      32  A3 B1 0004B 6$: CMPW    50(PACKET), #5                  ; 0597
                                    07 1B 0004F     BLEQU   7$
                                    55 DD 00051     PUSHL   ABD                             ; 0599
          0000G CF                  01 FB 00053     CALLS   #1, READ_ATTRIBUTE
          5E                    54  06 E1 00058 7$: BBC     #6, FUNCTION, 14$               ; 0601
                        02      01  A2 E8 0005C     BLBS    1(FIB), 8$                      ; 0605
                                    14 BF 00060     CHMU    #20                             ; 0607
                                    2C BB 00062 8$: PUSHR   #^M<R2,R3,R5>                   ; 0609
          0000G CF                  03 FB 00064     CALLS   #3, ACCESS_NEW_FILE
                                    04 00069     RET                                    ; 0592
                          0000G CF  D5 0006A 9$: TSTL    CURRENT_WCB                     ; 0622
                                    0F 13 0006E     BEQL    11$
                        05      32  A3 B1 00070     CMPW    50(PACKET), #5                  ; 0626
                                    01 1A 00074     BGTRU   10$
                                    04 00076     RET
                                    55 DD 00077 10$: PUSHL   ABD                            ; 0628
          0000G CF                  01 FB 00079     CALLS   #1, READ_ATTRIBUTE
                                    04 0007E     RET                                    ; 0624
                              0A    A2 B5 0007F 11$: TSTW    10(FIB)                         ; 0637
                                    1D 13 00082     BEQL    13$
          0000V CF                  00 FB 00084     CALLS   #0, CHECK_FIND                  ; 0640
                15                  50 E8 00089     BLBS    R0, 13$
                                    54 95 0008C     TSTB    FUNCTION                        ; 0644
                                    04 19 0008E     BLSS    12$
                            0910    8F BF 00090     CHMU    #2320                           ; 0646
          0000G CF          0619    8F 3C 00094 12$: MOVZWL  #1561, USER_STATUS             ; 0648
          0000G CF                  00 FB 0009B     CALLS   #0, MTA_CREATE                  ; 0649
                                    04 000A0     RET                                    ; 0655
                7E          08      A2 3C 000A1 13$: MOVZWL  8(FIB), -(SP)
```

```
                                                04   A2  DD 000A5          PUSHL   4(FIB)                                          ;
                                      0000G     CF       02  FB 000A8     CALLS   #2, POSITION_BY_FID                             ;
                                                05   32  A3  B1 000AD     CMPW    50(PACKET), #5                                  ; 0657
                                                         07  1B 000B1     BLEQU   14$                                             ;
                                                         55  DD 000B3     PUSHL   ABD                                             ; 0659
                                      0000G     CF       01  FB 000B5     CALLS   #1, READ_ATTRIBUTE                              ;
                                5C              54       06  E1 000BA 14$: BBC     #6, FUNCTION, 16$                               ; 0664
                                                         01  DD 000BE     PUSHL   #1                                              ; 0667
                                      0000G     CF       01  FB 000C0     CALLS   #1, CHECK_FILE_ACC                              ;
        7E      01      A2                  01       00  EF 000C5     EXTZV   #0, #1, 1(FIB), -(SP)                           ; 0668
                                      0000G     CF       01  FB 000CB     CALLS   #1, CHECK_ACCESS                                ;
                                                   2E    AB  95 000D0     TSTB    46(CURRENT_VCB)                                 ; 0670
                                                         07  12 000D3     BNEQ    15$                                             ;
                                                         01  DD 000D5     PUSHL   #1                                              ; 0672
                                      0000G     CF       01  FB 000D7     CALLS   #1, SPACE_TM                                    ;
                                                         52  DD 000DC 15$: PUSHL   FIB                                             ; 0674
                                                         01  DD 000DE     PUSHL   #1                                              ;
                                                         5E  DD 000E0     PUSHL   SP                                              ;
                                                0000G    CF  9F 000E2     PUSHAB  SET_USER_EOT                                    ;
                                   00000000G    9F       04  FB 000E6     CALLS   #4, @#SYS$CMKRNL                                ;
                                                         55  DD 000ED     PUSHL   ABD                                             ; 0676
        7E      01      A2                  01       00  EF 000EF     EXTZV   #0, #1, 1(FIB), -(SP)                           ;
                                                         01  DD 000F5     PUSHL   #1                                              ;
                                                   0C    A3  DD 000F7     PUSHL   12(PACKET)                                      ;
                                                         62  DD 000FA     PUSHL   (FIB)                                           ;
                                                         05  DD 000FC     PUSHL   #5                                              ;
                                                         5E  DD 000FE     PUSHL   SP                                              ;
                                                0000G    CF  9F 00100     PUSHAB  ACCESS_FILE                                     ;
                                   00000000G    9F       08  FB 00104     CALLS   #8, @#SYS$CMKRNL                                ;
                                                         7E  D4 0010B     CLRL    -(SP)                                           ; 0677
                                                         5E  DD 0010D     PUSHL   SP                                              ;
                                                0000G    CF  9F 0010F     PUSHAB  START_VIO                                       ;
                                   00000000G    9F       03  FB 00113     CALLS   #3, @#SYS$CMKRNL                                ;
                                                            04 0011A 16$: RET                                                    ; 0682

