G 3

```
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN      NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN      NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN      NNN  GGG
EEEEEEEEEEEE              XXX     CCC              HHHHHHHHHHHHHHH  NNN   NNN   NNN  GGG
EEEEEEEEEEEE              XXX     CCC              HHHHHHHHHHHHHHH  NNN   NNN   NNN  GGG
EEEEEEEEEEEE              XXX     CCC              HHHHHHHHHHHHHHH  NNN   NNN   NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN      NNNNNN  GGG   GGGGGGGGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN      NNNNNN  GGG   GGGGGGGGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN      NNNNNN  GGG   GGGGGGGGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG         GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG         GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN         NNN  GGG         GGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGG
```

```
**FILE**ID**EXCMOUN                                   N 3

EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    000000    UU      UU  NN      NN
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    000000    UU      UU  NN      NN
EE          XX      XX  CC          MMMM  MMMM  00      00  UU      UU  NN      NN
EE          XX      XX  CC          MMMM  MMMM  00      00  UU      UU  NN      NN
EE            XX  XX    CC          MM  MM  MM  00      00  UU      UU  NNNN    NN
EE            XX  XX    CC          MM  MM  MM  00      00  UU      UU  NNNN    NN
EEEEEEEE        XX      CC          MM      MM  00      00  UU      UU  NN  NN  NN
EEEEEEEE        XX      CC          MM      MM  00      00  UU      UU  NN  NN  NN
EE            XX  XX    CC          MM      MM  00      00  UU      UU  NN    NNNN
EE            XX  XX    CC          MM      MM  00      00  UU      UU  NN    NNNN
EE          XX      XX  CC          MM      MM  00      00  UU      UU  NN      NN      ....
EE          XX      XX  CC          MM      MM  00      00  UU      UU  NN      NN      ....
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    000000    UUUUUUUUUU  NN      NN      ....
EEEEEEEEEE  XX      XX    CCCCCCCC  MM      MM    000000    UUUUUUUUUU  NN      NN      ....

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

```
0 MODULE  exch$moun                                 %TITLE 'MOUNT verb dispatch and misc routines'
0                  (
0                  IDENT = 'V04-000',
0                  ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
0                  ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:     EXCHANGE - Foreign volume interchange facility
1 !
1 ! ABSTRACT:     Primary action routines for MOUNT verb
1 !
1 ! ENVIRONMENT:  VAX/VMS User mode
1 !
1 ! AUTHOR:       CW Hobbs          CREATION DATE: 19-July-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-003 CWH3003         CW Hobbs                26-Jul-1984
1 !               Message chosen for CWH3002 needs an FAO parm, supply it.
1 !
1 !       V03-002 CWH3002         CW Hobbs                12-Apr-1984
1 !               Give explicit message for mount on remote link.  If a
1 !               TU58, attempt to read and write block 1 so that we can
1 !               sense whether the device is write-locked.  The TU58
1 !               driver does not report write-lock status until a
1 !               write is actually done.
1 !
1 !
1 !--
1
1 ! Include files:
1 !
1 MACRO $module_name_string = 'exch$moun' %;      ! The require file needs to know our module name
```

```
:   58         0058  1 REQUIRE 'SRC$:EXCREQ'                       ! Facility-wide require file
:   59         0059  1       ;
```

```
%SBTTL 'Module table of contents'

! Module table of contents:
!
FORWARD ROUTINE
    exch$moun_dismount,                              ! Main entry routine for DISMOUNT verb
    exch$moun_dismount_action,                       ! Action routine for DISMOUNT
        moun_foreign,                                ! Specific routine to do foreign device mounts
    exch$moun_implied_mount,                         ! Perform an automatic mount
        moun_init : NOVALUE,                         ! Setups common to Moun_implied_mount and Moun_mount
    exch$moun_mount,                                 ! Main action routine for MOUNT
        moun_virtual,                                ! Specific routine to do virtual device mounts
    exch$moun_vms_mount                              ! Call the $MOUNT service to do an implied $ MOUNT /FOREIGN
    ;

! EXCHANGE facility routines
!
EXTERNAL ROUTINE
    exch$cmd_namb_clone,                             ! Make a duplicate of a namb
    exch$cmd_parse_filespec,                         ! Parse a file specification
    exch$cmd_unwind_cli_syntax,                      ! Unwind out of a CLI$PRESENT call if qualifier not allowed
    exch$dos11_mount,                                ! DOS-11 volume mount processing
    exch$io_rt11_read,                               ! Read a block from a foreign disk
    exch$io_rt11_write,                              ! Write a block to a foreign disk
    exch$rt11_dircache_stop     : NOVALUE,           ! Flush RT-11 caches during dismount
    exch$rt11_mount,                                 ! RT-11 volume mount processing
    exch$util_file_error,                            ! Signal RMS error
    exch$util_namb_release      : NOVALUE,           ! Release name block
    exch$util_vm_allocate_zeroed,                    ! Allocate virtual memory
    exch$util_vol_getdvi,                            ! Get device information
    exch$util_volb_release      : NOVALUE,           ! Release volume block
    exch$util_volb_allocate                          ! Allocate volume block
    ;

! Equated symbols:
!
!LITERAL
!    ;

! Bound declarations:
!
BIND
    ascid_devicename = %ASCID 'DEVICENAME'
    ;
```

```
1 GLOBAL ROUTINE exch$moun_dismount =      %SBTTL 'exch$moun_dismount'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Entry routine for the dismount verb, parses and performs main control functions for dismount
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       Command parameters and qualifiers as returned from CLI$xxx routines.
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       device will be dismounted
2 !--
2
2 $dbgtrc_prefix ('moun_dismount> ');
2
2 LOCAL
2     namb        : $ref_bblock,                     ! Local pointer to a namb
2     volb        : $ref_bblock,                     ! Local pointer to a volb
2     status
2     ;
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock  ! pointer to our work area
2     ;
```

```
2 ! Allocate and/or initialize the work area
2 !
2 moun_init ();
2
2 ! Parse the device name parameter into a newly allocated $NAMB, there are no defaults
2 !
2 status = exch$cmd_parse_filespec (ascid_devicename, 0, 0, moun [moun$q_device], namb);
2 moun [moun$a_namb] = .namb;                          ! Save it in the work area too
2 IF NOT .status
2 THEN
2     $exch_signal_return (exch$_parseerr, 1, moun [moun$q_device], .status);
2 IF NOT .namb [namb$v_explicit_device]
2 THEN
2     $exch_signal_return (exch$_nodevice, 1, moun [moun$q_device]);
2 IF .namb [namb$v_explicit_node]
2 THEN
2     $exch_signal_return (exch$_noremote, 1, moun [moun$q_device]);
2 IF    .namb [namb$v_explicit_directory] OR .namb [namb$v_explicit_name]
2   OR .namb [namb$v_explicit_type] OR .namb [namb$v_explicit_version]
2 THEN
2     $exch_signal (exch$_devonly, 1, moun [moun$q_device]);
2
2 ! Make sure that the volume is already mounted
2 !
3 IF (.namb [namb$a_assoc_volb] NEQ 0)
2 THEN
2     status = exch$moun_dismount_action (.namb [namb$a_assoc_volb])
2
2 ELSE
2
2     ! Signal and return the error
2     !
2     $exch_signal (exch$_notmounted, 1, namb [namb$q_device]);
2
2 ! Release the namb
2 !
2 exch$util_namb_release (.namb);
2
2 RETURN .status;
1 END;


                                                    .TITLE  EXCH$MOUN MOUNT verb dispatch and misc routines
                                                    .IDENT  \V04-000\

                                                    .PSECT  EXCH$MOUN_PLIT,NOWRT,2

00 00 45 4D 41 4E 45 43 49 56 45 44  00000 P.AAB:   .ASCII  \DEVICENAME\<0><0>                  ;
                            010E000A  0000C P.AAA:   .LONG   17694730                            ;
                           00000000'  00010          .ADDRESS P.AAB                              ;

                                           ASCID_DEVICENAME=    P.AAA
                                                    .EXTRN  EXCH$CMD_NAMB_CLONE
                                                    .EXTRN  EXCH$CMD_PARSE_FILESPEC
                                                    .EXTRN  EXCH$CMD_UNWIND_CLI_SYNTAX
                                                    .EXTRN  EXCH$DOST1_MOUNT
                                                    .EXTRN  EXCH$IO_RTT1_READ
```

```
                                                               .EXTRN   EXCH$IO_RT11_WRITE
                                                               .EXTRN   EXCH$RT11_DIRCACHE_STOP
                                                               .EXTRN   EXCH$RT11_MOUNT
                                                               .EXTRN   EXCH$UTIL_FILE_ERROR
                                                               .EXTRN   EXCH$UTIL_NAMB_RELEASE
                                                               .EXTRN   EXCH$UTIL_VM_ALLOCATE_ZEROED
                                                               .EXTRN   EXCH$UTIL_VOC_GETDVI
                                                               .EXTRN   EXCH$UTIL_VOLB_RELEASE
                                                               .EXTRN   EXCH$UTIL_VOLB_ALLOCATE
                                                               .EXTRN   EXCH$A_GBL, EXCH$_PARSEERR
                                                               .EXTRN   EXCH$_NODEVICE, EXCH$_NOREMOTE
                                                               .EXTRN   EXCH$_DEVONLY, EXCH$_NOTMOUNTED

                                                               .PSECT   EXCH$MOUN_CODE,NOWRT,2

                                        00FC 00000             .ENTRY   EXCH$MOUN_DISMOUNT, Save R2,R3,R4,R5,R6,R7  ; 0200
                      57 00000000G  00    9E 00002             MOVAB    LIB$SIGNAL, R7                              ;
                      5E              04  C2 00009             SUBL2    #4, SP                                      ;
       53 00000000G   EF              14  C1 0000C             ADDL3    #20, EXCH$A_GBL, R3                         ; 0242
             0000V    CF              00  FB 00014             CALLS    #0, MOUN_INIT                               ; 0246
                                      5E  DD 00019             PUSHL    SP                                          ; 0250
       55             63              0C  C1 0001B             ADDL3    #12, (R3), R5                               ;
                                      55  DD 0001F             PUSHL    R5                                          ;
                                      7E  7C 00021             CLRQ     -(SP)                                       ;
                           0000'  CF  9F 00023                 PUSHAB   ASCID_DEVICENAME                            ;
         00000000G    EF              05  FB 00027             CALLS    #5, EXCH$CMD_PARSE_FILESPEC                 ;
                      56              50  D0 0002E             MOVL     R0, STATUS                                  ;
                      52              6E  D0 00031             MOVL     NAMB, R2                                    ; 0251
            00        B3              52  D0 00034             MOVL     R2, @0(R3)                                  ;
                      15              56  E8 00038             BLBS     STATUS, 1$                                  ; 0252
                      53 00000000G    8F  D0 0003B             MOVL     #EXCH$_PARSEERR, TEMP                       ; 0254
                      7E              55  7D 00042             MOVQ     R5, -(SP)                                   ;
                                      01  DD 00045             PUSHL    #1                                          ;
                                      53  DD 00047             PUSHL    TEMP                                        ;
                      67              04  FB 00049             CALLS    #4, LIB$SIGNAL                              ;
                      50              53  D0 0004C             MOVL     TEMP, R0                                    ;
                                          04 0004F             RET                                                  ;
                      53         6C   A2  9E 00050  1$:        MOVAB    108(R2), R3                                 ; 0255
                                      63  95 00054             TSTB     (R3)                                        ;
                                      09  19 00056             BLSS     2$                                          ;
                      54 00000000G    8F  D0 00058             MOVL     #EXCH$_NODEVICE, TEMP                       ; 0257
                                      0B  11 0005F             BRB      3$                                          ;
       14             63              06  E1 00061  2$:        BBC      #6, (R3), 4$                                ; 0258
                      54 00000000G    8F  D0 00065             MOVL     #EXCH$_NOREMOTE, TEMP                       ; 0260
                                      55  DD 0006C  3$:        PUSHL    R5                                          ;
                                      01  DD 0006E             PUSHL    #1                                          ;
                                      54  DD 00070             PUSHL    TEMP                                        ;
                      67              03  FB 00072             CALLS    #3, LIB$SIGNAL                              ;
                      50              54  D0 00075             MOVL     TEMP, R0                                    ;
                                          04 00078             RET                                                  ;
                      0C         01   A3  E8 00079  4$:        BLBS     1(R3), 5$                                   ; 0261
       08             63              09  E0 0007D             BBS      #9, (R3), 5$                                ;
       04             63              0A  E0 00081             BBS      #10, (R3), 5$                               ; 0262
       0D             63              0B  E1 00085             BBC      #11, (R3), 6$                               ;
                                      55  DD 00089  5$:        PUSHL    R5                                          ; 0264
                                      01  DD 0008B             PUSHL    #1                                          ;
                         00000000G    8F  DD 0008D             PUSHL    #EXCH$_DEVONLY                              ;
                         67          03  FB 00093          CALLS   #3, LIB$SIGNAL
                                74   A2  D5 00096 6$:      TSTL    116(R2)                               ; 0268
                                     0D  13 00099          BEQL    7$                                    ;
                                74   A2  DD 0009B          PUSHL   116(R2)                               ; 0270
                    0000V  CF        01  FB 0009E          CALLS   #1, EXCH$MOUN_DISMOUNT_ACTION         ;
                           56        50  D0 000A3          MOVL    R0, STATUS                            ;
                                     0E  11 000A6          BRB     8$                                    ;
                                40   A2  9F 000A8 7$:      PUSHAB  64(R2)                                ; 0276
                                     01  DD 000AB          PUSHL   #1                                    ;
                            00000000G  8F  DD 000AD        PUSHL   #EXCH$_NOTMOUNTED                     ;
                           67        03  FB 000B3          CALLS   #3, LIB$SIGNAL                        ;
                                     52  DD 000B6 8$:      PUSHL   R2                                    ; 0280
                00000000G  EF        01  FB 000B8          CALLS   #1, EXCH$UTIL_NAMB_RELEASE            ;
                           50        56  D0 000BF          MOVL    STATUS, R0                            ; 0282
                                         04 000C2          RET                                           ; 0283

; Routine Size:  195 bytes,    Routine Base:  EXCH$MOUN_CODE + 0000
```

```
1 GLOBAL ROUTINE exch$moun_dismount_action (volb : $ref_bblock)  =          %SBTTL 'exch$moun_dismount_action (v
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Action routine for the dismount verb, cleans up and closes the volb
2 !
2 ! INPUTS:
2 !
2 !       volb - pointer to volume to be released
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       file will be closed, volb returned to internal tables
2 !--
2
2 $dbgtrc_prefix ('moun_dismount_action> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock, ! pointer to our work area
2     fab = volb [volb$a_fab] : $ref_bblock,
2     rab = volb [volb$a_rab] : $ref_bblock
2     ;
2
2 ! If we have global caching for an RT-11 volume, flush the cache
2 !
2 IF .exch$a_gbl [excg$v_q_cache]
2 THEN
2     IF .volb [volb$b_vol_format] EQL volb$k_vfmt_rt11
2     THEN
3         BEGIN
3
3         ! If we are in an exit handler, it is possible that I/O is active (likely if the device is a TU58),
3         ! so wait for it to complete.
3         !
4         IF NOT (status = $wait (rab = .rab))
```

```
;  249          0341  3             THEN
;  250          0342  3                 exch$util_file_error (exch$_waiterr, .status, .fab, .rab [rab$l_stv]);
;  251          0343  3
;  252          0344  3             exch$a_gbl [excg$v_q_cache] = false;     ! Kill global caching, otherwise nothing will happen
;  253          0345  3             exch$rtl1_dircache_stop (.volb);         ! Flush the caches
;  254          0346  3             exch$a_gbl [excg$v_q_cache] = true;      ! Reenable global caching
;  255          0347  2             END;
;  256          0348  2
;  257          0349  2 ! Close the RMS file associated with the device
;  258          0350  2 !
;  259          0351  2 $trace_print_fao ('closing, fab=!XL', .volb [volb$a_fab]);
;  260          0352  3 IF NOT (status = $close (FAB = .volb [volb$a_fab]))
;  261          0353  2 THEN
;  262          0354  2     $exch_signal_stop (.status);
;  263          0355  2
;  264          0356  2 ! Tell them it is gone unless we are dismounting from the exit handler.  Note that the MOUN work area is not
;  265          0357  2 ! valid if we are in an exit handler.
;  266          0358  2 !
;  267          0359  2 IF NOT .exch$a_gbl [excg$v_exiting]
;  268          0360  2 THEN
;  269          0361  2     IF .moun [moun$v_q_message]
;  270          0362  2     THEN
;  271        P 0363  2         $exch_signal (exch$_dismounted, 4, .volb [volb$l_vol_type_len], volb [volb$t_vol_type],
;  272          0364  2                                         .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
;  273          0365  2
;  274          0366  2 ! Release the $VOLB
;  275          0367  2 !
;  276          0368  2 exch$util_volb_release (.volb);
;  277          0369  2
;  278          0370  2 RETURN .status;
;  279          0371  1 END;


                                                              .EXTRN   SYS$WAIT, EXCH$_WAITERR
                                                              .EXTRN   SYS$CLOSE, LIB$STOP
                                                              .EXTRN   EXCH$_DISMOUNTED

                                              007C 00000      .ENTRY   EXCH$MOUN_DISMOUNT_ACTION, Save R2,R3,R4,-   ; 0284
                                                                       R5,R6                                        ;
                     56 00000000G    EF   9E 00002            MOVAB    EXCH$A_GBL, R6                               ;
        55           66              14   C1 00009            ADDL3    #20, EXCH$A_GBL, R5                          ; 0324
                     52          04  AC   D0 0000D            MOVL     VOLB, R2                                     ; 0325
        3F       00  B6              01   E1 00011            BBC      #1, @EXCH$A_GBL, 2$                          ; 0331
                     03          58  A2   91 00016            CMPB     88(R2), #3                                   ; 0333
                                     39   12 0001A            BNEQ     2$                                           ;
                     53          14  A2   D0 0001C            MOVL     20(R2), R3                                   ; 0340
                                     53   DD 00020            PUSHL    R3                                           ;
          00000000G  00              01   FB 00022            CALLS    #1, SYS$WAIT                                 ;
                     54              50   D0 00029            MOVL     R0, STATUS                                   ;
                     15              54   E8 0002C            BLBS     STATUS, 1$                                   ;
                                 0C  A3   DD 0002F            PUSHL    12(R3)                                       ; 0342
                                 10  A2   DD 00032            PUSHL    16(R2)                                       ;
                                     54   DD 00035            PUSHL    STATUS                                       ;
                        00000000G    8F   DD 00037            PUSHL    #EXCH$_WAITERR                               ;
          00000000G  EF              04   FB 0003D            CALLS    #4, EXCH$UTIL_FILE_ERROR                     ;
                 00  B6              02   8A 00044 1$:        BICB2    #2, @EXCH$A_GBL                              ; 0344
```

```
                                 52  DD 00048         PUSHL   R2                                          ; 0345
         00000000G  EF           01  FB 0004A         CALLS   #1, EXCH$RT11_DIRCACHE_STOP                 ;
                00  B6           02  88 00051         BISB2   #2, @EXCH$A_GBL                             ; 0346
                         10      A2  DD 00055 2$:     PUSHL   16(R2)                                      ; 0352
         00000000G  00           01  FB 00058         CALLS   #1, SYS$CLOSE                               ;
                    54           50  D0 0005F         MOVL    R0, STATUS                                  ;
                    0A           54  E8 00062         BLBS    STATUS, 3$                                  ;
                                 54  DD 00065         PUSHL   STATUS                                      ; 0354
         00000000G  00           01  FB 00067         CALLS   #1, LIB$STOP                                ;
                                 04  0006E            RET                                                 ;
      23        00  B6           04  E0 0006F 3$:     BBS     #4, @EXCH$A_GBL, 4$                         ; 0359
                    50           65  D0 00074         MOVL    (R5), R0                                    ; 0361
      1B        20  A0           04  E1 00077         BBC     #4, 32(R0), 4$                              ;
                         69      A2  9F 0007C         PUSHAB  105(R2)                                     ; 0364
                         65      A2  DD 0007F         PUSHL   101(R2)                                     ;
                         5D      A2  9F 00082         PUSHAB  93(R2)                                      ;
                         59      A2  DD 00085         PUSHL   89(R2)                                      ;
                                 04  DD 00088         PUSHL   #4                                          ;
                   00000000G     8F  DD 0008A         PUSHL   #EXCH$_DISMOUNTED                           ;
         00000000G  00           06  FB 00090         CALLS   #6, LIB$SIGNAL                              ;
                                 52  DD 00097 4$:     PUSHL   R2                                          ; 0368
         00000000G  EF           01  FB 00099         CALLS   #1, EXCH$UTIL_VOLB_RELEASE                  ;
                    50           54  D0 000A0         MOVL    STATUS, R0                                  ; 0370
                                 04  000A3            RET                                                 ; 0371

; Routine Size:  164 bytes,    Routine Base:  EXCH$MOUN_CODE + 00C3
```

```
1 GLOBAL ROUTINE moun_foreign =   %SBTTL 'moun_foreign'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Make a foreign volume known to EXCHANGE.  If the device is not mounted in the VMS sense, then mount
2 !       foreign with the $mount service.
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       namb - name block describing the device
2 !       write - flag saying if we allow writes
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       volb - volume block which will describe the mounted volume
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       lots
2 !--
2
2 $dbgtrc_prefix ('moun_foreign> ');
2
2 LOCAL
2     ptr : $ref_bblock,                           ! Pointer to scan along the queue
2     status
2
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock, ! pointer to our work area
2     namb = .moun [moun$a_namb] : $bblock,              ! Pointer to exchange NAMB structure
2     volb = .moun [moun$a_volb] : $bblock,              ! Pointer to exchange VOLB structure
2     fab  = .volb [volb$a_fab] : $bblock,               ! File Access Block for the volume
2     rab  = .volb [volb$a_rab] : $bblock,               ! Record Access Block for the volume
2     nam  = .volb [volb$a_nam] : $bblock,               ! RMS name block for the volume
2     dev_desc = namb [namb$q_device] : $desc_block      ! Pointer to the device name
2     ;
2
2 $block_check (2, .moun, moun, 433);
2 $block_check (2, namb, namb, 434);
2 $block_check (2, volb, volb, 435);
2
2 ! Get the device information
```

```
2 !
3 IF NOT (status = exch$util_vol_getdvi (dev_desc, volb))
2 THEN
3     BEGIN
3     $exch_signal (exch$_accessfail, 1, dev_desc, .status);
3     RETURN .status;
2     END;
2
2 ! Check to make sure that this physical device is not already mounted as a foreign.  This could
2 ! happen if we are dealing with hidden device names.
2 !
2 ptr = .exch$a_gbl [excg$a_volb_use_flink];      ! Start at the front of the queue
2 WHILE .ptr NEQA exch$a_gbl [excg$q_volb_use]    ! And work to the end
2 DO
3     BEGIN
3
3     $block_check (2, .ptr, volb, 540);          ! If these aren't volbs we are in deep trouble
3
3     IF .ptr NEQ volb                            ! Our volb is already in queue, so ignore it
3     THEN
3
3         ! If the physical names match, then this device has already been mounted via a concealed device name
3         !
3         IF CH$EQL (.ptr [volb$l_devnamlen], ptr [volb$t_devnam], .volb [volb$l_devnamlen], volb [volb$t_devn
3         THEN
4             BEGIN
4             LOCAL
4                 desc : VECTOR [2, LONG];
4             desc [0] = .volb [volb$l_devnamlen];
4             desc [1] = volb [volb$t_devnam];
4             $exch_signal_return (exch$_volmount, 1, desc);
3             END;
3
3     ptr = .ptr [volb$a_flink];                  ! Advance to next volb in the in-use queue
2     END;
2
2 ! Look at the device characteristics and make some decisions
2 !
3 BEGIN   ! scope "devbits"
3     BIND
3         devbits = volb [volb$l_devchar] : $bblock;
3     REGISTER
3         must_have, cannot_have;                 ! masks for device tests
3
3     ! We need to make sure that the thing is at least similar to a disk or tape.  First define masks for all
3     ! required and all prohibited device characteristics
3     !
3     IF .devbits [dev$v_rnd]
3     THEN
4         BEGIN                                   ! bits for "disks"
4         must_have = (dev$m_rnd OR dev$m_fod OR dev$m_shr OR dev$m_avl OR dev$m_idv OR dev$m_odv OR dev$m_dir
5         cannot_have = (dev$m_rec OR dev$m_ccl OR dev$m_trm OR dev$m_sdi OR dev$m_sqd OR dev$m_spl OR dev$m_o
4                     OR dev$m_net OR dev$m_gen OR dev$m_mbx OR dev$m_dmt OR dev$m_rtm);
4         END
3     ELSE
4         BEGIN                                   ! bits for "tapes"
4         must_have = (dev$m_sqd OR dev$m_fod OR dev$m_avl OR dev$m_idv OR dev$m_odv);
```

```
  5         cannot_have = (dev$m_ccl OR dev$m_trm OR dev$m_spl OR dev$m_opr
  4                        OR dev$m_net OR dev$m_gen OR dev$m_mbx OR dev$m_dmt OR dev$m_rtm);
  3         END;
  3
  3     ! If we are missing any "must_have" items or if we have any "cannot_have" items, scream and shout
  3     !
  4     IF  (((.volb [volb$l_devchar] XOR .must_have) AND .must_have) NEQ 0)
  3       OR
  4         ((.volb [volb$l_devchar] AND .cannot_have) NEQ 0)
  3     THEN
  3         $exch_signal_return (exch$_devnotsuit, 1, dev_desc);
  3
  3     ! If the device is not mounted in the VMS sense, then we must do that
  3     ! and recursively call ourself
  3     !
  3     IF NOT .devbits [dev$v_mnt]
  3     THEN
  4         BEGIN
  4         IF NOT exch$moun_vms_mount (volb, dev_desc)
  4         THEN
  4             RETURN false;
  4         RETURN moun_foreign ();
  3         END;
  3
  3     ! The device must be mounted foreign
  3     !
  3     IF NOT .devbits [dev$v_for]                     ! If the volume is write-locked
  3     THEN
  3         $exch_signal_return (exch$_opnotperf11, 1, namb [namb$q_device]);
  3
  2 END;    : scope "devbits"
  2
  2 ! Now set the unique ident field of this volb
  2 !
P 2 $trace_print_fao ('volb devnam "!AF", namb device "!AF", namb volid "!AF", concealed !UL',
P 2         .volb [volb$l_devnamlen], volb [volb$t_devnam],
P 2         (BIND ndev = namb [namb$q_device] : $desc_block; .ndev [dsc$w_length]),
P 2         (BIND ndev = namb [namb$q_device] : $desc_block; .ndev [dsc$a_pointer]),
P 2         .namb [namb$l_vol_ident_len], namb [namb$t_vol_ident],
  2         .namb [namb$v_concealed_device]);
  2 CH$MOVE (volb$s_vol_ident, namb [namb$t_vol_ident], volb [volb$t_vol_ident]);
  2 volb [volb$l_vol_ident_len] = .namb [namb$l_vol_ident_len];
  2
L 2 %IF switch_debug                                    ! Debugging trace code
U 2 %THEN
U 2     BEGIN
U 2     LOCAL
U 2         tmp_desc : $desc_block;
U 2     $stat_str_desc_init (tmp_desc, .volb [volb$l_devnamlen], volb [volb$t_devnam]);
U 2     $trace_print_fao ('Getdvi for name "!AS" resolved to device "!AS"', dev_desc, tmp_desc);
U 2     END;
  2 %FI
  2
  2 ! Init the RMS blocks for the volume
  2 !
P 2 $fab_init (
P 2         FAB = fab,                              ! Volume FAB
:   452          P 0543  2             FAC = (BIO,GET),                       !  Block I/O, read-only
:   453          P 0544  2             FNA = volb [volb$t_vol_ident],         !  Set name addr
:   454          P 0545  2             FNS = .volb [volb$l_vol_ident_len],    !  Set name size
:   455          P 0546  2             FOP = NFS,                             !  Non-File Structured
:   456            0547  2             NAM = nam);                            !  Name block
:   457          P 0548  2 $rab_init (
:   458          P 0549  2             RAB = rab,                             ! Volume RAB
:   459          P 0550  2             ROP = BIO,                             !  Block I/O
:   460            0551  2             FAB = fab);                            !  FAB addr
:   461          P 0552  2 $nam_init (
:   462          P 0553  2             NAM = nam,                             ! File name block
:   463          P 0554  2             RSA = .volb [volb$a_rsbuf],            !  Result name addr
:   464          P 0555  2             RSS = nam$c_maxrss,                    !  Result name size
:   465          P 0556  2             ESA = .volb [volb$a_esbuf],            !  Expanded name addr
:   466            0557  2             ESS = nam$c_maxrss);                   !  Expanded name size
:   467            0558  2
:   468            0559  2 ! Make any adjustments to the RMS blocks as necessary for dynamic conditions
:   469            0560  2 !
:   470            0561  2 fab [fab$v_put] = .moun [moun$l_q_write];
:   471            0562  2
:   472            0563  2 ! Open and connect to the volume
:   473            0564  2 !
:   474            0565  2 $trace_print_fao ('opening, fab=!XL', fab);
:   475            0566  3 IF NOT (status = $open (fab = fab))
:   476            0567  2 THEN
:   477            0568  2     RETURN exch$util_file_error (exch$_openforeign, .status, fab, .fab [fab$l_stv]);
:   478            0569  2 $check_call (4, exch$dbg_fab_dump, fab);
:   479            0570  2
:   480            0571  2 volb [volb$w_channel] = .fab [fab$l_stv];       ! Save the channel number (NFS ==> user mode channel)
:   481            0572  2
:   482            0573  3 IF NOT (status = $connect (rab = rab))
:   483            0574  2 THEN
:   484            0575  2     RETURN exch$util_file_error (exch$_openforeign, .status, fab, .rab [rab$l_stv]);
:   485            0576  2
:   486            0577  2 ! Fill in the rest of the state
:   487            0578  2 !
:   488            0579  2 volb [volb$v_connected] = true;
:   489            0580  2 volb [volb$v_foreign] = true;
:   490            0581  3 volb [volb$v_write] = (BIND devbits = fab [fab$l_dev] : $bblock;        ! Device can't be writelocked
:   491            0582  2                 (NOT .devbits [dev$v_swl]) AND .fab [fab$v_put]);       !  and the put bit must be set
:   492            0583  2
:   493            0584  2 ! The TU58 driver does not tell us if the cartridge is writelocked, therefore let us read and then
:   494            0585  2 ! attempt to write block 1 of the cartridge.
:   495            0586  2 !
:   496            0587  2 IF .volb [volb$v_write]
:   497            0588  2 THEN
:   498            0589  2     IF .volb [volb$l_devclass] EQL dc$_disk
:   499            0590  2       AND
:   500            0591  2        .volb [volb$l_devtype] EQL dt$_tu58
:   501            0592  2     THEN
:   502            0593  3         BEGIN
:   503            0594  3         LOCAL
:   504            0595  3             buf : $bvector [512];
:   505            0596  3         rab [rab$l_ubf] = buf;                     ! User buffer address
:   506            0597  3         rab [rab$w_usz] = 512;                     ! User buffer size
:   507            0598  3         rab [rab$l_bkt] = 1;                       ! LBN 1
:   508            0599  4         IF NOT (status = $read (rab = rab))
```

```
;  509        0600  3             THEN
;  510        0601  4                 BEGIN
;  511        0602  4                 LOCAL
;  512        0603  4                     tmp_desc : $desc_block;
;  513        0604  4                 $stat_str_desc_init (tmp_desc, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  514        0605  4                 $exch_signal_return (exch$_accessfail, 1, tmp_desc, .status, .rab [rab$l_stv]);
;  515        0606  3                 END;
;  516        0607  3             rab [rab$l_rbf] = buf;                  ! Record buffer address
;  517        0608  3             rab [rab$w_rsz] = 512;                  ! Record buffer size
;  518        0609  3             rab [rab$l_bkt] = 1;
;  519        0610  4             IF NOT (status = $write (rab = rab))
;  520        0611  3             THEN
;  521        0612  4                 BEGIN
;  522        0613  4                 $trace_print_fao ('status from TU-58 write check is !XL, stv !XL', .status, .rab [rab$l_stv]);
;  523        0614  4                 IF .status EQL rms$_wlk
;  524        0615  4                 THEN
;  525        0616  4                     volb [volb$v_write] = false
;  526        0617  4                 ELSE
;  527        0618  5                     BEGIN
;  528        0619  5                     LOCAL
;  529        0620  5                         tmp_desc : $desc_block;
;  530        0621  5                     $stat_str_desc_init (tmp_desc, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  531        0622  5                     $exch_signal_return (exch$_accessfail, 1, tmp_desc, .status, .rab [rab$l_stv]);
;  532        0623  4                     END;
;  533        0624  3                 END;
;  534        0625  2             END;
;  535        0626  2
;  536        0627  2 ! Set the volume format
;  537        0628  2 !
;  538        0629  2 volb [volb$b_vol_format] = .namb [namb$b_vol_format],
;  539        0630  2 volb [volb$v_vfmt_explicit] = .namb [namb$v_vfmt_explicit];
;  540        0631  2
;  541        0632  2 RETURN true;
;  542        0633  1 END;


                                                                  .EXTRN  EXCH$UTIL_BLOCK_CHECK
                                                                  .EXTRN  EXCH$_ACCESSFAIL
                                                                  .EXTRN  EXCH$_VOLMOUNT, EXCH$_DEVNOTSUIT
                                                                  .EXTRN  EXCH$_OPNOTPERF11
                                                                  .EXTRN  SYS$OPEN, EXCH$_OPENFOREIGN
                                                                  .EXTRN  SYS$CONNECT, SYS$READ
                                                                  .EXTRN  SYS$WRITE

                                                   0FFC 00000     .ENTRY  MOUN_FOREIGN, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  0372
                                                                          R10,R11                                    ;
                              5E      FDF4    CE   9E 00002       MOVAB   -524(SP), SP                               ;
             50 00000000G     EF        14    C1 00007            ADDL3   #20, EXCH$A_GBL, R0                        ;  0415
                              5B              60   D0 0000F       MOVL    (R0), R11                                  ;  0416
                              5A              6B   D0 00012       MOVL    (R11), R10                                 ;
                              56        04    AB   D0 00015       MOVL    4(R11), R6                                 ;  0417
                              58        10    A6   D0 00019       MOVL    16(R6), R8                                 ;  0418
                              57        14    A6   D0 0001D       MOVL    20(R6), R7                                 ;  0419
                              59        18    A6   D0 00021       MOVL    24(R6), R9                                 ;  0420
                              55        40    AA   9E 00025       MOVAB   64(R10), R5                                ;  0421
                              52 002400F8      8F   D0 00029      MOVL    #2359544, R2                               ;  0424
```

```
                                51      01B1  8F  3C 00030        MOVZWL  #433, R1                                   ;
                                50            5B  D0 00035        MOVL    R11, R0                                    ;
                                   00000000G  EF  16 00038        JSB     EXCH$UTIL_BLOCK_CHECK                      ;
                                52 010A00F7   8F  D0 0003E        MOVL    #17432823, R2                              ; 0425
                                51      01B2  8F  3C 00045        MOVZWL  #434, R1                                   ;
                                50            5A  D0 0004A        MOVL    R10, R0                                    ;
                                   00000000G  EF  16 0004D        JSB     EXCH$UTIL_BLOCK_CHECK                      ;
                                52 041B00F3   8F  D0 00053        MOVL    #68878579, R2                              ; 0426
                                51      01B3  8F  3C 0005A        MOVZWL  #435, R1                                   ;
                                50            56  D0 0005F        MOVL    R6, R0                                     ;
                                   00000000G  EF  16 00062        JSB     EXCH$UTIL_BLOCK_CHECK                      ;
                                7E            55  7D 00068        MOVQ    R5, -(SP)                                  ; 0430
                      00000000G EF            02  FB 0006B        CALLS   #2, EXCH$UTIL_VOL_GETDVI                   ;
                                6E            50  D0 00072        MOVL    R0, STATUS                                 ;
                                17            6E  E8 00075        BLBS    STATUS, 1$                                 ;
                                              6E  DD 00078        PUSHL   STATUS                                     ; 0433
                                              55  DD 0007A        PUSHL   R5                                         ;
                                              01  DD 0007C        PUSHL   #1                                         ;
                                   00000000G  8F  DD 0007E        PUSHL   #EXCH$_ACCESSFAIL                          ;
                      00000000G 00            04  FB 00084        CALLS   #4, LIB$SIGNAL                             ;
                                50            6E  D0 0008B        MOVL    STATUS, R0                                 ; 0434
                                                  04 0008E        RET                                                ;
                                50 00000000G  EF  D0 0008F  1$:   MOVL    EXCH$A_GBL, R0                             ; 0440
                                54      00C0  C0  D0 00096        MOVL    192(R0), PTR                               ;
                   50 00000000G EF 000000C0   8F  C1 0009B  2$:   ADDL3   #192, EXCH$A_GBL, R0                       ; 0441
                                50            54  D1 000A7        CMPL    PTR, R0                                    ;
                                44            13 000AA            BEQL    4$                                         ;
                                52 041B00F3   8F  D0 000AC        MOVL    #68878579, R2                              ; 0445
                                51      021C  8F  3C 000B3        MOVZWL  #540, R1                                   ;
                                50            54  D0 000B8        MOVL    PTR, R0                                    ;
                                   00000000G  EF  16 000BB        JSB     EXCH$UTIL_BLOCK_CHECK                      ;
                                56            54  D1 000C1        CMPL    PTR, R6                                    ; 0447
                                              25  13 000C4        BEQL    3$                                         ;
     38   A6           00    00E9 C4        38  A4  2D 000C6      CMPC5   56(PTR), 233(PTR), #0, 56(R6), 233(R6)     ; 0452
                                      00E9  C6     000CF                                                             ;
                                              17  12 000D2        BNEQ    3$                                         ;
                          F8    AD      38    A6  D0 000D4        MOVL    56(R6), DESC                               ; 0457
                          FC    AD    00E9    C6  9E 000D9        MOVAB   233(R6), DESC+4                            ; 0458
                                52 00000000G  8F  D0 000DF        MOVL    #EXCH$_VOLMOUNT, TEMP                      ; 0459
                                        F8    AD  9F 000E6        PUSHAB  DESC                                       ;
                                              68  11 000E9        BRB     12$                                        ;
                                54            64  D0 000EB  3$:   MOVL    (PTR), PTR                                 ; 0462
                                              AB  11 000EE        BRB     2$                                         ; 0441
                   10          2F A6          04  E1 000F0  4$:   BBC     #4, 47(R6), 5$                             ; 0476
                                50 1C054008   8F  D0 000F5        MOVL    #470106120, MUST_HAVE                      ; 0479
                                51 203220F7   8F  D0 000FC        MOVL    #540156151, CANNOT_HAVE                    ; 0480
                                              0E  11 00103        BRB     6$                                         ; 0476
                                50 0C044020   8F  D0 00105  5$:   MOVL    #201605152, MUST_HAVE                      ; 0485
                                51 203220C6   8F  D0 0010C        MOVL    #540156102, CANNOT_HAVE                    ; 0486
                   52          2C A6          50  CD 00113  6$:   XORL3   MUST_HAVE, 44(R6), R2                      ; 0492
                                50            52  D3 00118        BITL    R2, MUST_HAVE                              ;
                                06            12 0011B            BNEQ    7$                                         ;
                                51      2C    A6  D3 0011D        BITL    44(R6), CANNOT_HAVE                        ; 0494
                                09            13 00121            BEQL    8$                                         ;
                                52 00000000G  8F  D0 00123  7$:   MOVL    #EXCH$_DEVNOTSUIT, TEMP                    ; 0496
                                              25  11 0012A        BRB     11$                                        ;
                   15          2E A6          03  E0 0012C  8$:   BBS     #3, 46(R6), 10$                            ; 0501
```

```
                                                  55  DD 00131           PUSHL   R5                                          ; 0504
                                                  56  DD 00133           PUSHL   R6                                          ;
                              0000V  CF           02  FB 00135           CALLS   #2, EXCHSMOUN_VMS_MOUNT                     ;
                                     03           50  E8 0013A           BLBS    R0, 9$                                      ;
                                             0190 31  0013D              BRW     20$                                         ;
                              FEBB   CF           00  FB 00140 9$:       CALLS   #0, MOUN_FOREIGN                            ; 0507
                                                      04 00145           RET                                                 ;
                                     18       2F  A6  E8 00146 10$:      BLBS    47(R6), 13$                                 ; 0512
                                     52 00000000G 8F  D0 0014A           MOVL    #EXCH$_OPNOTPERF1', TEMP                    ; 0514
                                                  55  DD 00151 11$:      PUSHL   R5                                          ;
                                                  01  DD 00153 12$:      PUSHL   #1                                          ;
                                                  52  DD 00155           PUSHL   TEMP                                        ;
                          00000000G  00           03  FB 00157           CALLS   #3, LIB$SIGNAL                              ;
                                     50           52  D0 0015E           MOVL    TEMP, R0                                    ;
                                                      04 00161           RET                                                 ;
                69    A6      008A   CA      0080 8F  28 00162 13$:      MOVC3   #128, 138(R10), 105(R6)                     ; 0524
                                65   A6      0086 CA  D0 0016B           MOVL    134(R10), 101(R6)                           ; 0527
     0050   8F          00           6E           00  2C 00171           MOVC5   #0, (SP), #0, #80, (R8)                     ; 0547
                                                  68     00178                                                               ;
                                     68      5003 8F  B0 00179           MOVW    #20483, (R8)                                ;
                                04   A8 00010000  8F  D0 0017E           MOVL    #65536, 4(R8)                               ;
                                16   A8           22  90 00186           MOVB    #34, 22(R8)                                 ;
                                1F   A8           02  90 0018A           MOVB    #2, 31(R8)                                  ;
                                28   A8           59  D0 0018E           MOVL    R9, 40(R8)                                  ;
                                2C   A8        69 A6  9E 00192           MOVAB   105(R6), 44(R8)                             ;
                                34   A8        65 A6  90 00197           MOVB    101(R6), 52(R8)                             ;
     0044   8F          00           6E           00  2C 0019C           MOVC5   #0, (SP), #0, #68, (R7)                     ; 0551
                                                  67     001A3                                                               ;
                                     67      4401 8F  B0 001A4           MOVW    #17409, (R7)                                ;
                                04   A7      0800 8F  3C 001A9           MOVZWL  #2048, 4(R7)                                ;
                                3C   A7           58  D0 001AF           MOVL    R8, 60(R7)                                  ;
     0060   8F          00           6E           00  2C 001B3           MOVC5   #0, (SP), #0, #96, (R9)                     ; 0557
                                                  69     001BA                                                               ;
                                     69      6002 8F  B0 001BB           MOVW    #24578, (R9)                                ;
                                02   A9           01  8E 001C0           MNEGB   #1, 2(R9)                                   ;
                                04   A9        20 A6  D0 001C4           MOVL    32(R6), 4(R9)                               ;
                                0A   A9           01  8E 001C9           MNEGB   #1, 10(R9)                                  ;
                                0C   A9        1C A6  D0 001CD           MOVL    28(R6), 12(R9)                              ;
       16   A8          01           00        1C AB  F0 001D2           INSV    28(R11), #0, #1, 22(R8)                     ; 0561
                                                  58  DD 001D9           PUSHL   R8                                          ; 0566
                          00000000G  00           01  FB 001DB           CALLS   #1, SYS$OPEN                                ;
                                     6E           50  D0 001E2           MOVL    R0, STATUS                                  ;
                                     05           6E  E8 001E5           BLBS    STATUS, 14$                                 ;
                                               0C A8  DD 001E8           PUSHL   12(R8)                                      ; 0568
                                                  17  11 001EB           BRB     15$                                         ;
                                4A   A6        0C A8  B0 001ED 14$:      MOVW    12(R8), 74(R6)                              ; 0571
                                                  57  DD 001F2           PUSHL   R7                                          ; 0573
                          00000000G  00           01  FB 001F4           CALLS   #1, SYS$CONNECT                             ;
                                     6E           50  D0 001FB           MOVL    R0, STATUS                                  ;
                                     16           6E  E8 001FE           BLBS    STATUS, 16$                                 ;
                                               0C A7  DD 00201           PUSHL   12(R7)                                      ; 0575
                                                  58  DD 00204 15$:      PUSHL   R8                                          ;
                                               08 AE  DD 00206           PUSHL   STATUS                                      ;
                                        00000000G 8F  DD 00209           PUSHL   #EXCH$_OPENFOREIGN                          ;
                          00000000G  EF           04  FB 0020F           CALLS   #4, EXCH$UTIL_FILE_ERROR                    ;
                                                      04 00216           RET                                                 ;
                                     52        48 A6  9E 00217 16$:      MOVAB   72(R6), R2                                  ; 0579
                                62         09 88 0021B          BISB2   #9, (R2)                                          ; 0580
      50      16   A8           01         00 EF 0021E          EXTZV   #0, #1, 22(R8), R0                                ; 0582
      51      43   A8           01         01 EF 00224          EXTZV   #1, #1, 67(R8), R1                                ;
                                50         51 8A 0022A          BICB2   R1, R0                                            ;
      62           01           05         50 F0 0022D          INSV    R0, #5, #1, (R2)                                  ;
                   54           62         05 E1 00232          BBC     #5, (R2), 17$                                     ; 0587
                                01    30   A6 D1 00236          CMPL    48(R6), #1                                        ; 0589
                                           7F 12 0023A          BNEQ    19$                                               ;
                                0E    3C   A6 D1 0023C          CMPL    60(R6), #14                                       ; 0591
                                           79 12 00240          BNEQ    19$                                               ;
                          24    A7    0C   AE 9E 00242          MOVAB   BUF, 36(R7)                                       ; 0596
                          20    A7  0200   8F B0 00247          MOVW    #512, 32(R7)                                      ; 0597
                          38    A7         01 D0 0024D          MOVL    #1, 56(R7)                                        ; 0598
                                           57 DD 00251          PUSHL   R7                                                ; 0599
                   00000000G    00         01 FB 00253          CALLS   #1, SYS$READ                                      ;
                                6E         50 D0 0025A          MOVL    R0, STATUS                                        ;
                                2C         6E E9 0025D          BLBC    STATUS, 18$                                       ;
                          28    A7    0C   AE 9E 00260          MOVAB   BUF, 40(R7)                                       ; 0607
                          22    A7  0200   8F B0 00265          MOVW    #512, 34(R7)                                      ; 0608
                          38    A7         01 D0 0026B          MOVL    #1, 56(R7)                                        ; 0609
                                           57 DD 0026F          PUSHL   R7                                                ; 0610
                   00000000G    00         01 FB 00271          CALLS   #1, SYS$WRITE                                     ;
                                6E         50 D0 00278          MOVL    R0, STATUS                                        ;
                                3D         6E E8 0027B          BLBS    STATUS, 19$                                       ;
                   000182BA     8F         6E D1 0027E          CMPL    STATUS, #99002                                    ; 0614
                                           05 12 00285          BNEQ    18$                                               ;
                                62         20 8A 00287          BICB2   #32, (R2)                                         ; 0616
                                           2F 11 0028A 17$:     BRB     19$                                               ;
                          06    AE  010E   8F B0 0028C 18$:     MOVW    #270, DESC+2                                      ; 0621
                          04    AE    65   A6 B0 00292          MOVW    101(R6), DESC                                     ;
                          08    AE    69   A6 9E 00297          MOVAB   105(R6), DESC+4                                   ;
                                53 00000000G 8F D0 0029C        MOVL    #EXCH$_ACCESSFAIL, TEMP                           ; 0622
                                      0C   A7 DD 002A3          PUSHL   12(R7)                                            ;
                                      04   AE DD 002A6          PUSHL   STATUS                                            ;
                                      0C   AE 9F 002A9          PUSHAB  TMP_DESC                                          ;
                                           01 DD 002AC          PUSHL   #1                                                ;
                                           53 DD 002AE          PUSHL   TEMP                                              ;
                   00000000G    00         05 FB 002B0          CALLS   #5, LIB$SIGNAL                                    ;
                                50         53 D0 002B7          MOVL    TEMP, R0                                          ;
                                              04 002BA          RET                                                       ;
                          58    A6    7A   AA 90 002BB 19$:     MOVB    122(R10), 88(R6)                                  ; 0629
      50    0085   CA           01         02 EF 002C0          EXTZV   #2, #1, 133(R10), R0                              ; 0630
      62           01           06         50 F0 002C7          INSV    R0, #6, #1, (R2)                                  ;
                                50         01 D0 002CC          MOVL    #1, R0                                            ; 0632
                                              04 002CF          RET                                                       ;
                                           50 D4 002D0 20$:     CLRL    R0                                                ; 0633
                                              04 002D2          RET                                                       ;

; Routine Size:  723 bytes,    Routine Base:  EXCH$MOUN_CODE + 0167
```

```
1 GLOBAL ROUTINE exch$moun_implied_mount (namb : $ref_bblock) =   %SBTTL 'exch$moun_implied_mount'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform an implied mount.  This routine is called by verb routines when a request
2 !       is made to operate on an unmounted volume.  In particular, this is necessary for
2 !       EXCHANGE to work as a "foreign" DCL command (e.g. $EXCH DIRE CSA1:).
2 !
2 ! INPUTS:
2 !
2 !       namb - a pointer to a completed namb
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       namb [namb$a_assoc_volb] receives the address of the volb
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       a volb is allocated and mounted
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       A device may be added to internal tables.  It might also be mounted to VMS.
2 !--
2
2 $dbgtrc_prefix ('moun_implied_mount> ');
2
2 LOCAL
2     volb        : $ref_bblock,                  ! Local pointer to the volb
2     status
2     ;
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock  ! pointer to our work area
2     ;
2
2 $block_check (2, .namb, namb, 436);
2
2 ! Allocate and/or initialize the work area
2 !
2 moun_init ();
2
2 ! Get or default all the qualifiers
2 !
2 moun [moun$v_q_foreign] = true;
2 moun [moun$v_q_virtual] = false;
2 moun [moun$l_q_write]   = true;
```

```
2
2 ! Make sure that the volume is not already mounted
2 !
2 $logic_check (2, (.namb [namb$a_assoc_volb] EQL 0), 115);
2
2 ! Allocate a $VOLB to describe the volume
2 !
2 volb = exch$util_volb_allocate ();
2 moun [moun$a_volb] = .volb;
2
2 ! The moun routine expects to access the namb from the moun block
2 !
2 moun [moun$a_namb] = .namb;
2
2 ! Set some state in the volb.  Since the RX01 and RX02 drives do not support read and write checking, if we
2 ! see that one of these has been mounted /DATA_CHECK, we will do it by hand.
2 !
3 BEGIN
3 BIND
3     devbits = namb [namb$l_fabdev] : $bblock;
3 LOCAL
3     check;
4 check = (.devbits [dev$v_rck]
3         AND ((.namb [namb$b_devtype] EQL dt$_rx01) OR (.namb [namb$b_devtype] EQL dt$_rx02)));
3 volb [volb$v_read_check] = .moun [moun$v_q_read_check] OR .check;
4 check = (.devbits [dev$v_wck]
3         AND ((.namb [namb$b_devtype] EQL dt$_rx01) OR (.namb [namb$b_devtype] EQL dt$_rx02)));
3 volb [volb$v_write_check] = .moun [moun$v_q_write_check] OR .check;
3 $trace_print_fao ('read_check !UL, write_check !UL', .volb [volb$v_read_check], .volb [volb$v_write_check]);
3 END;
2
2 ! Now dispatch to foreign mount to complete the task
2 !
2 status = moun_foreign ();
2
2 ! If this worked, attempt the volume-specific mount
2 !
2 IF .status
2 THEN
3     BEGIN
3
3     CASE .volb [volb$b_vol_format] FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF
3     SET
3         [volb$k_vfmt_dos11]      : status = exch$dos11_mount (.volb);
3         [volb$k_vfmt_rt11]       : status = exch$rt11_mount (.volb);
3 !\      [volb$k_vfmt_rtmt]       : $exch_signal_stop (exch$_notimplement);
3         [OUTRANGE,INRANGE]       : $logic_check (0, (false), 228);
3     TES;
3
2     END;
2
2 ! If the foreign mount or volume-specific mount failed, then deallocate our VOLB
2 !
2 IF NOT .status
2 THEN
3     BEGIN
3     $trace_print_fao ('closing, fab=!XL', .volb [volb$a_fab]);
```

```
;  658          0748 3       $close (fab=.volb [volb$a_fab]);              ! Close it, ignore any errors
;  659          0749 3       exch$util_volb_release (.volb);
;  660          0750 3       exch$util_namb_release (.namb);
;  661          0751 3       END
;  662          0752 3   !
;  663          0753 3   ! Otherwise, signal the implied mount and return the address of the volb
;  664          0754 3   !
;  665          0755 2   ELSE
;  666          0756 3       BEGIN
;  667          0757 3
;  668          0758 3       IF  NOT .volb [volb$v_write]
;  669          0759 3         AND
;  670          0760 3           .moun [moun$v_q_message]
;  671          0761 3       THEN
;  672          0762 3           $exch_signal (exch$_writelock, 2, .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
;  673          0763 3
;  674          0764 3       $debug_print_fao ('volb address !XL', .volb);
;  675          0765 3
;  676          0766 3       IF .moun [moun$v_q_message]
;  677          0767 3       THEN
;  678        P 0768 3           $exch_signal (exch$_mounted, 4, .volb [volb$l_vol_type_len], volb [volb$t_vol_type],
;  679          0769 3                                           .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
;  680          0770 3       namb [namb$a_assoc_volb] = .volb;
;  681          0771 3
;  682          0772 3       ! Make a copy of the namb, since someone else owns the input namb
;  683          0773 3       !
;  684          0774 3       volb [volb$a_assoc_namb] = exch$cmd_namb_clone (.namb);
;  685          0775 3
;  686          0776 2       END;
;  687          0777 2
;  688          0778 2 RETURN .status;
;  689          0779 1 END;


                                                              .EXTRN  EXCH$_BADLOGIC, EXCH$_WRITELOCK
                                                              .EXTRN  EXCH$_MOUNTED

                                          01FC 00000          .ENTRY  EXCH$MOUN_IMPLIED_MOUNT, Save R2,R3,R4,R5,- ; 0634
                                                                      R6,R7,R8                                    ;
                      58 00000000G  00   9E 00002          MOVAB   LIB$SIGNAL, R8                              ;
                      57 00000000G  00   9E 00009          MOVAB   LIB$STOP, R7                                ;
                      56 00000000G  8F   D0 00010          MOVL    #EXCH$_BADLOGIC, R6                         ;
         53 00000000G EF            14   C1 00017          ADDL3   #20, EXCH$A_GBL, R3                         ; 0677
                      54        04  AC   D0 0001F          MOVL    NAMB, R4                                    ; 0680
                      52 010A00F7  8F   D0 00023          MOVL    #17432823, R2                               ;
                      51      01B4  8F   3C 0002A          MOVZWL  #436, R1                                    ;
                      50            54   D0 0002F          MOVL    R4, R0                                      ;
                         00000000G  EF   16 00032          JSB     EXCH$UTIL_BLOCK_CHECK                       ;
               0000V  CF            00   FB 00038          CALLS   #0, MOUN_INIT                               ; 0684
                      53            63   D0 0003D          MOVL    (R3), R3                                    ; 0688
                      55        20  A3   9E 00040          MOVAB   32(R3), R5                                  ;
                      65            04   88 00044          BISB2   #4, (R5)                                    ;
                      65            08   8A 00047          BICB2   #8, (R5)                                    ; 0689
                  1C  A3            01   D0 0004A          MOVL    #1, 28(R3)                                  ; 0690
                                74  A4   D5 0004E          TSTL    116(R4)                                     ; 0694
                                    0B   13 00051          BEQL    1$                                          ;
```

```
                                   7E         73    8F   9A 00053          MOVZBL  #115, -(SP)                           :
                                                    01   DD 00057          PUSHL   #1                                    :
                                                    56   DD 00059          PUSHL   R6                                    :
                                   67               03   FB 0005B          CALLS   #3, LIB$STOP                          :
                        00000000G  EF               00   FB 0005E 1$:      CALLS   #0, EXCH$UTIL_VOLB_ALLOCATE           : 0698
                                   52               50   D0 00065          MOVL    R0, VOLB                              :
                             04    A3               52   D0 00068          MOVL    VOLB, 4(R3)                           : 0699
                                   63               54   D0 0006C          MOVL    R4, (R3)                              : 0703
                                                    51   D4 0006F          CLRL    R1                                    : 0714
                                   10         79    A4   91 00071          CMPB    121(R4), #16                          :
                                                    02   12 00075          BNEQ    2$                                    :
                                                    51   D6 00077          INCL    R1                                    :
                                                    50   D4 00079 2$:      CLRL    R0                                    :
                                   0B         79    A4   91 0007B          CMPB    121(R4), #11                          :
                                                    02   12 0007F          BNEQ    3$                                    :
                                                    50   D6 00081          INCL    R0                                    :
                                   50               51   C8 00083 3$:      BISL2   R1, R0                                :
          53       6B      A4      01               06   EF 00086          EXTZV   #6, #1, 107(R4), CHECK                :
                                   53               53   D2 0008C          MCOML   CHECK, CHECK                          :
                           53      50               53   CB 0008F          BICL3   CHECK, R0, CHECK                      :
          50               65      01               00   EF 00093          EXTZV   #0, #1, (R5), R0                      : 0715
                                   50               53   88 00098          BISB2   CHECK, R0                             :
     48   A2               01      01               50   F0 0009B          INSV    R0, #1, #1, 72(VOLB)                  :
                                                    51   D4 000A1          CLRL    R1                                    : 0717
                                   10         79    A4   91 000A3          CMPB    121(R4), #16                          :
                                                    02   12 000A7          BNEQ    4$                                    :
                                                    51   D6 000A9          INCL    R1                                    :
                                                    50   D4 000AB 4$:      CLRL    R0                                    :
                                   0B         79    A4   91 000AD          CMPB    121(R4), #11                          :
                                                    02   12 000B1          BNEQ    5$                                    :
                                                    50   D6 000B3          INCL    R0                                    :
                                   50               51   C8 000B5 5$:      BISL2   R1, R0                                :
          53       6B      A4      01               07   EF 000B8          EXTZV   #7, #1, 107(R4), CHECK                :
                                   53               53   D2 000BE          MCOML   CHECK, CHECK                          :
                           53      50               53   CB 000C1          BICL3   CHECK, R0, CHECK                      :
          50               65      01               01   EF 000C5          EXTZV   #1, #1, (R5), R0                      : 0718
                                   50               53   88 000CA          BISB2   CHECK, R0                             :
     48   A2               01      02               50   F0 000CD          INSV    R0, #2, #1, 72(VOLB)                  :
                            FC55   CF               00   FB 000D3          CALLS   #0, MOUN_FOREIGN                      : 0724
                                   53               50   D0 000D8          MOVL    R0, STATUS                            :
                                   34               53   E9 000DB          BLBC    STATUS, 12$                           : 0728
                           03      00         58    A2   8F 000DE          CASEB   88(VOLB), #0, #3                      : 0732
        0020             0008    0015           0008        000E3 6$:      .WORD   7$-6$,-                               :
                                                                                   8$-6$,-                               :
                                                                                   7$-6$,-                               :
                                                                                   9$-6$                                 :
                                   7E         E4    8F   9A 000EB 7$:      MOVZBL  #228, -(SP)                           : 0737
                                                    01   DD 000EF          PUSHL   #1                                    :
                                                    56   DD 000F1          PUSHL   R6                                    :
                                   67               03   FB 000F3          CALLS   #3, LIB$STOP                          :
                                                    17   11 000F6          BRB     11$                                   :
                                                    52   DD 000F8 8$:      PUSHL   VOLB                                  : 0734
                        00000000G  EF               01   FB 000FA          CALLS   #1, EXCH$DOS11_MOUNT                  :
                                                    09   11 00101          BRB     10$                                   :
                                                    52   DD 00103 9$:      PUSHL   VOLB                                  : 0735
                        00000000G  EF               01   FB 00105          CALLS   #1, EXCH$RT11_MOUNT                   :
                                   53               50   D0 0010C 10$:     MOVL    R0, STATUS                            :
```

```
                         1E          53  E8 0010F 11$:     BLBS    STATUS, 13$                                  ; 0744
                   10    A2          DD 00112 12$:         PUSHL   16(VOLB)                                     ; 0748
        00000000G  00                01  FB 00115          CALLS   #1, SYS$CLOSE
                                     52  DD 0011C          PUSHL   VOLB                                         ; 0749
        00000000G  EF                01  FB 0011E          CALLS   #1, EXCH$UTIL_VOLB_RELEASE
                                     54  DD 00125          PUSHL   R4                                           ; 0750
        00000000G  EF                01  FB 00127          CALLS   #1, EXCH$UTIL_NAMB_RELEASE
                                     46  11 0012E          BRB     16$                                          ; 0744
   15      48    A2                  05  E0 00130 13$:     BBS     #5, 72(VOLB), 14$                            ; 0758
   2C            65                  04  E1 00135          BBC     #4, (R5), 15$                                ; 0760
                   69    A2          9F 00139          PUSHAB  105(VOLB)                                    ; 0762
                   65    A2          DD 0013C          PUSHL   101(VOLB)
                                     02  DD 0013F          PUSHL   #2
                00000000G    8F      DD 00141          PUSHL   #EXCH$_WRITELOCK
                         68          04  FB 00147          CALLS   #4, LIB$SIGNAL
   17            65                  04  E1 0014A 14$:     BBC     #4, (R5), 15$                                ; 0766
                   69    A2          9F 0014E          PUSHAB  105(VOLB)                                    ; 0769
                   65    A2          DD 00151          PUSHL   101(VOLB)
                   5D    A2          9F 00154          PUSHAB  93(VOLB)
                   59    A2          DD 00157          PUSHL   89(VOLB)
                                     04  DD 0015A          PUSHL   #4
                00000000G    8F      DD 0015C          PUSHL   #EXCH$_MOUNTED
                         68          06  FB 00162          CALLS   #6, LIB$SIGNAL
           74    A4                  52  D0 00165 15$:     MOVL    VOLB, 116(R4)                                ; 0770
                                     54  DD 00169          PUSHL   R4                                           ; 0774
        00000000G  EF                01  FB 0016B          CALLS   #1, EXCH$CMD_NAMB_CLONE
           24    A2                  50  D0 00172          MOVL    R0, 36(VOLB)
                 50                  53  D0 00176 16$:     MOVL    STATUS, R0                                   ; 0778
                                         04 00179          RET                                                  ; 0779

; Routine Size:  378 bytes,    Routine Base:  EXCH$MOUN_CODE + 043A
```

```
1 GLOBAL ROUTINE moun_init : NOVALUE =     %SBTTL 'moun_init'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !        Perform setups common to both EXCH$MOUN_MOUNT and EXCH$MOUN_IMPLIED_MOUNT, also EXCH$MOUN_DISMOUNT
2 !
2 ! INPUTS:
2 !
2 !        none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !        global environment
2 !
2 ! OUTPUTS:
2 !
2 !        none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !        none
2 !
2 ! ROUTINE VALUE:
2 !
2 !        none
2 !
2 ! SIDE EFFECTS:
2 !
2 !        memory might be allocated for the moun control block
2 !--
2
2 $dbgtrc_prefix ('moun_init> ');
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock  ! pointer to our work area
2     ;
2
2 LOCAL
2     message
2     ;
2
2 ! If this is an implied mount, then the /MESSAGE qualifier will not be allowed.  Enable the condition handle
2 ! so that it will be ignored.
2 !
2 ENABLE
2     exch$cmd_unwind_cli_syntax;
2
2 ! If our pointer is null, we need to allocate and initialize the work area
2 !
2 IF .moun EQL 0
2 THEN
3     BEGIN
3
3     ! Get the right sized chunk of memory, conveniently set to nulls
3     !
```

```
;   748        0837  3      moun = exch$util_vm_allocate_zeroed (exchblk$s_moun);
;   749        0838  3
;   750        0839  3      ! Set the ident fields
;   751        0840  3      !
;   752        0841  3      $block_init (.moun, moun);
;   753        0842  3
;   754        0843  3      ! Set the descriptors up
;   755        0844  3      !
;   756        0845  3      $dyn_str_desc_init (moun [moun$q_device]);
;   757        0846  3      $dyn_str_desc_init (moun [moun$q_filename]);
;   758        0847  3
;   759        0848  2      END;
;   760        0849  2
;   761        0850  2  ! Make sure that our work area is valid
;   762        0851  2  !
;   763        0852  2  $block_check (2, .moun, moun, 437);
;   764        0853  2
;   765        0854  2  ! Set the flag for printing mount and dismount messages.
;   766        0855  2  !
;   767        0856  2  moun [moun$v_q_message] = .exch$a_gbl [excg$v_q_message];        ! Default to external state
;   768        0857  2  message = cli$present (%ASCID 'MESSAGE');                        ! Find the flag state for the
;   769        0858  2
;   770        0859  2  !**************************************************************************************************
;   771        0860  2  !*  NOTE:  On an implied mount we will UNWIND out of the above call to CLI$PRESENT.  Therefore, none of the
;   772        0861  2  !*  which follows this comment will be executed during an implied mount!!!
;   773        0862  2  !**************************************************************************************************
;   774        0863  2
;   775        0864  2  IF   .message EQL cli$_present                      ! Either /MESSAGE or /NOMESSAGE must be specified in order t
;   776        0865  2    OR                                                !  override the external default
;   777        0866  2      .message EQL cli$_negated
;   778        0867  2  THEN
;   779        0868  2      moun [moun$v_q_message] = .message;
;   780        0869  2
;   781        0870  2  RETURN;
;   782        0871  1  END;


                                                            .PSECT  EXCH$MOUN_PLIT,NOWRT,2

                00  45  47  41  53  53  45  4D  00014 P.AAD:  .ASCII  \MESSAGE\<0>                                      :
                                    010E0007  0001C P.AAC:  .LONG   17694727                                         :
                                   00000000'  00020         .ADDRESS P.AAD                                           :

                                                            .EXTRN  EXCH$GQ_DYN_STR_TEMPLATE
                                                            .EXTRN  CLI$PRESENT, CLI$_PRESENT
                                                            .EXTRN  CLI$_NEGATED

                                                            .PSECT  EXCH$MOUN_CODE,NOWRT,2

                                      001C 00000            .ENTRY  MOUN_INIT, Save R2,R3,R4                    : 0780
                     54 00000000G  EF   9E 00002            MOVAB   TMPL, R4                                         :
             52 00000000G  EF      14   C1 00009            ADDL3   #20, EXCH$A_GBL, R2                          : 0816
                        6D     0072  CF   DE 00011            MOVAL   4$, (FP)                                         :
                                   62   D5 00016            TSTL    (R2)                                        : 0831
                                   22   12 00018            BNEQ    1$                                               :
                                   24   DD 0001A            PUSHL   #36                                         : 0837
```

```
                                                                 N  5
EXCH$MOUN        MOUNT verb dispatch and misc routines          16-Sep-1984 01:08:34    VAX-11 Bliss-32 V4.0-742             Page  26
V04-000          moun_init                                      14-Sep-1984 12:29:06    [EXCHNG.SRC]EXCMOUN.B32;1                  (8)

                          00000000G  EF           01  FB 0001C          CALLS   #1, EXCH$UTIL_VM_ALLOCATE_ZEROED               ;
                                     62           50  D0 00023          MOVL    R0, (R2)                                       ;
                                 08  A0           24  B0 00026          MOVW    #36, 8(R0)                                     ; 0841
                                 0A  A0           08  8E 0002A          MNEGB   #8, 10(R0)                                     ;
                       50            62           0C  C1 0002E          ADDL3   #12, (R2), R0                                  ; 0845
                                     60           64  7D 00032          MOVQ    TMPL, (R0)                                     ;
                       50            62           14  C1 00035          ADDL3   #20, (R2), R0                                  ; 0846
                                     60           64  7D 00039          MOVQ    TMPL, (R0)                                     ;
                                     53           62  D0 0003C 1$:      MOVL    (R2), R3                                       ; 0852
                                     52 002400F8  8F  D0 0003F          MOVL    #2359544, R2                                   ;
                                     51     01B5  8F  3C 00046          MOVZWL  #437, R1                                       ;
                                     50           53  D0 0004B          MOVL    R3, R0                                         ;
                                        00000000G EF  16 0004E          JSB     EXCH$UTIL_BLOCK_CHECK                          ;
           50 00000000G  FF          01           02  EF 00054          EXTZV   #2, #1, @EXCH$A_GBL, R0                        ; 0856
      20   A3            01          04           50  F0 0005D          INSV    R0, #4, #1, 32(R3)                             ;
                                            0000' CF  9F 00063          PUSHAB  P.AAC                                          ; 0857
                          00000000G  00           01  FB 00067          CALLS   #1, CLI$PRESENT                                ;
                          00000000G  8F           50  D1 0006E          CMPL    MESSAGE, #CLI$_PRESENT                         ; 0864
                                                  09  13 00075          BEQL    2$                                             ;
                          00000000G  8F           50  D1 00077          CMPL    MESSAGE, #CLI$_NEGATED                         ; 0866
                                                  06  12 0007E          BNEQ    3$                                             ;
      20   A3            01          04           50  F0 00080 2$:      INSV    MESSAGE, #4, #1, 32(R3)                        ; 0868
                                                      04 00086 3$:      RET                                                    ; 0871
                                                    0000 00087 4$:      .WORD   Save nothing                                   ; 0816
                                                  7E  D4 00089          CLRL    -(SP)                                          ;
                                                  5E  DD 0008B          PUSHL   SP                                             ;
                                     7E       04  AC  7D 0008D          MOVQ    4(AP), -(SP)                                   ;
                          00000000G  EF           03  FB 00091          CALLS   #3, EXCH$CMD_UNWIND_CLI_SYNTAX                 ;
                                                      04 00098          RET                                                    ;

; Routine Size:  153 bytes,    Routine Base:  EXCH$MOUN_CODE + 05B4
```

```
GLOBAL ROUTINE exch$moun_mount =        %SBTTL 'exch$moun_mount'
BEGIN
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       Action routine for the MOUNT verb, parses and performs main control functions for MOUNT
!
! INPUTS:
!
!       none
!
! IMPLICIT INPUTS:
!
!       Command parameters and qualifiers as returned from CLI$xxx routines.
!
! OUTPUTS:
!
!       none
!
! IMPLICIT OUTPUTS:
!
!       none
!
! ROUTINE VALUE:
!
!       Success or worst error encountered.
!
! SIDE EFFECTS:
!
!       A device may be added to internal tables.  It might also be mounted to VMS.
!--

$dbgtrc_prefix ('moun_mount> ');

LOCAL
    namb        : $ref_bblock,                  ! Local pointer to a namb
    volb        : $ref_bblock,                  ! Local pointer to a volb
    status
    ;

BIND
    moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock  ! pointer to our work area
    ;


! Allocate and/or initialize the work area
!
moun_init ();

! Get the individual boolean qualifiers.
!
moun [moun$v_q_read_check]  = cli$present (%ASCID 'DATA_CHECK.READ');
moun [moun$v_q_write_check] = cli$present (%ASCID 'DATA_CHECK.WRITE');
moun [moun$v_q_foreign]     = (cli$present (%ASCID 'FOREIGN') EQL cli$_present);        ! Only set if explic
moun [moun$v_q_virtual]     = cli$present (%ASCID 'VIRTUAL');
moun [moun$l_q_write]       = cli$present (%ASCID 'WRITE');
```

```
2
2  ! Do some consistency checks on the qualifiers
2  !
2  IF .moun [moun$v_q_foreign] AND .moun [moun$v_q_virtual]        ! /FOR/VIR is not allowed
2  THEN
2      $exch_signal_return (exch$_confqual);
2  moun [moun$v_q_foreign] = NOT .moun [moun$v_q_virtual];
2
2  ! Parse the device name parameter into a newly allocated $NAMB, there are no defaults
2  !
2  status = exch$cmd_parse_filespec (ascid_devicename, 0, 0, moun [moun$q_device], namb);
2  moun [moun$a_namb] = .namb;                              ! Save it in the work area too
2  IF NOT .status
2  THEN
2      $exch_signal_return (exch$_parseerr, 1, moun [moun$q_device], .status);
2  IF NOT .namb [namb$v_explicit_device]
2  THEN
2      $exch_signal_return (exch$_nodevice, 1, moun [moun$q_device]);
2  IF .namb [namb$v_explicit_node]
2  THEN
2      $exch_signal_return (exch$_noremote, 1, moun [moun$q_device]);
2  IF   .namb [namb$v_explicit_directory] OR .namb [namb$v_explicit_name]
2    OR .namb [namb$v_explicit_type] OR .namb [namb$v_explicit_version]
2  THEN
2      $exch_signal (exch$_devonly, 1, moun [moun$q_device]);
2
2  ! Make sure that the volume is not already mounted
2  !
3  IF (.namb [namb$a_assoc_volb] NEQ 0)
2  THEN
3      BEGIN
3
3      ! Signal and return the error
3      !
3      $exch_signal (exch$_volmount, 1, namb [namb$q_device]);
3      status = exch$_volmount;
3
3      END
2  ELSE
3      BEGIN
3
3      ! Allocate a $VOLB to describe the volume
3      !
3      volb = exch$util_volb_allocate ();
3      moun [moun$a_volb] = .volb;
3
3      ! Set some state in the volb.  Since the RX01 and RX02 drives do not support read and write checking, if
3      ! see that one of these has been mounted /DATA_CHECK, we will do it by hand.
3      !
4      BEGIN
4      BIND
4          devbits = namb [namb$l_fabdev] : $bblock;
4      LOCAL
4          check;
5      check = (.devbits [dev$v_rck]
4              AND ((.namb [namb$b_devtype] EQL dt$_rx01) OR (.namb [namb$b_devtype] EQL dt$_rx02)));
4      volb [volb$v_read_check] = .moun [moun$v_q_read_check] OR .check;
```

```
5      check = (.devbits [dev$v_wck]
4              AND ((.namb [namb$b_devtype] EQL dt$_rx01) OR (.namb [namb$b_devtype] EQL dt$_rx02)));
4      volb [volb$v_write_check] = .moun [moun$v_q_write_check] OR .check;
4      $trace_print_fao ('read_check !UL, write_check !UL', .volb [volb$v_read_check], .volb [volb$v_write_chec
3      END;
3
3      ! Now dispatch to either the virtual mount or to foreign mount to complete the task
3      !
3      IF .moun [moun$v_q_virtual]
3      THEN
3          status = moun_virtual ()
3      ELSE
3          status = moun_foreign ();
3
3      !******************************************************************************************
3      !*  moun_virtual will have reparsed the device name and created a new namb.  All references to namb must
3      !*  use moun [moun$a_namb] from this point on!!
3      !******************************************************************************************
3
3      ! If this worked, attempt the volume-specific mount
3      !
3      IF .status
3      THEN
4          BEGIN
4
4          CASE .volb [volb$b_vol_format] FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF
4          SET
4                  [volb$k_vfmt_dos11]     : status = exch$dos11_mount (.volb);
4                  [volb$k_vfmt_rt11]      : status = exch$rt11_mount (.volb);
4 !\               [volb$k_vfmt_rtmt]      : $exch_signal_stop (exch$_notimplement);
4                  [OUTRANGE,INRANGE]      : $logic_check (0, (false), 227);
4          TES;
3          END;
3
3      ! If the mount or the specific mount failed, then deallocate our VOLB
3      !
3      IF NOT .status
3      THEN
4          BEGIN
4          $trace_print_fao ('closing, fab=!XL', .volb [volb$a_fab]);
4          $close (fab=.volb [volb$a_fab]);         ! Close it, ignore any errors
4          exch$util_volb_release (.volb);
4          exch$util_namb_release (.moun [moun$a_namb]);
4          END
4
4      ! If it worked then finish the mount
4      !
3      ELSE
4          BEGIN
4
4          ! Save the namb so that we can look at it later
4          !
4          volb [volb$a_assoc_namb] = .moun [moun$a_namb];
4
4          ! If the volume is write-locked, signal this information
4          !
5          IF  (NOT .volb [volb$v_write])          ! If the volb is marked no write
```

```
;  955          1043  4             AND                                     !  and
;  956          1044  4                 .moun [moun$l_q_write]              !  we thought we were going to write
;  957          1045  4             THEN
;  958          1046  5                 BEGIN
;  959          1047  5                 LOCAL
;  960          1048  5                     status2;
;  961          1049  5
;  962          1050  5                 IF .moun [moun$l_q_write] EQL cli$_present
;  963          1051  5                 THEN
;  964          1052  6                     status2 = $warning_stat_copy (exch$_writelock)  ! If an explicit /WRITE, then warning status
;  965          1053  5                 ELSE
;  966          1054  5                     status2 = exch$_writelock;                      ! otherwise its info status.
;  967          1055  5                 $exch_signal (.status2, 2, .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
;  968          1056  5
;  969          1057  4                 END;
;  970          1058  4
;  971          1059  4             ! Now tell that we mounted it
;  972          1060  4             !
;  973          1061  4             IF .moun [moun$v_q_virtual]
;  974          1062  4             THEN
;  975          1063  5                 BEGIN
;  976          1064  5                 BIND
;  977          1065  5                     nam = .volb [volb$a_nam] : $bblock;      ! RMS nam block for the opened file
;  978          1066  5                 IF .moun [moun$v_q_message]
;  979          1067  5                 THEN
;  980        P 1068  5                     $exch_signal (exch$_mountvir, 4, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
;  981          1069  5                                             .nam [nam$b_rsl], .nam [nam$l_rsa]);
;  982          1070  5                 $debug_print_fao ('volb address !XL', .volb);
;  983          1071  5                 END
;  984          1072  4             ELSE
;  985          1073  5                 BEGIN
;  986          1074  5                 IF .moun [moun$v_q_message]
;  987          1075  5                 THEN
;  988        P 1076  5                     $exch_signal (exch$_mounted, 4, .volb [volb$l_vol_type_len], volb [volb$t_vol_type],
;  989          1077  5                                             .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
;  990          1078  5                 $debug_print_fao ('volb address !XL', .volb);
;  991          1079  4                 END;
;  992          1080  4             END
;  993          1081  2         END;
;  994          1082  2
;  995          1083  2 RETURN .status;
;  996          1084  1 END;



                                                                                  .PSECT  EXCH$MOUN_PLIT,NOWRT,2

44  41  45  52  2E  4B  43  45  48  43  5F  41  54  41  44  00024 P.AAF:   .ASCII  \DATA_CHECK.READ\<0>                          ;
                                                        00  00033                                                              ;
                                                  010E000F  00034 P.AAE:   .LONG   17694735                                      ;
                                                 00000000'  00038          .ADDRESS P.AAF                                        ;
54  49  52  57  2E  4B  43  45  48  43  5F  41  54  41  44  0003C P.AAH:   .ASCII  \DATA_CHECK.WRITE\                            ;
                                                        45  0004B                                                              ;
                                                  010E0010  0004C P.AAG:   .LONG   17694736                                      ;
                                                 00000000'  00050          .ADDRESS P.AAH                                        ;
                            00  4E  47  49  45  52  4F  46  00054 P.AAJ:   .ASCII  \FOREIGN\<0>                                  ;
                                                  010E0007  0005C P.AAI:   .LONG   17694727                                      ;
```

```
                                       00000000' 00060          .ADDRESS P.AAJ                                   ;
                  00  4C  41  55  54  52  49  56  00064 P.AAL:  .ASCII   \VIRTUAL\<0>                            ;
                                       010E0007  0006C P.AAK:   .LONG    17694727                                ;
                                       00000000' 00070          .ADDRESS P.AAL                                   ;
                  00  00  00  45  54  49  52  57  00074 P.AAN:  .ASCII   \WRITE\<0><0><0>                        ;
                                       010E0005  0007C P.AAM:   .LONG    17694725                                ;
                                       00000000' 00080          .ADDRESS P.AAN                                   ;

                                                                .EXTRN   EXCH$_MOUNTVIR

                                                                .PSECT   EXCH$MOUN_CODE,NOWRT,2

                                            0FFC 00000          .ENTRY   EXCH$MOUN_MOUNT, Save R2,R3,R4,R5,R6,R7,R8,-; 0872
                                                                         R9,R10,R11                              ;
                             5B      0000'  CF   9E 00002       MOVAB    P.AAE, R11                              ;
                             5A 00000000G   00   9E 00007       MOVAB    CLI$PRESENT, R10                        ;
                             59 00000000G   00   9E 0000E       MOVAB    LIB$SIGNAL, R9                          ;
                             5E             04   C2 00015       SUBL2    #4, SP                                  ;
                52 00000000G EF             14   C1 00018       ADDL3    #20, EXCH$A_GBL, R2                     ; 0914
                       FF42  CF             00   FB 00020       CALLS    #0, MOUN_INIT                           ; 0920
                             54             62   D0 00025       MOVL     (R2), R4                                ; 0924
                             55       20    A4   9E 00028       MOVAB    32(R4), R5                              ;
                                            5B   DD 0002C       PUSHL    R11                                     ;
                             6A             01   FB 0002E       CALLS    #1, CLI$PRESENT                         ;
        65      01           00             50   F0 00031       INSV     R0, #0, #1, (R5)                        ;
                                      18    AB   9F 00036       PUSHAB   P.AAG                                   ; 0925
                             6A             01   FB 00039       CALLS    #1, CLI$PRESENT                         ;
        65      01           01             50   F0 0003C       INSV     R0, #1, #1, (R5)                        ;
                                      28    AB   9F 00041       PUSHAB   P.AAI                                   ; 0926
                             6A             01   FB 00044       CALLS    #1, CLI$PRESENT                         ;
                                            51   D4 00047       CLRL     R1                                      ;
                00000000G    8F             50   D1 00049       CMPL     R0, #CLI$_PRESENT                       ;
                                            02   12 00050       BNEQ     1$                                      ;
                                            51   D6 00052       INCL     R1                                      ;
        65      01           02             51   F0 00054 1$:   INSV     R1, #2, #1, (R5)                        ;
                                      38    AB   9F 00059       PUSHAB   P.AAK                                   ; 0927
                             6A             01   FB 0005C       CALLS    #1, CLI$PRESENT                         ;
        65      01           03             50   F0 0005F       INSV     R0, #3, #1, (R5)                        ;
                                      48    AB   9F 00064       PUSHAB   P.AAM                                   ; 0928
                             6A             01   FB 00067       CALLS    #1, CLI$PRESENT                         ;
                        1C   A4             50   D0 0006A       MOVL     R0, 28(R4)                              ;
                12           65             02   E1 0006E       BBC      #2, (R5), 2$                            ; 0932
                0E           65             03   E1 00072       BBC      #3, (R5), 2$                            ;
                             52 00F812E0    8F   D0 00076       MOVL     #16257760, TEMP                         ; 0934
                                            52   DD 0007D       PUSHL    TEMP                                    ;
                             69             01   FB 0007F       CALLS    #1, LIB$SIGNAL                          ;
                                            3F   11 00082       BRB      3$                                      ;
        50      65           01             03   EF 00084 2$:   EXTZV    #3, #1, (R5), R0                        ; 0935
                             50             50   D2 00089       MCOML    R0, R0                                  ;
        65      01           02             50   F0 0008C       INSV     R0, #2, #1, (R5)                        ;
                                            5E   DD 00091       PUSHL    SP                                      ; 0939
                             56       0C    A4   9E 00093       MOVAB    12(R4), R6                              ;
                                            56   DD 00097       PUSHL    R6                                      ;
                                            7E   7C 00099       CLRQ     -(SP)                                   ;
                                      D8    AB   9F 0009B       PUSHAB   ASCID_DEVICENAME                        ;
                00000000G    EF             05   FB 0009E       CALLS    #5, EXCH$CMD_PARSE_FILESPEC             ;
                             58             50   D0 000A5       MOVL     R0, STATUS                              ;
```

```
                                    53            6E  D0 000A8        MOVL    NAMB, R3                      ; 0940
                                    64            53  D0 000AB        MOVL    R3, (R4)                      ;
                                    16            58  E8 000AE        BLBS    STATUS, 4$                    ; 0941
                                    52 00000000G  8F  D0 000B1        MOVL    #EXCH$_PARSEERR, TEMP         ; 0943
                                            0140  8F  BB 000B8        PUSHR   #^M<R6,R8>                    ;
                                                  01  DD 000BC        PUSHL   #1                            ;
                                                  52  DD 000BE        PUSHL   TEMP                          ;
                                    69            04  FB 000C0        CALLS   #4, LIB$SIGNAL                ;
                                    50            52  D0 000C3 3$:    MOVL    TEMP, R0                      ;
                                                      04 000C6        RET                                   ;
                                    52        6C  A3  9E 000C7 4$:    MOVAB   108(R3), R2                   ; 0944
                                                  62  95 000CB        TSTB    (R2)                          ;
                                                  09  19 000CD        BLSS    5$                            ;
                                    57 00000000G  8F  D0 000CF        MOVL    #EXCH$_NODEVICE, TEMP         ; 0946
                                                  0B  11 000D6        BRB     6$                            ;
                      14            62            06  E1 000D8 5$:    BBC     #6, (R2), 7$                  ; 0947
                                    57 00000000G  8F  D0 000DC        MOVL    #EXCH$_NOREMOTE, TEMP         ; 0949
                                                  56  DD 000E3 6$:    PUSHL   R6                            ;
                                                  01  DD 000E5        PUSHL   #1                            ;
                                                  57  DD 000E7        PUSHL   TEMP                          ;
                                    69            03  FB 000E9        CALLS   #3, LIB$SIGNAL                ;
                                    50            57  D0 000EC        MOVL    TEMP, R0                      ;
                                                      04 000EF        RET                                   ;
                                    0C        01  A2  E8 000F0 7$:    BLBS    1(R2), 8$                     ; 0950
                      08            62            09  E0 000F4        BBS     #9, (R2), 8$                  ;
                      04            62            0A  E0 000F8        BBS     #10, (R2), 8$                 ; 0951
                      0D            62            0B  E1 000FC        BBC     #11, (R2), 9$                 ;
                                                  56  DD 00100 8$:    PUSHL   R6                            ; 0953
                                                  01  DD 00102        PUSHL   #1                            ;
                                       00000000G  8F  DD 00104        PUSHL   #EXCH$_DEVONLY                ;
                                    69            03  FB 0010A        CALLS   #3, LIB$SIGNAL                ;
                                              74  A3  D5 0010D 9$:    TSTL    116(R3)                       ; 0957
                                                  18  13 00110        BEQL    10$                           ;
                                              40  A3  9F 00112        PUSHAB  64(R3)                        ; 0963
                                                  01  DD 00115        PUSHL   #1                            ;
                                       00000000G  8F  DD 00117        PUSHL   #EXCH$_VOLMOUNT               ;
                                    69            03  FB 0011D        CALLS   #3, LIB$SIGNAL                ;
                                    58 00000000G  8F  D0 00120        MOVL    #EXCH$_VOLMOUNT, STATUS       ; 0964
                                                0157  31 00127        BRW     30$                           ; 0957
                         00000000G  EF            00  FB 0012A 10$:   CALLS   #0, EXCH$UTIL_VOLB_ALLOCATE   ; 0972
                                    56            50  D0 00131        MOVL    R0, VOLB                      ;
                                04  A4            56  D0 00134        MOVL    VOLB, 4(R4)                   ; 0973
                                                  51  D4 00138        CLRL    R1                            ; 0984
                                    10        79  A3  91 0013A        CMPB    121(R3), #16                  ;
                                                  02  12 0013E        BNEQ    11$                           ;
                                                  51  D6 00140        INCL    R1                            ;
                                                  50  D4 00142 11$:   CLRL    R0                            ;
                                    0B        79  A3  91 00144        CMPB    121(R3), #11                  ;
                                                  02  12 00148        BNEQ    12$                           ;
                                                  50  D6 0014A        INCL    R0                            ;
                                    50            51  C8 0014C 12$:   BISL2   R1, R0                        ;
      52        6B    A3            01            06  EF 0014F        EXTZV   #6, #1, 107(R3), CHECK        ;
                                    52            52  D2 00155        MCOML   CHECK, CHECK                  ;
                      52            50            52  CB 00158        BICL3   CHECK, R0, CHECK              ;
      50              65            01            00  EF 0015C        EXTZV   #0, #1, (R5), R0              ; 0985
                                    50            52  88 00161        BISB2   CHECK, R0                     ;
48    A6              01            01            50  F0 00164        INSV    R0, #1, #1, 72(VOLB)          ;
```

```
                                                         H  6
EXCH$MOUN     MOUNT verb dispatch and misc routines     16-Sep-1984 01:08:34   VAX-11 Bliss-32 V4.0-742              Page  33
V04-000       exch$moun_mount                           14-Sep-1984 12:29:06   [EXCHNG.SRC]EXCMOUN.B32;1               (9)

                                                               51  D4 0016A         CLRL     R1                                  ; 0987
                                              10         79    A3  91 0016C         CMPB     121(R3), #16                        ;
                                                               02  12 00170         BNEQ     13$                                 ;
                                                               51  D6 00172         INCL     R1                                  ;
                                                               50  D4 00174  13$:   CLRL     R0                                  ;
                                              0B         79    A3  91 00176         CMPB     121(R3), #11                        ;
                                                               02  12 0017A         BNEQ     14$                                 ;
                                                               50  D6 0017C         INCL     R0                                  ;
                                              50               51  C8 0017E  14$:   BISL2    R1, R0                              ;
            52       6B    A3                 01               07  EF 00181         EXTZV    #7, #1, 107(R3), CHECK              ;
                                              52               52  D2 00187         MCOML    CHECK, CHECK                        ;
                           52                 50               52  CB 0018A         BICL3    CHECK, R0, CHECK                    ;
            50             65                 01               01  EF 0018E         EXTZV    #1, #1, (R5), R0                    ; 0988
                                              50               52  88 00193         BISB2    CHECK, R0                           ;
      48    A6             01                 02               50  F0 00196         INSV     R0, #2, #1, 72(VOLB)                ;
                           07                 65               03  E1 0019C         BBC      #3, (R5), 15$                       ; 0994
                                     0000V    CF               00  FB 001A0         CALLS    #0, MOUN_VIRTUAL                    ; 0996
                                                               05  11 001A5         BRB      16$                                 ;
                                      F96E    CF               00  FB 001A7  15$:   CALLS    #0, MOUN_FOREIGN                    ; 0998
                                              58               50  D0 001AC  16$:   MOVL     R0, STATUS                          ;
                                              3C               58  E9 001AF         BLBC     STATUS, 23$                         ; 1007
                           03                 00         58    A6  8F 001B2         CASEB    88(VOLB), #0, #3                    ; 1011
          0028           0008               001D             0008     001B7  17$:   .WORD    18$-17$,-                           ;
                                                                                             19$-17$,-                           ;
                                                                                             18$-17$,-                           ;
                                                                                             20$-17$                             ;
                                              7E         E3    8F  9A 001BF  18$:   MOVZBL   #227, -(SP)                         ; 1016
                                                               01  DD 001C3         PUSHL    #1                                  ;
                                                  00000000G    8F  DD 001C5         PUSHL    #EXCH$_BADLOGIC                     ;
                                 00000000G    00               03  FB 001CB         CALLS    #3, LIB$STOP                        ;
                                                               17  11 001D2         BRB      22$                                 ;
                                                               56  DD 001D4  19$:   PUSHL    VOLB                                ; 1013
                                 00000000G    EF               01  FB 001D6         CALLS    #1, EXCH$DOS11_MOUNT                ;
                                                               09  11 001DD         BRB      21$                                 ;
                                                               56  DD 001DF  20$:   PUSHL    VOLB                                ; 1014
                                 00000000G    EF               01  FB 001E1         CALLS    #1, EXCH$RT11_MOUNT                 ;
                                              58               50  D0 001E8  21$:   MOVL     R0, STATUS                          ;
                                              1E               58  E8 001EB  22$:   BLBS     STATUS, 24$                         ; 1022
                                                         10    A6  DD 001EE  23$:   PUSHL    16(VOLB)                            ; 1026
                                 00000000G    00               01  FB 001F1         CALLS    #1, SYS$CLOSE                       ;
                                                               56  DD 001F8         PUSHL    VOLB                                ; 1027
                                 00000000G    EF               01  FB 001FA         CALLS    #1, EXCH$UTIL_VOLB_RELEASE          ;
                                                               64  DD 00201         PUSHL    (R4)                                ; 1028
                                 00000000G    EF               01  FB 00203         CALLS    #1, EXCH$UTIL_NAMB_RELEASE          ;
                                                               75  11 0020A         BRB      30$                                 ; 1022
                                        24    A6               64  D0 0020C  24$:   MOVL     (R4), 36(VOLB)                      ; 1038
                           2E           48    A6               05  E0 00210         BBS      #5, 72(VOLB), 27$                   ; 1042
                                              2A         1C    A4  E9 00215         BLBC     28(R4), 27$                         ; 1044
                                 00000000G    8F         1C    A4  D1 00219         CMPL     28(R4), #CLI$_PRESENT               ; 1050
                                                               0C  12 00221         BNEQ     25$                                 ;
                                              50  00000000G    8F  D0 00223         MOVL     #EXCH$_WRITELOCK, STATUS2           ; 1052
                                              50               07  8A 0022A         BICB2    #7, STATUS2                         ;
                                                               07  11 0022D         BRB      26$                                 ;
                                              50  00000000G    8F  D0 0022F  25$:   MOVL     #EXCH$_WRITELOCK, STATUS2           ; 1054
                                                         69    A6  9F 00236  26$:   PUSHAB   105(VOLB)                           ; 1055
                                                         65    A6  DD 00239         PUSHL    101(VOLB)                           ;
                                                               02  DD 0023C         PUSHL    #2                                  ;
```

```
                                   50  DD 0023E          PUSHL   STATUS2                                 ;
                         69        04  FB 00240          CALLS   #4, LIB$SIGNAL                          ;
          1F             65        03  E1 00243 27$:     BBC     #3, (R5), 28$                           ; 1061
                         50     18 A6  D0 00247          MOVL    24(VOLB), R0                            ; 1065
          32             65        04  E1 0024B          BBC     #4, (R5), 30$                           ; 1066
                                04 A0  DD 0024F          PUSHL   4(R0)                                   ; 1069
                         7E     03 A0  9A 00252          MOVZBL  3(R0), -(SP)                            ;
                                69 A6  9F 00256          PUSHAB  105(VOLB)                               ;
                                65 A6  DD 00259          PUSHL   101(VOLB)                               ;
                                   04  DD 0025C          PUSHL   #4                                      ;
                          00000000G 8F DD 0025E          PUSHL   #EXCH$_MOUNTVIR                         ;
                                   18  11 00264          BRB     29$                                     ;
          17             65        04  E1 00266 28$:     BBC     #4, (R5), 30$                           ; 1074
                                69 A6  9F 0026A          PUSHAB  105(VOLB)                               ; 1077
                                65 A6  DD 0026D          PUSHL   101(VOLB)                               ;
                                5D A6  9F 00270          PUSHAB  93(VOLB)                                ;
                                59 A6  DD 00273          PUSHL   89(VOLB)                                ;
                                   04  DD 00276          PUSHL   #4                                      ;
                          00000000G 8F DD 00278          PUSHL   #EXCH$_MOUNTED                          ;
                         69        06  FB 0027E 29$:     CALLS   #6, LIB$SIGNAL                          ;
                         50        58  D0 00281 30$:     MOVL    STATUS, R0                              ; 1083
                                       04 00284          RET                                             ; 1084

; Routine Size:  645 bytes,    Routine Base:  EXCH$MOUN_CODE + 064D
```

EX

```
1 GLOBAL ROUTINE moun_virtual =   %SBTTL 'moun_virtual'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Make a virtual volume known to EXCHANGE.  An RMS file is opened on the volb.
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       q_namb - name block describing the device
2 !       q_write - flag saying if we allow writes
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       q_volb - volume block which will describe the mounted volume
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       lots
2 !--
2
2 $dbgtrc_prefix ('moun_virtual> ');
2
2 LOCAL
2     status,
2     xab : $bblock [xab$c_fhclen],                    ! file header char xab so that we can read the size of the f
2     namb : $ref_bblock,                              ! a pointer to the namb
2     vir_namb : $ref_bblock                           ! a local namb for the virtual name
2     ;
2
2 BIND
2     moun = exch$a_gbl [excg$a_moun_work] : $ref_bblock,  ! pointer to our work area
2     inpfile = moun [moun$q_filename] : $desc_block,      ! Name of virtual device filename as input
2     volb = .moun [moun$a_volb] : $bblock,                ! Pointer to exchange VOLB structure
2     fab  = .volb [volb$a_fab] : $bblock,                 ! File Access Block for the volume
2     rab  = .volb [volb$a_rab] : $bblock,                 ! Record Access Block for the volume
2     nam  = .volb [volb$a_nam] : $bblock                  ! RMS name block for the volume
2     ;
2
2 $block_check (2, .moun, moun, 438);
2 namb = .moun [moun$a_namb];                          ! Grab pointer to exchange NAMB structure for device name
2 $block_check (2, .namb, namb, 439);
2 $block_check (2, volb, volb, 440);
2
```

```
  2 ! Reparse the virtual volume name.  The "no_get_value" option is used to reparse the original name rather th
  2 ! calling CLI$GET_VALUE again.  We request a virtual device parse to prevent the device from being
  2 ! expanded, for example to prevent "R:" from becoming "_RTA0:" and "DM:" from becoming "_DMA0:".  The NAMB f
  2 ! the first parse has to be discarded.
  2 !
  2 exch$util_namb_release (.namb);
  2 status = exch$cmd_parse_filespec (moun [moun$q_device], 0, (prsopt$m_no_get_value OR prsopt$m_virtual_device
  2                                         moun [moun$q_device], namb);
  2 IF NOT .status                      !?? We probably don't need to check the status, since we have parsed it once
  2 THEN
  2     $exch_signal_return (exch$_parseerr, 1, moun [moun$q_device], .status);
  2 moun [moun$a_namb] = .namb;                         ! Save namb address in work area
  2 $logic_check (3, (.namb [namb$a_assoc_volb] EQL 0), 317);
  2
  2 ! Fetch the name of the file which is underneath the virtual volume
  2 !
  2 status = exch$cmd_parse_filespec (%ASCID 'FILENAME', %ASCID 'VIRTUAL.DSK', 0, inpfile, vir_namb);
  2 IF NOT .status
  2 THEN
  2     $exch_signal_return (exch$_parseerr, 1, inpfile, .status);
  2
  3 BEGIN   ! scope "fulfile"
  3     BIND
  3         fulfile = vir_namb [namb$q_fullname] : $desc_block;
  3
  3     ! Get the device information.  Note that we will later change the volb$l_devmaxblock to the actual in-us
  3     ! count for the file.
  3     !
  4     IF NOT (status = exch$util_vol_getdvi (fulfile, volb))
  3     THEN
  4         BEGIN
  4         $exch_signal (exch$_accessfail, 1, fulfile, .status);
  4         status = false;
  4         exch$util_namb_release (.vir_namb);
  4         RETURN .status;
  3         END;
  3
  3     ! Now set the unique ident field of this volb, we use the ident given by EXCH$CMD_PARSE_FILESPEC
  3     !
P 3     $trace_print_fao ('volb devnam "!AF", namb device "!AF", namb volid "!AF", concealed !UL',
P 3                 .volb [volb$l_devnamlen], volb [volb$t_devnam],
P 3                 (BIND ndev = namb [namb$q_device] : $desc_block; .ndev [dsc$w_length]),
P 3                 (BIND ndev = namb [namb$q_device] : $desc_block; .ndev [dsc$a_pointer]),
P 3                 .namb [namb$l_vol_ident_len], namb [namb$t_vol_ident],
  3                 .namb [namb$v_concealed_device]);
  3     CH$MOVE (volb$s_vol_ident, namb [namb$t_vol_ident], volb [volb$t_vol_ident]);
  3     volb [volb$l_vol_ident_len] = .namb [namb$l_vol_ident_len];
  3
L 3     %IF switch_debug                            ! Debugging trace code
U 3     %THEN
U 3         BEGIN
U 3         LOCAL
U 3             tmp_desc : $desc_block;
U 3         $stat_str_desc_init (tmp_desc, .volb [volb$l_devnamlen], volb [volb$t_devnam]);
U 3         $trace_print_fao ('Getdvi for name "!AS" resolved to device "!AS"', fulfile, tmp_desc);
U 3         END;
  3     %FI
```

```
     3
     3      ! Init the RMS blocks for the volume
     3      !
P    3      $fab_init (
P    3          FAB = fab,                                   ! Volume FAB
P    3          FAC = (BIO,GET),                             !  Block I/O, read-only
P    3          FNA = .fulfile [dsc$a_pointer],              !  Set name addr
P    3          FNS = .fulfile [dsc$w_length],               !  Set name size
P    3          NAM = nam,                                   !  Name block
     3          XAB = xab);                                  !  A file header char xab so that we can read the file size
P    3      $rab_init (
P    3          RAB = rab,                                   ! Volume RAB
P    3          ROP = BIO,                                   !  Block I/O
     3          FAB = fab);                                  !  FAB addr
P    3      $nam_init (
P    3          NAM = nam,                                   ! File name block
P    3          RSA = .volb [volb$a_rsbuf],                  !  Result name addr
P    3          RSS = nam$c_maxrss,                          !  Result name size
P    3          ESA = .volb [volb$a_esbuf],                  !  Expanded name addr
     3          ESS = nam$c_maxrss);                         !  Expanded name size
P    3      $xabfhc_init (                                   ! File header char xab so that we can read the file size
     3          XAB = xab);                                  !  RMS will fill it in when we open
     3
     3      ! Make any adjustments to the RMS blocks as necessary
     3      !
     3      fab [fab$v_put] = .moun [moun$l_q_write];
     3
     3      ! Open and connect to the volume
     3      !
     3      $trace_print_fao ('opening, fab=!XL', fab);
     4      IF NOT (status = $open (fab = fab))
     3      THEN
     4          BEGIN
     4
     4          ! If we failed with a privilege violation and we were going to write, try again /NOWRITE
     4          !
     4          IF (.status EQL rms$_prv) AND .fab [fab$v_put]
     4          THEN
     5              BEGIN
     5              fab [fab$v_put] = false;                 ! Cancel the write option
     5              $trace_print_fao ('try open again, fab=!XL', fab);
     6              IF NOT (status = $open (fab = fab))
     5              THEN
     6                  BEGIN
     6                  exch$util_file_error (exch$_openvirtual, .status, fab, .fab [fab$l_stv]);
     6                  status = false;
     6                  exch$util_namb_release (.vir_namb);
     6                  RETURN .status;
     5                  END;
     5              END
     4          ELSE
     5              BEGIN
     5              exch$util_file_error (exch$_openvirtual, .status, fab, .fab [fab$l_stv]);
     5              status = false;
     5              exch$util_namb_release (.vir_namb);
     5              RETURN .status;
     4              END;
```

```
3             END;
3
4         IF NOT (status = $connect (rab = rab))
3         THEN
4             BEGIN
4             exch$util_file_error (exch$_openvirtual, .status, fab, .rab [rab$l_stv]);
4             status = false;
4             exch$util_namb_release (.vir_namb);
4             RETURN .status;
3             END;
3
2     END;    ! scope "fulfile"
2
2     ! Fill in the rest of the state in the volb
2     !
2     volb [volb$v_connected] = true;
2     volb [volb$v_virtual] = true;
2     volb [volb$v_write] = .fab [fab$v_put];
2     volb [volb$l_devmaxblock] = .xab [xab$l_ebk] -  ! Put the file size in the device block size field
3                         (IF .xab [xab$w_ffb] NEQ 0  ! (Eof block is one too high if the first free byte is zero)
2                          THEN 0 ELSE 1);
2     volb [volb$l_volmaxblock] = .volb [volb$l_devmaxblock]; ! Make the physical
2     $trace_print_fao ('device max blocks !UL', .volb [volb$l_devmaxblock]);
2     fab [fab$l_xab] = 0;                                  ! Remove the xab from the fab, won't be valid after return
2
2     ! A virtual volume must be RT11.  If /VOL was specified, make sure that it was RT11, otherwise assume it
2     !
2     IF .namb [namb$v_vfmt_explicit]
2     THEN
3         BEGIN
3         IF .namb [namb$b_vol_format] NEQ volb$k_vfmt_rt11
3         THEN
3             status = exch$_invvolfmt
3         ELSE
4             BEGIN
4             volb [volb$v_vfmt_explicit] = true;
4             volb [volb$b_vol_format] = volb$k_vfmt_rt11;
3             END;
3         END
2     ELSE
3         BEGIN
3         volb [volb$v_vfmt_explicit] = false;
3         volb [volb$b_vol_format] = volb$k_vfmt_rt11;
2         END;
2
2     ! Release the namb
2     !
2     exch$util_namb_release (.vir_namb);
2
2     RETURN .status;
1     END;


                                                                  .PSECT  EXCH$MOUN_PLIT,NOWRT,2

          45  4D  41  4E  45  4C  49  46  00084 P.AAP:  .ASCII  \FILENAME\                                 ;
```

```
                                                010E0008  0008C P.AAO:     .LONG    17694728                                  :
                                                00000000' 00090            .ADDRESS P.AAP                                     :
        00  4B  53  44  2E  4C  41  55  54  52  49   56   00094 P.AAR:     .ASCII   \VIRTUAL.DSK\<0>                          :
                                                010E000B  000A0 P.AAQ:     .LONG    17694731                                  :
                                                00000000' 000A4            .ADDRESS P.AAR                                     :

                                                                           .EXTRN   EXCH$_OPENVIRTUAL
                                                                           .EXTRN   EXCH$_INVVOLFMT

                                                                           .PSECT   EXCH$MOUN_CODE,NOWRT,2

                                                     0FFC 00000            .ENTRY   MOUN_VIRTUAL, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  1085
                                                                                    R10,R11                                   :
                              5E                  3C  C2 00002             SUBL2    #60, SP                                   :
               50 00000000G   EF                  14  C1 00005             ADDL3    #20, EXCH$A_GBL, R0                       :  1129
                              57                  60  D0 0000D             MOVL     (R0), R7                                  :  1130
                              58          04      A7  D0 00010             MOVL     4(R7), R8                                 :  1131
                              56          10      A8  D0 00014             MOVL     16(R8), R6                                :  1132
                              5A          14      A8  D0 00018             MOVL     20(R8), R10                               :  1133
                              59          18      A8  D0 0001C             MOVL     24(R8), R9                                :  1134
                              52    002400F8      8F  D0 00020             MOVL     #2359544, R2                              :  1137
                              51        01B6      8F  3C 00027             MOVZWL   #438, R1                                  :
                              50                  57  D0 0002C             MOVL     R7, R0                                    :
                                  00000000G       EF  16 0002F             JSB      EXCH$UTIL_BLOCK_CHECK                     :
                      08      AE                  67  D0 00035             MOVL     (R7), NAMB                                :  1138
                              52    010A00F7      8F  D0 00039             MOVL     #17432823, R2                             :  1139
                              51        01B7      8F  3C 00040             MOVZWL   #439, R1                                  :
                              50          08      AE  D0 00045             MOVL     NAMB, R0                                  :
                                  00000000G       EF  16 00049             JSB      EXCH$UTIL_BLOCK_CHECK                     :
                              52    041B00F3      8F  D0 0004F             MOVL     #68878579, R2                             :  1140
                              51        01B8      8F  3C 00056             MOVZWL   #440, P1                                  :
                              50                  58  D0 0005B             MOVL     R8, R0                                    :
                                  00000000G       EF  16 0005E             JSB      EXCH$UTIL_BLOCK_CHECK                     :
                                          08      AE  DD 00064             PUSHL    NAMB                                      :  1147
                 00000000G    EF                  01  FB 00067             CALLS    #1, EXCH$UTIL_NAMB_RELEASE                :
                                          08      AE  9F 0006E             PUSHAB   NAMB                                      :  1149
                                          0C      A7  9F 00071             PUSHAB   12(R7)                                    :
                                                  03  DD 00074             PUSHL    #3                                        :
                                                  7E  D4 00076             CLRL     -(SP)                                     :
                                          0C      A7  9F 00078             PUSHAB   12(R7)                                    :  1148
                 00000000G    EF                  05  FB 0007B             CALLS    #5, EXCH$CMD_PARSE_FILESPEC               :  1149
                              6E                  50  D0 00082             MOVL     R0, STATUS                                :
                              0E                  6E  E8 00085             BLBS     STATUS, 1$                                :  1150
                              52    00000000G     8F  D0 00088             MOVL     #EXCH$_PARSEERR, TEMP                     :  1152
                                                  6E  DD 0008F             PUSHL    STATUS                                    :
                                          0C      A7  9F 00091             PUSHAB   12(R7)                                    :
                                                  30  11 00094             BRB      2$                                        :
                              5B          08      AE  D0 00096 1$:         MOVL     NAMB, R11                                 :  1153
                              67                  5B  D0 0009A             MOVL     R11, (R7)                                 :
                                          0C      AE  9F 0009D             PUSHAB   VIR_NAMB                                  :  1158
                                          14      A7  9F 000A0             PUSHAB   20(R7)                                    :
                                                  7E  D4 000A3             CLRL     -(SP)                                     :
                                        0000'     CF  9F 000A5             PUSHAB   P.AAQ                                     :
                                        0000'     CF  9F 000A9             PUSHAB   P.AAO                                     :
                 00000000G    EF                  05  FB 000AD             CALLS    #5, EXCH$CMD_PARSE_FILESPEC               :
                              6E                  50  D0 000B4             MOVL     R0, STATUS                                :
                              1B                  6E  E8 000B7             BLBS     STATUS, 3$                                :  1159
```

```
                                        52 00000000G   8F   D0 000BA          MOVL    #EXCH$_PARSEERR, TEMP                  ; 1161
                                                       6E   DD 000C1          PUSHL   STATUS                                 ;
                                                14     A7   9F 000C3          PUSHAB  20(R7)                                 ;
                                                       01   DD 000C6 2$:      PUSHL   #1                                     ;
                                                       52   DD 000C8          PUSHL   TEMP                                   ;
                               00000000G   00          04   FB 000CA          CALLS   #4, LIB$SIGNAL                         ;
                                           50          52   D0 000D1          MOVL    TEMP, R0                               ;
                                                            04 000D4          RET                                            ;
                          50         0C    AE          18   C1 000D5 3$:      ADDL3   #24, VIR_NAMB, R0                      ; 1165
                                     04    AE          60   9E 000DA          MOVAB   (R0), 4(SP)                            ;
                                                       58   DD 000DE          PUSHL   R8                                     ; 1170
                                                08     AE   DD 000E0          PUSHL   8(SP)                                  ;
                               00000000G   EF          02   FB 000E3          CALLS   #2, EXCH$UTIL_VOL_GETDVI               ;
                                           6E          50   D0 000EA          MOVL    R0, STATUS                             ;
                                           17          6E   E8 000ED          BLBS    STATUS, 4$                             ;
                                                       6E   DD 000F0          PUSHL   STATUS                                 ; 1173
                                                08     AE   DD 000F2          PUSHL   8(SP)                                  ;
                                                       01   DD 000F5          PUSHL   #1                                     ;
                                          00000000G    8F   DD 000F7          PUSHL   #EXCH$_ACCESSFAIL                      ;
                               00000000G   00          04   FB 000FD          CALLS   #4, LIB$SIGNAL                         ;
                                                     00DD   31 00104          BRW     8$                                     ; 1174
                     69    A8      008A    CB     0080 8F   28 00107 4$:      MOVC3   #128, 138(R11), 105(R8)                ; 1187
                                     65    A8     0086 CB   D0 00110          MOVL    134(R11), 101(R8)                      ; 1188
   0050    8F              00              6E          00   2C 00116          MOVC5   #0, (SP), #0, #80, (R6)                ; 1208
                                                       66      0011D                                                         ;
                                           66     5003 8F   B0 0011E          MOVW    #20483, (R6)                           ;
                                     16    A6          22   90 00123          MOVB    #34, 22(R6)                            ;
                                     1F    A6          02   90 00127          MOVB    #2, 31(R6)                             ;
                                     24    A6       10 AE   9E 0012B          MOVAB   XAB, 36(R6)                            ;
                                     28    A6          59   D0 00130          MOVL    R9, 40(R6)                             ;
                          50         04    AE          04   C1 00134          ADDL3   #4, 4(SP), R0                          ;
                                     2C    A6          60   D0 00139          MOVL    (R0), 44(R6)                           ;
                                     34    A6       04 BE   90 0013D          MOVB    @4(SP), 52(R6)                         ;
   0044    8F              00              6E          00   2C 00142          MOVC5   #0, (SP), #0, #68, (R10)               ; 1212
                                                       6A      00149                                                         ;
                                           6A     4401 8F   B0 0014A          MOVW    #17409, (R10)                          ;
                                     04    AA     0800 8F   3C 0014F          MOVZWL  #2048, 4(R10)                          ;
                                     3C    AA          56   D0 00155          MOVL    R6, 60(R10)                            ;
   0060    8F              00              6E          00   2C 00159          MOVC5   #0, (SP), #0, #96, (R9)                ; 1218
                                                       69      00160                                                         ;
                                           69     6002 8F   B0 00161          MOVW    #24578, (R9)                           ;
                                     02    A9          01   8E 00166          MNEGB   #1, 2(R9)                              ;
                                     04    A9       20 A8   D0 0016A          MOVL    32(R8), 4(R9)                          ;
                                     0A    A9          01   8E 0016F          MNEGB   #1, 10(R9)                             ;
                                     0C    A9       1C A8   D0 00173          MOVL    28(R8), 12(R9)                         ;
           2C              00              6E          00   2C 00178          MOVC5   #0, (SP), #0, #44, $RMS_PTR            ; 1220
                                                    10 AE      0017D                                                         ;
                                     10    AE     2C1D 8F   B0 0017F          MOVW    #11293, $RMS_PTR                       ;
     16    A6              01              00       1C A7   F0 00185          INSV    28(R7), #0, #1, 22(R6)                 ; 1224
                                                       56   DD 0018C          PUSHL   R6                                     ; 1229
                               00000000G   00          01   FB 0018E          CALLS   #1, SYS$OPEN                           ;
                                           6E          50   D0 00195          MOVL    R0, STATUS                             ;
                                           25          6E   E8 00198          BLBS    STATUS, 6$                             ;
                               0001829A    8F          6E   D1 0019B          CMPL    STATUS, #98970                         ; 1235
                                                       17   12 001A2          BNEQ    5$                                     ;
                                           13       16 A6   E9 001A4          BLBC    22(R6), 5$                             ;
                                     16    A6          01   8A 001A8          BICB2   #1, 22(R6)                             ; 1238
```

```
                                          56  DD 001AC         PUSHL   R6                                        ; 1240
                          00000000G  00   01  FB 001AE         CALLS   #1, SYS$OPEN                              ;
                                     6E   50  D0 001B5         MOVL    R0, STATUS                                ;
                                     05   6E  E8 001B8         BLBS    STATUS, 6$                                ;
                                  0C      A6  DD 001BB  5$:    PUSHL   12(R6)                                    ; 1251
                                          12  11 001BE         BRB     7$                                        ;
                                          5A  DD 001C0  6$:    PUSHL   R10                                       ; 1258
                          00000000G  00   01  FB 001C2         CALLS   #1, SYS$CONNECT                           ;
                                     6E   50  D0 001C9         MOVL    R0, STATUS                                ;
                                     19   6E  E8 001CC         BLBS    STATUS, 9$                                ;
                                  0C      AA  DD 001CF         PUSHL   12(R10)                                   ; 1261
                                          56  DD 001D2  7$:    PUSHL   R6                                        ;
                                  08      AE  DD 001D4         PUSHL   STATUS                                    ;
                           00000000G      8F  DD 001D7         PUSHL   #EXCH$_OPENVIRTUAL                        ;
                          00000000G  EF   04  FB 001DD         CALLS   #4, EXCH$UTIL_FILE_ERROR                  ;
                                          6E  D4 001E4  8$:    CLRL    STATUS                                    ; 1262
                                          4A  11 001E6         BRB     15$                                       ; 1263
                                51   48   A8  9E 001E8  9$:    MOVAB   72(R8), R1                                ; 1271
                                     61   11  88 001EC         BISB2   #17, (R1)                                 ; 1272
           61          01            05  16  A6  F0 001EF      INSV    22(R6), #5, #1, (R1)                      ; 1273
                                  24      AE  B5 001F5         TSTW    XAB+20                                    ; 1275
                                          04  13 001F8         BEQL    10$                                       ;
                                          50  D4 001FA         CLRL    R0                                        ;
                                          03  11 001FC         BRB     11$                                       ;
                                     50   01  D0 001FE  10$:   MOVL    #1, R0                                    ;
                40     A8       20   AE   50  C3 00201  11$:   SUBL3   R0, XAB+16, 64(R8)                        ;
                                44   A8  40   A8  D0 00207     MOVL    64(R8), 68(R8)                            ; 1277
                                  24      A6  D4 0020C         CLRL    36(R6)                                    ; 1279
                       15     0085   CB   02  E1 0020F         BBC     #2, 133(R11), 13$                         ; 1283
                                     03  7A   AB  91 00215     CMPB    122(R11), #3                              ; 1286
                                          09  13 00219         BEQL    12$                                       ;
                                6E 00000000G  8F  D0 0021B     MOVL    #EXCH$_INVVOLFMT, STATUS                  ; 1288
                                          0E  11 00222         BRB     15$                                       ;
                                     61  40   8F  88 00224 12$: BISB2  #64, (R1)                                 ; 1291
                                          04  11 00228         BRB     14$                                       ; 1292
                                     61  40   8F  8A 0022A 13$: BICB2  #64, (R1)                                 ; 1297
                                58   A8       03  90 0022E 14$: MOVB   #3, 88(R8)                                ; 1298
                                  0C      AE  DD 00232  15$:   PUSHL   VIR_NAMB                                  ; 1303
                          00000000G  EF   01  FB 00235         CALLS   #1, EXCH$UTIL_NAMB_RELEASE                ;
                                     50   6E  D0 0023C         MOVL    STATUS, R0                                ; 1305
                                              04 0023F         RET                                               ; 1306

; Routine Size:  576 bytes,    Routine Base:  EXCH$MOUN_CODE + 08D2
```

```
GLOBAL ROUTINE exch$moun_vms_mount (volb : $ref_bblock, devname : $ref_bblock) =       %SBTTL 'exch$moun_vm
BEGIN
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       Call the $MOUNT system service to perform a mount on VMS
!
! INPUTS:
!
!       volb    - address of volume block describing the volume to be mounted
!       devname - address of string descriptor containing the device name
!
! IMPLICIT INPUTS:
!
!       none
!
! OUTPUTS:
!
!       none
!
! IMPLICIT OUTPUTS:
!
!       none
!
! ROUTINE VALUE:
!
!       Success or worst error encountered.
!
! SIDE EFFECTS:
!
!       A device may be mounted to VMS.
!--

$dbgtrc_prefix ('moun_vms_mount> ');

LOCAL
    mnt_item : VECTOR [10, LONG],               ! item list for $mount
    status
    ;

OWN
    flags : INITIAL (mnt$m_foreign OR mnt$m_noassist OR mnt$m_nounload);

$block_check (2, .volb, volb, 633);

! Prepare the item list
!
mnt_item [0] = (mnt$_devnam^16 OR .devname [dsc$w_length]);
mnt_item [1] = .devname [dsc$a_pointer];
mnt_item [2] = 0;

mnt_item [3] = (mnt$_volnam^16 OR 8);
mnt_item [4] = UPLIT BYTE ('Exchange');
mnt_item [5] = 0;

mnt_item [6] = (mnt$_flags^16 OR 4);
```

```
; 1278          1364  2 mnt_item [7] = flags;
; 1279          1365  2 mnt_item [8] = 0;
; 1280          1366  2
; 1281          1367  2 mnt_item [9] = 0;                                  ! End of list
; 1282          1368  2
; 1283          1369  2 ! Do the $mount service
; 1284          1370  2 !
; 1285          1371  3 IF NOT (status = $mount (itmlst=mnt_item))
; 1286          1372  2 THEN
; 1287          1373  3     BEGIN
; 1288          1374  3     BIND
; 1289          1375  3         status2 = status : $bblock [4]:
; 1290          1376  3     status2 [sts$v_inhib_msg] = 0;
; 1291          1377  3     $exch_signal_return (exch$_mounterror, 1, .devname, .status2);
; 1292          1378  3     END
; 1293          1379  2 ELSE
; 1294          1380  2     $exch_signal (exch$_vmsmount, 1, .devname);
; 1295          1381  2
; 1296          1382  2 RETURN .status;
; 1297          1383  1 END;


                                                                     .PSECT  EXCH$MOUN_PLIT,NOWRT,2

                                              00100005  000A8 FLAGS:  .LONG   1048581                                  ;
                    65  67  6E  61  68  63  78  45  000AC P.AAS:  .ASCII  \Exchange\                               ;

                                                                     .EXTRN  SYS$MOUNT, EXCH$_MOUNTERROR
                                                                     .EXTRN  EXCH$_VMSMOUNT

                                                                     .PSECT  EXCH$MOUN_CODE,NOWRT,2

                                           003C 00000              .ENTRY  EXCH$MOUN_VMS_MOUNT, Save R2,R3,R4,R5     ; 1307
                           55 00000000G  00   9E 00002              MOVAB   LIB$SIGNAL, R5                           ;
                           5E            28   C2 00009              SUBL2   #40, SP                                  ;
                           52 041B00F3   8F   D0 0000C              MOVL    #68878579, R2                            ; 1351
                           51     0279   8F   3C 00013              MOVZWL  #633, R1                                 ;
                           50       04   AC   D0 00018              MOVL    VOLB, R0                                 ;
                              00000000G  EF   16 0001C              JSB     EXCH$UTIL_BLOCK_CHECK                    ;
                           53       08   AC   D0 00022              MOVL    DEVNAME, R3                              ; 1355
                           50            63   3C 00026              MOVZWL  (R3), R0                                 ;
                6E         50 00010000   8F   C9 00029              BISL3   #65536, R0, MNT_ITEM                     ;
                      04   AE       04   A3   D0 00031              MOVL    4(R3), MNT_ITEM+4                        ; 1356
                                    08   AE   D4 00036              CLRL    MNT_ITEM+8                               ; 1357
                      0C   AE 00020008   8F   D0 00039              MOVL    #131080, MNT_ITEM+12                     ; 1359
                      10   AE     0000'  CF   9E 00041              MOVAB   P.AAS, MNT_ITEM+16                       ; 1360
                                    14   AE   D4 00047              CLRL    MNT_ITEM+20                              ; 1361
                      18   AE 00040004   8F   D0 0004A              MOVL    #262148, MNT_ITEM+24                     ; 1363
                      1C   AE     0000'  CF   9E 00052              MOVAB   FLAGS, MNT_ITEM+28                       ; 1364
                                    20   AE   7C 00058              CLRQ    MNT_ITEM+32                              ; 1365
                                         5E   DD 0005B              PUSHL   SP                                       ; 1371
                 00000000G  00           01   FB 0005D              CALLS   #1, SYS$MOUNT                            ;
                           54            50   D0 00064              MOVL    R0, STATUS                               ;
                           19            54   E8 00067              BLBS    STATUS, 1$                               ;
     54            01      1C            00   F0 0006A              INSV    #0, #28, #1, STATUS2                     ; 1376
                           52 00000000G  8F   D0 0006F              MOVL    #EXCH$_MOUNTERROR, TEMP                  ; 1377
```

```
                                    18  BB 00076          PUSHR   #^M<R3,R4>                        ;
                                    01  DD 00078          PUSHL   #1                                ;
                                    52  DD 0007A          PUSHL   TEMP                              ;
                       65           04  FB 0007C          CALLS   #4, LIB$SIGNAL                    ;
                       50           52  D0 0007F          MOVL    TEMP, R0                          ;
                                        04 00082          RET                                       ;
                                    53  DD 00083 1$:      PUSHL   R3                                ;  1380
                                    01  DD 00085          PUSHL   #1                                ;
                          00000000G 8F  DD 00087          PUSHL   #EXCH$_VMSMOUNT                   ;
                       65           03  FB 0008D          CALLS   #3, LIB$SIGNAL                    ;
                       50           54  D0 00090          MOVL    STATUS, R0                        ;  1382
                                        04 00093          RET                                       ;  1383

; Routine Size:  148 bytes,    Routine Base:  EXCH$MOUN_CODE + 0B12
; 1299          1384  1 END
; 1300          1385  0 ELUDOM

                                                                  .EXTRN  LIB$SIGNAL, LIB$STOP

;                                    PSECT SUMMARY
;
;      Name                         Bytes                     Attributes
;
;   EXCH$MOUN_PLIT                    180  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   EXCH$MOUN_CODE                   2982  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                             Library Statistics
;
;                                            -------- Symbols --------       Pages        Processing
;      File                                  Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1            18619       144         0       1000           00:01.9
;   _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1      1151       119        10         79           00:01.3


;                                     COMMAND QUALIFIERS

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

; Size:          2982 code + 180 data bytes
; Run Time:         00:56.3
; Elapsed Time:     02:59.3
; Lines/CPU Min:     1475
; Lexemes/CPU-Min: 28251
; Memory Used:  296 pages
; Compilation Complete
```

0162 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY