
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

```
  CCCCCCCC  RRRRRRRR    EEEEEEEEEE    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
  CCCCCCCC  RRRRRRRR    EEEEEEEEEE    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
CC          RR      RR  EE          AA      AA      TT      EE
CC          RR      RR  EE          AA      AA      TT      EE
CC          RR      RR  EE          AA      AA      TT      EE
CC          RR      RR  EE          AA      AA      TT      EE
CC          RRRRRRRR    EEEEEEEE    AA      AA      TT      EEEEEEEE
CC          RRRRRRRR    EEEEEEEE    AA      AA      TT      EEEEEEEE
CC          RR  RR      EE          AAAAAAAAAA      TT      EE
CC          RR  RR      EE          AAAAAAAAAA      TT      EE
CC          RR    RR    EE          AA      AA      TT      EE              ....
CC          RR    RR    EE          AA      AA      TT      EE              ....
  CCCCCCCC  RR      RR  EEEEEEEEEE  AA      AA      TT      EEEEEEEEEE      ....
  CCCCCCCC  RR      RR  EEEEEEEEEE  AA      AA      TT      EEEEEEEEEE      ....

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
;     2          0002  0 MODULE CREATE (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 !
;    36          0036  1 !       This module executes the create function
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       Starlet operating system, including privileged system services
;    41          0041  1 !       and internal exec routines.
;    42          0042  1 !
;    43          0043  1 !--
;    44          0044  1
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  3-JUN-77
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-002 MMD0153         Meg Dumont,     26-Apr-1983  9:08
;    52          0052  1 !               Change reference to 80 to the symbol ANSI_LBLSZ. Add HDR4
;    53          0053  1 !               processsing which includes: 1) Changing the number of HDRs
;    54          0054  1 !               to look for when positioning to write a file. 2) Add the
;    55          0055  1 !               writing of HDR4 to the routine WRITE_HEADERS.
;    56          0056  1 !
;    57          0057  1 !       V03-001 MMD0001         Meg Dumont,     5-Nov-1982  16:22
```

```
;    58          0058  1 !                 Allow the setting of user handled EOT on a file create.
;    59          0059  1 !                 Add routine to set user handled EOT.
;    60          0060  1 !
;    61          0061  1 !       V02-012 DMW0019         David Michael Walp      11-Nov-1980
;    62          0062  1 !                 New BLISS compiler, FUNCTION declaration changed from
;    63          0063  1 !                 BBLOCK to BLOCK.  Old compiler use to give a longword
;    64          0064  1 !                 with a declaration of "BBLOCK [1]".
;    65          0065  1 !
;    66          0066  1 !       V02-011 MCN0018         Maria del C. Nasr       26-Aug-1980
;    67          0067  1 !                 Implement bug fix described in MCN0017
;    68          0068  1 !
;    69          0069  1 !       V02-010 MCN0017         Maria del C. Nasr       29-Jul-1980
;    70          0070  1 !                 Check if EOT was sensed after writing the header labels of the
;    71          0071  1 !                 file being created, instead of waiting for virtual IO to lock
;    72          0072  1 !                 the error.  Also, add a kernel mode routine which will update
;    73          0073  1 !                 the partfile flag in the VCB, to account for cancel IO.  This
;    74          0074  1 !                 fixes a problem in which creating empty files at the end of a
;    75          0075  1 !                 tape will cause a runaway tape situation.
;    76          0076  1 !
;    77          0077  1 !       V02-009 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    78          0078  1 !
;    79          0079  1 !       V02-008 SPR27361        Maria del C. Nasr       10-Jun-1980
;    80          0080  1 !                 The file should be accessed after the tape is positioned
;    81          0081  1 !                 right before the data starts, to make sure all IO has been
;    82          0082  1 !                 completed.  Also, eliminate user labels code.
;    83          0083  1 !
;    84          0084  1 !       A0007   MCN0011         Maria del C. Nasr       04-Feb-1980  9:05
;    85          0085  1 !                 Clear logical end of volume flag after spacing tape mark
;    86          0086  1 !                 back to fix bug.
;    87          0087  1 !
;    88          0088  1 !       A0006   MCN0003         Maria del C. Nasr       28-Sep-79   10:27
;    89          0089  1 !                 Add HDR3 processing
;    90          0090  1 !
;    91          0091  1 !       A0005   MCN0001         Maria del C. Nasr       21-Sep-79   10:15
;    92          0092  1 !                 Added argument to CHECK_FILE_ACC call to fix bug in
;    93          0093  1 !                 "create if" function.
;    94          0094  1 !
;    95          0095  1 !
;    96          0096  1 !**
;    97          0097  1
;    98          0098  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    99          0099  1
;   100          0100  1 REQUIRE 'SRC$:MTADEF.B32';
;   101          0484  1
;   102          0485  1 FORWARD ROUTINE
;   103          0486  1     MTA_CREATE          : NOPRES NOVALUE,       ! main routine to create a file
;   104          0487  1     ACCESS_NEW_FILE     : COMMON_CALL NOVALUE,  ! access a newly created file
;   105          0488  1     INIT_NEW_FILE       : COMMON_CALL NOVALUE,  ! initialize new file
;   106          0489  1     SET_USER_EOT        : COMMON_CALL NOVALUE,  ! set user wot handling for file
;   107          0490  1     WRITE_HEADERS       : NOVALUE L$WRITE_HEADER,       ! write hdr1, hdr2, and hdr3,hdr4
;   108          0491  1     UPD_PARTFILE        : COMMON_CALL NOVALUE;  ! update partfile flag in the VCB
;   109          0492  1
;   110          0493  1 EXTERNAL ROUTINE
;   111          0494  1     ACCESS_FILE         : COMMON_CALL,          ! access file
;   112          0495  1     CHECK_ACCESS        : COMMON_CALL,          ! check access to volume
;   113          0496  1     CHECK_FILE_ACC      : COMMON_CALL,          ! check access to file
;   114          0497  1     CLOSE_FILE          : L$CLOSE_FILE,         ! close file
```

```
;  115     0498  1      FORMAT_FID              : COMMON_CALL,                  ! format file identifier