; '  tine Size:  283 bytes,    Routine Base:  $CODE$ + 0000

;  301          0683  1
```

```
;  303          0684  1 ROUTINE CHECK_FIND : COMMON_CALL =
;  304          0685  1
;  305          0686  1 !++
;  306          0687  1 !
;  307          0688  1 ! FUNCTIONAL DESCRIPTION:
;  308          0689  1 !
;  309          0690  1 !       This routine calls the directory search and intercepts any error
;  310          0691  1 !       exits to handle the create if non-existent function. If the search
;  311          0692  1 !       is successful, the routine returns success; if the search fails with
;  312          0693  1 !       no such file and the create subfunction bit is set, it returns failure;
;  313          0694  1 !       all other errors are resignaled.
;  314          0695  1 !
;  315          0696  1 !
;  316          0697  1 ! CALLING SEQUENCE:
;  317          0698  1 !       CHECK_FIND ()
;  318          0699  1 !
;  319          0700  1 ! INPUT PARAMETERS:
;  320          0701  1 !       None
;  321          0702  1 !
;  322          0703  1 ! IMPLICIT INPUTS:
;  323          0704  1 !       None
;  324          0705  1 !
;  325          0706  1 ! OUTPUT PARAMETERS:
;  326          0707  1 !       None
;  327          0708  1 !
;  328          0709  1 ! IMPLICIT OUTPUTS:
;  329          0710  1 !       None
;  330          0711  1 !
;  331          0712  1 ! ROUTINE VALUE:
;  332          0713  1 !       1 if find is successful
;  333          0714  1 !       0 if file is to be created
;  334          0715  1 !
;  335          0716  1 ! SIDE EFFECTS:
;  336          0717  1 !       None
;  337          0718  1 !
;  338          0719  1 !--
;  339          0720  1
;  340          0721  2     BEGIN
;  341          0722  2
;  342          0723  2     EXTERNAL REGISTER
;  343          0724  2         COMMON_REG;
;  344          0725  2
;  345          0726  2     BUILTIN
;  346          0727  2         FP;
;  347          0728  2
;  348          0729  2     EXTERNAL ROUTINE
;  349          0730  2         FIND_FILE       : COMMON_CALL;          ! find file in directory
;  350          0731  2
;  351          0732  2     ! Establish the condition handler and call FIND. If we hear from it we
;  352          0733  2     ! return true. Any signals cause either unwind or resignal.
;  353          0734  2     !
;  354          0735  2     .FP = HANDLER;
;  355          0736  2     FIND_FILE();
;  356          0737  2     RETURN 1;
;  357          0738  2
;  358          0739  1     END;                                        ! end of routine CHECK_FIND
```

```
                                                .EXTRN  FIND_FILE

                                        0000 00000 CHECK_FIND:
                                                .WORD   Save nothing                    ; 0684
                    6D          0000V   CF  9E 00002    MOVAB   HANDLER, (FP)           ; 0735
                0000G   CF              00  FB 00007    CALLS   #0, FIND_FILE           ; 0736
                    50                  01  D0 0000C    MOVL    #1, R0                  ; 0737
                                            04 0000F    RET                             ; 0739

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 011B
```

```
;  360     0740  1 ROUTINE HANDLER (SIGNAL, MECHANISM) =
;  361     0741  1
;  362     0742  1 !++
;  363     0743  1 !
;  364     0744  1 ! FUNCTIONAL DESCRIPTION:
;  365     0745  1 !
;  366     0746  1 !       This routine is the condition handler for the conditional find call.
;  367     0747  1 !       It intercepts the error exit from FIND and unwinds to CHECK_FIND's
;  368     0748  1 !       caller when appropriate.
;  369     0749  1 !
;  370     0750  1 !
;  371     0751  1 ! CALLING SEQUENCE:
;  372     0752  1 !       HANDLER (ARG1, ARG2)
;  373     0753  1 !
;  374     0754  1 ! INPUT PARAMETERS:
;  375     0755  1 !       ARG1: address of signal array
;  376     0756  1 !       ARG2: address of mechanism array
;  377     0757  1 !
;  378     0758  1 ! IMPLICIT INPUTS:
;  379     0759  1 !       None
;  380     0760  1 !
;  381     0761  1 ! OUTPUT PARAMETERS:
;  382     0762  1 !       None
;  383     0763  1 !
;  384     0764  1 ! IMPLICIT OUTPUTS:
;  385     0765  1 !       None
;  386     0766  1 !
;  387     0767  1 ! ROUTINE VALUE:
;  388     0768  1 !       SS$_RESIGNAL or none if unwind
;  389     0769  1 !
;  390     0770  1 ! SIDE EFFECTS:
;  391     0771  1 !       None
;  392     0772  1 !
;  393     0773  1 !--
;  394     0774  1
;  395     0775  2     BEGIN
;  396     0776  2
;  397     0777  2     MAP
;  398     0778  2         SIGNAL          : REF BBLOCK,                   ! signal arg array
;  399     0779  2         MECHANISM       : REF BBLOCK;                   ! mechanism arg array
;  400     0780  2
;  401     0781  2     EXTERNAL ROUTINE
;  402     0782  2         SYS$UNWIND      : ADDRESSING_MODE (ABSOLUTE);   ! system unwind service
;  403     0783  2
;  404     0784  2     EXTERNAL
;  405     0785  2         LOCAL_FIB       : BBLOCK;                       ! copy of user's fib
;  406     0786  2
;  407     0787  2     ! If the condition is change mode to user (error exit) and the status is
;  408     0788  2     ! no such file, cause an unwind to return 0 to the access main line.
;  409     0789  2     ! Otherwise, just resignal the condition.
;  410     0790  2     !
;  411     0791  2
;  412     0792  2     IF .SIGNAL[CHF$L_SIG_NAME] EQL SS$_CMODUSER
;  413     0793  2         AND
;  414     0794  2         .SIGNAL[CHF$L_SIG_ARG1] EQL SS$_NOSUCHFILE
;  415     0795  2     THEN
;  416     0796  3         BEGIN
```

