C 6

```
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     000000000     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  000         000  UUU         UUU  NNNNNN      NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM  000         000  UUU         UUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     000000000     UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
_S

Sy
--
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI

LI
LI


LI
LI
LI
LI
LI
LI
LI
LI
LN
LN
LO
LO

LO
LO
LO
LO
MA

MO
MO
MO
MO
MO
MO
MO


 MO
 MO
```

FILEID**SRCVOL   K 15

```
  SSSSSSSS  RRRRRRRR     CCCCCCCC  VV        VV     000000    LL
  SSSSSSSS  RRRRRRRR     CCCCCCCC  VV        VV     000000    LL
SS          RR        RR  CC        VV        VV  00        00  LL
SS          RR        RR  CC        VV        VV  00        00  LL
SS          RR        RR  CC        VV        VV  00        00  LL
SS          RR        RR  CC        VV        VV  00        00  LL
  SSSSSS    RRRRRRRR      CC        VV        VV  00        00  LL
  SSSSSS    RRRRRRRR      CC        VV        VV  00        00  LL
        SS  RR    RR      CC        VV        VV  00        00  LL
        SS  RR    RR      CC        VV        VV  00        00  LL
        SS  RR      RR    CC          VV    VV    00        00  LL
        SS  RR      RR    CC          VV    VV    00        00  LL          ....
SSSSSSSS    RR        RR    CCCCCCCC      VV        000000    LLLLLLLLLL      ....
SSSSSSSS    RR        RR    CCCCCCCC      VV        000000    LLLLLLLLLL      ....
                                                                               ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE SRCVOL (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MOUNT Utility Structure Level 1
1 !
1 ! ABSTRACT:
1 !
1 !       This routine searches the device database for a particular volume.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  13-Oct-1977  20:09
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-008 HH0050          Hai Huang               17-Aug-1984
1 !               Return success status if MOUNT/CLUSTER on a volume
1 !               that is already mounted locally.
1 !
1 !       V03-007 HH0041          Hai Huang               24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
```

```
;   58          0058  1 !       V03-006 HH0036          Hai Huang               11-Jul-1984
;   59          0059  1 !               Allow MOUNT/CLUSTER on a volume that is already mounted
;   60          0060  1 !               on the local  node.
;   61          0061  1 !
;   62          0062  1 !       V03-005 HH0024          Hai Huang               18-Jun-1984
;   63          0063  1 !               Call IOC$CVT_DEVNAM to format the device name on a
;   64          0064  1 !               successful shared mount.
;   65          0065  1 !
;   66          0066  1 !       V03-004 HH0002          Hai Huang               01-Feb-1984
;   67          0067  1 !               Add job-wide mount support, i.e. deallocate mount list
;   68          0068  1 !               entries to paged-pool.
;   69          0069  1 !
;   70          0070  1 !       V03-003 CDS0001         Christian D. Saether    17-Jun-1983
;   71          0071  1 !               Fix SEARCH_VOL so it knows how to run through
;   72          0072  1 !               all system blocks, and therefore find all ddb's.
;   73          0073  1 !
;   74          0074  1 !       V03-002 DMW4041         DMWalp                   7-Jun-1983
;   75          0075  1 !               Remove (S)LOG_ENTRY
;   76          0076  1 !
;   77          0077  1 !       V03-001 STJ50311        Steven T. Jeffreys,     11-Feb-1983
;   78          0078  1 !               Make all uses of PHYS_NAME indexed by DEVICE_INDEX.
;   79          0079  1 !
;   80          0080  1 !       V02-004 STJ0195         Steven T. Jeffreys,     02-Feb-1982
;   81          0081  1 !               Removed SYSEXV declaration.
;   82          0082  1 !
;   83          0083  1 !       V02-003 STJ0165         Steven T. Jeffreys,     04-Jan-1982
;   84          0084  1 !               Remove obsolete calls to $SETEXV.
;   85          0085  1 !
;   86          0086  1 !       V02-002 ACG0169         Andrew C. Goldstein,    18-Apr-1980  13:58
;   87          0087  1 !               Bug check on internal errors
;   88          0088  1 !
;   89          0089  1 !       V02-001 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:39
;   90          0090  1 !               Previous revision history moved to MOUNT.REV
;   91          0091  1 !**
;   92          0092  1
;   93          0093  1
;   94          0094  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   95          0095  1 REQUIRE 'SRC$:MOUDEF.B32';
;   96          0627  1
;   97          0628  1
;   98          0629  1 FORWARD ROUTINE
;   99          0630  1         SEARCH_VOL,                     ! search for mounted volume
;  100          0631  1         SEARCH_HANDLER  : NOVALUE;      ! condition handler for above
```

```
GLOBAL ROUTINE SEARCH_VOL (NAME) =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine searches the device database for a particular volume.
!       Only file structured devices mounted /SYSTEM, /GROUP, or /SHARE
!       are considered.
!
!
! CALLING SEQUENCE:
!       SEARCH_VOL (ARG1)
!
! INPUT PARAMETERS:
!       ARG1: string descriptor of volume label
!
! IMPLICIT INPUTS:
!       MOUNT_OPTIONS: command option bits
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       UCB address of found volume or zero
!
! SIDE EFFECTS:
!       mount count of found volume is incremented if /SHARE mode mount
!
!--

BEGIN

MAP
        NAME            : REF VECTOR;   ! volume name string descriptor

LABEL
        SEARCH_LOOP;                    ! outer device search loop

LOCAL
        STATUS,                         ! random status value
        SB              : REF BBLOCK,   ! pointer to current SB
        DDB             : REF BBLOCK,   ! pointer to current DDB
        UCB             : REF BBLOCK,   ! pointer to current UCB
        VCB             : REF BBLOCK;   ! pointer to current VCB

EXTERNAL
        MOUNT_OPTIONS   : BITVECTOR,    ! MOUNT command options
        DEVICE_INDEX    : LONG,         ! index into PHYS_NAME vector
        PHYS_NAME       : VECTOR,       ! physical device name descriptor
        NAME_BUFFER     : VECTOR [,BYTE], ! physical device name buffer
        MTL_ENTRY       : REF BBLOCK,   ! address of mount list entry
        SMTL_ENTRY      : REF BBLOCK,   ! address of volume set mount list entry
        SCS$GQ_CONFIG   : ADDRESSING_MODE (ABSOLUTE),
2                                               ! I/O database listhead
2               MOUNT_ITMLST     : LONG;         ! address of the mount item list
2
2 LINKAGE
2               IOC_CVT_DEVNAM  = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 4,
2                                       REGISTER = 5; REGISTER = 1) :
2                                       PRESERVE (2,3,4,5,6,7);
2
2 EXTERNAL ROUTINE
2               LOCK_IODB,                       ! lock I/O database mutex
2               UNLOCK_IODB,                     ! unlock I/O database mutex
2               DEALLOCATE_MEM,                  ! deallocate dynamic memory
2               ALLOC_LOGNAME,                   ! create logical name and MTL entry
2               ENTER_LOGNAME,                   ! hook up logical name and MTL entry
2               IOC$CVT_DEVNAM   : IOC_CVT_DEVNAM ADDRESSING_MODE (GENERAL),
2                                                ! get fully expanded device name
2               MOUNT_CLUSTER;                   ! mount volume cluster-wide
2
2
2 ! Enable our condition handler.
2 ! Needless to say, the search must be done with the I/O database locked to
2 ! prevent list perturbations. We run down the DDB list, following the UCB
2 ! list off each one, looking for file structured devices that are mounted
2 ! but not allocated.
2 !
2
2 ENABLE SEARCH_HANDLER;
2
2 ! For a shared mount, preallocate logical name and mounted volume list entry
2 ! for the worst case (RVN 1), since we cannot tolerate failures after bumping
2 ! the mount count in a found VCB. If not needed, we discard them later. System
2 ! and Group mounts don't need this, since we will fail here one way or the other.
2 !
2
2 IF .MOUNT_OPTIONS[OPT_SHARE]
2 THEN
3     BEGIN
3     ALLOC_LOGNAME (0);
3     SMTL_ENTRY = .MTL_ENTRY;
3     MTL_ENTRY = 0;
3     ALLOC_LOGNAME (1);
2     END;
2
2 LOCK_IODB ();
2
2 SB = .SCS$GQ_CONFIG;
2
2 SEARCH_LOOP:
3 BEGIN
3 UNTIL .SB EQL SCS$GQ_CONFIG
3 DO
4 BEGIN
4
4 DDB = .SB [SB$L_DDB];
4
4 WHILE .DDB NEQ 0
4 DO
```

```
5       BEGIN
5       IF .DDB[DDB$B_TYPE] NEQ DYN$C_DDB
5       THEN BUG_CHECK (NOTDDBDDB, FATAL, 'Corrupted DDB list');
5       UCB = .DDB[DDB$L_UCB];
5
5       UNTIL .UCB EQL 0 DO
6           BEGIN
6           IF .UCB[UCB$B_TYPE] NEQ DYN$C_UCB
6           THEN BUG_CHECK (NOTUCBUCB, FATAL, 'Corrupted UCB list');
6
6           IF  .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_FOD]
6           AND .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_MNT]
6           AND NOT .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_ALL]
6           AND NOT .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_DMT]
6           THEN
7               BEGIN
7               VCB = .UCB[UCB$L_VCB];
7               IF .VCB[VCB$B_TYPE] NEQ DYN$C_VCB
7               THEN BUG_CHECK (NOTVCBUCB, FATAL, 'Not VCB pointer in UCB');
7               IF CH$EQL (.NAME[0], .NAME[1],
7                           VCB$S_VOLNAME, VCB[VCB$T_VOLNAME], ' ')
7               THEN LEAVE SEARCH_LOOP;
6               END;
6           UCB = .UCB[UCB$L_LINK];
5           END;                                ! end of UCB scan loop
5       DDB = .DDB[DDB$L_LINK];
4       END;                                    ! end of DDB scan loop
4
4   SB = .SB [SB$L_FLINK];
3   END;
2   END;                                        ! end of block SEARCH_LOOP
2
2   ! If we find a suitable volume that matches the name, the search is over.
2   ! If the mount is /SHARE, we increment the volume mount count here and now
2   ! to avoid race conditions. Finding a volume on a /SYSTEM or /GROUP mount
2   ! is an error.
2   !
2
2   STATUS = 0;
2   IF .UCB NEQ 0
2   THEN
3       BEGIN
3       IF .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_FOR] NEQ .MOUNT_OPTIONS[OPT_FOREIGN]
3       THEN STATUS = MOUN$_INCOMPAT
3
3       ELSE
4           BEGIN
4
4           IF NOT .MOUNT_OPTIONS[OPT_SHARE]
4           THEN
4               STATUS = MOUN$_VOLALRMNT
4           ELSE
5               BEGIN
5               STATUS = 1;
5               VCB[VCB$W_MCOUNT] = .VCB[VCB$W_MCOUNT] + 1;
4               END;
4           '
```

```
;  273          0803  4           For a successful shared mount or volume already mounted, format the
;  274          0804  4           expanded device name and set up the device name descriptor.
;  275          0805  4
;  276          0806  4           IOC$CVT_DEVNAM (NAMEBUF_LEN,                       ! Output buffer length
;  277          0807  4                       NAME_BUFFER +
;  278          0808  4                         (.DEVICE_INDEX*NAMEBUF_LEN),     ! Output buffer address
;  279          0809  4                       -1,                                ! Format device name in display form
;  280          0810  4                       .UCB;                              ! Address of the target UCB
;  281          0811  4                       PHYS_NAME [.DEVICE_INDEX*2]);      ! Returned length
;  282          0812  4         PHYS_NAME[(.DEVICE_INDEX*2)+1] = NAME_BUFFER +
;  283          0813  4                         (.DEVICE_INDEX*NAMEBUF_LEN);     ! Set up device name descriptor
;  284          0814  3         END;
;  285          0815  2     END;
;  286          0816  2
;  287          0817  2 UNLOCK_IODB ();
;  288          0818  2
;  289          0819  2 !
;  290          0820  2 ! If the /CLUSTER option is specified, send the request cluster-wide even
;  291          0821  2 ! if the volume is already mounted on the local node. Return with success
;  292          0822  2 ! in this case.
;  293          0823  2 !
;  294          0824  3 IF   ( .STATUS EQL MOUN$_VOLALRMNT)
;  295          0825  3 AND ( .MOUNT_OPTIONS [OPT_CLUSTER])
;  296          0826  2 THEN
;  297          0827  3     BEGIN
;  298          0828  3     MOUNT_CLUSTER (.MOUNT_ITMLST);
;  299          0829  3     RETURN 1;
;  300          0830  2     END;
;  301          0831  2
;  302          0832  2 IF NOT .STATUS THEN ERR_EXIT (.STATUS);
;  303          0833  2
;  304          0834  2 !
;  305          0835  2 ! We now have a successful shared mount. Fill in the logical name and MTL
;  306          0836  2 ! entries.
;  307          0837  2 !
;  308          0838  2 ! If what we found is RVN 1 of a volume set, keep the entries as is. Otherwise
;  309          0839  2 ! release one of each.
;  310          0840  2 !
;  311          0841  2
;  312          0842  2 IF .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_SQD]
;  313          0843  2 OR .VCB[VCB$W_RVN] NEQ 1
;  314          0844  2 THEN
;  315          0845  3     BEGIN
;  316          0846  3     DEALLOCATE_MEM (.MTL_ENTRY, 1);
;  317          0847  3     MTL_ENTRY = .SMTL_ENTRY;
;  318          0848  3     SMTL_ENTRY = 0;
;  319          0849  2     END;
;  320          0850  2
;  321          0851  2 ENTER_LOGNAME (.UCB, .VCB);
;  322          0852  2
;  323          0853  2 RETURN 1;
;  324          0854  2
;  325          0855  1 END;                                   ! end of routine SEARCH_VOL


                                                                    .TITLE  SRCVOL
                                                                    .IDENT  \V04-000\
                                                         .EXTRN   MOUNT_OPTIONS, DEVICE_INDEX
                                                         .EXTRN   PHYS_NAME, NAME_BUFFER
                                                         .EXTRN   MTL_ENTRY, SMTL_ENTRY
                                                         .EXTRN   SCS$GQ_CONFIG, MOUNT_ITMLST
                                                         .EXTRN   LOCK_IODB, UNLOCK_IODB
                                                         .EXTRN   DEALLOCATE_MEM, ALLOC_LOGNAME
                                                         .EXTRN   ENTER_LOGNAME, IOC$CVT_DEVNAM
                                                         .EXTRN   MOUNT_CLUSTER, BUG$_NOTDDBDDB
                                                         .EXTRN   BUG$_NOTUCBUCB, BUG$_NOTVCBUCB

                                                         .PSECT   $CODE$,NOWRT,2

                                    0FFC 00000           .ENTRY   SEARCH_VOL, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0632
                                                                  R10,R11                                       ;
                     5B      0000G   CF   9E 00002       MOVAB    SMTL_ENTRY, R11                               ;
                     5A      0000G   CF   9E 00007       MOVAB    MOUNT_OPTIONS, R10                            ;
                     59 00000000G    9F   9E 0000C       MOVAB    @#SCS$GQ_CONFIG, R9                           ;
                     58      0000G   CF   9E 00013       MOVAB    MTL_ENTRY, R8                                 ;
                     6D       0137   CF   DE 00018       MOVAL    21$, (FP)                                     ; 0666
          13         6A              06   E1 0001D       BBC      #6, MOUNT_OPTIONS, 1$                         ; 0723
                                     7E   D4 00021       CLRL     -(SP)                                         ; 0726
               0000G CF              01   FB 00023       CALLS    #1, ALLOC_LOGNAME                             ;
                     6B              68   D0 00028       MOVL     MTL_ENTRY, SMTL_ENTRY                         ; 0727
                                     68   D4 0002B       CLRL     MTL_ENTRY                                     ; 0728
                                     01   DD 0002D       PUSHL    #1                                            ; 0729
               0000G CF              01   FB 0002F       CALLS    #1, ALLOC_LOGNAME                             ;
               0000G CF              00   FB 00034 1$:   CALLS    #0, LOCK_IODB                                 ; 0732
                     57              69   D0 00039       MOVL     SCS$GQ_CONFIG, SB                             ; 0734
                     59              57   D1 0003C 2$:   CMPL     SB, R9                                        ; 0738
                                     60   13 0003F       BEQL     11$                                           ;
                     54         54   A7   D0 00041       MOVL     84(SB), DDB                                   ; 0742
                                     55   13 00045 3$:   BEQL     10$                                           ; 0744
                     06         0A   A4   91 00047       CMPB     10(DDB), #6                                   ; 0747
                                     04   13 0004B       BEQL     4$                                            ;
                                   FEFF 0004D            BUGW                                                   ; 0748
                                  0000* 0004F            .WORD    <BUG$_NOTDDBDDB!4>                            ;
                     55         04   A4   D0 00051 4$:   MOVL     4(DDB), UCB                                   ; 0749
                                     40   13 00055 5$:   BEQL     9$                                            ; 0751
                     10         0A   A5   91 00057       CMPB     10(UCB), #16                                  ; 0753
                                     04   13 0005B       BEQL     6$                                            ;
                                   FEFF 0005D            BUGW                                                   ; 0754
                                  0000* 0005F            .WORD    <BUG$_NOTUCBUCB!4>                            ;
          2B         39 A5           06   E1 00061 6$:   BBC      #6, 57(UCB), 8$                               ; 0756
          26         3A A5           03   E1 00066       BBC      #3, 58(UCB), 8$                               ; 0757
                                3A   A5   95 0006B       TSTB     58(UCB)                                       ; 0758
                                     21   19 0006E       BLSS     8$                                            ;
          1C         3A A5           05   E0 00070       BBS      #5, 58(UCB), 8$                               ; 0759
                     56         34   A5   D0 00075       MOVL     52(UCB), VCB                                  ; 0762
                     11         0A   A6   91 00079       CMPB     10(VCB), #17                                  ; 0763
                                     04   13 0007D       BEQL     7$                                            ;
                                   FEFF 0007F            BUGW                                                   ; 0764
                                  0000* 00081            .WORD    <BUG$_NOTVCBUCB!4>                            ;
                     50         04   AC   D0 00083 7$:   MOVL     NAME, R0                                      ; 0765
  0C      20         04 B0           60   2D 00087       CMPC5    (R0), @4(R0), #32, #12, 20(VCB)               ; 0766
                                14   A6      0008D                                                              ;
                                     10   13 0008F       BEQL     11$                                           ;
```

```
                                                                F 16
SRCVOL                                                          16-Sep-1984 01:33:17    VAX-11 Bliss-32 V4.0-742                Page   8
V04-000                                                         14-Sep-1984 12:45:35    DISK$VMSMASTER:[MOUNT.SRC]SRCVOL.B32;1   (2)

                                       55       30   A5 D0 00091 8$:      MOVL    48(UCB), UCB                                ; 0769
                                                     BE 11 00095          BRB     5$                                          ; 0751
                                       54            64 D0 00097 9$:      MOVL    (DDB), DDB                                  ; 0771
                                                     A9 11 0009A          BRB     3$                                          ; 0744
                                       57            67 D0 0009C 10$:     MOVL    (SB), SB                                    ; 0774
                                                     9B 11 0009F          BRB     2$                                          ; 0738
                                                     52 D4 000A1 11$:     CLRL    STATUS                                      ; 0784
                                                     55 D5 000A3          TSTL    UCB                                         ; 0785
                                                     5D 13 000A5          BEQL    15$                                         ;
           50       01   AA            01            03 EF 000A7          EXTZV   #3, #1, MOUNT_OPTIONS+1, R0                 ; 0788
           50       3B   A5            01            00 ED 000AD          CMPZV   #0, #1, 59(UCB), R0                         ;
                                                     09 13 000B3          BEQL    12$                                         ;
                                       52 007280AC   8F D0 000B5          MOVL    #7504044, STATUS                            ; 0789
                                                     46 11 000BC          BRB     15$                                         ;
                         09            6A            06 E0 000BE 12$:     BBS     #6, MOUNT_OPTIONS, 13$                      ; 0794
                                       52 007280B4   8F D0 000C2          MOVL    #7504052, STATUS                            ; 0796
                                                     06 11 000C9          BRB     14$                                         ;
                                       52            01 D0 000CB 13$:     MOVL    #1, STATUS                                  ; 0799
                                                4C   A6 B6 000CE          INCW    76(VCB)                                     ; 0800
                         50     0000G  CF            05 78 000D1 14$:     ASHL    #5, DEVICE_INDEX, R0                        ; 0808
                                       51     0000GCF40 9E 000D7          MOVAB   NAME_BUFFER[R0], R1                         ; 0807
                                       54            01 CE 000DD          MNEGL   #1, R4                                      ; 0811
                                       50            20 D0 000E0          MOVL    #32, R0                                     ;
                                          000000000G 00 16 000E3          JSB     IOC$CVT_DEVNAM                              ;
                         50     0000G  CF            01 78 000E9          ASHL    #1, DEVICE_INDEX, R0                        ;
                                0000GCF40            51 D0 000EF          MOVL    R1, PHYS_NAME[R0]                           ;
                         51     0000G  CF            05 78 000F5          ASHL    #5, DEVICE_INDEX, R1                        ; 0813
                                0000GCF40     0000GCF41 9E 000FB          MOVAB   NAME_BUFFER[R1], PHYS_NAME+4[R0]            ; 0812
                                0000G  CF            00 FB 00104 15$:     CALLS   #0, UNLOCK_IODB                             ; 0817
                            007280B4   8F            52 D1 00109          CMPL    STATUS, #7504052                            ; 0824
                                                     10 12 00110          BNEQ    16$                                         ;
                         0B       07   AA            06 E1 00112          BBC     #6, MOUNT_OPTIONS+7, 16$                    ; 0825
                                              0000G  CF DD 00117          PUSHL   MOUNT_ITMLST                                ; 0828
                                0000G  CF            01 FB 0011B          CALLS   #1, MOUNT_CLUSTER                           ;
                                                     2D 11 00120          BRB     20$                                         ; 0829
                                       09            52 E8 00122 16$:     BLBS    STATUS, 17$                                 ; 0832
                                                     52 DD 00125          PUSHL   STATUS                                      ;
                            000000000G 00            01 FB 00127          CALLS   #1, LIB$STOP                                ;
                         06       38   A5            05 E0 0012E 17$:     BBS     #5, 56(UCB), 18$                            ; 0842
                                       01       0E   A6 B1 00133          CMPW    14(VCB), #1                                 ; 0843
                                                     0E 13 00137          BEQL    19$                                         ;
                                                     01 DD 00139 18$:     PUSHL   #1                                          ; 0846
                                                     68 DD 0013B          PUSHL   MTL_ENTRY                                   ;
                                0000G  CF            02 FB 0013D          CALLS   #2, DEALLOCATE_MEM                          ;
                                       68            6B D0 00142          MOVL    SMTL_ENTRY, MTL_ENTRY                       ; 0847
                                                     6B D4 00145          CLRL    SMTL_ENTRY                                  ; 0848
                                       7E            55 7D 00147 19$:     MOVQ    UCB, -(SP)                                  ; 0851
                                0000G  CF            02 FB 0014A          CALLS   #2, ENTER_LOGNAME                           ;
                                       50            01 D0 0014F 20$:     MOVL    #1, R0                                      ; 0853
                                                        04 00152          RET                                                 ; 0855
                                                      0000 00153 21$:     .WORD   Save nothing                                ; 0666
                                                     7E D4 00155          CLRL    -(SP)                                       ;
                                                     5E DD 00157          PUSHL   SP                                          ;
                                       7E       04   AC 7D 00159          MOVQ    4(AP), -(SP)                                ;
                                0000V  CF            03 FB 0015D          CALLS   #3, SEARCH_HANDLER                          ;
                                                        04 00162          RET                                                 ;
```

```
; Routine Size:  355 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE SEARCH_HANDLER (SIGNAL, MECHANISM) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is the condition handler for the volume search
1 !       routine. It undoes any damage done so far and returns the error
1 !       status to the user mode caller.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       KERNEL_HANDLER (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of signal vector
1 !       ARG2: address of mechanism vector
1 !
1 ! IMPLICIT INPUTS:
1 !       global pointers to blocks allocated
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       stack unwound, allocations undone
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       SIGNAL          : REF BBLOCK,   ! signal vector
2       MECHANISM       : REF BBLOCK;   ! mechanism vector
2
2 EXTERNAL
2       MOUNT_OPTIONS   : BITVECTOR,    ! command parser options
2       MTL_ENTRY       : REF BBLOCK,   ! address of mount list entry
2       SMTL_ENTRY      : REF BBLOCK;   ! address of volume set mount list entry
2
2 EXTERNAL ROUTINE
2       UNLOCK_IODB,                    ! unlock I/O database mutex
2       DEALLOCATE_MEM;                 ! deallocate system dynamic memory
2
2
2 IF .SIGNAL[CHF$L_SIG_NAME] NEQ SS$_UNWIND
2 THEN
3     BEGIN
3
3     IF .SIGNAL[CHF$L_SIG_ARGS] NEQ 3
3     THEN BUG_CHECK (UNXSIGNAL, FATAL, 'Unexpected signal in MOUNT');
```

```
;  384         0913  3
;  385         0914  3
;  386         0915  3 ! Deallocate whatever control blocks exist to wherever they came from.
;  387         0916  3 !
;  388         0917  3
;  389         0918  3    IF .MTL_ENTRY NEQ 0
;  390         0919  3    THEN DEALLOCATE_MEM (.MTL_ENTRY, 1);
;  391         0920  3
;  392         0921  3    IF .SMTL_ENTRY NEQ 0
;  393         0922  3    THEN DEALLOCATE_MEM (.SMTL_ENTRY, 1);
;  394         0923  3
;  395         0924  3    MTL_ENTRY = SMTL_ENTRY = 0;
;  396         0925  3
;  397         0926  3 ! Return the condition code in R0.
;  398         0927  3 !
;  399         0928  3
;  400         0929  3    MECHANISM[CHF$L_MCH_SAVR0] = .SIGNAL[CHF$L_SIG_NAME];
;  401         0930  3    $UNWIND ();
;  402         0931  3
;  403         0932  2    END;
;  404         0933  1 END;                                  ! end of routine KERNEL_HANDLER



                                                                          .EXTRN  BUGS_UNXSIGNAL, SYS$UNWIND

                                                 0000 00000 SEARCH_HANDLER:
                                                                          .WORD   Save nothing                              ; 0856
                         50          04      AC   D0 00002                MOVL    SIGNAL, R0                                ; 0907
               00000920  8F          04      A0   D1 00006                CMPL    4(R0), #2336                              ;
                                             43   13 0000E                BEQL    4$                                        ;
                         03                  60   D1 00010                CMPL    (R0), #3                                  ; 0911
                                             04   13 00013                BEQL    1$                                        ;
                                              FEFF 00015                  BUGW                                              ; 0912
                                             0000* 00017                  .WORD   <BUG$_UNXSIGNAL!4>                        ;
                         50       0000G      CF   D0 00019 1$:            MOVL    MTL_ENTRY, R0                             ; 0918
                                             09   13 0001E                BEQL    2$                                        ;
                                             01   DD 00020                PUSHL   #1                                        ; 0919
                                             50   DD 00022                PUSHL   R0                                        ;
                   0000G CF                  02   FB 00024                CALLS   #2, DEALLOCATE_MEM                        ;
                         50       0000G      CF   D0 00029 2$:            MOVL    SMTL_ENTRY, R0                            ; 0921
                                             09   13 0002E                BEQL    3$                                        ;
                                             01   DD 00030                PUSHL   #1                                        ; 0922
                                             50   DD 00032                PUSHL   R0                                        ;
                   0000G CF                  02   FB 00034                CALLS   #2, DEALLOCATE_MEM                        ;
                                  0000G      CF   D4 00039 3$:            CLRL    SMTL_ENTRY                                ; 0924
                                  0000G      CF   D4 0003D                CLRL    MTL_ENTRY                                 ;
                         50          04      AC   7D 00041                MOVQ    SIGNAL, R0                                ; 0929
                      0C A1          04      A0   D0 00045                MOVL    4(R0), 12(R1)                             ;
                                             7E   7C 0004A                CLRQ    -(SP)                                     ; 0930
               00000000G 00                  02   FB 0004C                CALLS   #2, SYS$UNWIND                            ;
                                                  04 00053 4$:            RET                                               ; 0933

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 0163


;  405         0934  1
;  406           0935  1 END
;  407           0936  0 ELUDOM



                                                                                    .EXTRN  LIB$STOP

;                                              PSECT SUMMARY
;
;        Name                               Bytes                            Attributes
;
;   $CODE$                                       439  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        39          0      1000         00:01.9




;                                               COMMAND QUALIFIERS

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

; Size:         439 code + 0 data bytes
; Run Time:        00:16.4
; Elapsed Time:    00:43.9
; Lines/CPU Min:     3422
; Lexemes/CPU-Min: 29093
; Memory Used:  157 pages
; Compilation Complete
```

0246 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RDHOME
LIS

MWTUR2
LIS

RUJMAN
LIS

MWTUR1
LIS

SRCVOL
LIS

STACP
LIS

MRDBLK
LIS

REBUILD
LIS