;  116     0499  1      FORMAT_HDRS             : COMMON_CALL,                  ! format headers
;  117     0500  1      GET_FIB                 : COMMON_CALL,                  ! get copy of user's fib
;  118     0501  1      GET_START_HDR           : L$GET_START_HDR,              ! get header to start with
;  119     0502  1      MOUNT_VOL               : COMMON_CALL,                  ! mount a volume
;  120     0503  1      NEXT_VOL_WRITE          : L$NEXT_VOL_WRIT,              ! get next volume for write
;  121     0504  1      POSITION_TO_END         : COMMON_CALL,                  ! position to end of vol set
;  122     0505  1      READ_HDR                : COMMON_CALL,                  ! read header
;  123     0506  1      SPACE                   : COMMON_CALL,                  ! space given number of blocks
;  124     0507  1      SPACE_TM                : COMMON_CALL,                  ! space given number of tape marks
;  125     0508  1      START_VIO               : COMMON_CALL,                  ! start virtual io
;  126     0509  1      SYS$FAO                 : ADDRESSING_MODE (ABSOLUTE),   ! format ascii output
;  127     0510  1      UPDVCB_LEOV             : COMMON_CALL,                  ! to clear logical end of volume bit
;  128     0511  1      WRITE_BLOCK             : COMMON_CALL,                  ! write tape block
;  129     0512  1      WRITE_TM                : L$WRITE_TM;                   ! write tape mark
;  130     0513  1
;  131     0514  1  EXTERNAL
;  132     0515  1      CURRENT_WCB             : REF BBLOCK,                   ! address of current window control block
;  133     0516  1      IO_PACKET               : REF BBLOCK,                   ! address of io request packet
;  134     0517  1      IO_STATUS,                                              ! IO status returned
;  135     0518  1      HDR1                    : REF BBLOCK,                   ! address of HDR1(EOF1) label
;  136     0519  1      HDR2                    : REF BBLOCK,                   ! address of HDR2(EOF2) label
;  137     0520  1      HDR3                    : REF BBLOCK,                   ! address of HDR3 label(EOF3)
;  138     0521  1      HDR4                    : REF BBLOCK,                   ! address of HDR4 label(EOF4)
;  139     0522  1      USER_STATUS             : VECTOR;                       ! status to return the user
;  140     0523  1
```

```
;  142     0524  1 GLOBAL ROUTINE MTA_CREATE : NOPRES NOVALUE =
;  143     0525  1
;  144     0526  1 !++
;  145     0527  1 !
;  146     0528  1 ! FUNCTIONAL DESCRIPTION:
;  147     0529  1 !       This is the main processing routine for MTAACP create function
;  148     0530  1 !
;  149     0531  1 ! CALLING SEQUENCE:
;  150     0532  1 !       MTA_CREATE()
;  151     0533  1 !
;  152     0534  1 ! INPUT PARAMETERS:
;  153     0535  1 !       none
;  154     0536  1 !
;  155     0537  1 ! IMPLICIT INPUTS:
;  156     0538  1 !       HDR1      - address hdr1(eof1) label
;  157     0539  1 !       HDR2      - address hdr2(eof2) label
;  158     0540  1 !       HDR3      - address hdr3(eof3) label
;  159     0541  1 !       HDR4      - address hdr4(eof4) label
;  160     0542  1 !       IO_PACKET - address of current io request packet
;  161     0543  1 !       CURRENT_VCB - address of current vcb
;  162     0544  1 !
;  163     0545  1 ! OUTPUT PARAMETERS:
;  164     0546  1 !       none
;  165     0547  1 !
;  166     0548  1 ! IMPLICIT OUTPUTS:
;  167     0549  1 !       The header label set is written and the access to the file made if requested
;  168     0550  1 !
;  169     0551  1 ! ROUTINE VALUE:
;  170     0552  1 !       none
;  171     0553  1 !
;  172     0554  1 ! SIDE EFFECTS:
;  173     0555  1 !       none
;  174     0556  1 !
;  175     0557  1 ! USER ERRORS:
;  176     0558  1 !       SS$_BADPARAM    - bad input parameters to create or must ask to write file
;  177     0559  1 !       SS$_FILALRACC   - only one file at a time open on magnetic tape
;  178     0560  1 !       SS$_FILENOTEXP  - the file about to be overlayed is not expired
;  179     0561  1 !       SS$_NOPRIV      - the user does not have necessary privileges
;  180     0562  1 !--
;  181     0563  1
;  182     0564  2     BEGIN
;  183     0565  2
;  184     0566  2     EXTERNAL REGISTER
;  185     0567  2         COMMON_REG;
;  186     0568  2
;  187     0569  2     LOCAL
;  188     0570  2         INC_SEQ_NUM,                        ! indicator for init_new_file routine
;  189     0571  2         PACKET          : REF BBLOCK,       ! address of current io request packet
;  190     0572  2
;  191     0573  2         ! address of buffer descriptors
;  192     0574  2         !
;  193     0575  2         ABD             : REF BBLOCKVECTOR [, ABD$C_LENGTH],
;  194     0576  2         FIB             : REF BBLOCK,       ! address of copy of user's fib
;  195     0577  2         FUNCTION        : BLOCK [1];        ! io function
;  196     0578  2
;  197     0579  2     ! setup pointers to interesting structures
;  198     0580  2     !
```

```
;  199          0581  2      PACKET = .IO_PACKET;
;  200          0582  2
;  201          0583  2      ! address of buffer descriptors
;  202          0584  2      !
;  203          0585  2      ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  204          0586  2      FIB = GET_FIB(.ABD);        ! get copy of user's file information block
;  205          0587  2
;  206          0588  2      ! get io function code
;  207          0589  2      !
;  208          0590  2      FUNCTION = .PACKET[IRP$W_FUNC];
;  209          0591  2
;  210          0592  2      ! is the function requested a valid one?
;  211          0593  2      !
;  212          0594  2
;  213          0595  2      IF .FIB[FIB$V_TRUNC] OR .FUNCTION[IO$V_DELETE]
;  214          0596  3          OR ( NOT .FUNCTION[IO$V_CREATE])
;  215          0597  2          AND
;  216          0598  3          (.FIB[FIB$V_EXTEND] OR .PACKET[IRP$W_BCNT] GTRU ABD$C_ATTRIB
;  217          0599  3                  OR .FUNCTION[IO$V_ACCESS])
;  218          0600  2          OR NOT .FIB[FIB$V_WRITE]
;  219          0601  2          AND
;  220          0602  2          .FUNCTION[IO$V_ACCESS]
;  221          0603  2      THEN
;  222          0604  2          ERR_EXIT(SS$_BADPARAM);
;  223          0605  2
;  224          0606  2      ! make sure that a create is not done on a volume set that has an out
;  225          0607  2      ! standing access
;  226          0608  2      !