```
;  417     0797  3
;  418     0798  3          IF .LOCAL_FIB[FIB$V_WILD]
;  419     0799  3          THEN
;  420     0800  4              BEGIN
;  421     0801  4              IF .LOCAL_FIB[FIB$L_WCC] NEQ 0
;  422     0802  4              THEN
;  423     0803  5                  BEGIN
;  424     0804  5                  SIGNAL[CHF$L_SIG_ARG1] = SS$_NOMOREFILES;
;  425     0805  5
;  426     0806  5                  ! clear context if at end of search
;  427     0807  5                  !
;  428     0808  5                  LOCAL_FIB[FIB$L_WCC] = 0;
;  429     0809  4                  END;
;  430     0810  4              END
;  431     0811  3          ELSE
;  432     0812  4              BEGIN
;  433     0813  4
;  434     0814  4              ! clear context if at end of search
;  435     0815  4              !
;  436     0816  4              LOCAL_FIB[FIB$L_WCC] = 0;
;  437     0817  4
;  438     0818  4              MECHANISM[CHF$L_MCH_SAVR0] = 0;
;  439     0819  4              SYS$UNWIND(0, 0);
;  440     0820  3              END;
;  441     0821  3
;  442     0822  2          END;
;  443     0823  2
;  444     0824  2      RETURN SS$_RESIGNAL;                ! status is irrelevant if unwinding
;  445     0825  2
;  446     0826  1      END;                                        ! end of routine HANDLER


                                                .EXTRN  SYS$UNWIND, LOCAL_FIB

                                    0004 00000 HANDLER:.WORD    Save R2                         ; 0740
                         52   0000G  CF   9E 00002          MOVAB   LOCAL_FIB+16, R2
                         50     04   AC   D0 00007          MOVL    SIGNAL, R0                      ; 0792
              00000424   8F     04   A0   D1 0000B          CMPL    4(R0), #1060
                                     2E   12 00013          BNEQ    2$
              00000910   8F     08   A0   D1 00015          CMPL    8(R0), #2320                    ; 0794
                                     24   12 0001D          BNEQ    2$
                         0E     05   A2   E9 0001F          BLBC    LOCAL_FIB+21, 1$                ; 0798
                                     62   D5 00023          TSTL    LOCAL_FIB+16                    ; 0801
                                     1C   13 00025          BEQL    2$
                   08    A0   0930   8F   3C 00027          MOVZWL  #2352, 8(R0)                    ; 0804
                                     62   D4 0002D          CLRL    LOCAL_FIB+16                    ; 0808
                                     12   11 0002F          BRB     2$                              ; 0798
                                     62   D4 00031 1$:      CLRL    LOCAL_FIB+16                    ; 0816
                         50     08   AC   D0 00033          MOVL    MECHANISM, R0                   ; 0818
                                0C   A0   D4 00037          CLRL    12(R0)
                                     7E   7C 0003A          CLRQ    -(SP)                           ; 0819
              00000000G  9F          02   FB 0003C          CALLS   #2, @#SYS$UNWIND
                         50   0918   8F   3C 00043 2$:      MOVZWL  #2328, R0                       ; 0824
                                     04   00048          RET                                     ; 0826

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 012B
```

```
;  447          0827  1 END
;  448          0828  1
;  449          0829  0 ELUDOM



;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         372  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        35         0        1000         00:01.9



;                                     COMMAND QUALIFIERS

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

; Size:          372 code + 0 data bytes
; Run Time:        00:11.9
; Elapsed Time:    00:43.0
; Lines/CPU Min:    4179
; Lexemes/CPU-Min: 19709
; Memory Used:  142 pages
; Compilation Complete
```

CALDAYNO
LIS

SYSFAC
DAT

ACPCTR
LIS

ALLOCB
LIS

SYSMSG
LIS

CHKACC
LIS

MTAACP

ACCESS
LIS

MTAAACP
MAP

ACCFL
LIS

CALCTV
LIS

BLOCK
LIS

CLOSFL
LIS

MTADEF
B32 -

CHKDMO
LIS