```
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
```

```
_$

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**MOUPAR

G 11

```
MM      MM    OOOOOO    UU      UU  PPPPPPPP      AAAAAA    RRRRRRRR
MM      MM    OOOOOO    UU      UU  PPPPPPPP      AAAAAA    RRRRRRRR
MMMM  MMMM  OO      OO  UU      UU  PP      PP  AA      AA  RR      RR
MMMM  MMMM  OO      OO  UU      UU  PP      PP  AA      AA  RR      RR
MM  MM  MM  OO      OO  UU      UU  PP      PP  AA      AA  RR      RR
MM  MM  MM  OO      OO  UU      UU  PP      PP  AA      AA  RR      RR
MM      MM  OO      OO  UU      UU  PPPPPPPP    AA      AA  RRRRRRRR
MM      MM  OO      OO  UU      UU  PPPPPPPP    AA      AA  RRRRRRRR
MM      MM  OO      OO  UU      UU  PP          AAAAAAAAAA  RR  RR
MM      MM  OO      OO  UU      UU  PP          AAAAAAAAAA  RR  RR
MM      MM  OO      OO  UU      UU  PP          AA      AA  RR    RR        ....
MM      MM  OO      OO  UU      UU  PP          AA      AA  RR    RR        ....
MM      MM    OOOOOO    UUUUUUUUUU  PP          AA      AA  RR      RR      ....
MM      MM    OOOOOO    UUUUUUUUUU  PP          AA      AA  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
```

MOU
V04

```
MODULE MOUPAR (
                LANGUAGE (BLISS32),
                IDENT = 'V04-000'
                ) =
BEGIN

!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************

!++
!
! FACILITY:  MOUNT Utility Structure Level 1
!
! ABSTRACT:
!
!       This module contains the data base and utilities used to acquire the
!       MOUNT command line from the CLI parser.
!
! ENVIRONMENT:
!
!       STARLET operating system, including privileged system services
!       and internal exec routines.
!
!--
!
!
! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  29-Sep-1977  16:58
!
! MODIFIED BY:
!
!       V03-013 HH0041          Hai Huang                       24-Jul-1984
!               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
!
!       V03-012 DAS0001         David Solomon                   10-Jul-1984
!               Add support for MNT$V_NOREBUILD (/NOREBUILD qualifier).
!
```

```
1 !     V03-011 HH0009          Hai Huang                       27-Mar-1984
1 !             Add security auditing support, i.e. move MOUNT_FLAGS to
1 !             $GLOBALS psect.
1 !
1 !     V03-010 HH0004          Hai Huang                       09-Mar-1984
1 !             Add cluster-wide mount support (/CLUSTER qualifier).
1 !
1 !     V03-009 DAS0001         David Solomon                   29-Nov-1983
1 !             Add support for specifying maximum journal record size
1 !             with a new keyword, /JOURNAL=(RECORD_SIZE=n).
1 !
1 !     V03-008 MMD0189         Meg Dumont,     7-Jul-1983  10:00
1 !             Make the default for AVL/AVR the same from the DCL call
1 !             and from the system service call.
1 !
1 !     V03-007 MMD0114         Meg Dumont,     29-Mar-1983  0:38
1 !             Add support for new qualifiers for AVL, AVR and VMS protection
1 !
1 !     V03-006 STJ48132        Steven T. Jeffreys,     09-Oct-1982
1 !             Initialize MOUNT_FLAGS before using it.
1 !
1 !     V03-005 STJ0317 Steven T. Jeffreys,     15-Aug-1982
1 !             Added support for journalling qualifers.
1 !
1 !     V03-004 STJ0304 Steven T. Jeffreys,     18-May-1982
1 !             Added support for /NOUNLOAD qualifier.
1 !
1 !     V03-003 STJ0260 Steven T. Jeffreys,     14-Apr-1982
1 !             Removed the device list inspection code added in update V02-016
1 !
1 !     V03-002 STJ0250         Steven T. Jeffreys,     01-Apr-1982
1 !             - Check for invalid device name status from $GETDEV.
1 !
1 !     V03-001 STJ0238         Steven T. Jeffreys,     17-Mar-1982
1 !             - Check for CMKRNL privilege if /PROCESSOR=<filename> specified.
1 !
1 !     V02-018 STJ0230         Steven T. Jeffreys,     01-Mar-1982
1 !             - If user is trying to access a non-existant drive,
1 !               return a status to indicate such.  This is an improvement
1 !               to update #16 to this module.
1 !             - Close buffer probe protection hole.
1 !
1 !     V02-017 STJ0225         Steven T. Jeffreys,     17-Feb-1982
1 !             Fix incorrect probe of the users input parameters.
1 !
1 !     V02-016 STJ0217         Steven t. Jeffreys,     15-Feb-1982
1 !             Sweep through the device list to make sure that each
1 !             specified device exists and is file-oriented.
1 !
1 !     V02-015 STJ0200         Steven T. Jeffreys,     11-Feb-1982
1 !             Add support for the /COMMENT qualifier.
1 !
1 !     V02-014 STJ0185         Steven T. Jeffreys,     25-Jan-1982
1 !             - Removed privilege restirctions for /MOUNT_VERIFICATION.
1 !             - Zero GLOBAL and OWN storage to guaranty restartability.
1 !
1 !     V02-013 STJ0173         Steven T. Jeffreys,     13-Jan-1982
```

```
;  115          0115   1 !                  Modified to use the new $MOUNT interface.
;  116          0116   1 !
;  117          0117   1 !          V02-012 STJ0157          Steven T. Jeffreys,     04-Jan-1982
;  118          0118   1 !                  Added support for the /OVERRIDE=LOCK, /NOCACHE, and /MOUNTVER
;  119          0119   1 !                  qualifiers.  Changed reference of OPT_SIGNAL to OPT_MESSAGE.
;  120          0120   1 !
;  121          0121   1 !          V02-011 STJ0152          Steven T. Jeffreys,     02-Jan-1981
;  122          0122   1 !                  Extensive rewrite to support the $MOUNT system service.
;  123          0123   1 !
;  124          0124   1 !          V02-010 STJ0148          Steven T. Jeffreys,     16-Dec-1981
;  125          0125   1 !                  Add a "." to the front of the VOLSET parameter reference.
;  126          0126   1 !
;  127          0127   1 !          V02-009 BLS0080          Benn Schreiber          14-Sep-1981
;  128          0128   1 !                  Reference 'BIND' rather than 'VOLSET' for /BIND qualifier
;  129          0129   1 !
;  130          0130   1 !          V02-008 STJ0109          Steven T. Jeffreys,     28-Aug-1981
;  131          0131   1 !                  Liberal rewrite as part of $MOUNT support.
;  132          0132   1 !
;  133          0133   1 !          V02-007 STJ0036          Steven T. Jeffreys,     11-May-1981
;  134          0134   1 !                  Added support for /ASSIST qualifier.
;  135          0135   1 !
;  136          0136   1 !          V02-006 ACG0167          Andrew C. Goldstein,    18-Apr-1980  13:38
;  137          0137   1 !                  Previous revision history moved to MOUNT.REV
;  138          0138   1 !**
;  139          0139   1
;  140          0140   1
;  141          0141   1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  142          0142   1 REQUIRE 'SRC$:MOUDEF.B32';
;  143          0674   1 LIBRARY 'SYS$LIBRARY:CLIMAC.L32';
;  144          0675   1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
```

```
!+
!
! Impure data area. This area contains the MOUNT parameters extracted from
! the command line by the associated parsing routines.
!
!-


GLOBAL
        GLOBAL_START    : VECTOR [0],   ! mark start of global storage
        MOUNT_OPTIONS   : BITVECTOR [64], ! option flags
        MOUNT_FLAGS     : BBLOCK [4],   ! Mount option flags
        PROTECTION,                     ! value of /PROTECTION switch
        OWNER_UIC,                      ! value of /OWNER_UIC switch
        USER_UIC,                       ! value of /USER_UIC switch
        EXTENSION,                      ! value of /EXTENSION switch
        WINDOW,                         ! value of /WINDOW switch
        ACCESSED,                       ! value of /ACCESSED switch
        BLOCKSIZE,                      ! value of /BLOCK switch
        RECORDSZ,                       ! value of /RECORD switch
        FID_CACHE,                      ! space to allocate for file ID cache
        EXT_CACHE,                      ! space to allocate for extent cache
        QUO_CACHE,                      ! space to allocate for quota cache
        EXT_LIMIT,                      ! limit of disk free space to cache
        DEVICE_COUNT,                   ! number of devices specified
        LABEL_COUNT,                    ! number of volume labels specified
        VID_STRING      : VECTOR [2],   ! descriptor of VISUAL_ID string
        COMMENT_STRING  : VECTOR [2],   ! descriptor of COMMENT string
        ACP_STRING      : VECTOR [2],   ! descriptor of ACP device or name string
        LOG_NAME        : VECTOR [2],   ! descriptor of logical name string
        STRUCT_NAME     : VECTOR [2],   ! descriptor of volume set name
                                        ! (value of /BIND qualifier)
        DRIVE_COUNT     : VECTOR [DEVMAX], ! value of /DRIVES switch
        DEVICE_STRING   : VECTOR [DEVMAX*2], ! descriptors of device name strings
        LABEL_STRING    : VECTOR [LABMAX*2], ! descriptors of volume label strings
        JRNL_SIZE,                      ! Recovery Unit Journal (RUJ)  initial size
        JRNL_EXTEND,                    ! RUJ default file extension size
        JRNL_QUOTA,                     ! RUJ byte quota per recovery unit
        JRNL_RECORD_SIZE,                       ! RUJ maximum record size
        GLOBAL_END      : VECTOR [0];   ! Mark end of global storage

LITERAL
        COMMENT_SIZE    = 80,                   ! maximum length of a user comment
        BUFFER_SIZE     = 63,                   ! maximum length of a string buffer
        ITEM_SIZE       = 12;                   ! length of each item descriptor

OWN
        OWN_START       : VECTOR [0],                   ! Mark start of OWN storage
        DEVICE_BUFFER   : BBLOCK [BUFFER_SIZE*DEVMAX],  ! Buffer for all device names
        LABEL_BUFFER    : BBLOCK [BUFFER_SIZE*DEVMAX],  ! Buffer for all volume labels
        LOG_NAME_BUFFER : BBLOCK [BUFFER_SIZE],         ! Buffer for the volume logical name
        ACP_NAME_BUFFER : BBLOCK [BUFFER_SIZE],         ! Buffer for the user-specified ACP name
        VOLSET_BUFFER   : BBLOCK [BUFFER_SIZE],         ! Buffer for the volume set name
        COMMENT_BUFFER  : BBLOCK [COMMENT_SIZE],        ! Buffer for the user comment
        TPARSE_BLOCK    : BBLOCK [TPA$K_LENGTH0],       ! TPARSE control block
        OWN_END         : VECTOR [0];                   ! Mark end of OWN storage
```