;  227          0609  2
;  228          0610  2      IF .CURRENT_WCB NEQ 0
;  229          0611  2          AND
;  230          0612  2          .FUNCTION[IO$V_CREATE]
;  231          0613  2      THEN
;  232          0614  2          ERR_EXIT(SS$_FILALRACC);
;  233          0615  2
;  234          0616  2      ! now for the create
;  235          0617  2      !
;  236          0618  2
;  237          0619  2      IF .FUNCTION[IO$V_CREATE]
;  238          0620  2      THEN
;  239          0621  3          BEGIN
;  240          0622  3
;  241          0623  3          ! if there is a partial file then it  is closed.  if the user wants to
;  242          0624  3          ! access a partial file, the access function io must be used
;  243          0625  3          !
;  244          0626  3
;  245          0627  3          IF .CURRENT_VCB[VCB$V_PARTFILE]
;  246          0628  3          THEN
;  247          0629  4              CLOSE_FILE();
;  248          0630  3
;  249          0631  3          CHECK_ACCESS(1);        ! MUST HAVE WRITE PRIVILEGE JUST TO CREATE
;  250          0632  3
;  251          0633  3          ! now position the tape as requested unless this is a create if not
;  252          0634  3          ! found
;  253          0635  3          !
;  254          0636  3
;  255          0637  3          IF .PACKET[IRP$V_FCODE] EQL IO$_ACCESS  ! if create if not found
```

```
;  256     0638  3            OR
;  257     0639  4            NOT (.FIB[FIB$V_REWIND]
;  258     0640  4            OR
;  259     0641  4            .FIB[FIB$V_CURPOS])
;  260     0642  3        THEN
;  261     0643  3            POSITION_TO_END()
;  262     0644  3        ELSE
;  263     0645  4            BEGIN
;  264     0646  4
;  265     0647  4            IF .FIB[FIB$V_REWIND]
;  266     0648  4            THEN
;  267     0649  5                BEGIN
;  268     0650  5                MOUNT_VOL(1, $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_LBLCHECK));
;  269     0651  5
;  270     0652  5                IF NOT READ_HDR()
;  271     0653  5                THEN
;  272     0654  5                    ERR_EXIT(SS$_TAPEPOSLOST);
;  273     0655  5
;  274     0656  5                END
;  275     0657  4            ELSE
;  276     0658  4
;  277     0659  4                ! create function with positioning indicators
;  278     0660  4                !
;  279     0661  4                GET_START_HDR();
;  280     0662  4
;  281     0663  3            END;
;  282     0664  3
;  283     0665  3        ! check expiration on overlayed files and position tape
;  284     0666  3        !
;  285     0667  3
;  286     0668  3        IF NOT .CURRENT_VCB[VCB$V_LOGICEOVS]
;  287     0669  3        THEN
;  288     0670  4            BEGIN                   ! overlaying file if not at end of volume set
;  289     0671  4            INC_SEQ_NUM = 0;        ! do not increment file number
;  290     0672  4            CHECK_FILE_ACC(0);      ! can file be overlayed? (0)=mta_create
;  291     0673  4
;  292     0674  4            ! position to where file is to be written
;  293     0675  4            !
;  294     0676  4
;  295     0677  4            IF .CURRENT_VCB[VCB$B_TM] EQL 0
;  296     0678  4            THEN
;  297     0679  4
;  298     0680  5                IF (.HDR2[HD2$L_HD2LID] EQL 'HDR2')
;  299     0681  4                    AND
;  300     0682  5                    (.CURRENT_VCB[VCB$V_STARFILE])
;  301     0683  4                THEN
;  302     0684  4                    IF .CURRENT_VCB[VCB$B_LBLCNT] GTR 3
;  303     0685  4                        THEN SPACE(-4)          ! hdr1,hdr2,hdr3, hdr4 (or user label)
;  304     0686  4                        ELSE SPACE(-3)          ! hdr1, hdr2, and hdr3 lbls
;  305     0687  4                ELSE
;  306     0688  4                    SPACE(-2)           ! not VMS file, space hdr1 and hdr2
;  307     0689  4            ELSE
;  308     0690  5                BEGIN
;  309     0691  5                SPACE_TM(-1);                   ! backspace tm
;  310     0692  5
;  311     0693  5                IF .HDR2[HD2$L_HD2LID] EQL 'HDR2'       ! check if hdr2 found
;  312     0694  5                THEN
```

```
;  313     0695  5                SPACE(-2)                      ! backspace hdr1 and hdr2
;  314     0696  5            ELSE
;  315     0697  5                SPACE(-1);                     ! no hdr2, backspace hdr1 only
;  316     0698  5
;  317     0699  4                END;
;  318     0700  4
;  319     0701  4            END
;  320     0702  3        ELSE
;  321     0703  4            BEGIN
;  322     0704  4            SPACE_TM(-1);                      ! logical end indicated by tape mark
;  323     0705  4            KERNEL_CALL(UPDVCB_LEOV, 0);       ! not at end of vol anymore
;  324     0706  4            INC_SEQ_NUM = 1;                   ! increment file number
;  325     0707  3            END;
;  326     0708  3
;  327     0709  3        ! initialize new file and write headers
;  328     0710  3        !
;  329     0711  3        FORMAT_HDRS();
;  330     0712  3        KERNEL_CALL(INIT_NEW_FILE, .INC_SEQ_NUM);
;  331     0713  3        WRITE_HEADERS();                       ! write hdr1, hdr2, hdr3 and hdr4
;  332     0714  3
;  333     0715  3        ! return fid to user in fib
;  334     0716  3        !
;  335     0717  3        FORMAT_FID(FIB[FIB$W_FID_NUM]);
;  336     0718  3        FIB[FIB$W_FID_RVN] = .CURRENT_VCB[VCB$B_CUR_RVN];
;  337     0719  3
;  338     0720  3        ! now if newly created file should be accessed do so
;  339     0721  3        !
;  340     0722  3
;  341     0723  3        IF .FUNCTION[IO$V_ACCESS]
;  342     0724  3        THEN
;  343     0725  3            ACCESS_NEW_FILE(.FIB, .PACKET, .ABD);
;  344     0726  3
;  345     0727  2        END;
;  346     0728  2
;  347     0729  1    END;                                       ! end of routine


                .TITLE  CREATE
                .IDENT  \V04-000\

                .EXTRN  ACCESS_FILE, CHECK_ACCESS
                .EXTRN  CHECK_FILE_ACC, CLOSE_FILE
                .EXTRN  FORMAT_FID, FORMAT_HDRS
                .EXTRN  GET_FIB, GET_START_HDR
                .EXTRN  MOUNT_VOL, NEXT_VOL_WRITE
                .EXTRN  POSITION_TO_END
                .EXTRN  READ_HDR, SPACE
                .EXTRN  SPACE_TM, START_VIO
                .EXTRN  SYS$FAO, UPDVCB_LEOV
                .EXTRN  WRITE_BLOCK, WRITE_TM
                .EXTRN  CURRENT_WCB, IO_PACKET
                .EXTRN  IO_STATUS, HDR1
                .EXTRN  HDR2, HDR3, HDR4
                .EXTRN  USER_STATUS, SYS$CMKRNL

                .PSECT  $CODE$,NOWRT,2
```

```
                                    0000 00000          .ENTRY  MTA_CREATE, Save nothing                        ; 0524
                              53    0000G CF   D0 00002       MOVL    IO_PACKET, PACKET                       ; 0581
                              56      2C  B3   D0 00007       MOVL    @44(PACKET), ABD                        ; 0585
                                          56   DD 0000B       PUSHL   ABD                                     ; 0586
                        0000G CF                  01   FB 0000D       CALLS   #1, GET_FIB                     ;
                              52                  50   D0 00012       MOVL    R0, FIB                         ;
                              55      20  A3   3C 00015       MOVZWL  32(PACKET), FUNCTION                    ; 0590
                              1F      17  A2   E8 00019       BLBS    23(FIB), 2$                             ; 0595
                    1B        55          08   E0 0001D       BBS     #8, FUNCTION, 2$                        ;
                                          55   95 00021       TSTB    FUNCTION                                ; 0596
                                          0F   19 00023       BLSS    1$                                      ;
                                      16  A2   95 00025       TSTB    22(FIB)                                 ; 0598
                                          12   19 00028       BLSS    2$                                      ;
                              05      32  A3   B1 0002A       CMPW    50(PACKET), #5                          ;
                                          0C   1A 0002E       BGTRU   2$                                      ;
                    08        55          06   E0 00030       BBS     #6, FUNCTION, 2$                        ; 0599
                              06      01  A2   E8 00034 1$:    BLBS    1(FIB), 3$                              ; 0600
                    02        55          06   E1 00038       BBC     #6, FUNCTION, 3$                        ; 0602
                                          14   BF 0003C 2$:    CHMU    #20                                     ; 0604
                                    0000G CF   D5 0003E 3$:    TSTL    CURRENT_WCB                             ; 0610
                                          08   13 00042       BEQL    4$                                      ;
                                          55   95 00044       TSTB    FUNCTION                                ; 0612
                                          04   18 00046       BGEQ    4$                                      ;
                                    00A4  8F   BF 00048       CHMU    #164                                    ; 0614
                                          55   95 0004C 4$:    TSTB    FUNCTION                                ; 0619
                                          01   19 0004E       BLSS    5$                                      ;
                                               04 00050       RET                                             ;
                              03      0B  AB   E9 00051 5$:    BLBC    11(CURRENT_VCB), 6$                     ; 0627
                                    0000G 30   00055       BSBW    CLOSE_FILE                              ; 0629
                                          01   DD 00058 6$:    PUSHL   #1                                      ; 0631
                        0000G CF                  01   FB 0005A       CALLS   #1, CHECK_ACCESS                ;
32        20  A3                  06              00   ED 0005F       CMPZV   #0, #6, 32(PACKET), #50         ; 0637
                                          08   13 00065       BEQL    7$                                      ;
                    0F                62    03   E0 00067       BBS     #3, (FIB), 9$                           ; 0639
                    07                62    04   E0 0006B       BBS     #4, (FIB), 8$                           ; 0641
                        0000G CF                  00   FB 0006F 7$:    CALLS   #0, POSITION_TO_END             ; 0643
                                          1E   11 00074       BRB     11$                                     ;
                    17                62    03   E1 00076 8$:    BBC     #3, (FIB), 10$                          ; 0647
                                          03   DD 0007A 9$:    PUSHL   #3                                      ; 0650
                                          01   DD 0007C       PUSHL   #1                                      ;
                        0000G CF                  02   FB 0007E       CALLS   #2, MOUNT_VOL                   ;
                        0000G CF                  00   FB 00083       CALLS   #0, READ_HDR                    ; 0652
                                    09            50   E8 00088       BLBS    R0, 11$                         ;
                                          0224 8F   BF 0008B       CHMU    #548                                    ; 0654
                                          03   11 0008F       BRB     11$                                     ; 0647
                                    0000G 30   00091 10$:    BSBW    GET_START_HDR                           ; 0661
                    4F        0B  AB              01   E0 00094 11$:    BBS     #1, 11(CURRENT_VCB), 16$        ; 0668
                                          54   D4 00099       CLRL    INC_SEQ_NUM                             ; 0671
                                          7E   D4 0009B       CLRL    -(SP)                                   ; 0672
                        0000G CF                  01   FB 0009D       CALLS   #1, CHECK_FILE_ACC              ;
                                      2E  AB   95 000A2       TSTB    46(CURRENT_VCB)                         ; 0677
                                          24   12 000A5       BNEQ    14$                                     ;
                 32524448 8F        0000G DF   D1 000A7       CMPL    @HDR2, #844252232                       ; 0680
                                          14   12 000B0       BNEQ    13$                                     ;
                              10      2D  AB   E9 000B2       BLBC    45(CURRENT_VCB), 13$                    ; 0682
                              03      48  AB   91 000B6       CMPB    72(CURRENT_VCB), #3                     ; 0684
                                          05   1B 000BA       BLEQU   12$                                     ;
```

```
                7E              04   CE 000BC          MNEGL   #4, -(SP)                          ; 0685
                                20   11 000BF          BRB     15$
                7E              03   CE 000C1 12$:     MNEGL   #3, -(SP)                          ; 0686
                                1B   11 000C4          BRB     15$
                7E              02   CE 000C6 13$:     MNEGL   #2, -(SP)                          ; 0688
                                16   11 000C9          BRB     15$
                7E              01   CE 000CB 14$:     MNEGL   #1, -(SP)                          ; 0691
          0000G CF              01   FB 000CE          CALLS   #1, SPACE_TM
 32524448   8F      0000G       DF   D1 000D3          CMPL    @HDR2, #824252232                  ; 0693
                                E8   13 000DC          BEQL    13$
                7E              01   CE 000DE          MNEGL   #1, -(SP)                          ; 0697
          0000G CF              01   FB 000E1 15$:     CALLS   #1, SPACE
                                1B   11 000E6          BRB     17$                                ; 0668
                7E              01   CE 000E8 16$:     MNEGL   #1, -(SP)                          ; 0704
          0000G CF              01   FB 000EB          CALLS   #1, SPACE_TM
                7E              01   7D 000F0          MOVQ    #1, -(SP)                          ; 0705
                                5E   DD 000F3          PUSHL   SP
                        0000G   CF   9F 000F5          PUSHAB  UPDVCB_LEOV
 00000000G      9F              04   FB 000F9          CALLS   #4, @#SYS$CMKRNL
                54              01   D0 00100          MOVL    #1, INC_SEQ_NUM                    ; 0706
          0000G CF              00   FB 00103 17$:     CALLS   #0, FORMAT_ADRS                    ; 0711
                                54   DD 00108          PUSHL   INC_SEQ_NUM                        ; 0712
                                01   DD 0010A          PUSHL   #1
                                5E   DD 0010C          PUSHL   SP
                        0000V   CF   9F 0010E          PUSHAB  INIT_NEW_FILE
 00000000G      9F              04   FB 00112          CALLS   #4, @#SYS$CMKRNL
                          0000V 30   00119          BSBW    WRITE_HEADERS                      ; 0713
                     04   A2   9F   0011C          PUSHAB  4(FIB)                             ; 0717
                0000G   CF        01   FB 0011F          CALLS   #1, FORMAT_FID
                  08    A2        2F   AB 9B 00124       MOVZBW  47(CURRENT_VCB), 8(FIB)            ; 0718
          09            55        06   E1 00129          BBC     #6, FUNCTION, 18$                  ; 0723
                              004C  8F   BB 0012D          PUSHR   #^M<R2,R3,R6>                      ; 0725
                0000V   CF        03   FB 00131          CALLS   #3, ACCESS_NEW_FILE
                                       04 00136 18$:     RET                                        ; 0729

; Routine Size:  311 bytes,    Routine Base:  $CODE$ + 0000

;  348          0730  1
```

```
;  350     0731  1 GLOBAL ROUTINE ACCESS_NEW_FILE (FIB, PACKET, ABD) : COMMON_CALL NOVALUE =
;  351     0732  1
;  352     0733  1 !++
;  353     0734  1 !
;  354     0735  1 ! FUNCTIONAL DESCRIPTION:
;  355     0736  1 !       this routine accesses a newly created file
;  356     0737  1 !
;  357     0738  1 ! CALLING SEQUENCE:
;  358     0739  1 !       access_new_file(arg1,arg2,arg3)
;  359     0740  1 !
;  360     0741  1 ! INPUT PARAMETERS:
;  361     0742  1 !       arg1 - address of copy of user file identification block
;  362     0743  1 !       arg2 - address of current io request packet
;  363     0744  1 !       arg3 - address of buffer descriptor vectors
;  364     0745  1 !
;  365     0746  1 ! IMPLICIT INPUTS:
;  366     0747  1 !       none
;  367     0748  1 !
;  368     0749  1 ! OUTPUT PARAMETERS:
;  369     0750  1 !       none
;  370     0751  1 !
;  371     0752  1 ! IMPLICIT OUTPUTS:
;  372     0753  1 !       Changes made to system data base to allow virtual io.
;  373     0754  1 !       Tape mark is written.
;  374     0755  1 !
;  375     0756  1 ! ROUTINE VALUE:
;  376     0757  1 !       none
;  377     0758  1 !
;  378     0759  1 ! SIDE EFFECTS:
;  379     0760  1 !       none
;  380     0761  1 !
;  381     0762  1 !--
;  382     0763  1
;  383     0764  2     BEGIN
;  384     0765  2
;  385     0766  2     EXTERNAL REGISTER
;  386     0767  2         COMMON_REG;
;  387     0768  2
;  388     0769  2     EXTERNAL
;  389     0770  2         USER_STATUS     : VECTOR;               ! status to return to user
;  390     0771  2
;  391     0772  2     MAP
;  392     0773  2         FIB     : REF BBLOCK,                   ! file information block
;  393     0774  2         PACKET  : REF BBLOCK;                   ! address of io request block
;  394     0775  2
;  395     0776  2     ! Write tape mark, create window, setup data base connections, complete io,
;  396     0777  2     ! and startup virtual io
;  397     0778  2     !
;  398     0779  2
;  399     0780  2     WRITE_TM();
;  400     0781  2
;  401     0782  2     ! If the USEREOT flag is set in the FIB block of the I/O then as long
;  402     0783  2     ! as this file is accessed allow user eot handling
;  403     0784  2
;  404     0785  2     KERNEL_CALL(SET_USER_EOT, .FIB);
;  405     0786  2
;  406     0787  2
```

```
;  407          0788  2     ! If EOT is sensed, request a next volume, and clear the partial file flag
;  408          0789  2     ! since the end of volume labels will be written. If user handling of EOT
;  409          0790  2     ! is enabled then inform user of the condition.
;  410          0791  2
;  411          0792  2     IF NOT .CURRENT_VCB[VCB$V_ENUSEREOT]
;  412          0793  2     THEN
;  413          0794  3       BEGIN
;  414          0795  3
;  415          0796  3       IF .IO_STATUS EQL SS$_ENDOFTAPE
;  416          0797  3       THEN
;  417          0798  4         BEGIN
;  418          0799  4         KERNEL_CALL (UPD_PARTFILE, 0);
;  419          0800  4         NEXT_VOL_WRITE();
;  420          0801  3         END;
;  421          0802  3       END
;  422          0803  2     ELSE
;  423          0804  2       USER_STATUS[0] = .IO_STATUS;                ! return status to user
;  424          0805  2                                                   ! I/O's will be blocked
;  425          0806  2                                                   ! and completed in error
;  426          0807  2                                                   ! by the driver
;  427          0808  2
;  428          0809  2     KERNEL_CALL(ACCESS_FILE, .FIB[FIB$L_ACCTL], .PACKET[IRP$L_PID], 0, 1, .ABD);
;  429          0810  2     KERNEL_CALL(START_VIO);
;  430          0811  1     END;                                          ! end of routine


                                07FC 00000          .ENTRY  ACCESS_NEW_FILE, Save R2,R3,R4,R5,R6,R7,R8,-; 0731
                                                             R9,R10                                       ;
                            0000G 30 00002          BSBW    WRITE_TM                                     ; 0780
                         04   AC  DD 00005          PUSHL   FIB                                          ; 0785
                              01  DD 00008          PUSHL   #1                                           ;
                              5E  DD 0000A          PUSHL   SP                                           ;
                        0000V CF  9F 0000C          PUSHAB  SET_USER_EOT                                 ;
              00000000G 9F        04 FB 00010       CALLS   #4, a#SYS$CMKRNL                             ;
           20    2D   AB          01 E0 00017       BBS     #1, 45(CURRENT_VCB), 1$                      ; 0792
              00000878  8F  0000G CF D1 0001C       CMPL    IO_STATUS, #2168                             ; 0796
                              1C  12 00025          BNEQ    2$                                           ;
                        7E        01 7D 00027       MOVQ    #1, -(SP)                                    ; 0799
                              5E  DD 0002A          PUSHL   SP                                           ;
                        0000V CF  9F 0002C          PUSHAB  UPD_PARTFILE                                 ;
              00000000G 9F        04 FB 00030       CALLS   #4, a#SYS$CMKRNL                             ;
                            0000G 30 00037          BSBW    NEXT_VOL_WRITE                               ; 0800
                              07  11 0003A          BRB     2$                                           ; 0792
                   0000G  CF  0000G CF D0 0003C 1$: MOVL    IO_STATUS, USER_STATUS                       ; 0804
                         0C   AC  DD 00043 2$:      PUSHL   ABD                                          ; 0809
                              01  DD 00046          PUSHL   #1                                           ;
                              7E  D4 00048          CLRL    -(SP)                                        ;
                        50        08 AC D0 0004A    MOVL    PACKET, R0                                   ;
                                  0C A0 DD 0004E    PUSHL   12(R0)                                       ;
                                  04 BC DD 00051    PUSHL   @FIB                                         ;
                                  05 DD 00054       PUSHL   #5                                           ;
                                  5E DD 00056       PUSHL   SP                                           ;
                                0000G CF 9F 00058   PUSHAB  ACCESS_FILE                                  ;
              00000000G 9F        08 FB 0005C       CALLS   #8, a#SYS$CMKRNL                             ;
```

```
                                7E  D4 00063          CLRL    -(SP)                                   ; 0810
                                5E  DD 00065          PUSHL   SP                                      ;
                          0000G CF  9F 00067          PUSHAB  START_VIO                               ;
                00000000G 9F      03  FB 0006B          CALLS   #3, @#SYS$CMKRNL                        ;
                                    04  00072          RET                                             ; 0811

; Routine Size:  115 bytes,    Routine Base:  $CODE$ + 0137

;  431          0812  1
```

```
;  433     0813  1 GLOBAL ROUTINE SET_USER_EOT(FIB) : COMMON_CALL NOVALUE =
;  434     0814  1 !++
;  435     0815  1 !
;  436     0816  1 ! FUNCTIONAL DESCRIPTION:
;  437     0817  1 !
;  438     0818  1 !       This routine sets the Mode of the to user enabled EOT processing
;  439     0819  1 !       when the FIB$C_USEREOT bit is set in the FIB$W_CNTRLFUNC
;  440     0820  1 !
;  441     0821  1 ! CALLING SEQUENCE:
;  442     0822  1 !       set_user_eot(arg1)
;  443     0823  1 !
;  444     0824  1 ! INPUT PARAMETERS:
;  445     0825  1 !       Arg1 - address of copy of user file identification block
;  446     0826  1 !
;  447     0827  1 ! IMPLICIT INPUTS:
;  448     0828  1 !       none
;  449     0829  1 !
;  450     0830  1 ! OUTPUT PARAMETERS:
;  451     0831  1 !
;  452     0832  1 !       none
;  453     0833  1 !
;  454     0834  1 ! IMPLICIT OUTPUTS:
;  455     0835  1 !       VCB$W_MODE may be modified
;  456     0836  1 !
;  457     0837  1 ! ROUTINE VALUE:
;  458     0838  1 !       none
;  459     0839  1 !
;  460     0840  1 ! SIDE EFFECTS:
;  461     0841  1 !       none
;  462     0842  1 !
;  463     0843  1 !--
;  464     0844  2     BEGIN
;  465     0845  2         EXTERNAL REGISTER
;  466     0846  2             COMMON_REG;
;  467     0847  2
;  468     0848  2         MAP
;  469     0849  2             FIB  : REF BBLOCK;                  ! File information block
;  470     0850  2
;  471     0851  2         IF .FIB[FIB$W_CNTRLFUNC] EQL FIB$C_USEREOT
;  472     0852  2         THEN
;  473     0853  2           CURRENT_VCB[VCB$V_ENUSEREOT] = 1
;  474     0854  1     END;


                                    0000 00000          .ENTRY  SET_USER_EOT, Save nothing                  ; 0813
                          50    04  AC  D0 00002          MOVL    FIB, R0                                     ; 0851
                          0F    16  A0  B1 00006          CMPW    22(R0), #15                                 ;
                                    04  12 0000A          BNEQ    1$                                          ;
                        2D    AB        02  88 0000C          BISB2   #2, 45(CURRENT_VCB)                     ; 0853
                                        04 00010 1$:        RET                                             ; 0854

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 01AA
```

```
;  475          0855  1
```

```
;  477     0856  1 GLOBAL ROUTINE WRITE_HEADERS : L$WRITE_HEADER NOVALUE =
;  478     0857  1
;  479     0858  1 !++
;  480     0859  1 !
;  481     0860  1 ! FUNCTIONAL DESCRIPTION:
;  482     0861  1 !       this routine inserts the file section and sequence number into hdr1,
;  483     0862  1 !       inserts the header label identifiers, zeroes the block count, and
;  484     0863  1 !       writes hdr1, hdr2, hdr3, and hdr4.
;  485     0864  1 !
;  486     0865  1 ! CALLING SEQUENCE:
;  487     0866  1 !       write_headers()
;  488     0867  1 !
;  489     0868  1 ! INPUT PARAMETERS:
;  490     0869  1 !       none
;  491     0870  1 !
;  492     0871  1 ! IMPLICIT INPUTS:
;  493     0872  1 !       hdr1, hdr2, hdr3, and hdr4 partially formatted
;  494     0873  1 !
;  495     0874  1 ! OUTPUT PARAMETERS:
;  496     0875  1 !
;  497     0876  1 !       none
;  498     0877  1 !
;  499     0878  1 ! IMPLICIT OUTPUTS:
;  500     0879  1 !       hdr1, hdr2, hdr3 and hdr4 written
;  501     0880  1 !
;  502     0881  1 ! ROUTINE VALUE:
;  503     0882  1 !       none
;  504     0883  1 !
