
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

```
  AAAAAA      CCCCCCCC  PPPPPPPP     CCCCCCCC  TTTTTTTTTT  RRRRRRRR
  AAAAAA      CCCCCCCC  PPPPPPPP     CCCCCCCC  TTTTTTTTTT  RRRRRRRR
AA      AA  CC          PP      PP  CC            TT      RR      RR
AA      AA  CC          PP      PP  CC            TT      RR      RR
AA      AA  CC          PP      PP  CC            TT      RR      RR
AA      AA  CC          PP      PP  CC            TT      RR      RR
AA      AA  CC          PPPPPPPP    CC            TT      RRRRRRRR
AA      AA  CC          PPPPPPPP    CC            TT      RRRRRRRR
AAAAAAAAAA  CC          PP          CC            TT      RR  RR
AAAAAAAAAA  CC          PP          CC            TT      RR  RR
AA      AA  CC          PP          CC            TT      RR    RR      ....
AA      AA  CC          PP          CC            TT      RR    RR      ....
AA      AA    CCCCCCCC  PP            CCCCCCCC    TT      RR      RR    ....
AA      AA    CCCCCCCC  PP            CCCCCCCC    TT      RR      RR    ....

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 ACPCTR (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 handles acp control functions.
;    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:  09-JUL-1977
;    47          0047  1 !
;    48          0048  1 ! MODIFIED BY:
;    49          0049  1 !
;    50          0050  1 !       V03-005 MMD0236         Meg Dumont,     4-Feb-1984  15:13
;    51          0051  1 !               Add support for FIB$C_CLSEREXCP when set with IO$_ACPCONTROL.
;    52          0052  1 !
;    53          0053  1 !       V03-004 MMD0171         Meg Dumont,     9-May-1983  15:12
;    54          0054  1 !               Fix to make USER_STATUS defined consistently within module
;    55          0055  1 !
;    56          0056  1 !       V03-003 MMD0149         Meg Dumont,     26-Apr-1983  8:51
;    57          0057  1 !               Change references to 80 to the symbol ANSI_LBLSZ
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-002 MMD0001         Meg Dumont,     3-Jan-1983  15:22
;    60          0060  1 !               Added a call to stop access to a file if the trailer labels
;    61          0061  1 !               have been read.
;    62          0062  1 !
;    63          0063  1 !       V03-001 STJ0309         Steven T. Jeffreys      1-Jun-1982
;    64          0064  1 !               Added handler for REMOUNT control function.  It's a NOP.
;    65          0065  1 !
;    66          0066  1 !       V02-011 DMW00077        David Michael Walp      8-Feb-1982
;    67          0067  1 !               Stored account and user name during mount time
;    68          0068  1 !
;    69          0069  1 !       V02-010 DMW00034        David Michael Walp      15-Sep-1981
;    70          0070  1 !               Fixed Cancel I/O vs Dismount race condition
;    71          0071  1 !
;    72          0072  1 !       V02-009 DMW00024        David Michael Walp      20-Jul-1981
;    73          0073  1 !               Change to RET_FREE_PAGE to not contract region P0 every time
;    74          0074  1 !               space is returned.
;    75          0075  1 !
;    76          0076  1 !       V02-008 DMW00009        David Michael Walp      14-Mar-1981
;    77          0077  1 !               Changed calculation of CCB address
;    78          0078  1 !
;    79          0079  1 !       V02-007 DMW00001        David Michael Walp      11-Nov-1980
;    80          0080  1 !               New BLISS compiler, FUNCTION declaration changed from
;    81          0081  1 !               BBLOCK to BLOCK. Old compiler used to give a longword
;    82          0082  1 !               with a declaration of "BBLOCK [1]".
;    83          0083  1 !
;    84          0084  1 !       V02-006 REFORMAT        Maria del C. Nasr       30-Jun-1980
;    85          0085  1 !
;    86          0086  1 !       V02-005 MCN0017         Maria del C. Nasr       18-Jun-1980
;    87          0087  1 !               Add a call to START_VIO after completing the next volume
;    88          0088  1 !               write.  This is part of the fix for multivolume processing,
;    89          0089  1 !               in which a new volume should be requested when the EOT is
;    90          0090  1 !               sensed in writing the header labels, and not wait for the
;    91          0091  1 !               data to be written.
;    92          0092  1 !
;    93          0093  1 !       V02-004 SPR27361        Maria del C. Nasr       10-Jun-1980
;    94          0094  1 !               Add a call to START_VIO after completing the next volume
;    95          0095  1 !               read.  This is part of the general fix which delays IO
;    96          0096  1 !               posting until all IO is successfully completed.
;    97          0097  1 !
;    98          0098  1 !       A0103   MCN0007         Maria del C. Nasr       13-Nov-1979   19:35
;    99          0099  1 !               Set single directory structured device bit (DEV$M_SDI)
;   100          0100  1 !
;   101          0101  1 !
;   102          0102  1 !**
;   103          0103  1
;   104          0104  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   105          0105  1
;   106          0106  1 REQUIRE 'SRC$:MTADEF.B32';
;   107          0490  1
;   108          0491  1 FORWARD ROUTINE
;   109          0492  1     MTA_ACPCNTRL        : NOPRES NOVALUE, ! control function dispatch
;   110          0493  1     MTA_MOUNT   : NOPRES NOVALUE,         ! mount function
;   111          0494  1     CANCEL_IO   : COMMON_CALL NOVALUE,    ! cancel i/o control
;   112          0495  1     DO_CANCEL   : COMMON_CALL NOVALUE,    ! do actual cancelation of io
;   113          0496  1     MOUNT       : COMMON_CALL NOVALUE,    ! mount control function, kernel mode
;   114          0497  1     STALL       : COMMON_CALL NOVALUE,    ! stall cancel
```

```
;  115     0498  1     TERMINATE_VOL: COMMON_CALL NOVALUE;   ! terminate mount volume request
;  116     0499  1
;  117     0500  1 EXTERNAL ROUTINE
;  118     0501  1     CANCEL_OP_REPLY     : COMMON_CALL,                ! cancel reply from operator
;  119     0502  1     IO_DONE,                                          ! complete io
;  120     0503  1     NEXT_VOL_READ       : L$NEXT_VOL_READ NOVALUE,    ! get next vol for read
;  121     0504  1     NEXT_VOL_WRITE      : L$NEXT_VOL_WRIT NOVALUE,    ! get next vol for write
;  122     0505  1     READ_BLOCK  : COMMON_CALL,                        ! read a tape block
;  123     0506  1     RET_FREE_PAGE       : COMMON_CALL,     ! return virtual page to free list
;  124     0507  1     RETURN_ALL_ERR      : COMMON_CALL,     ! return blocked virtual io in error
;  125     0508  1     SEND_ERRLOG,
;  126     0509  1     SPACE_TM    : COMMON_CALL,                        ! space tape mark
;  127     0510  1     START_VIO   : COMMON_CALL,                        ! start up virtual io
;  128     0511  1     STOP_VIO    : COMMON_CALL,                        ! Disallow VIO's
;  129     0512  1     SYS$QIOW    : ADDRESSING_MODE (ABSOLUTE),
;  130     0513  1     ZERO_CHANNEL        : COMMON_CALL;                ! zero channel
;  131     0514  1
;  132     0515  1 EXTERNAL
;  133     0516  1     SCH$GL_PCBVEC       : REF VECTOR ADDRESSING_MODE (ABSOLUTE),
;  134     0517  1     CURRENT_UCB : REF BBLOCK,
;  135     0518  1     CURRENT_WCB : REF BBLOCK,   ! address of current window control block
;  136     0519  1     HDR1        : REF BBLOCK,   ! hdr1(eof1) label
;  137     0520  1     IO_CHANNEL,
;  138     0521  1     IO_PACKET   : REF BBLOCK,   ! address of current io packet
;  139     0522  1     USER_STATUS : VECTOR [2];   ! address of user status
;  140     0523  1
```

```
;  142     0524  1 GLOBAL ROUTINE MTA_ACPCNTRL : NOPRES NOVALUE =
;  143     0525  1
;  144     0526  1 !++
;  145     0527  1 !
;  146     0528  1 ! FUNCTIONAL DESCRIPTION:
;  147     0529  1 !       This routine handles the acp control function.
;  148     0530  1 !
;  149     0531  1 ! CALLING SEQUENCE:
;  150     0532  1 !       MTA_ACPCNTRL()
;  151     0533  1 !
;  152     0534  1 ! INPUT PARAMETERS:
;  153     0535  1 !       None
;  154     0536  1 !
;  155     0537  1 ! IMPLICIT INPUTS:
;  156     0538  1 !       CURRENT_UCB - address of current unit control block
;  157     0539  1 !       CURRENT_VCB - address of current volume control block
;  158     0540  1 !       IO_PACKET - address of current io request packet
;  159     0541  1 !       QUEUE_HEAD  - address of acp queue
;  160     0542  1 !
;  161     0543  1 ! OUTPUT PARAMETERS:
;  162     0544  1 !       None
;  163     0545  1 !
;  164     0546  1 ! IMPLICIT OUTPUTS:
;  165     0547  1 !       LOCAL_FIB - copy of user's fib
;  166     0548  1 !
;  167     0549  1 ! ROUTINE VALUE:
;  168     0550  1 !       None
;  169     0551  1 !
;  170     0552  1 ! SIDE EFFECTS:
;  171     0553  1 !       None
;  172     0554  1 !
;  173     0555  1 !--
;  174     0556  1
;  175     0557  2     BEGIN
;  176     0558  2
;  177     0559  2     EXTERNAL REGISTER
;  178     0560  2         COMMON_REG;
;  179     0561  2
;  180     0562  2     EXTERNAL ROUTINE
;  181     0563  2         ISSUE_IO : L$ISSUE_IO,                  ! Send an io to the tape drive
;  182     0564  2         GET_FIB : COMMON_CALL,          ! get user's file information block
;  183     0565  2         POSITION_TO_END : COMMON_CALL,          ! position volume set to end
;  184     0566  2         SPACE_IN_FILE   : COMMON_CALL,          ! space within file
;  185     0567  2         REWIND_FILE     : COMMON_CALL,          ! rewind file
;  186     0568  2         REWIND_VOL_SET  : COMMON_CALL;          ! rewind volume set
;  187     0569  2
;  188     0570  2     EXTERNAL
;  189     0571  2
;  190     0572  2         ! address of current unit control block
;  191     0573  2         !
;  192     0574  2         CURRENT_UCB     : REF BBLOCK,
;  193     0575  2         IO_PACKET       : REF BBLOCK,           ! address of current io request
;  194     0576  2                                                 !   packet
;  195     0577  2         QUEUE_HEAD      : REF BBLOCK;           ! address of acp queue head
;  196     0578  2
;  197     0579  2     LOCAL
;  198     0580  2         FIB             : REF BBLOCK,           ! address of copy of user's
```

```
;  199     0581  2          FUNCTION        : BLOCK [1],                ! file info block
;  200     0582  2                                                      ! io function code and
;  201     0583  2                                                      !  modifiers
;  202     0584  2          PACKET  : REF BBLOCK;                       ! address of io request packet
;  203     0585  2
;  204     0586  2      PACKET = .IO_PACKET;                            ! get address  of io packet
;  205     0587  2      FUNCTION = .PACKET[IRP$W_FUNC];     ! get function code and modifiers
;  206     0588  2
;  207     0589  2      IF .FUNCTION[IO$V_DMOUNT]
;  208     0590  2          OR
;  209     0591  2          .FUNCTION[IO$V_MOUNT]
;  210     0592  2          OR
;  211     0593  2          .FUNCTION[IO$V_REMOUNT]
;  212     0594  2      THEN
;  213     0595  2          RETURN;
;  214     0596  2
;  215     0597  2      IF NOT .PACKET[IRP$V_VIRTUAL]
;  216     0598  2      THEN
;  217     0599  3          BEGIN
;  218     0600  3          KERNEL_CALL(CANCEL_IO);
;  219     0601  3
;  220     0602  4          IF (.CURRENT_VCB[VCB$V_WAIMOUVOL]
;  221     0603  4              AND
;  222     0604  4              NOT CANCEL_OP_REPLY())
;  223     0605  4              OR
;  224     0606  4              .CURRENT_VCB[VCB$V_WAIUSRLBL]
;  225     0607  4          THEN
;  226     0608  5              BEGIN
;  227     0609  5              ERROR(SS$_CANCEL);
;  228     0610  5              KERNEL_CALL(DO_CANCEL);
;  229     0611  5              END;
;  230     0612  4
;  231     0613  4          ! Stall cancel until rewind or mount vol complete so cancels are not
;  232     0614  4          ! continuously issued.
;  233     0615  4          !
;  234     0616  4
;  235     0617  4          IF .CURRENT_VCB[VCB$V_WAIREWIND]
;  236     0618  4              OR
;  237     0619  4              .CURRENT_VCB[VCB$V_WAIMOUVOL]
;  238     0620  4          THEN
;  239     0621  5              KERNEL_CALL(STALL);
;  240     0622  4
;  241     0623  4          RETURN;
;  242     0624  4
;  243     0625  3          END;
;  244     0626  2
;  245     0627  2      FIB = GET_FIB(.BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT]);
;  246     0628  2
;  247     0629  2      IF .CURRENT_VCB[VCB$V_WAIUSRLBL]
;  248     0630  2      THEN
;  249     0631  3          ERR_EXIT(SS$_WAITUSRLBL);
;  250     0632  2
;  251     0633  2      IF .CURRENT_VCB[VCB$V_MUSTCLOSE]
;  252     0634  2      THEN
;  253     0635  3          ERR_EXIT(SS$_MUSTCLOSEFL);
;  254     0636  2
;  255     0637  2      ! Allow the user to clear the serious exception from the tape drive
```

```
;  256     0638  2     ! by issuing a sensemode, which is effectively a NOP. This gives the
;  257     0639  2     ! user the capibility to write blocks beyond EOT and before the
;  258     0640  2     ! EOV labels. It also allows the user to read blocks beyond the
;  259     0641  2     ! EOT and before the EOV labels. The user is never allowed to
;  260     0642  2     ! read the EOV labels.
;  261     0643  2     !
;  262     0644  2     ! Please note that the case statement works on the assumption that
;  263     0645  2     ! the variables are within a certain range. FIB$C_CLSEREXCP does
;  264     0646  2     ! not fall in that range.
;  265     0647  2
;  266     0648  2     IF .FIB[FIB$W_CNTRLFUNC] EQL FIB$C_CLSEREXCP
;  267     0649  2         THEN
;  268     0650  3             BEGIN
;  269     0651  3             ISSUE_IO(IO$_SENSEMODE, 0, 0);
;  270     0652  3             RETURN;
;  271     0653  3             END;
;  272     0654  2
;  273     0655  2     CASE .FIB[FIB$W_CNTRLFUNC] FROM FIB$C_REWINDVOL TO FIB$C_REWINDFIL OF
;  274     0656  2         SET
;  275     0657  2
;  276     0658  2         [FIB$C_REWINDFIL] :
;  277     0659  2             REWIND_FILE();
;  278     0660  2
;  279     0661  2         [FIB$C_POSEND] :
;  280     0662  2             POSITION_TO_END();
;  281     0663  2
;  282     0664  2         [FIB$C_NEXTVOL] :
;  283     0665  3             BEGIN
;  284     0666  3
;  285     0667  3             ! file must be accessed
;  286     0668  3             !
;  287     0669  3
;  288     0670  3             IF .CURRENT_WCB EQL 0
;  289     0671  3             THEN
;  290     0672  3                 ERR_EXIT(SS$_FILNOTACC);
;  291     0673  3
;  292     0674  3             ! if not in data area, not appropriate time to be doing a next
;  293     0675  3             ! volume
;  294     0676  3             !
;  295     0677  3
;  296     0678  3             IF .CURRENT_VCB[VCB$B_TM] NEQ 1
;  297     0679  3             THEN
;  298     0680  3                 ERR_EXIT(SS$_ILLSEQOP);
;  299     0681  3
;  300     0682  3             KERNEL_CALL(STOP_VIO);
;  301     0683  3
;  302     0684  3             IF .CURRENT_WCB[WCB$V_READ]
;  303     0685  3             THEN
;  304     0686  4                 BEGIN                               ! read case
;  305     0687  4                 SPACE_TM(1);                        ! space to trailer record
;  306     0688  4
;  307     0689  4                 IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
;  308     0690  4                 THEN
;  309     0691  4                     ERR_EXIT(SS$_TAPEPOSLOST);
;  310     0692  4
;  311     0693  4                 IF .HDR1[EO1$L_EO1LID] EQL 'EOF1'
;  312     0694  4                 THEN
```

```
;  313     0695  4                    ERR_EXIT(SS$_ENDOFFILE);
;  314     0696  4
;  315     0697  4                NEXT_VOL_READ();
;  316     0698  4                END
;  317     0699  3            ELSE
;  318     0700  3                NEXT_VOL_WRITE();               ! write case
;  319     0701  3            KERNEL_CALL(START_VIO);             ! requeue blocked io
;  320     0702  2            END;
;  321     0703  2
;  322     0704  2        [FIB$C_SPACE] :
;  323     0705  2            SPACE_IN_FILE();
;  324     0706  2
;  325     0707  2        [FIB$C_REWINDVOL] :
;  326     0708  2            REWIND_VOL_SET();
;  327     0709  2
;  328     0710  2        [OUTRANGE] :
;  329     0711  2            ERR_EXIT(SS$_ILLCNTRFUNC);
;  330     0712  2
;  331     0713  2        [INRANGE] :
;  332     0714  2            ERR_EXIT(SS$_ILLCNTRFUNC);
;  333     0715  2        TES;
;  334     0716  2
;  335     0717  1    END;


                                .TITLE  ACPCTR
                                .IDENT  \V04-000\

                                .EXTRN  CANCEL_OP_REPLY
                                .EXTRN  IO_DONE, NEXT_VOL_READ
                                .EXTRN  NEXT_VOL_WRITE, READ_BLOCK
                                .EXTRN  RET_FREE_PAGE, RETURN_ALL_ERR
                                .EXTRN  SEND_ERRLOG, SPACE_TM
                                .EXTRN  START_VIO, STOP_VIO
                                .EXTRN  SYS$QIOW, ZERO_CHANNEL
                                .EXTRN  SCH$GL_PCBVEC, CURRENT_UCB
                                .EXTRN  CURRENT_WCB, HDR1
                                .EXTRN  IO_CHANNEL, IO_PACKET
                                .EXTRN  USER_STATUS, ISSUE_IO
                                .EXTRN  GET_FIB, POSITION_TO_END
                                .EXTRN  SPACE_IN_FILE, REWIND_FILE
                                .EXTRN  REWIND_VOL_SET, QUEUE_HEAD
                                .EXTRN  SYS$CMKRNL

                                .PSECT  $CODE$,NOWRT,2

                          0000 00000            .ENTRY  MTA_ACPCNTRL, Save nothing                      ; 0524
                    52  0000G CF   D0 00002    MOVL    IO_PACKET, PACKET                               ; 0586
                    50    20  A2   3C 00007    MOVZWL  32(PACKET), FUNCTION                            ; 0587
01                  50        0A   E1 0000B    BBC     #10, FUNCTION, 1$                               ; 0589
                                   04 0000F    RET                                                     ;
01                  50        09   E1 00010 1$: BBC     #9, FUNCTION, 2$                                ; 0591
                                   04 00014    RET                                                     ;
01                  50        0B   E1 00015 2$: BBC     #11, FUNCTION, 3$                               ; 0593
                                   04 00019    RET                                                     ;
4D        2A        A2        04   E0 0001A 3$: BBS     #4, 42(PACKET), 8$                              ; 0597
                              7E   D4 0001F    CLRL    -(SP)                                           ; 0600
```

```
                                 5E  DD 00021          PUSHL   SP                                                      ;
                           0000V CF  9F 00023          PUSHAB  CANCEL_IO                                               ;
                00000000G  9F        03 FB 00027       CALLS   #3, @#SYS$CMKRNL                                        ;
          08       0B  AB            02 E1 0002E       BBC     #2, 11(CURRENT_VCB), 4$                                 ; 0602
             0000G  CF                00 FB 00033       CALLS   #0, CANCEL_OP_REPLY                                     ; 0604
                    05                50 E9 00038       BLBC    R0, 5$                                                  ;
          16       0B  AB            04 E1 0003B 4$:   BBC     #4, 11(CURRENT_VCB), 6$                                 ; 0606
             0000G  CF        0830    8F B0 00040 5$:   MOVW    #2096, USER_STATUS                                      ; 0609
                                      7E D4 00047       CLRL    -(SP)                                                   ; 0610
                                      5E DD 00049       PUSHL   SP                                                      ;
                                0000V CF  9F 0004B      PUSHAB  DO_CANCEL                                               ;
                00000000G  9F        03 FB 0004F       CALLS   #3, @#SYS$CMKRNL                                        ;
          06       0B  AB            03 E0 00056 6$:   BBS     #3, 11(CURRENT_VCB), 7$                                 ; 0617
          01       0B  AB            02 E0 0005B       BBS     #2, 11(CURRENT_VCB), 7$                                 ; 0619
                                      04 00060          RET                                                             ;
                                      7E D4 00061 7$:   CLRL    -(SP)                                                   ; 0621
                                      5E DD 00063       PUSHL   SP                                                      ;
                                0000V CF  9F 00065      PUSHAB  STALL                                                   ;
                                  00B4    31 00069      BRW     22$                                                     ;
                                    2C    B2 DD 0006C 8$:   PUSHL   @44(PACKET)                                       ; 0627
                             0000G  CF        01 FB 0006F       CALLS   #1, GET_FIB                                     ;
                                    52        50 D0 00074       MOVL    R0, FIB                                         ;
          04       0B  AB                    04 E1 00077       BBC     #4, 11(CURRENT_VCB), 9$                         ; 0629
                                0950    8F  BF 0007C          CHMU    #2384                                           ; 0631
          04       0B  AB                    06 E1 00080 9$:   BBC     #6, 11(CURRENT_VCB), 10$                        ; 0633
                                0948    8F  BF 00085          CHMU    #2376                                           ; 0635
                                    11    16  A2 B1 00089 10$:   CMPW    22(FIB), #17                                    ; 0648
                                          0B  12 0008D        BNEQ    11$                                             ;
                                          7E  7C 0008F        CLRQ    -(SP)                                           ; 0651
                                          27  DD 00091        PUSHL   #39                                             ;
                                        0000G 30 00093        BSBW    ISSUE_IO                                        ;
                                       5E       0C  C0 00096       ADDL2   #12, SP                                     ;
                                                04 00099          RET                                                 ; 0650
                                 05             01  16  A2 AF 0009A 11$:   CASEW   22(FIB), #1, #5                    ; 0655
                0089          001B          0015      008F     0009F 12$:   .WORD   24$-12$,-                          ;
                                            000F      0095     000A7                14$-12$,-                          ;
                                                                                    15$-12$,-                          ;
                                                                                    23$-12$,-                          ;
                                                                                    25$-12$,-                          ;
                                                                                    13$-12$                            ;
                                          0086 31 000AB          BRW     25$                                         ; 0711
                                 0000G  CF    00 FB 000AE 13$:   CALLS   #0, REWIND_FILE                             ; 0659
                                              04 000B3          RET                                                 ;
                                 0000G  CF    00 FB 000B4 14$:   CALLS   #0, POSITION_TO_END                         ; 0662
                                              04 000B9          RET                                                 ;
                                       0000G  CF D5 000BA 15$:   TSTL    CURRENT_WCB                                 ; 0670
                                              04 12 000BE        BNEQ    16$                                         ;
                                       00AC   8F BF 000C0        CHMU    #172                                        ; 0672
                                    01    2E   AB 91 000C4 16$:   CMPB    46(CURRENT_VCB), #1                         ; 0678
                                              04 13 000C8        BEQL    17$                                         ;
                                       02DC   8F BF 000CA        CHMU    #732                                        ; 0680
                                              7E D4 000CE 17$:   CLRL    -(SP)                                       ; 0682
                                              5E DD 000D0        PUSHL   SP                                          ;
                                       0000G  CF 9F 000D2        PUSHAB  STOP_VIO                                    ;
                          00000000G  9F        03 FB 000D6       CALLS   #3, @#SYS$CMKRNL                            ;
                                    50  0000G  CF D0 000DD       MOVL    CURRENT_WCB, R0                             ; 0684
                                    2F    0B  A0 E9 000E2       BLBC    11(R0), 20$                                 ;
```

```
                                    01  DD 000E6          PUSHL   #1                                          ; 0687
                          0000G     CF  01  FB 000E8          CALLS   #1, SPACE_TM
                                7E      50  8F  9A 000ED      MOVZBL  #80, -(SP)                              ; 0689
                              0000G     CF  DD 000F1          PUSHL   HDR1
                          0000G     CF  02  FB 000F5          CALLS   #2, READ_BLOCK
                                04      50  E8 000FA          BLBS    R0, 18$
                                    0224    8F  BF 000FD      CHMU    #548                                    ; 0691
                        31464F45    8F  0000G   DF  D1 00101 18$:    CMPL    @HDR1, #826691397               ; 0693
                                    04  12 0010A             BNEQ    19$
                                    0870    8F  BF 0010C      CHMU    #2160                                   ; 0695
                                        0000G   30 00110 19$:       BSBW    NEXT_VOL_READ                   ; 0697
                                        03  11 00113          BRB     21$                                     ; 0684
                                        0000G   30 00115 20$:       BSBW    NEXT_VOL_WRITE                  ; 0700
                                        7E  D4 00118 21$:       CLRL    -(SP)                               ; 0701
                                        5E  DD 0011A          PUSHL   SP
                                    0000G   CF  9F 0011C      PUSHAB  START_VIO
                        00000000G   9F      03  FB 00120 22$:   CALLS   #3, @#SYS$CMKRNL
                                            04  00127          RET                                         ; 0655
                          0000G     CF      00  FB 00128 23$:   CALLS   #0, SPACE_IN_FILE                   ; 0705
                                            04  0012D          RET
                          0000G     CF      00  FB 0012E 24$:   CALLS   #0, REWIND_VOL_SET                  ; 0708
                                            04  00133          RET
                                    00E4    8F  BF 00134 25$:   CHMU    #228                                ; 0714
                                            04  00138          RET                                         ; 0717

; Routine Size:  313 bytes,    Routine Base:  $CODE$ + 0000

;  336          0718  1
```

```
;  338     0719  1 ROUTINE MOUNT : COMMON_CALL NOVALUE =
;  339     0720  1
;  340     0721  1 !++
;  341     0722  1 !
;  342     0723  1 ! FUNCTIONAL DESCRIPTION:
;  343     0724  1 !       This routine gets a virtual page for the mounted volume to use
;  344     0725  1 !
;  345     0726  1 ! CALLING SEQUENCE:
;  346     0727  1 !       Mount(), must be called in kernel mode
;  347     0728  1 !
;  348     0729  1 ! INPUT PARAMETERS:
;  349     0730  1 !       None
;  350     0731  1 !
;  351     0732  1 ! IMPLICIT INPUTS:
;  352     0733  1 !       CURRENT_UCB - address of currnet unit control block
;  353     0734  1 !       CURRENT_VCB - address of current volume control block
;  354     0735  1 !
;  355     0736  1 ! OUTPUT PARAMETERS:
;  356     0737  1 !       None
;  357     0738  1 !
;  358     0739  1 ! IMPLICIT OUTPUTS:
;  359     0740  1 !       Virtual page for volume to use
;  360     0741  1 !
;  361     0742  1 ! ROUTINE VALUE:
;  362     0743  1 !       None
;  363     0744  1 !
;  364     0745  1 ! SIDE EFFECTS:
;  365     0746  1 !       None
;  366     0747  1 !
;  367     0748  1 !--
;  368     0749  1
;  369     0750  2     BEGIN
;  370     0751  2
;  371     0752  2     EXTERNAL REGISTER
;  372     0753  2         COMMON_REG;
;  373     0754  2
;  374     0755  2     EXTERNAL ROUTINE
;  375     0756  2         GET_FREE_PAGE   : COMMON_CALL;          ! get free virtual page
;  376     0757  2
;  377     0758  2     EXTERNAL
;  378     0759  2
;  379     0760  2         ! address of current unit control block
;  380     0761  2         !
;  381     0762  2         CURRENT_UCB     : REF BBLOCK;
;  382     0763  2
;  383     0764  2     LOCAL
;  384     0765  2         JIB     : REF BBLOCK,
;  385     0766  2         PCB     : REF BBLOCK,
;  386     0767  2         VPAGE   : REF BBLOCK;   ! address of virtual page for volume set
;  387     0768  2
;  388     0769  2     ! get virtual page for use by the volume set
;  389     0770  2     !
;  390     0771  2     GET_FREE_PAGE(1, VPAGE);
;  391     0772  2     VPAGE[VVP$B_TYPE] = VVP_TYPE;
;  392     0773  2     INSQUE(.VPAGE, CURRENT_VCB[VCB$L_VPFL]);
;  393     0774  2     VPAGE[VVP$L_STALLIOFL] = VPAGE[VVP$L_STALLIOFL];
;  394     0775  2     VPAGE[VVP$L_STALLIOBL] = VPAGE[VVP$L_STALLIOFL];
```

```
;  395          0776  2
;  396          0777  2        CURRENT_UCB[UCB$L_DEVCHAR] = .CURRENT_UCB[UCB$L_DEVCHAR]
;  397          0778  2                                     OR
;  398          0779  2                                     (DEV$M_MNT OR DEV$M_DIR OR DEV$M_SDI);
;  399          0780  2
;  400          0781  2        ! save the Account and User names
;  401          0782  2        !
;  402          0783  2        PCB = .SCH$GL_PCBVEC [ .(IO_PACKET[IRP$L_PID])<0, 16> ];
;  403          0784  2        JIB = .PCB [ PCB$L_JIB ];
;  404          0785  2        CH$MOVE ( VVP$S_USERNAME, JIB [JIB$T_USERNAME], VPAGE [VVP$T_USERNAME] );
;  405          0786  2        CH$MOVE ( VVP$S_ACCOUNT,  JIB [JIB$T_ACCOUNT],  VPAGE [VVP$T_ACCOUNT] );
;  406          0787  1        END;                                            ! end of page


                                                                .EXTRN  GET_FREE_PAGE

                                                00FC 00000 MOUNT:   .WORD   Save R2,R3,R4,R5,R6,R7                  ; 0719
                5E                          04   C2 00002           SUBL2   #4, SP
                                            5E   DD 00005           PUSHL   SP                                      ; 0771
                                            01   DD 00007           PUSHL   #1
          0000G CF                          02   FB 00009           CALLS   #2, GET_FREE_PAGE
                50                          6E   D0 0000E           MOVL    VPAGE, R0                               ; 0772
            0A  A0                          02   90 00011           MOVB    #2, 10(R0)
                51              3C          AB   9E 00015           MOVAB   60(R11), R1                             ; 0773
                61                          60   0E 00019           INSQUE  (R0), (R1)
                57                          6E   D0 0001C           MOVL    VPAGE, R7                               ; 0774
                50          01A4            C7   9E 0001F           MOVAB   420(R7), R0
                60                          50   D0 00024           MOVL    R0, (R0)
            01A8 C7                         50   D0 00027           MOVL    R0, 424(R7)                             ; 0775
                50          0000G           CF   D0 0002C           MOVL    CURRENT_UCB, R0                         ; 0777
            38  A0      00080018            8F   C8 00031           BISL2   #524312, 56(R0)                         ; 0779
                51      00000000G           9F   D0 00039           MOVL    @#SCH$GL_PCBVEC, R1                     ; 0783
                50          0000G           CF   D0 00040           MOVL    IO_PACKET, R0
                50                          0C   C0 00045           ADDL2   #12, R0                                 ;
                50                          60   3C 00048           MOVZWL  (R0), R0                                ;
                50                        6140   D0 0004B           MOVL    (R1)[R0], PCB                           ;
                56          0080            C0   D0 0004F           MOVL    128(PCB), JIB                           ; 0784
          01B0  C7          0C    A6              0C   28 00054     MOVC3   #12, 12(JIB), 432(R7)                   ; 0785
          01BC  C7          18    A6              08   28 0005B     MOVC3   #8, 24(JIB), 444(R7)                    ; 0786
                                                    04 00062     RET                                             ; 0787

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 0139
```

```
;  408          0788  1 ROUTINE CANCEL_IO : COMMON_CALL NOVALUE =
;  409          0789  1
;  410          0790  1 !++
;  411          0791  1 !
;  412          0792  1 ! FUNCTIONAL DESCRIPTION:
;  413          0793  1 !       This routine sets the cancel io indicator if a file is accessed.
;  414          0794  1 !
;  415          0795  1 ! CALLING SEQUENCE:
;  416          0796  1 !       CANCEL_IO()
;  417          0797  1 !
;  418          0798  1 ! INPUT PARAMETERS:
;  419          0799  1 !       None
;  420          0800  1 !
;  421          0801  1 ! IMPLICIT INPUTS:
;  422          0802  1 !       CURRENT_VCB - address of current volume control block
;  423          0803  1 !
;  424          0804  1 ! OUTPUT PARAMETERS:
;  425          0805  1 !       None
;  426          0806  1 !
;  427          0807  1 ! IMPLICIT OUTPUTS:
;  428          0808  1 !       None
;  429          0809  1 !
;  430          0810  1 ! ROUTINE VALUE:
;  431          0811  1 !       None
;  432          0812  1 !
;  433          0813  1 ! SIDE EFFECTS:
;  434          0814  1 !       None
;  435          0815  1 !
;  436          0816  1 ! USER ERRORS:
;  437          0817  1 !       None
;  438          0818  1 !
;  439          0819  1 !--
;  440          0820  1
;  441          0821  2     BEGIN
;  442          0822  2
;  443          0823  2     EXTERNAL REGISTER
;  444          0824  2         COMMON_REG;
;  445          0825  2
;  446          0826  2     IF .CURRENT_VCB[VCB$L_WCB] NEQ 0
;  447          0827  2         OR
;  448          0828  2         .CURRENT_VCB[VCB$V_WAIREWIND]
;  449          0829  2         OR
;  450          0830  2         .CURRENT_VCB[VCB$V_WAIMOUVOL]
;  451          0831  2     THEN
;  452          0832  2
;  453          0833  2         ! remember that cancel was issued
;  454          0834  2         !
;  455          0835  2         CURRENT_VCB[VCB$V_CANCELIO] = 1;
;  456          0836  2
;  457          0837  1     END;


                                0000 00000 CANCEL_IO:
                                                .WORD   Save nothing                    ; 0788
```

```
                                38   AB  D5 00002          TSTL    56(CURRENT_VCB)                          ; 0826
                                     0A  12 00005          BNEQ    1$                                       ;
                    05      0B   AB      03  E0 00007      BBS     #3, 11(CURRENT_VCB), 1$                  ; 0828
                    04      0B   AB      02  E1 0000C      BBC     #2, 11(CURRENT_VCB), 2$                  ; 0830
                            0B   AB      20  88 00011 1$:  BISB2   #32, 11(CURRENT_VCB)                     ; 0835
                                         04  00015 2$:     RET                                              ; 0837

; Routine Size:  22 bytes,    Routine Base:  $CODE$ + 019C

;  458          0838  1
```

```
;  460     0839  1 GLOBAL ROUTINE DO_CANCEL : COMMON_CALL NOVALUE =
;  461     0840  1
;  462     0841  1 !++
;  463     0842  1 !
;  464     0843  1 ! FUNCTIONAL DESCRIPTION:
;  465     0844  1 !       This routine cancels all blocked io and acp functions.
;  466     0845  1 !
;  467     0846  1 ! CALLING SEQUENCE:
;  468     0847  1 !       DO_CANCEL(), called in kernel mode
;  469     0848  1 !
;  470     0849  1 ! INPUT PARAMETERS:
;  471     0850  1 !       None
;  472     0851  1 !
;  473     0852  1 ! IMPLICIT INPUTS:
;  474     0853  1 !       CURRENT_VCB - address of current volume control block
;  475     0854  1 !       USER_STATUS - contains error code which blocked io should be returned with
;  476     0855  1 !
;  477     0856  1 ! OUTPUT PARAMETERS:
;  478     0857  1 !       USER_STATUS - reset to normal status
;  479     0858  1 !
;  480     0859  1 ! IMPLICIT OUTPUTS:
;  481     0860  1 !       None
;  482     0861  1 !
;  483     0862  1 ! ROUTINE VALUE:
;  484     0863  1 !       None
;  485     0864  1 !
;  486     0865  1 ! SIDE EFFECTS:
;  487     0866  1 !       None
;  488     0867  1 !
;  489     0868  1 ! USER ERRORS:
;  490     0869  1 !       None
;  491     0870  1 !
;  492     0871  1 !--
;  493     0872  1
;  494     0873  2     BEGIN
;  495     0874  2
;  496     0875  2     EXTERNAL REGISTER
;  497     0876  2         COMMON_REG;
;  498     0877  2
;  499     0878  2     EXTERNAL ROUTINE
;  500     0879  2         CHECK_DISMOUNT  : COMMON_CALL;          ! get free virtual page
;  501     0880  2
;  502     0881  2     MAP
;  503     0882  2         USER_STATUS : LONG;
;  504     0883  2
;  505     0884  2     LOCAL
;  506     0885  2         ABD     : REF BBLOCKVECTOR [, ABD$C_LENGTH],
;  507     0886  2
;  508     0887  2         ! address containing information about blocked request
;  509     0888  2         !
;  510     0889  2         BLOCK_PAGE,
;  511     0890  2         FUNCTION        : BLOCK [1],
;  512     0891  2         PACKET  : REF BBLOCK,           ! address of io blocked request
;  513     0892  2         WINDOW;                         ! address of window for this request
;  514     0893  2
;  515     0894  2     ! If the process does not have a virtual page containing the information
;  516     0895  2     ! describing the blocked request then have  fatal error
```

```
;  517          0896  2      !
;  518          0897  2
;  519          0898  2      IF .CURRENT_VCB[VCB$L_VPFL] EQLA .CURRENT_VCB[VCB$L_VPBL]
;  520          0899  2      THEN
;  521          0900  2          BUG_CHECK(NOBVPVCB);
;  522          0901  2
;  523          0902  2      REMQUE(.CURRENT_VCB[VCB$L_VPBL], BLOCK_PAGE);
;  524          0903  2      PACKET = .(.BLOCK_PAGE + VVP$K_LENGTH + IO_PACKET - USER_STATUS);
;  525          0904  2      RET_FREE_PAGE(.BLOCK_PAGE,FALSE);   ! return page(s) to virtual memory
;  526          0905  2      RETURN_ALL_ERR();                   ! return all blocked physical io in error
;  527          0906  2
;  528          0907  2      IF .CURRENT_VCB[VCB$V_WAIMOUVOL]
;  529          0908  2          OR
;  530          0909  2          .CURRENT_VCB[VCB$V_WAIREWIND]
;  531          0910  2      THEN
;  532          0911  2          TERMINATE_VOL(.CURRENT_VCB[VCB$L_WCB]);
;  533          0912  2
;  534          0913  2      ! If fib descriptor present, zero count so the fib is not returned.
;  535          0914  2      ! complete i/o.
;  536          0915  2      !
;  537          0916  2
;  538          0917  2      IF .PACKET NEQ 0
;  539          0918  2      THEN
;  540          0919  3          BEGIN
;  541          0920  3
;  542          0921  3          !
;  543          0922  3          !
;  544          0923  3
;  545          0924  3          IF .PACKET[IRP$V_COMPLX]
;  546          0925  3          THEN
;  547          0926  4              BEGIN
;  548          0927  4              FUNCTION = .PACKET[IRP$W_FUNC];
;  549          0928  4              ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  550          0929  4
;  551          0930  4              IF .FUNCTION[IO$V_ACCESS]
;  552          0931  4              THEN
;  553          0932  4                  ZERO_CHANNEL(.PACKET)
;  554          0933  4              ELSE
;  555          0934  5                  BEGIN
;  556          0935  5                  FUNCTION = .PACKET[IRP$V_FCODE];
;  557          0936  5
;  558          0937  5                  IF .FUNCTION NEQ IO$_DEACCESS
;  559          0938  5                  THEN
;  560          0939  5                      ABD[ABD$C_WINDOW, ABD$W_COUNT] = 0;
;  561          0940  5
;  562          0941  4                  END;
;  563          0942  4
;  564          0943  4              ABD[ABD$C_FIB, ABD$W_COUNT] = 0;
;  565          0944  3              END;
;  566          0945  3
;  567          0946  3          IO_DONE(.PACKET);
;  568          0947  2          END;
;  569          0948  2
;  570          0949  2      ! return stalled i/o with cancel
;  571          0950  2      !
;  572          0951  2
;  573          0952  2      WHILE 1
```

```
;  574     0953  2     DO
;  575     0954  3         BEGIN
;  576     0955  3
;  577     0956  3         LOCAL
;  578     0957  3             SAVE_STATUS;
;  579     0958  3
;  580     0959  3         IF REMQUE(.BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_STALLIOFL], PACKET)
;  581     0960  3         THEN
;  582     0961  3             EXITLOOP;
;  583     0962  3
;  584     0963  3         IF .PACKET[IRP$V_COMPLX]
;  585     0964  3         THEN
;  586     0965  4             BEGIN
;  587     0966  4             FUNCTION = .PACKET[IRP$W_FUNC];
;  588     0967  4             ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  589     0968  4
;  590     0969  4             IF .FUNCTION[IO$V_ACCESS]
;  591     0970  4             THEN
;  592     0971  4                 ZERO_CHANNEL(.PACKET)
;  593     0972  4             ELSE
;  594     0973  4                 ABD[ABD$C_WINDOW, ABD$W_COUNT] = 0;
;  595     0974  4
;  596     0975  4             ABD[ABD$C_FIB, ABD$W_COUNT] = 0;
;  597     0976  3             END;
;  598     0977  3
;  599     0978  3         ! If this is a cancel request, return is with normal status
;  600     0979  3         !
;  601     0980  3         SAVE_STATUS = .USER_STATUS;
;  602     0981  3         FUNCTION = .PACKET[IRP$V_FCODE];
;  603     0982  3
;  604     0983  3         IF .FUNCTION EQL IO$_ACPCONTROL
;  605     0984  3             AND
;  606     0985  3             NOT .PACKET[IRP$V_VIRTUAL]
;  607     0986  3         THEN
;  608     0987  3             USER_STATUS = 1;
;  609     0988  3
;  610     0989  3         IO_DONE(.PACKET);
;  611     0990  3         USER_STATUS = .SAVE_STATUS;
;  612     0991  2         END;
;  613     0992  2
;  614     0993  2     ! If no file is accessed, turn off cancel I/O bit now.
;  615     0994  2     !
;  616     0995  2
;  617     0996  2     IF .CURRENT_VCB[VCB$L_WCB] EQL 0
;  618     0997  2     THEN
;  619     0998  3         BEGIN
;  620     0999  3         CURRENT_VCB [ VCB$V_CANCELIO ] = 0;
;  621     1000  3
;  622     1001  3         ! If while the cancel I/O was pending a dismount could have been issued
;  623     1002  3         ! and refused waiting for cancel I/O to complete.  Check for dismount.
;  624     1003  3         !
;  625     1004  3         CHECK_DISMOUNT ( .BBLOCK [ .CURRENT_VCB[VCB$L_RVT], RVT$L_UCBLST ] );
;  626     1005  2         END;
;  627     1006  2
;  628     1007  2     CURRENT_VCB[VCB$V_WAIREWIND] = 0;          ! no longer waiting
;  629     1008  2     CURRENT_VCB[VCB$V_WAIUSRLBL] = 0;
;  630     1009  2     CURRENT_VCB[VCB$V_WAIMOUVOL] = 0;
```

```
;  631          1010  2     ERROR(SS$_NORMAL);          ! cancel function should complete normally
;  632          1011  1     END;


                                                        .EXTRN  CHECK_DISMOUNT, BUG$_NOBVPVCB

                                        007C 00000      .ENTRY  DO_CANCEL, Save R2,R3,R4,R5,R6          ; 0839
                              56  0000G CF  9E 00002    MOVAB   USER_STATUS, R6                         ;
                        40    AB    3C    AB  D1 00007    CMPL    60(CURRENT_VCB), 64(CURRENT_VCB)        ; 0898
                                          04  12 0000C    BNEQ    1$                                      ;
                                        FEFF  0000E     BUGW                                            ; 0900
                                       0000* 00010      .WORD   <BUG$_NOBVPVCB!4>                       ;
                              50    40    BB  0F 00012 1$: REMQUE  @64(CURRENT_VCB), BLOCK_PAGE           ; 0902
                              52  0000GCF40  9E 00016    MOVAB   IO_PACKET+12[BLOCK_PAGE], R2            ; 0903
                              51          66  9E 0001C    MOVAB   USER_STATUS, R1                         ;
                              52          51  C2 0001F    SUBL2   R1, R2                                  ;
                              52          62  D0 00022    MOVL    (R2), PACKET                            ;
                                          7E  D4 00025    CLRL    -(SP)                                   ; 0904
                              50          DD  00027       PUSHL   BLOCK_PAGE                              ;
                        0000G CF          02  FB 00029    CALLS   #2, RET_FREE_PAGE                       ;
                        0000G CF          00  FB 0002E    CALLS   #0, RETURN_ACL_ERR                      ; 0905
                  05      0B  AB          02  E0 00033    BBS     #2, 11(CURRENT_VCB), 2$                 ; 0907
                  08      0B  AB          03  E1 00038    BBC     #3, 11(CURRENT_VCB), 3$                 ; 0909
                        38    AB    DD  0003D 2$:   PUSHL   56(CURRENT_VCB)                         ; 0911
                        0000V CF          01  FB 00040    CALLS   #1, TERMINATE_VOL                       ;
                                          52  D5 00045 3$:  TSTL    PACKET                              ; 0917
                                          32  13 00047    BEQL    7$                                      ;
                        26      2A  A2    03  E1 00049    BBC     #3, 42(PACKET), 6$                      ; 0924
                              55    20  A2  3C  0004E    MOVZWL  32(PACKET), FUNCTION                    ; 0927
                              53    2C  B2  D0  00052    MOVL    @44(PACKET), ABD                        ; 0928
                        09    55          06  E1 00056    BBC     #6, FUNCTION, 4$                        ; 0930
                                          52  DD 0005A    PUSHL   PACKET                                  ; 0932
                        0000G CF          01  FB 0005C    CALLS   #1, ZERO_CHANNEL                        ;
                                          0E  11 00061    BRB     5$                                      ;
55      20  A2          06          00  EF 00063 4$:  EXTZV   #0, #6, 32(PACKET), FUNCTION            ; 0935
                              34          55  D1 00069    CMPL    FUNCTION, #52                           ; 0937
                                          03  13 0006C    BEQL    5$                                      ;
                                    02    A3  B4 0006E    CLRW    2(ABD)                                  ; 0939
                                    0A    A3  B4 00071 5$:  CLRW    10(ABD)                             ; 0943
                                          52  DD 00074 6$:  PUSHL   PACKET                              ; 0946
                        0000G CF          01  FB 00076    CALLS   #1, IO_DONE                             ;
                              50    3C    AB  D0 0007B 7$:  MOVL    60(CURRENT_VCB), R0                   ; 0959
                              52  01A4    D0  0F 0007F    REMQUE  @420(R0), PACKET                        ;
                                          42  1D 00084    BVS     12$                                     ;
                        1B      2A  A2    03  E1 00086    BBC     #3, 42(PACKET), 10$                     ; 0963
                              55    20  A2  3C  0008B    MOVZWL  32(PACKET), FUNCTION                    ; 0966
                              53    2C  B2  D0  0008F    MOVL    @44(PACKET), ABD                        ; 0967
                        09    55          06  E1 00093    BBC     #6, FUNCTION, 8$                        ; 0969
                                          52  DD 00097    PUSHL   PACKET                                  ; 0971
                        0000G CF          01  FB 00099    CALLS   #1, ZERO_CHANNEL                        ;
                                          03  11 0009E    BRB     9$                                      ;
                                    02    A3  B4 000A0 8$:  CLRW    2(ABD)                              ; 0973
                                    0A    A3  B4 000A3 9$:  CLRW    10(ABD)                             ; 0975
                              54          66  D0 000A6 10$: MOVL    USER_STATUS, SAVE_STATUS            ; 0980
55      20  A2          06          00  EF 000A9       EXTZV   #0, #6, 32(PACKET), FUNCTION            ; 0981
                              38          55  D1 000AF       CMPL    FUNCTION, #56                       ; 0983
```

```
                                                08  12  000B2          BNEQ    11$                                             ;
                        03      2A      A2      04  E0  000B4          BBS     #4, 42(PACKET), 11$                             ; 0985
                                66      66      01  D0  000B9          MOVL    #1, USER_STATUS                                 ; 0987
                                                52  DD  000BC  11$:     PUSHL   PACKET                                          ; 0989
                                0000G   CF      01  FB  000BE          CALLS   #1, IO_DONE
                                66              54  D0  000C3          MOVL    SAVE_STATUS, USER_STATUS                        ; 0990
                                                B3  11  000C6          BRB     7$                                              ; 0952
                                        38      AB  D5  000C8  12$:     TSTL    56(CURRENT_VCB)                                 ; 0996
                                                10  12  000CB          BNEQ    13$
                                0B      AB      20  8A  000CD          BICB2   #32, 11(CURRENT_VCB)                           ; 0999
                                50      20      AB  D0  000D1          MOVL    32(CURRENT_VCB), R0                             ; 1004
                                        44      A0  DD  000D5          PUSHL   68(R0)
                                0000G   CF      01  FB  000D8          CALLS   #1, CHECK_DISMOUNT
                                0B      AB      1C  8A  000DD  13$:     BICB2   #28, 11(CURRENT_VCB)                           ; 1009
                                66              01  B0  000E1          MOVW    #1, USER_STATUS                                 ; 1010
                                                    04  000E4          RET                                                     ; 1011

; Routine Size:  229 bytes,    Routine Base:  $CODE$ + 01B2

;  633          1012  1
```

```
;  635     1013  1 GLOBAL ROUTINE TERMINATE_VOL (WINDOW) : COMMON_CALL NOVALUE =
;  636     1014  1
;  637     1015  1 !++
;  638     1016  1 !
;  639     1017  1 ! FUNCTIONAL DESCRIPTION:
;  640     1018  1 !       This routine terminates a mount request.  If a file is open
;  641     1019  1 !       then the user must close the file.  The write indicator is cleared so
;  642     1020  1 !       that eof trailers are not written on deaccess.  The volume is marked
;  643     1021  1 !       not mounted and the volume position is marked ambiguous.
;  644     1022  1 !
;  645     1023  1 ! CALLING SEQUENCE:
;  646     1024  1 !       TERMINATE_MOUNT(WINDOW), called in kernel mode
;  647     1025  1 !
;  648     1026  1 ! INPUT PARAMETERS:
;  649     1027  1 !       ARG1 - address of window for request
;  650     1028  1 !
;  651     1029  1 ! IMPLICIT INPUTS:
;  652     1030  1 !       None
;  653     1031  1 !
;  654     1032  1 ! OUTPUT PARAMETERS:
;  655     1033  1 !       None
;  656     1034  1 !
;  657     1035  1 ! IMPLICIT OUTPUTS:
;  658     1036  1 !       None
;  659     1037  1 !
;  660     1038  1 ! ROUTINE VALUE:
;  661     1039  1 !       None
;  662     1040  1 !
;  663     1041  1 ! SIDE EFFECTS:
;  664     1042  1 !       None
;  665     1043  1 !
;  666     1044  1 ! USER ERRORS:
;  667     1045  1 !       None
;  668     1046  1 !
;  669     1047  1 !--
;  670     1048  1
;  671     1049  2     BEGIN
;  672     1050  2
;  673     1051  2     EXTERNAL ROUTINE
;  674     1052  2         GET_CCB;
;  675     1053  2
;  676     1054  2     EXTERNAL REGISTER
;  677     1055  2         COMMON_REG;
;  678     1056  2
;  679     1057  2     MAP
;  680     1058  2         WINDOW  : REF BBLOCK;           ! address of window control block
;  681     1059  2
;  682     1060  2     LOCAL
;  683     1061  2         MVL_ENTRY       : REF BBLOCK;                   ! address of MVL entry
;  684     1062  2
;  685     1063  2     IF .WINDOW NEQ 0
;  686     1064  2     THEN                                            ! a file is open
;  687     1065  2         BEGIN
;  688     1066  3         CURRENT_VCB[VCB$V_NOWRITE] = 1;
;  689     1067  3         CURRENT_VCB[VCB$V_MUSTCLOSE] = 1;       ! the file must be closed
;  690     1068  2         END;
;  691     1069  2
```

```
;  692     1070  2      IF .CURRENT_VCB[VCB$V_WAIMOUVOL]
;  693     1071  2      THEN
;  694     1072  2          BEGIN
;  695     1073  2
;  696     1074  2          LOCAL
;  697     1075  2              CCB : REF BBLOCK,
;  698     1076  2              UCB : REF VECTOR;
;  699     1077  2
;  700     1078  2          MVL_ENTRY = .CURRENT_VCB[VCB$L_MVL] + MVL$K_FIXLEN +
;  701     1079  2              ((.CURRENT_VCB[VCB$B_CUR_RVN] - 1)*MVL$K_LENGTH);
;  702     1080  2          MVL_ENTRY[MVL$V_MOUNTED] = 0;           ! volume is not mounted
;  703     1081  2          UCB = BBLOCK[.CURRENT_VCB[VCB$L_RVT], RVT$L_UCBLST];
;  704     1082  2          UCB = .UCB[.CURRENT_VCB[VCB$W_RVN]];
;  705     1083  2          CCB = GET_CCB ( .IO_CHANNEL );
;  706     1084  2          CCB[CCB$L_UCB] = .UCB;
;  707     1085  2          SYS$QIOW(0, .IO_CHANNEL,
;  708     1086  2              IO$_REWINDOFF
;  709     1087  2              OR
;  710     1088  2              IO$M_NOWAIT
;  711     1089  2              OR
;  712     1090  2              IO$M_CLSEREXCP, 0, 0, 0, 0, 0, 0, 0, 0, 0);
;  713     1091  2          SEND_ERRLOG(0, .UCB);
;  714     1092  2          CURRENT_VCB[VCB$B_CUR_RVN] = 0;         ! no volume is current
;  715     1093  2
;  716     1094  2          ! no file is current, ie: start at beginning
;  717     1095  2          !
;  718     1096  2          CURRENT_VCB[VCB$L_CUR_FID] = 0;
;  719     1097  2          END;
;  720     1098  2
;  721     1099  1      END;                                    ! end of routine TERMINATE_MOUNT


                                                .EXTRN  GET_CCB

                                        0004 00000      .ENTRY  TERMINATE_VOL, Save R2                          ; 1013
                                    04  AC   D5 00002    TSTL    WINDOW                                          ; 1063
                                    05  13   00005    BEQL    1$
                                0B  AB      C0  8F   88 00007    BISB2   #192, 11(CURRENT_VCB)                   ; 1067
                            58  0B  AB          02   E1 0000C 1$: BBC     #2, 11(CURRENT_VCB), 2$                 ; 1070
                                50      2F  AB  9A   00011    MOVZBL  47(CURRENT_VCB), R0                     ; 1079
                                50      34 BB40 7E   00015    MOVAQ   @52(CURRENT_VCB)[R0], MVL_ENTRY         ; 1078
                                50          1C  C0   0001A    ADDL2   #28, MVL_ENTRY
                            07  A0          01  8A   0001D    BICB2   #1, 7(MVL_ENTRY)                        ; 1080
                            52  20  AB 00000044 8F   C1 00021    ADDL3   #68, 32(CURRENT_VCB), UCB               ; 1081
                                50      0E  AB  3C   0002A    MOVZWL  14(CURRENT_VCB), R0                     ; 1082
                                52      6240    D0   0002E    MOVL    (UCB)[R0], UCB
                                    0000G   CF  DD   00032    PUSHL   IO_CHANNEL                              ; 1083
                            0000G   CF          01   FB 00036    CALLS   #1, GET_CCB
                                60          52  D0   0003B    MOVL    UCB, (CCB)                              ; 1084
                                            7E  7C   0003E    CLRQ    -(SP)                                   ; 1085
                                            7E  7C   00040    CLRQ    -(SP)
                                            7E  7C   00042    CLRQ    -(SP)
                                            7E  7C   00044    CLRQ    -(SP)
                                            7E  D4   00046    CLRL    -(SP)
                                7E      02A2    8F   3C 00048    MOVZWL  #674, -(SP)                             ; 1089
                                    0000G   CF  DD   0004D    PUSHL   IO_CHANNEL                              ; 1085
```

```
                                 7E  D4 00051          CLRL    -(SP)                                   ;
                    00000000G    9F                   0C  FB 00053          CALLS   #12, @#SYS$QIOW                 ;
                                                      52  DD 0005A          PUSHL   UCB                             ; 1091
                                                      7E  D4 0005C          CLRL    -(SP)                           ;
                         0000G   CF                   02  FB 0005E          CALLS   #2, SEND_ERRLOG                 ;
                                                  2F  AB  94 00063          CLRB    47(CURRENT_VCB)                 ; 1092
                                                  24  AB  D4 00066          CLRL    36(CURRENT_VCB)                 ; 1096
                                                      04  00069 2$:         RET                                     ; 1099

; Routine Size:  106 bytes,    Routine Base:  $CODE$ + 0297

;  722          1100  1
```

```
;  724     1101  1 GLOBAL ROUTINE MTA_MOUNT : NOPRES NOVALUE =
;  725     1102  1
;  726     1103  1 !++
;  727     1104  1 !
;  728     1105  1 ! FUNCTIONAL DESCRIPTION:
;  729     1106  1 !       This routine checks the validity of the mount request and
;  730     1107  1 !       sets up a virtual page for this volume set.
;  731     1108  1 !
;  732     1109  1 !
;  733     1110  1 ! CALLING SEQUENCE:
;  734     1111  1 !       MTA_MOUNT()
;  735     1112  1 !
;  736     1113  1 ! INPUT PARAMETERS:
;  737     1114  1 !       None
;  738     1115  1 !
;  739     1116  1 ! IMPLICIT INPUTS:
;  740     1117  1 !       CURRENT_UCB     - address of current unit control block
;  741     1118  1 !       QUEUE_HEAD      - address of queue head for ACP
;  742     1119  1 !
;  743     1120  1 ! OUTPUT PARAMETERS:
;  744     1121  1 !       None
;  745     1122  1 !
;  746     1123  1 ! IMPLICIT OUTPUTS:
;  747     1124  1 !       one page of virtual memory is devoted to this volume set
;  748     1125  1 !
;  749     1126  1 ! ROUTINE VALUE:
;  750     1127  1 !       None
;  751     1128  1 !
;  752     1129  1 ! SIDE EFFECTS:
;  753     1130  1 !       None
;  754     1131  1 !
;  755     1132  1 !--
;  756     1133  1
;  757     1134  2     BEGIN
;  758     1135  2
;  759     1136  2     EXTERNAL
;  760     1137  2         CURRENT_UCB     : REF BBLOCK,
;  761     1138  2         QUEUE_HEAD      : REF BBLOCK;
;  762     1139  2
;  763     1140  2     EXTERNAL REGISTER
;  764     1141  2         COMMON_REG;
;  765     1142  2
;  766     1143  2     IF NOT .BBLOCK[CURRENT_UCB[UCB$L_DEVCHAR], DEV$V_SQD]
;  767     1144  2         OR
;  768     1145  2         .QUEUE_HEAD[AQB$B_ACPTYPE] NEQ AQB$K_MTA
;  769     1146  2     THEN
;  770     1147  2         ERR_EXIT(SS$_WRONGACP);
;  771     1148  2
;  772     1149  2     KERNEL_CALL(MOUNT);
;  773     1150  1     END;                                    ! end of routine MTA_MOUNT


                                0000 00000          .ENTRY  MTA_MOUNT, Save nothing                 ; 1101
                50      0000G   CF   D0 00002       MOVL    CURRENT_UCB, R0                         ; 1143
```

```
                 0B      38  A0                 05  E1 00007          BBC     #5, 56(R0), 1$                  ;
                             50      0000G      CF  D0 0000C          MOVL    QUEUE_HEAD, R0                  ; 1145
                             03        15       A0  91 00011          CMPB    21(R0), #3                      ;
                                                04  13 00015          BEQL    2$                              ;
                                      031C      8F  BF 00017 1$:      CHMU    #796                            ; 1147
                                                7E  D4 0001B 2$:      CLRL    -(SP)                           ; 1149
                                                5E  DD 0001D          PUSHL   SP                              ;
                                      FE15      CF  9F 0001F          PUSHAB  MOUNT                           ;
                          00000000G     9F          03  FB 00023          CALLS   #3, @#SYS$CMKRNL            ;
                                                    04  0002A          RET                                     ; 1150

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0301

;  774          1151  1
```

```
;  776          1152  1 ROUTINE STALL : COMMON_CALL NOVALUE =
;  777          1153  1
;  778          1154  1 !++
;  779          1155  1 !
;  780          1156  1 ! FUNCTIONAL DESCRIPTION:
;  781          1157  1 !
;  782          1158  1 !       This routine puts the cancel request packet on the stalled queue.
;  783          1159  1 !
;  784          1160  1 ! CALLING SEQUENCE:
;  785          1161  1 !       STALL(), called in KERNEL mode
;  786          1162  1 !
;  787          1163  1 ! INPUT PARAMETERS:
;  788          1164  1 !       None
;  789          1165  1 !
;  790          1166  1 ! IMPLICIT INPUTS:
;  791          1167  1 !       None
;  792          1168  1 !
;  793          1169  1 ! OUTPUT PARAMETERS:
;  794          1170  1 !       None
;  795          1171  1 !
;  796          1172  1 ! IMPLICIT OUTPUTS:
;  797          1173  1 !       cancel request queued to stall I/O queue
;  798          1174  1 !
;  799          1175  1 ! ROUTINE VALUE:
;  800          1176  1 !       None
;  801          1177  1 !
;  802          1178  1 ! SIDE EFFECTS:
;  803          1179  1 !       None
;  804          1180  1 !
;  805          1181  1 !--
;  806          1182  1
;  807          1183  2     BEGIN
;  808          1184  2
;  809          1185  2     EXTERNAL
;  810          1186  2         IO_PACKET       : REF BBLOCK;           ! address of current I/O packet
;  811          1187  2
;  812          1188  2     EXTERNAL REGISTER
;  813          1189  2         COMMON_REG;
;  814          1190  2
;  815          1191  2     LOCAL
;  816          1192  2         VPAGE    : REF BBLOCK;
;  817          1193  2
;  818          1194  2     VPAGE = .CURRENT_VCB[VCB$L_VPFL];
;  819          1195  2     INSQUE(.IO_PACKET, .VPAGE[VVP$L_STALLIOBL]);
;  820          1196  2     IO_PACKET = 0;
;  821          1197  1     END;


                                        0000 00000 STALL:   .WORD   Save nothing                          ; 1152
                        50      3C  AB  D0 00002            MOVL    60(CURRENT_VCB), VPAGE               ; 1194
                01A8    D0  0000G   DF  0E 00006            INSQUE  @IO_PACKET, @424(VPAGE)              ; 1195
                            0000G   CF  D4 0000D            CLRL    IO_PACKET                            ; 1196
                                        04 00011            RET                                          ; 1197
```

```
; Routine Size:  18 bytes,    Routine Base:  $CODE$ + 032C

;  822          1198  1 END
;  823          1199  1
;  824          1200  0 ELUDOM



;                                PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                         830  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        74         0      1000          00:01.9



;                                    COMMAND QUALIFIERS

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

; Size:          830 code + 0 data bytes
; Run Time:          00:20.6
; Elapsed Time:      00:56.8
; Lines/CPU Min:      3490
; Lexemes/CPU-Min:  19902
; Memory Used:  158 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