```
;  203          0733  1 LITERAL
;  204          0734  1         !
;  205          0735  1         ! Define the lengths of the GLOBAL and OWN storage areas.
;  206          0736  1         !
;  207          0737  1         GLOBAL_LENGTH   = GLOBAL_END - GLOBAL_START,
;  208          0738  1         OWN_LENGTH      = OWN_END - OWN_START;
;  209          0739  1
;  210          0740  1 FORWARD ROUTINE
;  211          0741  1         COPY_ITEM;                              ! Copy an item to iternal storage
;  212          0742  1
;  213          0743  1 EXTERNAL
;  214          0744  1         DEVCHAR_DESC    : VECTOR,
;  215          0745  1         DEVICE_CHAR     : BBLOCK,
;  216          0746  1         NEWLINE         : BBLOCK,
;  217          0747  1         CTL$GL_PHD      : ADDRESSING_MODE (GENERAL);
;  218          0748  1
;  219          0749  1 FORWARD
;  220          0750  1         ACP_KTB         : VECTOR [0],           ! TPARSE table label.
;  221          0751  1         ACP_STB         : VECTOR [0];           ! TPARSE table label.
;  222          0752  1
;  223          0753  1 MACRO
;  224          0754  1         !
;  225          0755  1         ! Symbolic offsets into an item descriptor
;  226          0756  1         !
;  227          0757  1         LENGTH          = 0, 0, 16, 0%,         ! length of item (in bytes)
;  228          0758  1         CODE            = 2, 0, 16, 0%,         ! item code
;  229          0759  1         ADDRESS         = 4, 0, 32, 0%,         ! item address
;  230          0760  1         UNUSED          = 8, 0, 32, 0%;         ! reserved for future use
;  231          0761  1
```

```
1 GLOBAL ROUTINE CHECK_PARAMS (ITEM_LIST) =
1 !++
1 ! Functional description:
1 !
1 !       This routine is responible for validating the parameters passed
1 !       by the $MOUNT system service call, and copyin them to internal
1 !       storage areas.  The validation consits of:
1 !
1 !               - Probing each parameter for read access.
1 !               - Copying each parameter to an internal area.
1 !               - Sanity checks on the parameter values.
1 !
1 ! Input:
1 !
1 !       None.
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       Mount's internal data storage is initialized.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       SS$_NORMAL       : Normal successful completion
1 !       SS$_ACCVIO       : A parameter was not readable
1 !       SS$_BADPARAM     : Bad parameter value
1 !       MOUN$_CONFQUAL   : The command has conflicting qualifiers
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       AP,                                     ! Arguement pointer
2       PROBER;                                 ! Probe for read access
2
2 EXTERNAL ROUTINE
2       LIB$TPARSE       : ADDRESSING_MODE (GENERAL);    ! Finite state parser
2
2 LOCAL
2       PHD              : REF BBLOCK,          ! Process PHD
2       ITEM             : REF BBLOCK,          ! Pointer to item list
2       PTR              : LONG,                ! Temporary storage
2       STR_DESCRIPTOR   : REF BBLOCK,          ! Temporary descriptor
2       JRNL_ITEM_SEEN   : LONG,                ! A boolean
2       DENSITY          : LONG,                ! Tape volume density
```

```
2            STATUS           : LONG;                    ! Store routine status code
2
2    !
2    ! Zero the GLOBAL and OWN storage ares to guaranty the
2    ! restartablity of the code.
2    !
2    CH$FILL (0, GLOBAL_LENGTH, GLOBAL_START);
2    CH$FILL (0, OWN_LENGTH, OWN_START);
2    MOUNT_FLAGS [0, 0, 32, 0] = 0;
2    !
2    ! Initialize the TPARSE control block.
2    !
2    TPARSE_BLOCK [TPA$L_COUNT]       = TPA$K_COUNT0;
2    TPARSE_BLOCK [TPA$L_OPTIONS]     = (TPA$M_BLANKS OR TPA$M_ABBREV);
2
2
2    !*********************************************************************!
2    !                                                                     !
2    ! Interpret each item in the item list.  After doing some             !
2    ! sanity checks for each parameter, copy it to the internal           !
2    ! storage area.  This is done to protect the data from corruption     !
2    ! by malicious users.  The item list is terminated by an item         !
2    ! descriptor with an item code of 0.  Items with a zero length are    !
2    ! rejected.                                                           !
2    !                                                                     !
2    ! Since a given item may appear in the list more than once, we must be !
2    ! careful to reset any item context for each version of the item.  A  !
2    ! good example of where this is necessary is for the MNT$_DENSITY item. !
2    !                                                                     !
2    !*********************************************************************!
2
2    MOUNT_OPTIONS = MOUNT_OPTIONS+4 = 0;
2    STATUS = 1;
2    JRNL_ITEM_SEEN = 0;
2    DEVICE_COUNT = 0;
2    LABEL_COUNT = 0;
2    ITEM = .ITEM_LIST;
2    IF NOT PROBER (%REF (0), %REF (4), ITEM [0, 0, 32, 0])  ! The first 4 bytes must be readable
2    THEN
2        RETURN (SS$_ACCVIO);
2
2    WHILE .STATUS AND (.ITEM [CODE] NEQ 0) DO
3        BEGIN
3        !
3        ! PROBE the next 12 bytes, which includes the remainder of this item
3        ! descriptor and the first longword of the next one.  If duplicate
3        ! items are declared, then the value last encountered will be used.
3        ! The exception to this are device names and volume labels.
3        !
3        ! As new items are defined, a corresponding entry must be made in
3        ! the select table defined below.
3        !
3        IF NOT PROBER (%REF (0), %REF (ITEM_SIZE), .ITEM)
3        THEN
3            STATUS = SS$_ACCVIO
3        ELSE
4            BEGIN
```

```
5           STATUS = (SELECTONE .ITEM [CODE] OF
5                   SET
5                   !
5                   ! The following items are character strings.
5                   !
6                   [MNT$_DEVNAM]       :(IF .DEVICE_COUNT LSS DEVMAX
6                                           THEN
7                                               BEGIN
7                                               MOUNT_OPTIONS [OPT_DEVICE] = 1;
7                                               DEVICE_COUNT = .DEVICE_COUNT + 1;
7                                               DEVICE_STRING [(.DEVICE_COUNT-1)*2] = MIN (.ITEM [LENGTH], BUF
7                                               DEVICE_STRING [((.DEVICE_COUNT-1)*2)+1] = DEVICE_BUFFER+(BUFFE
7                                               COPY_ITEM (.ITEM, BUFFER_SIZE, .DEVICE_STRING [((.DEVICE_COUNT
7                                               END
6                                           ELSE
5                                               MOUN$_MAXDEV);
5
6                   [MNT$_VOLNAM]       :(IF .LABEL_COUNT LSS DEVMAX
6                                           THEN
7                                               BEGIN
7                                               MOUNT_OPTIONS [OPT_LABEL] = 1;
'                                               LABEL_COUNT = .LABEL_COUNT + 1;
7                                               LABEL_STRING [(.LABEL_COUNT-1)*2] = MIN (.ITEM [LENGTH], BUFFE
7                                               LABEL_STRING [((.LABEL_COUNT-1)*2)+1] = LABEL_BUFFER + (BUFFER
7                                               COPY_ITEM (.ITEM, BUFFER_SIZE, .LABEL_STRING [((.LABEL_COUNT-1
7                                               END
6                                           ELSE
5                                               MOUN$_MAXLAB);
5
6                   [MNT$_LOGNAM]       : BEGIN
6                                         MOUNT_OPTIONS [OPT_LOG_NAME] = 1;             ! Mark the logical n
6                                         LOG_NAME [0] = MIN (.ITEM [LENGTH], BUFFER_SIZE);
6                                         LOG_NAME [1] = LOG_NAME_BUFFER;
6                                         STATUS = COPY_ITEM (.ITEM, BUFFER_SIZE, LOG_NAME_BUFFER);
6                                         !
6                                         ! Scan for a trailing or imbedded colon.  If found,
6                                         ! use the string preceding the colon.
6                                         !
6                                         PTR = CH$FIND_CH (.LOG_NAME [0], .LOG_NAME [1], ':');
6                                         IF NOT CH$FAIL (.PTR)
6                                         THEN
6                                             LOG_NAME [0] = .PTR - .LOG_NAME [1];
6                                         .STATUS
5                                         END;
5
6                   [MNT$_PROCESSOR]    : BEGIN
6                                         ACP_STRING [0] = MIN (.ITEM [LENGTH], BUFFER_SIZE);
6                                         ACP_STRING [1] = ACP_NAME_BUFFER;
6                                         COPY_ITEM (.ITEM, BUFFER_SIZE, ACP_NAME_BUFFER)
5                                         END;
5
6                   [MNT$_VOLSET]       : BEGIN
6                                         STRUCT_NAME [0] = MIN (.ITEM [LENGTH], BUFFER_SIZE);
6                                         STRUCT_NAME [1] = VOLSET_BUFFER;
6                                         MOUNT_OPTIONS [OPT_BIND] = 1;
6                                         COPY_ITEM (.ITEM, BUFFER_SIZE, VOLSET_BUFFER)
5                                         END;
```

```
        !
        ! The following items are integer values,
        ! and are all longwords (4 bytes in length).
        !

        [MNT$_FLAGS]        : COPY_ITEM (.ITEM, 4, MOUNT_FLAGS);

        [MNT$_ACCESSED]     : BEGIN
                              MOUNT_OPTIONS [OPT_ACCESSED] = 1;
                              COPY_ITEM (.ITEM, 4, ACCESSED)
                              END;

        [MNT$_BLOCKSIZE]    : BEGIN
                              MOUNT_OPTIONS [OPT_BLOCKSIZE] = 1;
                              IF (STATUS = COPY_ITEM (.ITEM, 4, BLOCKSIZE))
                              THEN
                                  IF .BLOCKSIZE GTR 65534
                                  THEN
                                      STATUS = MOUN$_SZTOOBIG;
                              .STATUS
                              END;

        [MNT$_DENSITY]      : BEGIN
                              MOUNT_OPTIONS [OPT_DENSITY] = 1;
                              MOUNT_OPTIONS [OPT_DENS_800] = 0;
                              MOUNT_OPTIONS [OPT_DENS_1600] = 0;
                              IF (STATUS = COPY_ITEM (.ITEM, 4, DENSITY))
                              THEN
                                  SELECTONE .DENSITY OF
                                      SET
                                      [800]          : MOUNT_OPTIONS [OPT_DENS_800] = 1;
                                      [1600]         : MOUNT_OPTIONS [OPT_DENS_1600] = 1;
                                      [6250]         : 1;
                                      [OTHERWISE]    : STATUS = MOUN$_BADDENS;
                                      TES;
                              .STATUS
                              END;

        [MNT$_EXTENT]       : BEGIN
                              STATUS = COPY_ITEM (.ITEM, 4, EXT_CACHE);
                              MOUNT_OPTIONS [OPT_NOEXT_C] = 0;
                              MOUNT_OPTIONS [OPT_CACHE] = 0;
                              IF .EXT_CACHE LEQ 0
                              THEN
                                  MOUNT_OPTIONS [OPT_NOEXT_C] = 1
                              ELSE
                                  MOUNT_OPTIONS [OPT_CACHE] = 1;
                              .STATUS
                              END;

        [MNT$_FILEID]       : BEGIN
                              STATUS = COPY_ITEM (.ITEM, 4, FID_CACHE);
                              MOUNT_OPTIONS [OPT_NOFID_C] = 0;
                              MOUNT_OPTIONS [OPT_CACHE] = 0;
                              IF .FID_CACHE LEQ 1
                              THEN
                                  MOUNT_OPTIONS [OPT_NOFID_C] = 1
```