;  505     0884  1 ! SIDE EFFECTS:
;  506     0885  1 !       none
;  507     0886  1 !
;  508     0887  1 !--
;  509     0888  1
;  510     0889  2     BEGIN
;  511     0890  2
;  512     0891  2     EXTERNAL REGISTER
;  513     0892  2         COMMON_REG;
;  514     0893  2
;  515     0894  2     LOCAL
;  516     0895  2
;  517     0896  2         ! descriptor for file section and sequence number
;  518     0897  2         !
;  519     0898  2         DESCR   : VECTOR [2];
;  520     0899  2
;  521     0900  2     ! store file section and sequence number
;  522     0901  2     !
;  523     0902  2     DESCR[0] = HD1$S_FILESECNO + HD1$S_FILESEQNO;
;  524     0903  2     DESCR[1] = HDR1[HD1$T_FILESECNO];
;  525     0904  2     SYS$FAO(DESCRIPTOR('!4ZW!4ZW'),
;  526     0905  2                 0,
;  527     0906  2                 DESCR,
;  528     0907  2                 .CURRENT_VCB[VCB$W_CUR_SEQ],
;  529     0908  2                 .CURRENT_VCB[VCB$W_CUR_NUM]);
;  530     0909  2
;  531     0910  2     ! zero block count
;  532     0911  2     !
;  533     0912  2     CH$FILL('0', HD1$S_BLOCKCNT, HDR1[HD1$T_BLOCKCNT]);
```

```
;  534       0913  2
;  535       0914  2         ! inserts header label identifiers
;  536       0915  2         !
;  537       0916  2         HDR1[HD1$L_HD1LID] = 'HDR1';
;  538       0917  2         HDR2[HD2$L_HD2LID] = 'HDR2';
;  539       0918  2         HDR3[HD3$L_HD3LID] = 'HDR3';
;  540       0919  2         HDR4[HD4$L_HD4LID] = 'HDR4';
;  541       0920  2
;  542       0921  2         ! now write headers
;  543       0922  2         !
;  544       0923  2         WRITE_BLOCK(.HDR1, ANSI_LBLSZ);
;  545       0924  2         WRITE_BLOCK(.HDR2, ANSI_LBLSZ);
;  546       0925  2
;  547       0926  2         ! write hdr3 and hdr4 only if mount switch allows it
;  548       0927  2         !
;  549       0928  2
;  550       0929  2         IF NOT .CURRENT_VCB[VCB$V_NOHDR3] AND .CURRENT_VCB[VCB$B_LBLCNT] GTR 2
;  551       0930  2         THEN
;  552       0931  3             BEGIN
;  553       0932  3                 WRITE_BLOCK(.HDR3, ANSI_LBLSZ);
;  554       0933  3                 IF .CURRENT_VCB[VCB$B_LBLCNT] GTR 3
;  555       0934  3                     THEN WRITE_BLOCK(.HDR4, ANSI_LBLSZ);
;  556       0935  2             END;
;  557       0936  2
;  558       0937  1         END;                                        ! end of routine


                             57 5A 34 21 57 5A 34 21  001BB P.AAB:    .ASCII   \!4ZW!4ZW\                        ;
                                                      001C3           .BLKB    1
                                         00000008     001C4 P.AAA:    .LONG    8                                 ;
                                         00000000'    001C8           .ADDRESS P.AAB


                                007C        8F  BB 00000 WRITE_HEADERS::
                                                                      PUSHR    #^M<R2,R3,R4,R5,R6>               ; 0856
                            5E              04  C2 00004              SUBL2    #4, SP
                                            08  DD 00007              PUSHL    #8                                ; 0902
         04   AE     0000G  CF              1B  C1 00009              ADDL3    #27, HDR1, DESCR+4                ; 0903
                            7E        24    AB  3C 00010              MOVZWL   36(CURRENT_VCB), -(SP)            ; 0908
                            7E        26    AB  3C 00014              MOVZWL   38(CURRENT_VCB), -(SP)            ; 0907
                                      08    AE  9F 00018              PUSHAB   DESCR                             ; 0904
                                            7E  D4 0001B              CLRL     -(SP)                             ;
                                      D8    AF  9F 0001D              PUSHAB   P.AAA                             ;
                 00000000G  9F              05  FB 00020              CALLS    #5, @#SYS$FAO                     ;
                            56       0000G  CF  D0 00027              MOVL     HDR1, R6                          ; 0912
    06           30         6E              00  2C 0002C              MOVC5    #0, (SP), #48, #6, 54(R6)         ;
                                      36    A6     00031
                            66 31524448     8F  D0 00033              MOVL     #827475016, (R6)                  ; 0916
                 0000G      DF 32524448     8F  D0 0003A              MOVL     #844252232, @HDR2                 ; 0917
                 0000G      DF 33524448     8F  D0 00043              MOVL     #861029448, @HDR3                 ; 0918
                 0000G      DF 34524448     8F  D0 0004C              MOVL     #877806664, @HDR4                 ; 0919
                            7E        50    8F  9A 00055              MOVZBL   #80, -(SP)                        ; 0923
                                            56  DD 00059              PUSHL    R6                                ;
                 0000G      CF              02  FB 0005B              CALLS    #2, WRITE_BLOCK                   ;
                            7E        50    8F  9A 00060              MOVZBL   #80, -(SP)                        ; 0924
```

```
                          0000G  CF   DD 00064          PUSHL   HDR2                                                    ;
                 0000G    CF        02 FB 00068          CALLS   #2, WRITE_BLOCK                                        ;
                            2C     AB 95 0006D          TSTB    44(CURRENT_VCB)                                        ; 0929
                                   26 19 00070          BLSS    1$                                                     ;
                            02     48 AB 91 00072       CMPB    72(CURRENT_VCB), #2                                    ;
                                   20 1B 00076          BLEQU   1$                                                     ;
                            7E     50 8F 9A 00078       MOVZBL  #80, -(SP)                                             ; 0932
                          0000G    CF DD 0007C          PUSHL   HDR3                                                   ;
                 0000G    CF        02 FB 00080          CALLS   #2, WRITE_BLOCK                                        ;
                            03     48 AB 91 00085       CMPB    72(CURRENT_VCB), #3                                    ; 0933
                                   0D 1B 00089          BLEQU   1$                                                     ;
                            7E     50 8F 9A 0008B       MOVZBL  #80, -(SP)                                             ; 0934
                          0000G    CF DD 0008F          PUSHL   HDR4                                                   ;
                 0000G    CF        02 FB 00093          CALLS   #2, WRITE_BLOCK                                        ;
                            5E        08 C0 00098 1$:    ADDL2   #8, SP                                                 ; 0937
                          007C     8F BA 0009B          POPR    #^M<R2,R3,R4,R5,R6>                                    ;
                                      05 0009F          RSB                                                            ;

; Routine Size:  160 bytes,    Routine Base:  $CODE$ + 01CC

;  559          0938  1
```

```
;   561     0939  1 ROUTINE INIT_NEW_FILE (INC_SEQ_NUM) : COMMON_CALL NOVALUE =
;   562     0940  1
;   563     0941  1 !++
;   564     0942  1 !
;   565     0943  1 ! FUNCTIONAL DESCRIPTION:
;   566     0944  1 !       this routine initializes the file identifier and sets status
;   567     0945  1 !       indicators.
;   568     0946  1 !
;   569     0947  1 ! CALLING SEQUENCE:
;   570     0948  1 !       INIT_NEW_FILE(ARG1), called in kernel mode
;   571     0949  1 !
;   572     0950  1 ! INPUT PARAMETERS:
;   573     0951  1 !       INC_SEQ_NUM - iNDICATES IF THE CURRENT FILE NUMBER IN THE vcb
;   574     0952  1 !                     should be incremented or not.
;   575     0953  1 !
;   576     0954  1 ! IMPLICIT INPUTS:
;   577     0955  1 !       CURRENT_VCB - address of current volume control block
;   578     0956  1 !
;   579     0957  1 ! OUTPUT PARAMETERS:
;   580     0958  1 !       none
;   581     0959  1 !
;   582     0960  1 ! IMPLICIT OUTPUTS:
;   583     0961  1 !       current file id, starlet file indicator updated
;   584     0962  1 !
;   585     0963  1 ! ROUTINE VALUE:
;   586     0964  1 !       none
;   587     0965  1 !
;   588     0966  1 ! SIDE EFFECTS:
;   589     0967  1 !       none
;   590     0968  1 !
;   591     0969  1 !--
;   592     0970  1
;   593     0971  2     BEGIN
;   594     0972  2
;   595     0973  2     EXTERNAL REGISTER
;   596     0974  2         COMMON_REG;
;   597     0975  2
;   598     0976  2     ! if end of volume set or dummy file, inc file number
;   599     0977  2     !
;   600     0978  2
;   601     0979  2     IF .INC_SEQ_NUM
;   602     0980  2         OR
;   603     0981  3         (.CURRENT_VCB[VCB$B_CUR_RVN] EQL 1 AND .CURRENT_VCB[VCB$W_CUR_NUM] EQL 0)
;   604     0982  2     THEN
;   605     0983  2         CURRENT_VCB[VCB$W_CUR_NUM] = .CURRENT_VCB[VCB$W_CUR_NUM] + 1;
;   606     0984  2
;   607     0985  2     CURRENT_VCB[VCB$W_CUR_SEQ] = 1;
;   608     0986  2     CURRENT_VCB[VCB$V_STARFILE] = 1;
;   609     0987  2     CURRENT_VCB[VCB$B_LBLCNT] = NO_OF_SUPPORT_ANSI_LABELS;
;   610     0988  2
;   611     0989  2     ! set status indicators to show partial file
;   612     0990  2     !
;   613     0991  2     CURRENT_VCB[VCB$V_PARTFILE] = 1;
;   614     0992  1     END;                                        ! end of routine
```

```
                                0000 00000 INIT_NEW_FILE:
                                           .WORD   Save nothing                                    ; 0939
                      0B    04  AC  E8 00002          BLBS    INC_SEQ_NUM, 1$                     ; 0979
                      01    2F  AB  91 00006          CMPB    47(CURRENT_VCB), #1                 ; 0981
                                08  12 0000A          BNEQ    2$                                  ;
                            24  AB  B5 0000C          TSTW    36(CURRENT_VCB)                     ;
                                03  12 0000F          BNEQ    2$                                  ;
                            24  AB  B6 00011 1$:      INCW    36(CURRENT_VCB)                     ; 0983
                  26    AB          01  B0 00014 2$:  MOVW    #1, 38(CURRENT_VCB)                 ; 0985
                  2D    AB          01  88 00018      BISB2   #1, 45(CURRENT_VCB)                 ; 0986
                  48    AB          04  90 0001C      MOVB    #4, 72(CURRENT_VCB)                 ; 0987
                  0B    AB          01  88 00020      BISB2   #1, 11(CURRENT_VCB)                 ; 0991
                                    04  00024         RET                                         ; 0992

; Routine Size:  37 bytes,    Routine Base:  $CODE$ + 026C

;  615          0993  1
```

```
1 ROUTINE UPD_PARTFILE (BIT_VALUE) : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine clears or sets the partial file flag in the VCB.
1 !
1 ! CALLING SEQUENCE:
1 !       UPD_PARTFILE(ARG1), called in kernel mode
1 !
1 ! INPUT PARAMETERS:
1 !       Value to which flag should be set to:
1 !               0 - clear flag
1 !               1 - set flag
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       PARTFILE flag in VCB is updated
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     ! Update Partial file flag in the VCB
2     !
2     CURRENT_VCB[VCB$V_PARTFILE] = .BIT_VALUE;
2
1     END;                                              ! end of routine


                                 0000 00000 UPD_PARTFILE:
                                                 .WORD   Save nothing                              ; 0994
  0B   AB        01         00       04   AC  F0 00002   INSV    BIT_VALUE, #0, #1, 11(CURRENT_VCB)        ; 1033
                                              04 00009   RET                                               ; 1035

; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 0291


1
1 END
```

```
;  661          1038  1
;  662          1039  0 ELUDOM



;                                PSECT SUMMARY
;
;      Name                    Bytes                          Attributes
;
;  $CODE$                        667  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         50         0      1000          00:01.8



;                                       COMMAND QUALIFIERS

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

; Size:         650 code + 17 data bytes
; Run Time:         00:16.3
; Elapsed Time:     00:54.4
; Lines/CPU Min:     3822
; Lexemes/CPU-Min: 19074
; Memory Used:  153 pages
; Compilation Complete
```

FREEPG
LIS

DEACCS
LIS

FIND
LIS

DATECN
LIS

FRMOD1
LIS

CREATE
LIS

GETREQ
LIS

EXPIRE
LIS

CNTRL
LIS

FRMHDR
LIS

HEADER
LIS

COMLABPRC
LIS

ENDVOL
LIS

GETFIB
LIS