```
:   461         0990  6                                       ELSE
:   462         0991  6                                           MOUNT_OPTIONS [OPT_CACHE] = 1;
:   463         0992  6                                       .STATUS
:   464         0993  5                                       END;
:   465         0994  5
:   466         0995  5             [MNT$_LIMIT]          : COPY_ITEM (.ITEM, 4, EXT_LIMIT);
:   467         0996  5
:   468         0997  6             [MNT$_OWNER]          : BEGIN
:   469         0998  6                                       MOUNT_OPTIONS [OPT_OWNER_UIC] = 1;
:   470         0999  6                                       COPY_ITEM (.ITEM, 4, OWNER_UIC)
:   471         1000  5                                       END;
:   472         1001  5
:   473         1002  6             [MNT$_VPROT]          : BEGIN
:   474         1003  6                                       MOUNT_OPTIONS [OPT_PROTECTION] = 1;
:   475         1004  6                                       COPY_ITEM (.ITEM, 4, PROTECTION)
:   476         1005  5                                       END;
:   477         1006  5
:   478         1007  6             [MNT$_QUOTA]          : BEGIN
:   479         1008  6                                       STATUS = COPY_ITEM (.ITEM, 4, QUO_CACHE);
:   480         1009  6                                       MOUNT_OPTIONS [OPT_NOQUO_C] = 0;
:   481         1010  6                                       MOUNT_OPTIONS [OPT_CACHE] = 0;
:   482         1011  6                                       IF .QUO_CACHE LEQ 0
:   483         1012  6                                       THEN
:   484         1013  6                                           MOUNT_OPTIONS [OPT_NOQUO_C] = 1
:   485         1014  6                                       ELSE
:   486         1015  6                                           MOUNT_OPTIONS [OPT_CACHE] =1;
:   487         1016  6                                       .STATUS
:   488         1017  5                                       END;
:   489         1018  5
:   490         1019  6             [MNT$_RECORDSIZ]      : BEGIN
:   491         1020  6                                       MOUNT_OPTIONS [OPT_RECORDSZ] = 1;
:   492         1021  7                                       IF (STATUS = COPY_ITEM (.ITEM, 4, RECORDSZ))
:   493         1022  6                                       THEN
:   494         1023  6                                           IF .RECORDSZ GTR 65534
:   495         1024  6                                           THEN
:   496         1025  6                                               STATUS = MOUN$_SZTOOBIG;
:   497         1026  6                                       .STATUS
:   498         1027  5                                       END;
:   499         1028  5
:   500         1029  6             [MNT$_WINDOW]         : BEGIN
:   501         1030  6                                       MOUNT_OPTIONS [OPT_WINDOW] = 1;
:   502         1031  6                                       COPY_ITEM (.ITEM, 4, WINDOW)
:   503         1032  5                                       END;
:   504         1033  5
:   505         1034  6             [MNT$_EXTENSION]      : BEGIN
:   506         1035  6                                       MOUNT_OPTIONS [OPT_EXTENSION] = 1;
:   507         1036  6                                       COPY_ITEM (.ITEM, 4, EXTENSION)
:   508         1037  5                                       END;
:   509         1038  5
:   510         1039  6             [MNT$_COMMENT]        : BEGIN
:   511         1040  6                                       !
:   512         1041  6                                       ! Append a newline (<cr><lf>) to the front of the comment string.
:   513         1042  6                                       !
:   514         1043  6                                       MOUNT_OPTIONS [OPT_COMMENT] = 1;
:   515         1044  6                                       COMMENT_STRING [0] = MIN (.ITEM [LENGTH], COMMENT_SIZE) + .NEWLINE
:   516         1045  6                                       COMMENT_STRING [1] = COMMENT_BUFFER;
:   517         1046  6                                       CH$MOVE (.NEWLINE [DSC$W_LENGTH], .NEWLINE [DSC$A_POINTER], COMMEN
6                                               COPY_ITEM (.ITEM,
6                                                       COMMENT_SIZE-.NEWLINE [DSC$W_LENGTH],
6                                                       COMMENT_BUFFER+.NEWLINE [DSC$W_LENGTH]
6                                                       )
5                                                END;
5
6                               [MNT$_JRNLSIZE]   : BEGIN
6                                                   JRNL_ITEM_SEEN = 1;
6                                                   COPY_ITEM (.ITEM, 4, JRNL_SIZE)
5                                                   END;
5
6                               [MNT$_JRNLEXTEND] : BEGIN
6                                                   JRNL_ITEM_SEEN = 1;
6                                                   COPY_ITEM (.ITEM, 4, JRNL_EXTEND)
5                                                   END;
5
6                               [MNT$_JRNLQUOTA]  : BEGIN
6                                                   JRNL_ITEM_SEEN = 1;
6                                                   COPY_ITEM (.ITEM, 4, JRNL_QUOTA)
5                                                   END;
5
6                               [MNT$_JRNLRECORD_SIZE]: BEGIN
6                                                   JRNL_ITEM_SEEN = 1;
6                                                   COPY_ITEM (.ITEM, 4, JRNL_RECORD_SIZE)
5                                                   END;
5
5                               [OTHERWISE]       : SS$_BADPARAM;
5                               TES
4                           );
4           ITEM = .ITEM + ITEM_SIZE;       ! Point to the next item in the list.
3           END;
2       END;
2
2 !
2 ! If something went awry, stop right now.
2 !
2 IF NOT .STATUS
2 THEN
2       RETURN (.STATUS);
2 !
```

```
2 !*********************************************************************!
2 !                                                                     !
2 ! Now perform some additional sanity checks on each of the parameters. !
2 ! The ordering of the processing is not important.  Note that some     !
2 ! privilege checking is done here, and explicit "no xxxx privilege"    !
2 ! status codes are returned to make $MOUNT more friendly.              !
2 !                                                                     !
2 !*********************************************************************!
2
2 !
2 ! Set the appropriate mount option flags.
2 !
2 MOUNT_OPTIONS [OPT_CLUSTER]     = .MOUNT_FLAGS [MNT$V_CLUSTER];
2 MOUNT_OPTIONS [OPT_SYSTEM]      = .MOUNT_FLAGS [MNT$V_SYSTEM];
2 MOUNT_OPTIONS [OPT_GROUP]       = .MOUNT_FLAGS [MNT$V_GROUP];
2 MOUNT_OPTIONS [OPT_MOUNTVER]    = NOT .MOUNT_FLAGS [MNT$V_NOMNTVER];
2 MOUNT_OPTIONS [OPT_NOQUOTA]     = .MOUNT_FLAGS [MNT$V_NODISKQ];
2 MOUNT_OPTIONS [OPT_NOHDR3]      = .MOUNT_FLAGS [MNT$V_NOHDR3];
2 MOUNT_OPTIONS [OPT_SHARE]       = .MOUNT_FLAGS [MNT$V_SHARE];
2 MOUNT_OPTIONS [OPT_WRITE]       = NOT .MOUNT_FLAGS [MNT$V_NOWRITE];
2 MOUNT_OPTIONS [OPT_NOCACHE]     = .MOUNT_FLAGS [MNT$V_NOCACHE];
2 MOUNT_OPTIONS [OPT_OVR_LOCK]    = .MOUNT_FLAGS [MNT$V_OVR_LOCK];
2 MOUNT_OPTIONS [OPT_MESSAGE]     = .MOUNT_FLAGS [MNT$V_MESSAGE];
2 MOUNT_OPTIONS [OPT_READCHECK]   = .MOUNT_FLAGS [MNT$V_READCHECK];
2 MOUNT_OPTIONS [OPT_WRITECHECK]  = .MOUNT_FLAGS [MNT$V_WRITECHECK];
2 MOUNT_OPTIONS [OPT_WTHRU]       = .MOUNT_FLAGS [MNT$V_WRITETHRU];
2 MOUNT_OPTIONS [OPT_ASSIST]      = NOT .MOUNT_FLAGS [MNT$V_NOASSIST];
2 MOUNT_OPTIONS [OPT_FOREIGN]     = .MOUNT_FLAGS [MNT$V_FOREIGN];
2 MOUNT_OPTIONS [OPT_OVR_EXP]     = .MOUNT_FLAGS [MNT$V_OVR_EXP];
2 MOUNT_OPTIONS [OPT_OVR_ID]      = .MOUNT_FLAGS [MNT$V_OVR_IDENT];
2 MOUNT_OPTIONS [OPT_OVR_SETID]   = .MOUNT_FLAGS [MNT$V_OVR_SETID];
2 MOUNT_OPTIONS [OPT_OVR_ACC]     = .MOUNT_FLAGS [MNT$V_OVR_ACCESS];
2 MOUNT_OPTIONS [OPT_BLOCK]       = 1;
2 MOUNT_OPTIONS [OPT_NOUNLOAD]    = .MOUNT_FLAGS [MNT$V_NOUNLOAD];
2 MOUNT_OPTIONS [OPT_NOJRNL]      = .MOUNT_FLAGS [MNT$V_NOJRNL];
2 MOUNT_OPTIONS [OPT_NEWJRNL]     = .MOUNT_FLAGS [MNT$V_NEWJRNL];
2 MOUNT_OPTIONS [OPT_NOAUTO]      = .MOUNT_FLAGS [MNT$V_NOAUTO];
2 MOUNT_OPTIONS [OPT_INIT_ALL]    = .MOUNT_FLAGS [MNT$V_INIT_ALL];
2 MOUNT_OPTIONS [OPT_INIT_CONT]   = .MOUNT_FLAGS [MNT$V_INIT_CONT];
2 MOUNT_OPTIONS [OPT_OVR_VOLO]    = .MOUNT_FLAGS [MNT$V_OVR_VOLO];
2 MOUNT_OPTIONS [OPT_INTERCHG]    = .MOUNT_FLAGS [MNT$V_INTERCHG];
2 MOUNT_OPTIONS [OPT_NOREBUILD]   = .MOUNT_FLAGS [MNT$V_NOREBUILD];
2
2 !++
2 ! Process the journalling qualifers.  The valued parameters may only
2 ! be specified if MNT$V_NEWJRNL is specified.  It is the responsibility
2 ! of the actual journaling code to check the user supplied values for
2 ! validity, and to apply default values where appropriat.
2 !
2 ! Note that MNT$V_NEWJRNL and MNT$V_NOJRNL are mutually exclusive.
2 !--
2
3 IF (.MOUNT_OPTIONS [OPT_NEWJRNL] AND .MOUNT_OPTIONS [OPT_NOJRNL])
3 OR (.JRNL_ITEM_SEEN AND NOT .MOUNT_OPTIONS [OPT_NEWJRNL])
2 THEN
2     RETURN (MOUN$_CONFQUAL);
2
```

```
2  !++
2  ! Parse the ACP name, if one was provided.
2  !--
2
2  IF .ACP_STRING NEQ 0
2  THEN
3      BEGIN
3      !
3      ! Call TPARSE to interpret this parameter.  All the work
3      ! is done by the action routine.
3      !
3      STR_DESCRIPTOR = ACP_STRING;
3      TPARSE_BLOCK [TPA$L_STRINGCNT] = .STR_DESCRIPTOR [DSC$W_LENGTH];
3      TPARSE_BLOCK [TPA$L_STRINGPTR] = .STR_DESCRIPTOR [DSC$A_POINTER];
4      IF NOT (STATUS = LIB$TPARSE (TPARSE_BLOCK, ACP_STB, ACP_KTB))
3      THEN
3          RETURN (.STATUS);
3      !
3      ! If the ACP string is a device or file name, copy it to internal storage.
3      !
3      IF .MOUNT_OPTIONS [OPT_SAMEACP]
3      OR .MOUNT_OPTIONS [OPT_FILEACP]
3      THEN
4          BEGIN
4          ACP_STRING [0] = .STR_DESCRIPTOR [DSC$W_LENGTH];
4          ACP_STRING [1] = .STR_DESCRIPTOR [DSC$A_POINTER];
3          END;
3
2      END;
2
2  !
2  ! If the volume is being mounted CLUSTER without the /SYSTEM or /GROUP
2  ! qualifier, force it to be mounted SYSTEM.
2  !
2  IF  .MOUNT_OPTIONS [OPT_CLUSTER]
3  AND ( NOT .MOUNT_OPTIONS [OPT_GROUP] )
2  THEN
2      MOUNT_OPTIONS [OPT_SYSTEM] = 1;
2
2  !
2  ! If the volume is not being mounted SHARE, GROUP, or SYSTEM,
2  ! then assume that the volume is being mounted NOSHARE (the default).
2  !
3  IF  (NOT .MOUNT_OPTIONS [OPT_SHARE])
3  AND (NOT .MOUNT_OPTIONS [OPT_GROUP])
3  AND (NOT .MOUNT_OPTIONS [OPT_SYSTEM])
2  THEN
2       MOUNT_OPTIONS [OPT_NOSHARE] = 1;
2
2  !
2  ! If the volume is being mounted GROUP, then it
2  ! cannot be mounted SHARE, or SYSTEM.
2  !
2  IF  .MOUNT_OPTIONS [OPT_GROUP]
3  AND (.MOUNT_OPTIONS [OPT_SHARE] OR .MOUNT_OPTIONS [OPT_SYSTEM])
2  THEN
2      RETURN (MOUN$_CONFQUAL);
```

```
2
2 !
2 ! If the volume is being mounted SYSTEM, then it
2 ! cannot be mounted SHARE, or GROUP.
2 !
2 IF   .MOUNT_OPTIONS [OPT_SYSTEM]
3 AND (.MOUNT_OPTIONS [OPT_SHARE] OR .MOUNT_OPTIONS [OPT_GROUP])
2 THEN
2     RETURN (MOUN$_CONFQUAL);
2
2 !
2 ! If the volume is being mounted SHARE then it
2 ! cannot be mounted SYSTEM or GROUP.
2 !
2 IF   .MOUNT_OPTIONS [OPT_SHARE]
3 AND (.MOUNT_OPTIONS [OPT_SYSTEM] OR .MOUNT_OPTIONS [OPT_GROUP])
2 THEN
2     RETURN (MOUN$_CONFQUAL);
2
2 !
2 ! Do not allow user to override volume identification if requesting
2 ! a SYSTEM, GROUP or SHARE mount.
2 !
2 IF   .MOUNT_OPTIONS [OPT_OVR_ID]
3 AND (.MOUNT_OPTIONS [OPT_SYSTEM] OR
3      .MOUNT_OPTIONS [OPT_GROUP]  OR
3      .MOUNT_OPTIONS [OPT_SHARE])
2 THEN
2     RETURN (MOUN$_CONFQUAL);
2
2 !
2 ! If no device names have been seen,
2 ! reject the mount request.
2 !
2 IF NOT .MOUNT_OPTIONS [OPT_DEVICE]
2 THEN
2     RETURN (SS$_BADPARAM);
2
2 !
2 ! If no volume label seen, make sure the user has
2 ! done this on purpose.  If not, exit.
2 !
2 IF  NOT .MOUNT_OPTIONS [OPT_LABEL]             ! Indicate volume label seen
3 AND NOT (.MOUNT_OPTIONS [OPT_OVR_ID]    OR
3          .MOUNT_OPTIONS [OPT_FOREIGN])
2 THEN
2     RETURN (SS$_BADPARAM);
2
2 !
2 ! If the device is being mounted /FOREIGN, then many of
2 ! the qualifiers having to do with FILES-11 are not legal.
2 !
2 IF   .MOUNT_OPTIONS [OPT_FOREIGN]
3 AND (.MOUNT_OPTIONS [OPT_ACCESSED]      OR
3      .MOUNT_OPTIONS [OPT_UNIQUEACP]     OR
3      .MOUNT_OPTIONS [OPT_FILEACP]       OR
3      .MOUNT_OPTIONS [OPT_SAMEACP]       OR
3         .MOUNT_OPTIONS [OPT_BIND]         OR
3         .MOUNT_OPTIONS [OPT_CACHE]        OR
3         .MOUNT_OPTIONS [OPT_WINDOW]       OR
3         .MOUNT_OPTIONS [OPT_EXTENSION]
3        )
2    THEN
2        RETURN (MOUN$_CONFQUAL);
2
2    !+
2    ! Perform some preliminary privilege checks.
2    !-
2
2    PHD = .CTL$GL_PHD;                                       ! Get the process header address
2
2    IF   .MOUNT_OPTIONS [OPT_GROUP]                          ! If /GROUP, user must have GRPNAM
2    AND  NOT .BBLOCK [PHD [PHD$Q_PRIVMSK], PRV$V_GRPNAM]
2    THEN
2        RETURN (SS$_NOGRPNAM);
2
2    IF   .MOUNT_OPTIONS [OPT_SYSTEM]                         ! If /SYSTEM, user must have SYSNAM
2    AND NOT .BBLOCK [PHD [PHD$Q_PRIVMSK], PRV$V_SYSNAM]
2    THEN
2        RETURN (SS$_NOSYSNAM);
2
3    IF   (.MOUNT_OPTIONS [OPT_UNIQUEACP]     OR              ! This check must be performed AFTER the /PROCESSOR
3         .MOUNT_OPTIONS [OPT_SAMEACP]       OR              ! qualifier has been processed.
3         .MOUNT_OPTIONS [OPT_FILEACP])
2    AND NOT .BBLOCK [PHD [PHD$Q_PRIVMSK], PRV$V_OPER]
2    THEN
2        RETURN (SS$_NOOPER);
2
2    IF   .MOUNT_OPTIONS [OPT_FILEACP]                        ! Must have CMK to use
2    AND NOT .BBLOCK [PHD [PHD$Q_PRIVMSK], PRV$V_CMKRNL]     ! a special ACP
2    THEN
2        RETURN (SS$_NOCMKRNL);
2
2
2    !++
2    ! If we get this far, the parameters have passed the
2    ! preliminary checks.  Return a successful status.
2    !--
2
2    SS$_NORMAL
2
1    END;                                                 ! End of routine CHECK_PARAMETERS

                                                          .TITLE  MOUPAR
                                                          .IDENT  \V04-000\

                                                          .PSECT  $OWN$,NOEXE,2

                                           00000 OWN_START:
                                                          .BLKB   0
                                           00000 DEVICE_BUFFER:
                                                          .BLKB   1008
                                           003F0 LABEL_BUFFER:
```

```
                .BLKB   1008
007E0 LOG_NAME_BUFFER:
                .BLKB   63
0081F           .BLKB   1
00820 ACP_NAME_BUFFER:
                .BLKB   63
0085F           .BLKB   1
00860 VOLSET_BUFFER:
                .BLKB   63
0089F           .BLKB   1
008A0 COMMENT_BUFFER:
                .BLKB   80
008F0 TPARSE_BLOCK:
                .BLKB   36
00914 OWN_END:.BLKB     0

                .PSECT  $GLOBAL$,NOEXE,2

00000 GLOBAL_START::
                .BLKB   0
00000 MOUNT_OPTIONS::
                .BLKB   8
00008 MOUNT_FLAGS::
                .BLKB   4
0000C PROTECTION::
                .BLKB   4
00010 OWNER_UIC::
                .BLKB   4
00014 USER_UIC::
                .BLKB   4
00018 EXTENSION::
                .BLKB   4
0001C WINDOW::.BLKB     4
00020 ACCESSED::
                .BLKB   4
00024 BLOCKSIZE::
                .BLKB   4
00028 RECORDSZ::
                .BLKB   4
0002C FID_CACHE::
                .BLKB   4
00030 EXT_CACHE::
                .BLKB   4
00034 QUO_CACHE::
                .BLKB   4
00038 EXT_LIMIT::
                .BLKB   4
0003C DEVICE_COUNT::
                .BLKB   4
00040 LABEL_COUNT::
                .BLKB   4
00044 VID_STRING::
                .BLKB   8
0004C COMMENT_STRING::
                .BLKB   8
00054 ACP_STRING::
                .BLKB   8
```

```
                                            0005C LOG_NAME::
                                                          .BLKB   8
                                            00064 STRUCT_NAME::
                                                          .BLKB   8
                                            0006C DRIVE_COUNT::
                                                          .BLKB   64
                                            000AC DEVICE_STRING::
                                                          .BLKB   128
                                            0012C LABEL_STRING::
                                                          .BLKB   128
                                            001AC JRNL_SIZE::
                                                          .BLKB   4
                                            001B0 JRNL_EXTEND::
                                                          .BLKB   4
                                            001B4 JRNL_QUOTA::
                                                          .BLKB   4
                                            001B8 JRNL_RECORD_SIZE::
                                                          .BLKB   4
                                            001BC GLOBAL_END::
                                                          .BLKB   0

                                                          .EXTRN  DEVCHAR_DESC, DEVICE_CHAR
                                                          .EXTRN  NEWLINE, CTL$GL_PHD
                                                          .EXTRN  LIB$TPARSE

                                                          .PSECT  $CODE$,NOWRT,2

                                       0FFC 00000         .ENTRY  CHECK_PARAMS, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  0762
                                                                  R10,R11                                    :
                          5B     0000'  CF   9E 00002     MOVAB   GLOBAL_START, R11                          :
                          5E            04   C2 00007     SUBL2   #4, SP                                     :
 01BC   8F          00    6E            00   2C 0000A     MOVC5   #0, (SP), #0, #444, GLOBAL_START           :  0825
                                        6B      00011                                                        :
 0914   8F          00    6E            00   2C 00012     MOVC5   #0, (SP), #0, #2324, OWN_START             :  0826
                                 0000'  CF      00019                                                        :
                 0000'    CF            08   D0 0001C     MOVL    #8, TPARSE_BLOCK                           :  0831
                 0000'    CF            03   D0 00021     MOVL    #3, TPARSE_BLOCK+4                         :  0832
                                 04     AB   7C 00026     CLRQ    MOUNT_OPTIONS+4                            :  0850
                                        6B   D4 00029     CLRL    MOUNT_OPTIONS                              :
                          58            01   7D 0002B     MOVQ    #1, STATUS                                 :  0851
                                 3C     AB   7C 0002E     CLRQ    DEVICE_COUNT                               :  0853
                          56     04     AC   D0 00031     MOVL    ITEM_LIST, ITEM                            :  0855
                    66    04            00   0C 00035     PROBER  #0, #4, (ITEM)                             :  0856
                                        04   12 00039     BNEQ    1$                                         :
                          50            0C   D0 0003B     MOVL    #12, R0                                    :  0858
                                             04 0003E     RET                                                :
                          03            58   E8 0003F 1$: BLBS    STATUS, 2$                                 :  0860
                                      04CD   31 00042     BRW     62$                                        :
                                 02     A6   B5 00045 2$: TSTW    2(ITEM)                                    :
                                        03   12 00048     BNEQ    3$                                         :
                                      032B   31 0004A     BRW     57$                                        :
                    66    0C            00   0C 0004D 3$: PROBER  #0, #12, (ITEM)                            :  0871
                                        05   12 00051     BNEQ    4$                                         :
                          58            0C   D0 00053     MOVL    #12, STATUS                                :  0873
                                        E7   11 00056     BRB     1$                                         :
                          52     02     A6   3C 00058 4$: MOVZWL  2(ITEM), R2                                :  0876
                          01            52   B1 0005C     CMPW    R2, #1                                     :  0881
```

```
                                                         L 12
MOUPAR                                                   16-Sep-1984 01:22:31    VAX-11 Bliss-32 V4.0-742             Page  18
V04-000                                                  14-Sep-1984 12:45:31    DISK$VMSMASTER:[MOUNT.SRC]MOUPAR.B32;1    (4)

                                     42  12 0005F          BNEQ    7$                                             :
                     10         3C   AB  D1 00061          CMPL    DEVICE_COUNT, #16                              :
                                     33  18 00065          BGEQ    6$                                             :
                03   AB         40   8F  88 00067          BISB2   #64, MOUNT_OPTIONS+3                           : 0884
                                3C   AB  D6 0006C          INCL    DEVICE_COUNT                                   : 0885
          50    3C   AB              01  78 0006F          ASHL    #1, DEVICE_COUNT, R0                           : 0886
                     51              66  3C 00074          MOVZWL  (ITEM), R1                                     :
                     3F              51  B1 00077          CMPW    R1, #63                                        :
                                     03  1B 0007A          BLEQU   5$                                             :
                     51              3F  D0 0007C          MOVL    #63, R1                                        :
           00A4 CB40                 51  D0 0007F 5$:      MOVL    R1, DEVICE_STRING-8[R0]                        :
          51    3C   AB              3F  C5 00085          MULL3   #63, DEVICE_COUNT, R1                          : 0887
           00A8 CB40          0000'CF41  9E 0008A          MOVAB   DEVICE_BUFFER[R1], DEVICE_STRING-4[R0]         :
                              00A8 CB40  DD 00093          PUSHL   DEVICE_STRING-4[R0]                            : 0888
                                     45  11 00098          BRB     9$                                             :
                58 00728084          8F  D0 0009A 6$:      MOVL    #7504004, STATUS                               : 0881
                                     45  11 000A1          BRB     11$                                            :
                     02              52  B1 000A3 7$:      CMPW    R2, #2                                         : 0893
                                     42  12 000A6          BNEQ    12$                                            :
                     10         40   AB  D1 000A8          CMPL    LABEL_COUNT, #16                               :
                                     33  18 000AC          BGEQ    10$                                            :
                03   AB         80   8F  88 000AE          BISB2   #128, MOUNT_OPTIONS+3                          : 0896
                                40   AB  D6 000B3          INCL    LABEL_COUNT                                    : 0897
          50    40   AB              01  78 000B6          ASHL    #1, LABEL_COUNT, R0                            : 0898
                     51              66  3C 000BB          MOVZWL  (ITEM), R1                                     :
                     3F              51  B1 000BE          CMPW    R1, #63                                        :
                                     03  1B 000C1          BLEQU   8$                                             :
                     51              3F  D0 000C3          MOVL    #63, R1                                        :
           0124 CB40                 51  D0 000C6 8$:      MOVL    R1, LABEL_STRING-8[R0]                         :
          51    40   AB              3F  C5 000CC          MULL3   #63, LABEL_COUNT, R1                           : 0899
           0128 CB40          0000'CF41  9E 000D1          MOVAB   LABEL_BUFFER[R1], LABEL_STRING-4[R0]           :
                              0128 CB40  DD 000DA          PUSHL   LABEL_STRING-4[R0]                             : 0900
                                     6D  11 000DF 9$:      BRB     18$                                            :
                58 0072808C          8F  D0 000E1 10$:     MOVL    #7504012, STATUS                               : 0893
                                     43  11 000E8 11$:     BRB     15$                                            :
                     03              52  B1 000EA 12$:     CMPW    R2, #3                                         : 0905
                                     41  12 000ED          BNEQ    16$                                            :
                03   AB              20  88 000EF          BISB2   #32, MOUNT_OPTIONS+3                           : 0906
                     50              66  3C 000F3          MOVZWL  (ITEM), R0                                     : 0907
                     3F              50  B1 000F6          CMPW    R0, #63                                        :
                                     03  1B 000F9          BLEQU   13$                                            :
                     50              3F  D0 000FB          MOVL    #63, R0                                        :
                5C   AB              50  D0 000FE 13$:     MOVL    R0, LOG_NAME                                   :
                60   AB         0000'    CF  9E 00102          MOVAB   LOG_NAME_BUFFER, LOG_NAME+4                    : 0908
                                0000'    CF  9F 00106          PUSHAB  LOG_NAME_BUFFER                                : 0909
                                     3F  DD 0010C          PUSHL   #63                                            :
                                     56  DD 0010E          PUSHL   ITEM                                           :
                0000V   CF           03  FB 00110          CALLS   #3, COPY_ITEM                                  :
                     58              50  D0 00115          MOVL    R0, STATUS                                     :
     60   BB    5C   AB              3A  3A 00118          LOCC    #58, LOG_NAME, @LOG_NAME+4                     : 0914
                                     02  12 0011E          BNEQ    14$                                            :
                                     51  D4 00120          CLRL    R1                                             :
                     5A              51  D0 00122 14$:     MOVL    R1, PTR                                        :
                                     06  13 00125          BEQL    15$                                            : 0915
     5C   AB         5A         60   AB  C3 00127          SUBL3   LOG_NAME+4, PTR, LOG_NAME                      : 0917
                                   0242  31 0012D 15$:     BRW     56$                                            : 0918
                     06              52  B1 00130 16$:     CMPW    R2, #6                                         : 0921
```

```
                                                   M 12
MOUPAR                                      16-Sep-1984 01:22:31    VAX-11 Bliss-32 V4.0-742                Page  19
V04-000                                     14-Sep-1984 12:45:31    DISK$VMSMASTER:[MOUNT.SRC]MOUPAR.B32;1  (4)

                            1B  12 00133          BNEQ    19$                                ;
              50            66  3C 00135          MOVZWL  (ITEM), R0                         ; 0922
              3F            50  B1 00138          CMPW    R0, #63                            ;
                            03  1B 0013B          BLEQU   17$                                ;
              50            3F  D0 0013D          MOVL    #63, R0                            ;
        54    AB            50  D0 00140 17$:     MOVL    R0, ACP_STRING                     ;
        58    AB    0000'   CF  9E 00144          MOVAB   ACP_NAME_BUFFER, ACP_STRING+4      ; 0923
                    0000'   CF  9F 0014A          PUSHAB  ACP_NAME_BUFFER                    ; 0924
                            22  11 0014E 18$:     BRB     21$                                ;
              07            52  B1 00150 19$:     CMPW    R2, #7                             ; 0927
                            22  12 00153          BNEQ    22$                                ;
              50            66  3C 00155          MOVZWL  (ITEM), R0                         ; 0928
              3F            50  B1 00158          CMPW    R0, #63                            ;
                            03  1B 0015B          BLEQU   20$                                ;
              50            3F  D0 0015D          MOVL    #63, R0                            ;
        64    AB            50  D0 00160 20$:     MOVL    R0, STRUCT_NAME                    ;
        68    AB    0000'   CF  9E 00164          MOVAB   VOLSET_BUFFER, STRUCT_NAME+4       ; 0929
        05    AB            01  88 0016A          BISB2   #1, MOUNT_OPTIONS+5                ; 0930
                    0000'   CF  9F 0016E          PUSHAB  VOLSET_BUFFER                      ; 0931
                            3F  DD 00172 21$:     PUSHL   #63                                ;
                          01EC  31 00174          BRW     54$                                ;
              04            52  B1 00177 22$:     CMPW    R2, #4                             ; 0938
                            05  12 0017A          BNEQ    23$                                ;
                        08  AB  9F 0017C          PUSHAB  MOUNT_FLAGS                        ;
                            0C  11 0017F          BRB     24$                                ;
              05            52  B1 00181 23$:     CMPW    R2, #5                             ; 0940
                            0A  12 00184          BNEQ    25$                                ;
        03    AB            02  88 00186          BISB2   #2, MOUNT_OPTIONS+3                ; 0941
                        20  AB  9F 0018A          PUSHAB  ACCESSED                           ; 0942
                          01D1  31 0018D 24$:     BRW     53$                                ;
              08            52  B1 00190 25$:     CMPW    R2, #8                             ; 0945
                            21  12 00193          BNEQ    26$                                ;
        02    AB            01  88 00195          BISB2   #1, MOUNT_OPTIONS+2                ; 0946
                        24  AB  9F 00199          PUSHAB  BLOCKSIZE                          ; 0947
                            04  DD 0019C          PUSHL   #4                                 ;
                            56  DD 0019E          PUSHL   ITEM                               ;
     0000V    CF            03  FB 001A0          CALLS   #3, COPY_ITEM                      ;
              58            50  D0 001A5          MOVL    R0, STATUS                         ;
              82            58  E9 001A8          BLBC    STATUS, 15$                        ;
  0000FFFE    8F        24  AB  D1 001AB          CMPL    BLOCKSIZE, #65534                  ; 0949
                          0111  31 001B3          BRW     40$                                ;
              09            52  B1 001B6 26$:     CMPW    R2, #9                             ; 0955
                            4D  12 001B9          BNEQ    29$                                ;
              6B            01  88 001BB          BISB2   #1, MOUNT_OPTIONS                  ; 0956
              6B            02  8A 001BE          BICB2   #2, MOUNT_OPTIONS                  ; 0957
        05    AB            08  8A 001C1          BICB2   #8, MOUNT_OPTIONS+5                ; 0958
                            5E  DD 001C5          PUSHL   SP                                 ; 0959
                            04  DD 001C7          PUSHL   #4                                 ;
                            56  DD 001C9          PUSHL   ITEM                               ;
     0000V    CF            03  FB 001CB          CALLS   #3, COPY_ITEM                      ;
              58            50  D0 001D0          MOVL    R0, STATUS                         ;
              7B            58  E9 001D3          BLBC    STATUS, 31$                        ;
              50            6E  D0 001D6          MOVL    DENSITY, R0                        ; 0961
  00000320    8F            50  D1 001D9          CMPL    R0, #800                           ; 0963
                            05  12 001E0          BNEQ    27$                                ;
              6B            02  88 001E2          BISB2   #2, MOUNT_OPTIONS                  ;
                            6A  11 001E5          BRB     31$                                ;
        00000640   8F               50  D1  001E7  27$:    CMPL    R0, #1600                          ; 0964
                                    06  12  001EE          BNEQ    28$                                ;
              05   AB               08  88  001F0          BISB2   #8, MOUNT_OPTIONS+5                ;
                                    5B  11  001F4          BRB     31$                                ;
        0000186A   8F               50  D1  001F6  28$:    CMPL    R0, #6250                          ; 0965
                                    52  13  001FD          BEQL    31$                                ;
                   58  00728014     8F  D0  001FF          MOVL    #7503892, STATUS                   ; 0966
                                    49  11  00206          BRB     31$                                ; 0961
                   0A               52  B1  00208  29$:    CMPW    R2, #10                            ; 0971
                                    20  12  0020B          BNEQ    30$                                ;
                              30    AB  9F  0020D          PUSHAB  EXT_CACHE                          ; 0972
                                    04  DD  00210          PUSHL   #4                                 ;
                                    56  DD  00212          PUSHL   ITEM                               ;
           0000V   CF               03  FB  00214          CALLS   #3, COPY_ITEM                      ;
                   58               50  D0  00219          MOVL    R0, STATUS                         ;
              05   AB         A0    8F  8A  0021C          BICB2   #160, MOUNT_OPTIONS+5              ; 0974
                              30    AB  D5  00221          TSTL    EXT_CACHE                          ; 0975
                                    78  14  00224          BGTR    37$                                ;
              05   AB         80    8F  88  00226          BISB2   #128, MOUNT_OPTIONS+5              ; 0977
                                    75  11  0022B          BRB     38$                                ;
                   0B               52  B1  0022D  30$:    CMPW    R2, #11                            ; 0983
                                    21  12  00230          BNEQ    32$                                ;
                              2C    AB  9F  00232          PUSHAB  FID_CACHE                          ; 0984
                                    04  DD  00235          PUSHL   #4                                 ;
                                    56  DD  00237          PUSHL   ITEM                               ;
           0000V   CF               03  FB  00239          CALLS   #3, COPY_ITEM                      ;
                   58               50  D0  0023E          MOVL    R0, STATUS                         ;
              05   AB       0120    8F  AA  00241          BICW2   #288, MOUNT_OPTIONS+5              ; 0985
                   01         2C    AB  D1  00247          CMPL    FID_CACHE, #1                      ; 0987
                                    51  14  0024B          BGTR    37$                                ;
              06   AB               01  88  0024D          BISB2   #1, MOUNT_OPTIONS+6                ; 0989
                                    7D  11  00251  31$:    BRB     41$                                ;
                   0C               52  B1  00253  32$:    CMPW    R2, #12                            ; 0995
                                    05  12  00256          BNEQ    33$                                ;
                              38    AB  9F  00258          PUSHAB  EXT_LIMIT                          ;
                                    0C  11  0025B          BRB     34$                                ;
                   0D               52  B1  0025D  33$:    CMPW    R2, #13                            ; 0997
                                    09  12  00260          BNEQ    35$                                ;
              02   AB               04  88  00262          BISB2   #4, MOUNT_OPTIONS+2                ; 0998
                              10    AB  9F  00266          PUSHAB  OWNER_UIC                          ; 0999
                                    74  11  00269  34$:    BRB     43$                                ;
                   0E               52  B1  0026B  35$:    CMPW    R2, #14                            ; 1002
                                    09  12  0026E          BNEQ    36$                                ;
              02   AB               02  88  00270          BISB2   #2, MOUNT_OPTIONS+2                ; 1003
                              0C    AB  9F  00274          PUSHAB  PROTECTION                         ; 1004
                                    75  11  00277          BRB     45$                                ;
                   0F               52  B1  00279  36$:    CMPW    R2, #15                            ; 1007
                                    26  12  0027C          BNEQ    39$                                ;
                              34    AB  9F  0027E          PUSHAB  QUO_CACHE                          ; 1008
                                    04  DD  00281          PUSHL   #4                                 ;
                                    56  DD  00283          PUSHL   ITEM                               ;
           0000V   CF               03  FB  00285          CALLS   #3, COPY_ITEM                      ;
                   58               50  D0  0028A          MOVL    R0, STATUS                         ;
              05   AB       0220    8F  AA  0028D          BICW2   #544, MOUNT_OPTIONS+5              ; 1009
                              34    AB  D5  00293          TSTL    QUO_CACHE                          ; 1011
                                    06  14  00296          BGTR    37$                                ;
              06   AB               02  88  00298          BISB2   #2, MOUNT_OPTIONS+6                ; 1013
```

```
                                        32  11 0029C        BRB     41$                                          ;
                 05   AB                20  88 0029E 37$:   BISB2   #32, MOUNT_OPTIONS+5                         ; 1015
                                        2C  11 002A2 38$:   BRB     41$                                          ; 1016
                      10                52  B1 002A4 39$:   CMPW    R2, #16                                      ; 1019
                                        2A  12 002A7        BNEQ    42$                                          ;
                 04   AB                20  88 002A9        BISB2   #32, MOUNT_OPTIONS+4                         ; 1020
                                  28    AB  9F 002AD        PUSHAB  RECORDSZ                                     ; 1021
                                        04  DD 002B0        PUSHL   #4                                           ;
                                        56  DD 002B2        PUSHL   ITEM                                         ;
              0000V   CF                03  FB 002B4        CALLS   #3, COPY_ITEM                                ;
                      58                50  D0 002B9        MOVL    R0, STATUS                                   ;
                      11                58  E9 002BC        BLBC    STATUS, 41$                                  ;
          0000FFFE    8F          28    AB  D1 002BF        CMPL    RECORDSZ, #65534                             ; 1023
                                        07  15 002C7 40$:   BLEQ    41$                                          ;
                      58 0072817C       8F  D0 002C9        MOVL    #7504252, STATUS                             ; 1025
                                     009F   31 002D0 41$:   BRW     56$                                          ; 1026
                      11                52  B1 002D3 42$:   CMPW    R2, #17                                      ; 1029
                                        09  12 002D6        BNEQ    44$                                          ;
                 03   AB                01  88 002D8        BISB2   #1, MOUNT_OPTIONS+3                          ; 1030
                                  1C    AB  9F 002DC        PUSHAB  WINDOW                                       ; 1031
                                        72  11 002DF 43$:   BRB     51$                                          ;
                      12                52  B1 002E1 44$:   CMPW    R2, #18                                      ; 1034
                                        0A  12 002E4        BNEQ    46$                                          ;
                 02   AB          80    8F  88 002E6        BISB2   #128, MOUNT_OPTIONS+2                        ; 1035
                                  18    AB  9F 002EB        PUSHAB  EXTENSION                                    ; 1036
                                        71  11 002EE 45$:   BRB     53$                                          ;
                      14                52  B1 002F0 46$:   CMPW    R2, #20                                      ; 1039
                                        36  12 002F3        BNEQ    48$                                          ;
                      6B                08  88 002F5        BISB2   #8, MOUNT_OPTIONS                            ; 1043
                      51                66  3C 002F8        MOVZWL  (ITEM), R1                                   ; 1044
               0050   8F                51  B1 002FB        CMPW    R1, #80                                      ;
                                        04  1B 00300        BLEQU   47$                                          ;
                      51          50    8F  9A 00302        MOVZBL  #80, R1                                      ;
                      57        0000G   CF  3C 00306 47$:   MOVZWL  NEWLINE, R7                                  ;
      4C     AB       51                57  C1 0030B        ADDL3   R7, R1, COMMENT_STRING                       ;
                      50    AB  0000'   CF  9E 00310        MOVAB   COMMENT_BUFFER, COMMENT_STRING+4             ; 1045
  0000'      CF    0000G    DF          57  28 00316        MOVC3   R7, @NEWLINE+4, COMMENT_BUFFER               ; 1046
                                0000'CF 47  9F 0031E        PUSHAB  COMMENT_BUFFER[R7]                           ; 1049
                                  B0    A7  9F 00323        PUSHAB  -80(R7)                                      ; 1048
                      6E                6E  CE 00326        MNEGL   (SP), (SP)                                   ;
                                        38  11 00329        BRB     54$                                          ; 1047
                      15                52  B1 0032B 48$:   CMPW    R2, #21                                      ; 1053
                                        09  12 0032E        BNEQ    49$                                          ;
                      59                01  D0 00330        MOVL    #1, JRNL_ITEM_SEEN                           ; 1054
                                01AC    CB  9F 00333        PUSHAB  JRNL_SIZE                                    ; 1055
                                        28  11 00337        BRB     53$                                          ;
                      16                52  B1 00339 49$:   CMPW    R2, #22                                      ; 1058
                                        09  12 0033C        BNEQ    50$                                          ;
                      59                01  D0 0033E        MOVL    #1, JRNL_ITEM_SEEN                           ; 1059
                                01B0    CB  9F 00341        PUSHAB  JRNL_EXTEND                                  ; 1060
                                        1A  11 00345        BRB     53$                                          ;
                      17                52  B1 00347 50$:   CMPW    R2, #23                                      ; 1063
                                        09  12 0034A        BNEQ    52$                                          ;
                      59                01  D0 0034C        MOVL    #1, JRNL_ITEM_SEEN                           ; 1064
                                01B4    CB  9F 0034F        PUSHAB  JRNL_QUOTA                                   ; 1065
                                        0C  11 00353 51$:   BRB     53$                                          ;
                      18                52  B1 00355 52$:   CMPW    R2, #24                                      ; 1068
```

```
                                             15 12 00358       BNEQ     55$                                       ;
                                59           01 D0 0035A       MOVL     #1, JRNL_ITEM_SEEN                        ; 1069
                                      01B8   CB 9F 0035D       PUSHAB   JRNL_RECORD_SIZE                          ; 1070
                                             04 DD 00361 53$:  PUSHL    #4                                        ;
                                             56 DD 00363 54$:  PUSHL    ITEM                                      ;
                         0000V  CF           03 FB 00365       CALLS    #3, COPY_ITEM                             ;
                                58           50 D0 0036A       MOVL     R0, STATUS                                ;
                                             03 11 0036D       BRB      56$                                       ;
                                58           14 D0 0036F 55$:  MOVL     #20, STATUS                               ; 1073
                                56           0C C0 00372 56$:  ADDL2    #12, ITEM                                 ; 1076
                                           FCC7 31 00375       BRW      1$                                        ; 0860
                                03           58 E8 00378 57$:  BLBS     STATUS, 58$                               ; 1083
                                           0194 31 0037B       BRW      62$                                       ;
     50     0B    AB            01           04 EF 0037E 58$:  EXTZV    #4, #1, MOUNT_FLAGS+3, R0                 ; 1099
07   AB           01            06           50 F0 00384       INSV     R0, #6, #1, MOUNT_OPTIONS+7               ;
     50     09    AB            01           06 EF 0038A       EXTZV    #6, #1, MOUNT_FLAGS+1, R0                 ; 1100
01   AB           01            00           50 F0 00390       INSV     R0, #0, #1, MOUNT_OPTIONS+1               ;
     50     08    AB            01           01 EF 00396       EXTZV    #1, #1, MOUNT_FLAGS, R0                   ; 1101
     6B           01            07           50 F0 0039C       INSV     R0, #7, #1, MOUNT_OPTIONS                 ;
     50     0A    AB            01           03 EF 003A1       EXTZV    #3, #1, MOUNT_FLAGS+2, R0                 ; 1102
                                50           50 D2 003A7       MCOML    R0, R0                                    ;
06   AB           01            06           50 F0 003AA       INSV     R0, #6, #1, MOUNT_OPTIONS+6               ;
     50     08    AB            01           03 EF 003B0       EXTZV    #3, #1, MOUNT_FLAGS, R0                   ; 1103
05   AB           01            02           50 F0 003B6       INSV     R0, #2, #1, MOUNT_OPTIONS+5               ;
     50     08    AB            01           04 EF 003BC       EXTZV    #4, #1, MOUNT_FLAGS, R0                   ; 1104
05   AB           01            04           50 F0 003C2       INSV     R0, #4, #1, MOUNT_OPTIONS+5               ;
     50     09    AB            01           04 EF 003C8       EXTZV    #4, #1, MOUNT_FLAGS+1, R0                 ; 1105
     6B           01            06           50 F0 003CE       INSV     R0, #6, #1, MOUNT_OPTIONS                 ;
     50     08    AB            01           06 EF 003D3       EXTZV    #6, #1, MOUNT_FLAGS, R0                   ; 1106
                                50           50 D2 003D9       MCOML    R0, R0                                    ;
01   AB           01            01           50 F0 003DC       INSV     R0, #1, #1, MOUNT_OPTIONS+1               ;
     50     0A    AB            01           01 EF 003E2       EXTZV    #1, #1, MOUNT_FLAGS+2, R0                 ; 1107
06   AB           01            04           50 F0 003E8       INSV     R0, #4, #1, MOUNT_OPTIONS+6               ;
     50     0A    AB            01           02 EF 003EE       EXTZV    #2, #1, MOUNT_FLAGS+2, R0                 ; 1108
06   AB           01            05           50 F0 003F4       INSV     R0, #5, #1, MOUNT_OPTIONS+6               ;
     50     09    AB            01           05 EF 003FA       EXTZV    #5, #1, MOUNT_FLAGS+1, R0                 ; 1109
06   AB           01            03           50 F0 00400       INSV     R0, #3, #1, MOUNT_OPTIONS+6               ;
     50     09    AB            01           03 EF 00406       EXTZV    #3, #1, MOUNT_FLAGS+1, R0                 ; 1110
04   AB           01            03           50 F0 0040C       INSV     R0, #3, #1, MOUNT_OPTIONS+4               ;
     50     09    AB            01           07 EF 00412       EXTZV    #7, #1, MOUNT_FLAGS+1, R0                 ; 1111
04   AB           01            04           50 F0 00418       INSV     R0, #4, #1, MOUNT_OPTIONS+4               ;
05   AB           01            06     0A    AB F0 0041E       INSV     MOUNT_FLAGS+2, #6, #1, MOUNT_OPTIONS+5    ; 1112
     50     08    AB            01           02 EF 00425       EXTZV    #2, #1, MOUNT_FLAGS, R0                   ; 1113
                                50           50 D2 0042B       MCOML    R0, R0                                    ;
06   AB           01            02           50 F0 0042E       INSV     R0, #2, #1, MOUNT_OPTIONS+6               ;
01   AB           01            03     08    AB F0 00434       INSV     MOUNT_FLAGS, #3, #1, MOUNT_OPTIONS+1      ; 1114
02   AB           01            04     09    AB F0 0043B       INSV     MOUNT_FLAGS+1, #4, #1, MOUNT_OPTIONS+2    ; 1115
     50     09    AB            01           01 EF 00442       EXTZV    #1, #1, MOUNT_FLAGS+1, R0                 ; 1116
02   AB           01            06           50 F0 00448       INSV     R0, #6, #1, MOUNT_OPTIONS+2               ;
     50     09    AB            01           02 EF 0044E       EXTZV    #2, #1, MOUNT_FLAGS+1, R0                 ; 1117
02   AB           01            05           50 F0 00454       INSV     R0, #5, #1, MOUNT_OPTIONS+2               ;
     50     08    AB            01           07 EF 0045A       EXTZV    #7, #1, MOUNT_FLAGS, R0                   ; 1118
04   AB           01            06           50 F0 00460       INSV     R0, #6, #1, MOUNT_OPTIONS+4               ;
                          01    AB     80    8F 88 00466       BISB2    #128, MOUNT_OPTIONS+1                     ; 1119
     50     0A    AB            01           04 EF 0046B       EXTZV    #4, #1, MOUNT_FLAGS+2, R0                 ; 1120
01   AB           01            02           50 F0 00471       INSV     R0, #2, #1, MOUNT_OPTIONS+1               ;
     50     0A    AB            01           05 EF 00477       EXTZV    #5, #1, MOUNT_FLAGS+2, R0                 ; 1121
```

```
      06   AB          01              07          50 F0 0047D         INSV    R0, #7, #1, MOUNT_OPTIONS+6                     ;
           50     0A   AB              01          06 EF 00483         EXTZV   #6, #1, MOUNT_FLAGS+2, R0                       ; 1122
      07   AB          01              00          50 F0 00489         INSV    R0, #0, #1, MOUNT_OPTIONS+7                     ;
           50     0A   AB              01          07 EF 0048F         EXTZV   #7, #1, MOUNT_FLAGS+2, R0                       ; 1123
      07   AB          01              01          50 F0 00495         INSV    R0, #1, #1, MOUNT_OPTIONS+7                     ;
      07   AB          01              02     0B   AB F0 0049B         INSV    MOUNT_FLAGS+3, #2, #1, MOUNT_OPTIONS+7          ; 1124
           50     0B   AB              01          01 EF 004A2         EXTZV   #1, #1, MOUNT_FLAGS+3, R0                       ; 1125
      07   AB          01              03          50 F0 004A8         INSV    R0, #3, #1, MOUNT_OPTIONS+7                     ;
           50     0B   AB              01          02 EF 004AE         EXTZV   #2, #1, MOUNT_FLAGS+3, R0                       ; 1126
      07   AB          01              04          50 F0 004B4         INSV    R0, #4, #1, MOUNT_OPTIONS+7                     ;
           50     0B   AB              01          03 EF 004BA         EXTZV   #3, #1, MOUNT_FLAGS+3, R0                       ; 1127
      07   AB          01              05          50 F0 004C0         INSV    R0, #5, #1, MOUNT_OPTIONS+7                     ;
           50     0B   AB              01          05 EF 004C6         EXTZV   #5, #1, MOUNT_FLAGS+3, R0                       ; 1128
      07   AB          01              07          50 F0 004CC         INSV    R0, #7, #1, MOUNT_OPTIONS+7                     ;
                                       08     07   AB E9 004D2         BLBC    MOUNT_OPTIONS+7, 60$                            ; 1139
                                              06   AB 95 004D6         TSTB    MOUNT_OPTIONS+6                                 ;
                                                   03 18 004D9         BGEQ    60$                                             ;
                                                 00E3 31 004DB 59$:    BRW     75$                                             ;
                                       04          59 E9 004DE 60$:    BLBC    JRNL_ITEM_SEEN, 61$                             ; 1140
                                       F6     07   AB E9 004E1         BLBC    MOUNT_OPTIONS+7, 59$                            ;
                                              54   AB D5 004E5 61$:    TSTL    ACP_STRING                                      ; 1148
                                                   3F 13 004E8         BEQL    65$                                             ;
                                       52     54   AB 9E 004EA         MOVAB   ACP_STRING, STR_DESCRIPTOR                      ; 1155
                                 0000'   CF        62 3C 004EE         MOVZWL  (STR_DESCRIPTOR), TPARSE_BLOCK+8                ; 1156
                                 0000'   CF   04   A2 D0 004F3         MOVL    4(STR_DESCRIPTOR), TPARSE_BLOCK+12              ; 1157
                                          0000V    CF 9F 004F9         PUSHAB  ACP_KTB                                         ; 1158
                                          0000V    CF 9F 004FD         PUSHAB  ACP_STB                                         ;
                                          0000'    CF 9F 00501         PUSHAB  TPARSE_BLOCK                                    ;
                             00000000G   00        03 FB 00505         CALLS   #3, LIB$TPARSE                                  ;
                                         58        50 D0 0050C         MOVL    R0, STATUS                                      ;
                                         04        58 E8 0050F         BLBS    STATUS, 63$                                     ;
                                         50        58 D0 00512 62$:    MOVL    STATUS, R0                                      ; 1160
                                                      04 00515         RET                                                     ;
                       05       03       AB        03 E0 00516 63$:    BBS     #3, MOUNT_OPTIONS+3, 64$                        ; 1164
                       09       03       AB        04 E1 0051B         BBC     #4, MOUNT_OPTIONS+3, 65$                        ; 1165
                                54       AB        62 3C 00520 64$:    MOVZWL  (STR_DESCRIPTOR), ACP_STRING                    ; 1168
                                58       AB   04   A2 D0 00524         MOVL    4(STR_DESCRIPTOR), ACP_STRING+4                 ; 1169
                       08       07       AB        06 E1 00529 65$:    BBC     #6, MOUNT_OPTIONS+7, 66$                        ; 1178
                                                   6B 95 0052E         TSTB    MOUNT_OPTIONS                                   ; 1179
                                                   04 19 00530         BLSS    66$                                             ;
                                01       AB        01 88 00532         BISB2   #1, MOUNT_OPTIONS+1                             ; 1181
                       0B                6B        06 E0 00536 66$:    BBS     #6, MOUNT_OPTIONS, 67$                          ; 1187
                                                   6B 95 0053A         TSTB    MOUNT_OPTIONS                                   ; 1188
                                                   07 19 0053C         BLSS    67$                                             ;
                                         03   01   AB E8 0053E         BLBS    MOUNT_OPTIONS+1, 67$                            ; 1189
                                         6B        10 88 00542         BISB2   #16, MOUNT_OPTIONS                              ; 1191
           52          6B                01        07 EF 00545 67$:    EXTZV   #7, #1, MOUNT_OPTIONS, R2                       ; 1197
                                         08        52 E9 0054A         BLBC    R2, 68$                                         ;
                       70                6B        06 E0 0054D         BBS     #6, MOUNT_OPTIONS, 75$                          ; 1198
                                         6C   01   AB E8 00551         BLBS    MOUNT_OPTIONS+1, 75$                            ;
           51     01   AB                01        00 EF 00555 68$:    EXTZV   #0, #1, MOUNT_OPTIONS+1, R1                     ; 1206
                                         07        51 E9 0055B         BLBC    R1, 69$                                         ;
                       5F                6B        06 E0 0055E         BBS     #6, MOUNT_OPTIONS, 75$                          ; 1207
                                         5C        52 E8 00562         BLBS    R2, 75$                                         ;
                       06                6B        06 E1 00565 69$:    BBC     #6, MOUNT_OPTIONS, 70$                          ; 1215
                                         55        51 E8 00569         BLBS    R1, 75$                                         ; 1216
                                         52        52 E8 0056C         BLBS    R2, 75$                                         ;
```

```
                 0A    02    AB          06  E1 0056F 70$:     BBC     #6, MOUNT_OPTIONS+2, 71$                     ; 1224
                             4A          51  E8 00574          BLBS    R1, 75$                                      ; 1225
                             47          52  E8 00577          BLBS    R2, 75$                                      ; 1226
                 43          6B          06  E0 0057A          BBS     #6, MOUNT_OPTIONS, 75$                       ; 1227
                 0F    03    AB          06  E1 0057E 71$:     BBC     #6, MOUNT_OPTIONS+3, 72$                     ; 1235
                                   03    AB  95 00583          TSTB    MOUNT_OPTIONS+3                              ; 1243
                                         0E  19 00586          BLSS    73$                                          ;
                 09    02    AB          06  E0 00588          BBS     #6, MOUNT_OPTIONS+2, 73$                     ; 1244
                 09    01    AB          03  E0 0058D          BBS     #3, MOUNT_OPTIONS+1, 74$                     ; 1245
                             50          14  D0 00592 72$:     MOVL    #20, R0                                      ; 1247
                                             04 00595          RET                                                  ;
                 2E    01    AB          03  E1 00596 73$:     BBC     #3, MOUNT_OPTIONS+1, 76$                     ; 1253
                 21    03    AB          01  E0 0059B 74$:     BBS     #1, MOUNT_OPTIONS+3, 75$                     ; 1254
                 1C    03    AB          02  E0 005A0          BBS     #2, MOUNT_OPTIONS+3, 75$                     ; 1255
                 17    03    AB          04  E0 005A5          BBS     #4, MOUNT_OPTIONS+3, 75$                     ; 1256
                 12    03    AB          03  E0 005AA          BBS     #3, MOUNT_OPTIONS+3, 75$                     ; 1257
                             0E    05    AB  E8 005AF          BLBS    MOUNT_OPTIONS+5, 75$                         ; 1258
                 09    05    AB          05  E0 005B3          BBS     #5, MOUNT_OPTIONS+5, 75$                     ; 1259
                             05    03    AB  E8 005B8          BLBS    MOUNT_OPTIONS+3, 75$                         ; 1260
                                   02    AB  95 005BC          TSTB    MOUNT_OPTIONS+2                              ; 1261
                                         08  18 005BF          BGEQ    76$                                          ;
                             50 0072802C 8F  D0 005C1 75$:     MOVL    #7503916, R0                                 ; 1264
                                             04 005C8          RET                                                  ;
                             50 00000000G 00 D0 005C9 76$:     MOVL    CTL$GL_PHD, PHD                              ; 1270
                             0A          52  E9 005D0          BLBC    R2, 77$                                      ; 1272
                 06          60          03  E0 005D3          BBS     #3, (PHD), 77$                               ; 1273
                             50    281C  8F  3C 005D7          MOVZWL  #10268, R0                                   ; 1275
                                             04 005DC          RET                                                  ;
                             0A          51  E9 005DD 77$:     BLBC    R1, 78$                                      ; 1277
                 06          60          02  E0 005E0          BBS     #2, (PHD), 78$                               ; 1278
                             50    2814  8F  3C 005E4          MOVZWL  #10260, R0                                   ; 1280
                                             04 005E9          RET                                                  ;
                 0A    03    AB          02  E0 005EA 78$:     BBS     #2, MOUNT_OPTIONS+3, 79$                     ; 1282
                 05    03    AB          03  E0 005EF          BBS     #3, MOUNT_OPTIONS+3, 79$                     ; 1283
                 18    03    AB          04  E1 005F4          BBC     #4, MOUNT_OPTIONS+3, 81$                     ; 1284
                 06          60          12  E0 005F9 79$:     BBS     #18, (PHD), 80$                              ; 1285
                             50    2894  8F  3C 005FD          MOVZWL  #10388, R0                                   ; 1287
                                             04 00602          RET                                                  ;
                 09    03    AB          04  E1 00603 80$:     BBC     #4, MOUNT_OPTIONS+3, 81$                     ; 1289
                             06          60  E8 00608          BLBS    (PHD), 81$                                   ; 1290
                             50    2804  8F  3C 0060B          MOVZWL  #10244, R0                                   ; 1292
                                             04 00610          RET                                                  ;
                             50          01  D0 00611 81$:     MOVL    #1, R0                                       ; 1302
                                             04 00614          RET                                                  ;

; Routine Size:  1557 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE COPY_ITEM (ITEM, DEST_SIZE, DEST_ADDR) =
1
1 !++
1 ! Functional description:
1 !
1 !       Given an item descriptor block, this routine will copy the
1 !       data described by the descriptor block to an internal storage
1 !       area.  The item descriptor block is assumed to already have been
1 !       probed for read access, and this routine is responsible for
1 !       probing the actual data for read access before copying it.
1 !
1 ! Input:
1 !
1 !       ITEM            : Address of an ITEM descriptor block.
1 !       DEST_SIZE       : The size, measured in bytes, of the internal storage area.
1 !       DEST_ADDR       : The address of the internal storage area.
1 !
1 ! Implicit Input:
1 !
1 !       None.
1 !
1 ! Output:
1 !
1 !       None.
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       SS$_ACCVIO      : The data item could not be read accessed.
1 !       SS$_BADPARAM    : The data item had a zero length.
1 !       SS$_BUFFEROVF   : The internal storage area was not large enough
1 !                         to contain the entire data item.
1 !       SS$_NORMAL      : The data was successfully copied.
1 !--
1
2 BEGIN                                         ! Start of COPY_ITEM
2
2 MAP
2       ITEM            : REF BBLOCK;           ! Item descriptor block
2
2 BUILTIN
2       PROBER;                                 ! MACRO-32 PROBE instruction
2
2 LOCAL
2       LOCAL_LENGTH,
2       LOCAL_ADDRESS;
2
2 !
2 ! Copy the item length and address to internal storage.
2 ! Reject items with a length of zero or length greater than 512.
```

```
;  833          1360  2 !
;  834          1361  2 LOCAL_ADDRESS = .ITEM [ADDRESS];
;  835          1362  2 LOCAL_LENGTH = .ITEM [LENGTH];
;  836          1363  3 IF (.LOCAL_LENGTH LEQ 0) OR (.LOCAL_LENGTH GTR 512)
;  837          1364  2 THEN
;  838          1365  2     RETURN (SS$_BADPARAM);
;  839          1366  2
;  840          1367  2 !
;  841          1368  2 ! Probe the data area for read access.
;  842          1369  2 !
;  843          1370  2 IF NOT PROBER (%REF (0), LOCAL_LENGTH, .LOCAL_ADDRESS)
;  844          1371  2 THEN
;  845          1372  2     RETURN (SS$_ACCVIO);
;  846          1373  2
;  847          1374  2 !
;  848          1375  2 ! Copy the data to the internal storage area. If the data item
;  849          1376  2 ! is too large it will be truncated, if it is too small, it will
;  850          1377  2 ! be zero-padded to fill the internal storge area.
;  851          1378  2 !
;  852          1379  2 CH$COPY (.LOCAL_LENGTH, .LOCAL_ADDRESS, 0, .DEST_SIZE, .DEST_ADDR);
;  853          1380  2
;  854          1381  2 !
;  855          1382  2 ! If the data item was too long to be copied in its entirety,
;  856          1383  2 ! return a status code to indicate such.
;  857          1384  2 !
;  858          1385  2 IF .LOCAL_LENGTH GTR .DEST_SIZE
;  859          1386  2 THEN
;  860          1387  2     RETURN (SS$_BUFFEROVF);
;  861          1388  2
;  862          1389  2 SS$_NORMAL                                        ! Normal exit value
;  863          1390  2
;  864          1391  1 END;                                              ! End of COPY_ITEM



                                                  007C 00000 COPY_ITEM:
                                                                  .WORD   Save R2,R3,R4,R5,R6                          ; 1303
                                 50        04  AC  D0 00002       MOVL    ITEM, R0                                     ; 1361
                                 51        04  A0  D0 00006       MOVL    4(R0), LOCAL_ADDRESS
                                 56            60  3C 0000A       MOVZWL  (R0), LOCAL_LENGTH                           ; 1362
                                               09  15 0000D       BLEQ    1$                                           ; 1363
                     00000200    8F            56  D1 0000F       CMPL    LOCAL_LENGTH, #512                           ;
                                               04  15 00016       BLEQ    2$                                           ;
                                 50            14  D0 00018 1$:   MOVL    #20, R0                                      ; 1365
                                                   04 0001B       RET                                                  ;
                      61         56            00  0C 0001C 2$:   PROBER  #0, LOCAL_LENGTH, (LOCAL_ADDRESS)            ; 1370
                                               04  12 00020       BNEQ    3$                                           ;
                                 50            0C  D0 00022       MOVL    #12, R0                                      ; 1372
                                                   04 00025       RET                                                  ;
      08    AC        00         61            56  2C 00026 3$:   MOVC5   LOCAL_LENGTH, (LOCAL_ADDRESS), #0, -         ; 1379
                                           0C  BC     0002C               DEST_SIZE, @DEST_ADDR                        ;
                            08   AC            56  D1 0002E       CMPL    LOCAL_LENGTH, DEST_SIZE                      ; 1385
                                               06  15 00032       BLEQ    4$                                           ;
                                 50      0601  8F  3C 00034       MOVZWL  #1537, R0                                    ; 1387
                                                   04 00039       RET                                                  ;
```

```
                                50            01  D0 0003A 4$:      MOVL    #1, R0                                 ; 1391
                                              04 0003D              RET                                            ;

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 0615
```

```
  1 !++
  1 ! What follows are the TPARSE table definitions.
  1 !--
  1
  1 !
  1 ! Parse /PROCESSOR options, set bits and store name.
  1 !
  1 $INIT_STATE (ACP_STB, ACP_KTB);
  1
P 1 $STATE   (,
P 1           ('UNIQUE',,,     1^OPT_UNIQUEACP, MOUNT_OPTIONS),
P 1           ((DEVICENAME),,,1^OPT_SAMEACP,    MOUNT_OPTIONS),
P 1           ((FILENAME),,,   1^OPT_FILEACP,    MOUNT_OPTIONS)
  1           );
  1
P 1 $STATE   (,
P 1           (TPA$_EOS, TPA$_EXIT)
  1           );
  1 !
  1 ! Syntax definition for a file name.
  1 !
P 1 $STATE   (FILENAME,
P 1           (TPA$_SYMBOL, FILENAME),
P 1           ('.', FILENAME),
P 1           (';', FILENAME),
P 1           (TPA$_LAMBDA, TPA$_EXIT)
  1           );
  1 !
  1 ! Syntax definition for a device name.
  1 !
P 1 $STATE   (DEVICENAME,
P 1           (TPA$_SYMBOL)
  1           );
  1
P 1 $STATE   (,
P 1           (':')
  1           );
  1
P 1 $STATE   (,
P 1           (TPA$_EOS, TPA$_EXIT)
  1           );
  1 END
  0 ELUDOM


                                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                                   00000 ;TPA$KEYST0
                                                         U.2:      .BLKB    0
                        45  55  51  49  4E  55  00000 ;TPA$KEYST
                                                         U.4:      .ASCII   \UNIQUE\                              ;
                                                FF  00006          .BYTE    -1                                    ;
                                                FF  00007 ;TPA$KEYFILL
                                                         U.18:     .BYTE    -1                                    ;

                                                                .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1
```

```
                  00000 ACP_STB::
                                  .BLKB    0
          6100    00000 ;TPA$TYPE
                        U.5:      .WORD    24832                                         ;
     00000000*    00002 ;TPA$ADDR
                        U.6:      .LONG    <<MOUNT_OPTIONS-U.6>-4>                       ;
     04000000     00006 ;TPA$MASK
                        U.7:      .LONG    67108864                                      ;
          69F8    0000A ;TPA$TYPE
                        U.8:      .WORD    27128                                         ;
         0000*    0000C ;TPA$SUBEXP
                        U.10:     .WORD    <<U.9-U.10>-2>                                ;
     00000000*    0000E ;TPA$ADDR
                        U.11:     .LONG    <<MOUNT_OPTIONS-U.11>-4>                      ;
     08000000     00012 ;TPA$MASK
                        U.12:     .LONG    134217728                                     ;
          6DF8    00016 ;TPA$TYPE
                        U.13:     .WORD    28152                                         ;
         0000*    00018 ;TPA$SUBEXP
                        U.15:     .WORD    <<U.14-U.15>-2>                               ;
     00000000*    0001A ;TPA$ADDR
                        U.16:     .LONG    <<MOUNT_OPTIONS-U.16>-4>                      ;
     10000000     0001E ;TPA$MASK
                        U.17:     .LONG    268435456                                     ;
          15F7    00022 ;TPA$TYPE
                        U.19:     .WORD    5623                                          ;
          FFFF    00024 ;TPA$TARGET
                        U.20:     .WORD    -1                                            ;
                  00026 ;FILENAME
                        U.14:     .BLKB    0
          11F1    00026 ;TPA$TYPE
                        U.21:     .WORD    4593                                          ;
         0000*    00028 ;TPA$TARGET
                        U.22:     .WORD    <<U.14-U.22>-2>                               ;
          102E    0002A ;TPA$TYPE
                        U.23:     .WORD    4142                                          ;
         0000*    0002C ;TPA$TARGET
                        U.24:     .WORD    <<U.14-U.24>-2>                               ;
          103B    0002E ;TPA$TYPE
                        U.25:     .WORD    4155                                          ;
         0000*    00030 ;TPA$TARGET
                        U.26:     .WORD    <<U.14-U.26>-2>                               ;
          15F6    00032 ;TPA$TYPE
                        U.27:     .WORD    5622                                          ;
          FFFF    00034 ;TPA$TARGET
                        U.28:     .WORD    -1                                            ;
                  00036 ;DEVICENAME
                        U.9:      .BLKB    0
          05F1    00036 ;TPA$TYPE
                        U.29:     .WORD    1521                                          ;
          043A    00038 ;TPA$TYPE
                        U.30:     .WORD    1082                                          ;
          15F7    0003A ;TPA$TYPE
                        U.31:     .WORD    5623                                          ;
          FFFF    0003C ;TPA$TARGET
                        U.32:     .WORD    -1                                            ;
```

```
                                                        .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

                                           00000 ACP_KTB::
                                                        .BLKB   0
                                           00000 ;TPA$KEY0
                                                 U.1:   .BLKB   0
                                     0000* 00000 ;TPA$KEY
                                                 U.3:   .WORD   <U.2-U.1>                                          ;



;                             PSECT SUMMARY
;
;    Name                      Bytes                    Attributes
;
;   $GLOBAL$                      444  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                        2324  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       1619  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   _LIB$KEY0$                      2  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   _LIB$STATE$                    62  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   _LIB$KEY1$                      8  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)



;                       Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;    File                                  Total    Loaded   Percent      Mapped      Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1          18619        88         0       1000        00:02.0
;   _$255$DUA28:[SYSLIB]CLIMAC.L32;1          14         0         0          9        00:00.1
;   _$255$DUA28:[SYSLIB]TPAMAC.L32;1          42        26        61         14        00:00.2




;                             COMMAND QUALIFIERS

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

; Size:          1619 code + 2840 data bytes
; Run Time:         00:42.0
; Elapsed Time:     01:23.6
; Lines/CPU Min:      2048
; Lexemes/CPU-Min: 30831
; Memory Used:  438 pages
; Compilation Complete
```

0245 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY