```
                                                       G  3

EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCCC  HHH         HHH  NNN         NNN     GGGGGGGGGGG
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
```

FILEID**EXCINIT

```
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII    NN      NN    IIIIII    TTTTTTTTTT
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII    NN      NN    IIIIII    TTTTTTTTTT
EE          XX      XX  CC              II      NN      NN      II          TT
EE          XX      XX  CC              II      NN      NN      II          TT
EE            XX  XX    CC              II      NNNN    NN      II          TT
EE            XX  XX    CC              II      NNNN    NN      II          TT
EEEEEEEE        XX      CC              II      NN  NN  NN      II          TT
EEEEEEEE        XX      CC              II      NN  NN  NN      II          TT
EE            XX  XX    CC              II      NN    NNNN      II          TT
EE            XX  XX    CC              II      NN    NNNN      II          TT
EE          XX      XX  CC              II      NN      NN      II          TT          ....
EE          XX      XX  CC              II      NN      NN      II          TT          ....
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII    NN      NN    IIIIII        TT          ....
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII    NN      NN    IIIIII        TT          ....

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
```

```
0001  0 MODULE  exch$init                               %TITLE 'INIT verb dispatch and misc routines'
0002  0                 (
0003  0                 IDENT = 'V04-000',
0004  0                 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
0005  0                 ) =
0006  1 BEGIN
0007  1 !
0008  1 !****************************************************************************
0009  1 !*                                                                          *
0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0012  1 !*  ALL RIGHTS RESERVED.                                                    *
0013  1 !*                                                                          *
0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0019  1 !*  TRANSFERRED.                                                            *
0020  1 !*                                                                          *
0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0023  1 !*  CORPORATION.                                                            *
0024  1 !*                                                                          *
0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0027  1 !*                                                                          *
0028  1 !*                                                                          *
0029  1 !****************************************************************************
0030  1
0031  1 !++
0032  1 ! FACILITY:     EXCHANGE - Foreign volume interchange facility
0033  1 !
0034  1 ! ABSTRACT:     Primary action routines for INIT verb
0035  1 !
0036  1 ! ENVIRONMENT:  VAX/VMS User mode
0037  1 !
0038  1 ! AUTHOR:       CW Hobbs          CREATION DATE: 04-Jan-1983
0039  1 !
0040  1 ! MODIFIED BY:
0041  1 !
0042  1 !       V03-002 CWH3002         CW Hobbs                12-Apr-1984
0043  1 !               Signal a specific error for an attempt to access a remote node
0044  1 !
0045  1 !
0046  1 !--
0047  1
0048  1 ! Include files:
0049  1 !
0050  1 MACRO $module_name_string = 'exch$init' %;      ! The require file needs to know our module name
0051  1 REQUIRE 'SRC$:EXCREQ'                           ! Facility-wide require file
0052  1     ;
```

```
1 %SBTTL 'Module table of contents'
1
1 ! Module table of contents:
1 !
1 FORWARD ROUTINE
1             init_dos11_init,                              ! DOS-11 volume init processing
1             init_foreign_close,                           ! Close a foreign volume
1             init_foreign_create,                          ! Open a file to a foreign virtual volume
1             init_foreign_open,                            ! Open a file to a foreign device
1             init_init : NOVALUE,                          ! Setups
1        exch$init_initialize,                              ! Main action routine
1             init_rt11_init,                               ! RT-11 volume init processing
1             init_zero_home_blocks                         ! Zero Files-11 home blocks
1        ;
1
1 ! EXCHANGE facility routines
1 !
1 EXTERNAL ROUTINE
1        exch$cmd_cli_get_integer,                          ! Get the value of an integer qualifier
1        exch$cmd_parse_filespec,                           ! Parse a file specification
1        exch$io_dos11_rewind,                              ! Rewind the sequential device
1        exch$io_dos11_set_density,                         ! Set magtape density
1        exch$io_dos11_write_tape_mark,                     ! Write a tape mark
1        exch$io_rt11_write,                                ! Write blocks to RT11 device
1        exch$moun_vms_mount,                               ! Perform VMS $mount service to mount volume
1        exch$rt11_format_current_date : NOVALUE jsb_r1,
1        exch$rtacp_verify_directory,                       ! Check for valid RT-11 directory
1        exch$util_file_error,                              ! Signal RMS error
1        exch$util_namb_release      : NOVALUE,             ! Release name block
1        exch$util_vm_allocate_zeroed,                      ! Allocate virtual memory
1        exch$util_vm_release        : NOVALUE,             ! Release memory
1        exch$util_vol_getdvi,                              ! Get device information
1        exch$util_volb_release      : NOVALUE,             ! Release volume block
1        exch$util_volb_allocate                            ! Allocate volume block
1        ;
1
1 ! Equated symbols:
1 !
1 !LITERAL
1 !     ;
1
1 ! Bound declarations:
1 !
1 !BIND
1 !     ;
```

```
  1 GLOBAL ROUTINE init_dos11_init =        %SBTTL 'init_dos11_init'
  2 BEGIN
  2 !++
  2 !
  2 ! FUNCTIONAL DESCRIPTION:
  2 !
  2 !       Perform dos11 volume specific init actions
  2 !
  2 ! INPUTS:
  2 !
  2 !       none
  2 !
  2 ! IMPLICIT INPUTS:
  2 !
  2 !       work area for INIT
  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 !       dos11 tape will be initialized
  2 !--
  2
  2 $dbgtrc_prefix ('init_dos11_init> ');
  2
  2 LOCAL
  2     dens,
  2     dosv : $ref_bblock,
  2     ent : $ref_bblock,
  2     status
  2     ;
  2
  2 BIND
  2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock, ! pointer to our work area
  2     volb = init [init$a_volb]            : $ref_bblock  ! pointer to exchange VOLB structure
  2     ;
  2
  2 $block_check (2, .init, init, 604);
  2 $block_check (2, .volb, volb, 605);
  2
  2 ! Make sure that we can do it
  2 !
  2 IF NOT .volb [volb$v_write]
  2 THEN
P 2     $exch_signal_return ($warning_stat_copy (exch$_writelock), 2,
  2                             .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
  2
```

```
2 ! Allocate and initialize our volb extension if it does not exist
2 !
2 dosv = .volb [volb$a_vfmt_specific];
2 IF .dosv EQL 0
2 THEN
3     BEGIN
3     dosv = exch$util_vm_allocate_zeroed (exchblk$s_dos11);      ! Get the memory
3     volb [volb$a_vfmt_specific] = .dosv;                        ! Stash the address in the volb
3     $block_init (.dosv, dos11);                                 ! Set the type
3     $queue_initialize (dosv [dos11$q_entry_header]);            ! Init the directory cache queue
2     END;
2
2 ! Rewind the magtape, then write two tape marks, then rewind the tape again
2 !
3 IF (status = exch$io_dos11_rewind (.volb))
2 THEN
3     IF (status = exch$io_dos11_write_tape_mark (.volb))
2     THEN
3         IF (status = exch$io_dos11_write_tape_mark (.volb))
2         THEN
2             status = exch$io_dos11_rewind (.volb);
2
2 ! If the /DENSITY qualifier is present, set the drive to the new density.  Tape must be at BOT to change den
2 !
2 IF .status
2 THEN
2     IF cli$present (%ASCID 'DENSITY')
2     THEN
2         status = exch$io_dos11_set_density (.volb);
2
2 ! If there is a cached "directory", release it
2 !
2 IF .dosv [dos11$a_entry_flink] NEQ 0
2 THEN
3     WHILE ((ent = $queue_remove_head (dosv [dos11$q_entry_header])) NEQ 0)
2     DO
2         exch$util_vm_release (dos11ent$k_length, .ent);
2
2 RETURN .status;
1 END;


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

                                                         .PSECT  EXCH$INIT_PLIT,NOWRT,2

    00  59  54  49  53  4E  45  44  00000 P.AAB:     .ASCII  \DENSITY\<0>                                    ;
                             010E0007  00008 P.AAA:     .LONG   17694727                                         ;
                             00000000' 0000C            .ADDRESS P.AAB                                           ;

                                                         .EXTRN  EXCH$CMD_CLI_GET_INTEGER
                                                         .EXTRN  EXCH$CMD_PARSE_FILESPEC
                                                         .EXTRN  EXCH$IO_DOS11_REWIND
                                                         .EXTRN  EXCH$IO_DOS11_SET_DENSITY
                                                         .EXTRN  EXCH$IO_DOS11_WRITE_TAPE_MARK
                                                         .EXTRN  EXCH$IO_RT11_WRITE
```

```
                                                              .EXTRN   EXCH$MOUN_VMS_MOUNT
                                                              .EXTRN   EXCH$RT11_FORMAT_CURRENT_DATE
                                                              .EXTRN   EXCH$RTACP_VERIFY_DIRECTORY
                                                              .EXTRN   EXCH$UTIL_FILE_ERROR
                                                              .EXTRN   EXCH$UTIL_NAMB_RELEASE
                                                              .EXTRN   EXCH$UTIL_VM_ALLOCATE_ZEROED
                                                              .EXTRN   EXCH$UTIL_VM_RELEASE
                                                              .EXTRN   EXCH$UTIL_VOC_GETDVI
                                                              .EXTRN   EXCH$UTIL_VOLB_RELEASE
                                                              .EXTRN   EXCH$UTIL_VOLB_ALLOCATE
                                                              .EXTRN   EXCH$A_GBL, EXCH$UTIL_BLOCK_CHECK
                                                              .EXTRN   EXCH$_WRITELOCK
                                                              .EXTRN   CLI$PRESENT

                                                              .PSECT   EXCH$INIT_CODE,NOWRT,2

                                    00FC 00000                .ENTRY   INIT_DOS11_INIT, Save R2,R3,R4,R5,R6,R7       ; 0194
            57 00000000G  EF   9E 00002                       MOVAB    EXCH$IO_DOS11_WRITE_TAPE_MARK, R7             ;
            56 00000000G  EF   9E 00009                       MOVAB    EXCH$IO_DOS11_REWIND, R6                      ;
            55 00000000G  EF   9E 00010                       MOVAB    EXCH$UTIL_BLOCK_CHECK, R5                     ;
 53 00000000G  EF          10  C1 00017                       ADDL3    #16, EXCH$A_GBL, R3                           ; 0237
 54            63          04  C1 0001F                       ADDL3    #4, (R3), R4                                  ; 0238
            52 002C00F9   8F   D0 00023                       MOVL     #2883833, R2                                  ; 0241
            51      025C  8F   3C 0002A                       MOVZWL   #604, R1                                      ;
            50            63   D0 0002F                       MOVL     (R3), R0                                      ;
                          65   16 00032                       JSB      EXCH$UTIL_BLOCK_CHECK                         ;
            53            64   D0 00034                       MOVL     (R4), R3                                      ; 0242
            52 041B00F3   8F   D0 00037                       MOVL     #68878579, R2                                 ;
            51      025D  8F   3C 0003E                       MOVZWL   #605, R1                                      ;
            50            53   D0 00043                       MOVL     R3, R0                                        ;
                          65   16 00046                       JSB      EXCH$UTIL_BLOCK_CHECK                         ;
 22         48  A3          05  E0 00048                      BBS      #5, 72(R3), 1$                                ; 0246
            50 00000000G  8F   D0 0004D                       MOVL     #EXCH$_WRITELOCK, STATUS2                     ; 0249
            50            07   8A 00054                       BICB2    #7, STATUS2                                   ;
            52            50   D0 00057                       MOVL     STATUS2, TEMP                                 ;
                    69    A3   9F 0005A                       PUSHAB   105(R3)                                       ;
                    65    A3   DD 0005D                       PUSHL    101(R3)                                       ;
                          02   DD 00060                       PUSHL    #2                                            ;
                          52   DD 00062                       PUSHL    TEMP                                          ;
      00000000G  00       04   FB 00064                       CALLS    #4, LIB$SIGNAL                                ;
            50            52   D0 0006B                       MOVL     TEMP, R0                                      ;
                               04 0006E                       RET                                                    ;
            52      54    A3   D0 0006F 1$:                   MOVL     84(R3), DOSV                                  ; 0253
                          23   12 00073                       BNEQ     2$                                            ; 0254
                          36   DD 00075                       PUSHL    #54                                           ; 0257
      00000000G  EF       01   FB 00077                       CALLS    #1, EXCH$UTIL_VM_ALLOCATE_ZEROED              ;
            52            50   D0 0007E                       MOVL     R0, DOSV                                      ;
        54  A3            52   D0 00081                       MOVL     DOSV, 84(R3)                                  ; 0258
        08  A2            36   B0 00085                       MOVW     #54, 8(DOSV)                                  ; 0259
        0A  A2            03   8E 00089                       MNEGB    #3, 10(DOSV)                                  ;
            50      12    A2   9E 0008D                       MOVAB    18(DOSV), R0                                  ; 0260
            60            50   D0 00091                       MOVL     R0, (R0)                                      ;
        04  A0            50   D0 00094                       MOVL     R0, 4(R0)                                     ;
                          53   DD 00098 2$:                   PUSHL    R3                                            ; 0265
            66            01   FB 0009A                       CALLS    #1, EXCH$IO_DOS11_REWIND                      ;
            54            50   D0 0009D                       MOVL     R0, STATUS                                    ;
            3B            54   E9 000A0                       BLBC     STATUS, 3$                                    ;
```

```
                                    53  DD 000A3          PUSHL   R3                                    ; 0267
                          67        01  FB 000A5          CALLS   #1, EXCH$IO_DOS11_WRITE_TAPE_MARK     ;
                          54        50  D0 000A8          MOVL    R0, STATUS                            ;
                          30        54  E9 000AB          BLBC    STATUS, 3$                            ;
                                    53  DD 000AE          PUSHL   R3                                    ; 0269
                          67        01  FB 000B0          CALLS   #1, EXCH$IO_DOS11_WRITE_TAPE_MARK     ;
                          54        50  D0 000B3          MOVL    R0, STATUS                            ;
                          25        54  E9 000B6          BLBC    STATUS, 3$                            ;
                                    53  DD 000B9          PUSHL   R3                                    ; 0271
                          66        01  FB 000BB          CALLS   #1, EXCH$IO_DOS11_REWIND              ;
                          54        50  D0 000BE          MOVL    R0, STATUS                            ;
                          1A        54  E9 000C1          BLBC    STATUS, 3$                            ; 0275
                          0000'     CF  9F 000C4          PUSHAB  P.AAA                                 ; 0277
                00000000G 00        01  FB 000C8          CALLS   #1, CLI$PRESENT                       ;
                          0C        50  E9 000CF          BLBC    R0, 3$                                ;
                                    53  DD 000D2          PUSHL   R3                                    ; 0279
                00000000G EF        01  FB 000D4          CALLS   #1, EXCH$IO_DOS11_SET_DENSITY         ;
                          54        50  D0 000DB          MOVL    R0, STATUS                            ;
                          12        A2  D5 000DE 3$:      TSTL    18(DOSV)                              ; 0283
                                    1C  13 000E1          BEQL    7$                                    ;
                          50  12    B2  0F 000E3 4$:      REMQUE  @18(DOSV), _T_                        ; 0285
                                    04  1C 000E7          BVC     5$                                    ;
                                    53  D4 000E9          CLRL    ENT                                   ;
                                    03  11 000EB          BRB     6$                                    ;
                          53        50  D0 000ED 5$:      MOVL    _T_, ENT                              ;
                                    0D  13 000F0 6$:      BEQL    7$                                    ;
                                    53  DD 000F2          PUSHL   ENT                                   ; 0287
                                    1C  DD 000F4          PUSHL   #28                                   ;
                00000000G EF        02  FB 000F6          CALLS   #2, EXCH$UTIL_VM_RELEASE              ;
                                    E4  11 000FD          BRB     4$                                    ;
                          50        54  D0 000FF 7$:      MOVL    STATUS, R0                            ; 0289
                                        04 00102          RET                                           ; 0290

; Routine Size:  259 bytes,    Routine Base:  EXCH$INIT_CODE + 0000
```

```
1 GLOBAL ROUTINE init_foreign_close =     %SBTTL 'init_foreign_close'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Close a temporarily opened foreign device.
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       INIT verb work area
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       work area
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error encountered.
2 !
2 ! SIDE EFFECTS:
2 !
2 !       A file is no longer open on the volb
2 !--
2
2 $dbgtrc_prefix ('init_foreign_close> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock, ! pointer to our work area
2     volb = .init [init$a_volb] : $bblock,               ! Pointer to exchange VOLB structure
2     fab  = .volb [volb$a_fab] : $bblock                 ! File Access Block for the volume
2     ;
2
2 $block_check (2, volb, volb, 575);
2
2 ! Close the open RMS link to the volume
2 !
3 IF NOT (status = $close (fab = fab))
2 THEN
2     RETURN exch$util_file_error (exch$_closeforeign  .status, fab, .fab [fab$l_stv]);
2
2 RETURN .status;
1 END;
```

```
                                                             .EXTRN  SYS$CLOSE, EXCH$_CLOSEFOREIGN

                                              000C 00000     .ENTRY  INIT_FOREIGN_CLOSE, Save R2,R3          ; 0291
             50 00000000G  EF            10  C1 00002        ADDL3   #16, EXCH$A_GBL, R0                     ; 0331
                           50            60  D0 0000A        MOVL    (R0), R0                                ; 0332
                           50        04  A0  D0 0000D        MOVL    4(R0), R0
                           53        10  A0  D0 00011        MOVL    16(R0), R3                              ; 0333
                           52 041B00F3   8F  D0 00015        MOVL    #68878579, R2                           ; 0336
                           51      023F  8F  3C 0001C        MOVZWL  #575, R1
                              00000000G  EF  16 00021        JSB     EXCH$UTIL_BLOCK_CHECK
                                             53  DD 00027    PUSHL   R3                                      ; 0340
                00000000G  00            01  FB 00029        CALLS   #1, SYS$CLOSE
                           52            50  D0 00030        MOVL    R0, STATUS
                           13            52  E8 00033        BLBS    STATUS, 1$
                                     0C  A3  DD 00036        PUSHL   12(R3)                                  ; 0342
                                         0C  BB 00039        PUSHR   #^M<R2,R3>
                              00000000G  8F  DD 0003B        PUSHL   #EXCH$_CLOSEFOREIGN
                00000000G  EF            04  FB 00041        CALLS   #4, EXCH$UTIL_FILE_ERROR
                                             04 00048        RET
                           50            52  D0 00049 1$:    MOVL    STATUS, R0                              ; 0344
                                             04 0004C        RET                                             ; 0345

; Routine Size:  77 bytes,    Routine Base:  EXCH$INIT_CODE + 0103
```

```
1 GLOBAL ROUTINE init_foreign_create =    %SBTTL 'init_foreign_create'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Create a foreign virtual volume with RMS so that we may initialize it.
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       namb - name block describing the device
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 $dbgtrc_prefix ('init_foreign_create> ');
2
2 LOCAL
2     len,
2     snum,
2     start,
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock, ! pointer to our work area
2     fildesc = init [init$q_device] : $bblock,           ! file name
2     namb = .init [init$a_namb] : $bblock,               ! Pointer to exchange NAMB structure
2     volb = .init [init$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 $trace_print_lit ('entry');
2 $block_check (2, .init, init, 630);
2 $block_check (2, namb, namb, 631);
2 $block_check (2, volb, volb, 632);
```

```
:   311          0403  2
:   312          0404  2 ! Copy the input name to the volb for the signal
:   313          0405  2 !
:   314          0406  2 len = MINU (volb$s_vol_ident, .fildesc [dsc$w_length]);
:   315          0407  2 CH$MOVE (.len, .fildesc [dsc$a_pointer], volb [volb$t_vol_ident]);
:   316          0408  2 volb [volb$l_vol_ident_len] = .len;
:   317          0409  2
:   318          0410  2 ! Determine the number of device blocks
:   319          0411  2 !
:   320          0412  4 len =  (BEGIN
:   321          0413  4         LOCAL
:   322          0414  4             bmax;
:   323          0415  4         bmax = MINU (65535, .init [init$l_q_allocation]);
:   324          0416  4         IF .bmax EQL 0
:   325          0417  4         THEN
:   326          0418  4             bmax = 494;                      ! Default to single density diskette
:   327          0419  4         IF .init [init$l_q_allocation] GTRU .bmax
:   328          0420  4         THEN
:   329          0421  4             $exch_signal (exch$_rt11_toomanyblk, 1, .bmax);
:   330          0422  4         .bmax
:   331          0423  2         END);
:   332          0424  2
:   333          0425  2 ! Determine the number of directory segments, so that we can put a floor on the size of the file
:   334          0426  2 !
:   335          0427  3 snum = (SELECTONE true OF
:   336          0428  3         SET
:   337          0429  3         [.init [init$l_q_segments] NEQ 0] :      .init [init$l_q_segments];
:   338          0430  3         [.len LEQU 512] :                           1;
:   339          0431  3         [.len LEQU 2048] :                          4;
:   340          0432  3         [.len LEQU 12288] :                        16;
:   341          0433  3         [OTHERWISE] :                              31;
:   342          0434  2         TES);
:   343          0435  2
:   344          0436  2 ! Apply the floor and determine the number of blocks
:   345          0437  2 !
:   346          0438  2 start = rt11$k_root_block + (2 * .snum);
:   347          0439  2 len = MAXU (.start+32, .len);                ! Make it at least 32 blocks for files
:   348          0440  2 volb [volb$l_devmaxblock] = .len;            ! We need to save it here too
:   349          0441  2 volb [volb$l_volmaxblock] = .len;            ! We need to save it here too
:   350          0442  2
:   351          0443  2 ! Init the RMS blocks for the volume
:   352          0444  2 !
:   353        P 0445  2 $fab_init (
:   354        P 0446  2         FAB = fab,                           ! Volume FAB
:   355        P 0447  2         ALQ = .len,                          !  Allocation quantity
:   356        P 0448  2         FAC = (BIO,GET,PUT),                 !  Block I/O, read and write
:   357        P 0449  2         FNA = .fildesc [dsc$a_pointer],      !  Set name addr
:   358        P 0450  2         FNS = .fildesc [dsc$w_length],       !  Set name size
:   359        P 0451  2         DNA = UPLIT BYTE ('VIRTUAL.DSK'),    !  Default name address
:   360        P 0452  2         DNS = 11,                            !  Default name size
:   361        P 0453  2         MRS = 512,                           !  Records size
:   362        P 0454  2         RAT = CR,                            !  Carriage return
    363        P 0455  2         RFM = FIX,                           !  Fixed length records
.   364          0456  2         NAM = nam);                          !  Name block
:   365        P 0457  2 $rab_init (
:   366        P 0458  2         RAB = rab,                           ! Volume RAB
:   367        P 0459  2         ROP = BIO,                           !  Block I/O
:   368          0460  2           FAB = fab);                            !  FAB addr
:   369        P 0461  2  $nam_init (
:   370        P 0462  2           NAM = nam,                             ! File name block
:   371        P 0463  2           RSA = .volb [volb$a_rsbuf],            !  Result name addr
:   372        P 0464  2           RSS = nam$c_maxrss,                    !  Result name size
:   373        P 0465  2           ESA = .volb [volb$a_esbuf],            !  Expanded name addr
:   374          0466  2           ESS = nam$c_maxrss);                   !  Expanded name size
:   375          0467  2
:   376          0468  2  ! Create and connect to the volume
:   377          0469  2  !
:   378          0470  3  IF NOT (status = $create (fab = fab))
:   379          0471  2  THEN
:   380          0472  2      RETURN exch$util_file_error (exch$_createvirt, .status, fab, .fab [fab$l_stv]);
:   381          0473  2
:   382          0474  2  ! Now put as much of the result name into the volb as we can
:   383          0475  2  !
:   384          0476  2  len = MINU (volb$s_vol_ident, .nam [nam$b_rsl]);
:   385          0477  2  CH$MOVE (.len, .nam [nam$l_rsa], volb [volb$t_vol_ident]);
:   386          0478  2  volb [volb$l_vol_ident_len] = .len;
:   387          0479  2
:   388          0480  2  volb [volb$w_channel] = .fab [fab$l_stv];       ! Save the channel number (NFS ==> user mode channel)
:   389          0481  2
:   390          0482  3  IF NOT (status = $connect (rab = rab))
:   391          0483  2  THEN
:   392          0484  2      RETURN exch$util_file_error (exch$_createvirt, .status, fab, .rab [rab$l_stv]);
:   393          0485  2
:   394          0486  2  ! Set the volume format and other bits and pieces
:   395          0487  2  !
:   396          0488  2  volb [volb$b_vol_format] = volb$k_vfmt_rt11;
:   397          0489  2  volb [volb$v_write] = true;
:   398          0490  2  volb [volb$v_virtual] = true;
:   399          0491  2
:   400          0492  2  ! Write the last block to set the eof block correctly
:   401          0493  2  !
:   402          0494  3  IF NOT (status = exch$io_rt11_write (volb, .volb [volb$l_volmaxblock]-1, 1, exch$io_rt11_write))
:   403          0495  2  THEN
:   404          0496  2      RETURN .status;
:   405          0497  2
:   406          0498  2  RETURN true;
:   407          0499  1  END;



                                                                        .PSECT  EXCH$INIT_PLIT,NOWRT,2

             4B  53  44  2E  4C  41  55  54  52  49  56  00010 P.AAC:   .ASCII  \VIRTUAL.DSK\                                ;

                                                                        .EXTRN  EXCH$_RT11_TOOMANYBLK
                                                                        .EXTRN  SYS$CREATE, EXCH$_CREATEVIRT
                                                                        .EXTRN  SYS$CONNECT

                                                                        .PSECT  EXCH$INIT_CODE,NOWRT,2

                                                    0FFC 00000          .ENTRY  INIT_FOREIGN_CREATE, Save R2,R3,R4,R5,R6,-   ; 0346
                                                                                R7,R8,R9,R10,R11                             ;
                    50 00000000G  EF          10  C1 00002              ADDL3   #16, EXCH$A_GBL, R0                          ; 0389
                                  58          60  D0 0000A              MOVL    (R0), R8                                     ; 0390
```

```
                                    0C   A8  9F 0000D           PUSHAB  12(R8)                              ; 0391
                           53            68  D0 00010           MOVL    (R8), R3                            ; 0392
                           57       04   A8  D0 00013           MOVL    4(R8), R7                           ; 0393
                           56       10   A7  D0 00017           MOVL    16(R7), R6                          ; 0394
                           5A       14   A7  D0 0001B           MOVL    20(R7), R10                         ; 0395
                           59       18   A7  D0 0001F           MOVL    24(R7), R9                          ; 0400
                           52 002C00F9   8F  D0 00023           MOVL    #2883833, R2                        ; 0400
                           51     0276   8F  3C 0002A           MOVZWL  #630, R1                            ;
                           50            58  D0 0002F           MOVL    R8, R0                              ;
                              00000000G  EF  16 00032           JSB     EXCH$UTIL_BLOCK_CHECK               ;
                           52 010A00F7   8F  D0 00038           MOVL    #17432823, R2                       ; 0401
                           51     0277   8F  3C 0003F           MOVZWL  #631, R1                            ;
                           50            53  D0 00044           MOVL    R3, R0                              ;
                              00000000G  EF  16 00047           JSB     EXCH$UTIL_BLOCK_CHECK               ;
                           52 041B00F3   8F  D0 0004D           MOVL    #68878579, R2                       ; 0402
                           51     0278   8F  3C 00054           MOVZWL  #632, R1                            ;
                           50            57  D0 00059           MOVL    R7, R0                              ;
                              00000000G  EF  16 0005C           JSB     EXCH$UTIL_BLOCK_CHECK               ;
                           50       00   BE  3C 00062           MOVZWL  @0(SP), R0                          ; 0406
                  0080     8F            50  B1 00066           CMPW    R0, #128                            ;
                                         04  1B 0006B           BLEQU   1$                                  ;
                           50       80   8F  9A 0006D           MOVZBL  #128, R0                            ;
                           5B            50  D0 00071 1$:       MOVL    R0, LEN                             ;
               7E          6E            04  C1 00074           ADDL3   #4, (SP), -(SP)                     ; 0407
                                         9E  DD 00078           PUSHL   @(SP)+                              ;
       69    A7            9E            5B  28 0007A           MOVC3   LEN, @(SP)+, 105(R7)                ;
                      65   A7            5B  D0 0007F           MOVL    LEN, 101(R7)                        ; 0408
                           50       1C   A8  D0 00083           MOVL    28(R8), R0                          ; 0415
              0000FFFF     8F            50  D1 00087           CMPL    R0, #65535                          ;
                                         05  1B 0008E           BLEQU   2$                                  ;
                           50     FFFF   8F  3C 00090           MOVZWL  #65535, R0                          ;
                           52            50  D0 00095 2$:       MOVL    R0, BMAX                            ;
                                         05  12 00098           BNEQ    3$                                  ; 0416
                           52     01EE   8F  3C 0009A           MOVZWL  #494, BMAX                          ; 0418
                           52       1C   A8  D1 0009F 3$:       CMPL    28(R8), BMAX                        ; 0419
                                         11  1B 000A3           BLEQU   4$                                  ;
                                         52  DD 000A5           PUSHL   BMAX                                ; 0421
                                         01  DD 000A7           PUSHL   #1                                  ;
                              00000000G  8F  DD 000A9           PUSHL   #EXCH$_RT11_TOOMANYBLK              ;
             00000000G     00            03  FB 000AF           CALLS   #3, LIB$SIGNAL                      ;
                           5B            52  D0 000B6 4$:       MOVL    BMAX, LEN                           ; 0422
                                    24   A8  D5 000B9           TSTL    36(R8)                              ; 0429
                                         06  13 000BC           BEQL    5$                                  ;
                           50       24   A8  D0 000BE           MOVL    36(R8), SNUM                        ;
                                         2D  11 000C2           BRB     9$                                  ;
              00000200     8F            5B  D1 000C4 5$:       CMPL    LEN, #512                           ; 0430
                                         05  1A 000CB           BGTRU   6$                                  ;
                           50            01  D0 000CD           MOVL    #1, SNUM                            ;
                                         1F  11 000D0           BRB     9$                                  ;
              00000800     8F            5B  D1 000D2 6$:       CMPL    LEN, #2048                          ; 0431
                                         05  1A 000D9           BGTRU   7$                                  ;
                           50            04  D0 000DB           MOVL    #4, SNUM                            ;
                                         11  11 000DE           BRB     9$                                  ;
              00003000     8F            5B  D1 000E0 7$:       CMPL    LEN, #12288                         ; 0432
                                         05  1A 000E7           BGTRU   8$                                  ;
                           50            10  D0 000E9           MOVL    #16, SNUM                           ;
                                         03  11 000EC           BRB     9$                                  ;
```

```
                                  50         1F  D0 000EE 8$:    MOVL    #31, SNUM                           ; 0433
                                  50         02  C4 000F1 9$:    MULL2   #2, START                           ; 0438
                                  50         26  C0 000F4        ADDL2   #38, R0                             ; 0439
                                  5B         50  D1 000F7        CMPL    R0, LEN                             ;
                                  03         1E  000FA           BGEQU   10$                                 ;
                                  50         5B  D0 000FC        MOVL    LEN, R0                             ;
                                  5B         50  D0 000FF 10$:   MOVL    R0, LEN                             ;
                              40  A7         5B  D0 00102        MOVL    LEN, 64(R7)                         ; 0440
                              44  A7         5B  D0 00106        MOVL    LEN, 68(R7)                         ; 0441
  0050   8F            00         6E         00  2C 0010A        MOVC5   #0, (SP), #0, #80, (R6)             ; 0456
                                             66     00111                                                    ;
                                  66   5003  8F  B0 00112        MOVW    #20483, (R6)                        ;
                              10  A6         5B  D0 00117        MOVL    LEN, 16(R6)                         ;
                              16  A6         23  90 0011B        MOVB    #35, 22(R6)                         ;
                              1E  A6   0102  8F  B0 0011F        MOVW    #258, 30(R6)                        ;
                              28  A6         59  D0 00125        MOVL    R9, 40(R6)                          ;
                       50         6E         04  C1 00129        ADDL3   #4, (SP), R0                        ;
                              2C  A6         60  D0 0012D        MOVL    (R0), 44(R6)                        ;
                              30  A6  0000'  CF  9E 00131        MOVAB   P.AAC, 48(R6)                       ;
                              34  A6     00  BE  90 00137        MOVB    @0(SP), 52(R6)                      ;
                              35  A6         0B  90 0013C        MOVB    #11, 53(R6)                         ;
                              36  A6   0200  8F  B0 00140        MOVW    #512, 54(R6)                        ;
  0044   8F            00         6E         00  2C 00146        MOVC5   #0, (SP), #0, #68, (R10)            ; 0460
                                             6A     0014D                                                    ;
                                  6A   4401  8F  B0 0014E        MOVW    #17409, (R10)                       ;
                              04  AA   0800  8F  3C 00153        MOVZWL  #2048, 4(R10)                       ;
                              3C  AA         56  D0 00159        MOVL    R6, 60(R10)                         ;
  0060   8F            00         6E         00  2C 0015D        MOVC5   #0, (SP), #0, #96, (R9)             ; 0466
                                             69     00164                                                    ;
                                  69   6002  8F  B0 00165        MOVW    #24578, (R9)                        ;
                              02  A9         01  8E 0016A        MNEGB   #1, 2(R9)                           ;
                              04  A9     20  A7  D0 0016E        MOVL    32(R7), 4(R9)                       ;
                              0A  A9         01  8E 00173        MNEGB   #1, 10(R9)                          ;
                              0C  A9     1C  A7  D0 00177        MOVL    28(R7), 12(R9)                      ;
                                             56  DD 0017C        PUSHL   R6                                  ; 0470
                       00000000G  00         01  FB 0017E        CALLS   #1, SYS$CREATE                      ;
                                  58         50  D0 00185        MOVL    R0, STATUS                          ;
                                  05         58  E8 00188        BLBS    STATUS, 11$                         ;
                                         0C  A6  DD 0018B        PUSHL   12(R6)                              ; 0472
                                             32  11 0018E        BRB     13$                                 ;
                                  50     03  A9  9A 00190 11$:   MOVZBL  3(R9), R0                           ; 0476
                              80  8F         50  91 00194        CMPB    R0, #128                            ;
                                             04  1B 00198        BLEQU   12$                                 ;
                                  50     80  8F  9A 0019A        MOVZBL  #128, R0                            ;
                                  5B         50  D0 0019E 12$:   MOVL    R0, LEN                             ;
                 69   A7          04  B9     5B  28 001A1        MOVC3   LEN, @4(R9), 105(R7)                ; 0477
                              65  A7         5B  D0 001A7        MOVL    LEN, 101(R7)                        ; 0478
                              4A  A7     0C  A6  B0 001AB        MOVW    12(R6), 74(R7)                      ; 0480
                                             5A  DD 001B0        PUSHL   R10                                 ; 0482
                       00000000G  00         01  FB 001B2        CALLS   #1, SYS$CONNECT                     ;
                                  58         50  D0 001B9        MOVL    R0, STATUS                          ;
                                  15         58  E8 001BC        BLBS    STATUS, 14$                         ;
                                         0C  AA  DD 001BF        PUSHL   12(R10)                             ; 0484
                                             56  DD 001C2 13$:   PUSHL   R6                                  ;
                                             58  DD 001C4        PUSHL   STATUS                              ;
                                  00000000G  8F  DD 001C6        PUSHL   #EXCH$_CREATEVIRT                   ;
                       00000000G  EF         04  FB 001CC        CALLS   #4, EXCH$UTIL_FILE_ERROR            ;
```

```
                                  04 001D3        RET                                          ;
          58    A7            03  90 001D4 14$:   MOVB    #3, 88(R7)                           ; 0488
          48    A7            30  88 001D8        BISB2   #48, 72(R7)                          ; 0490
                   00000000G  EF  9F 001DC        PUSHAB  EXCH$IO_RT11_WRITE                   ; 0494
                              01  DD 001E2        PUSHL   #1                                   ;
     7E   44    A7            01  C3 001E4        SUBL3   #1, 68(R7), -(SP)                    ;
                              57  DD 001E9        PUSHL   R7                                   ;
       00000000G  EF          04  FB 001EB        CALLS   #4, EXCH$IO_RT11_WRITE               ;
                58            50  D0 001F2        MOVL    R0, STATUS                           ;
                04            58  E8 001F5        BLBS    STATUS, 15$                          ;
                50            58  D0 001F8        MOVL    STATUS, R0                           ; 0496
                                  04 001FB        RET                                          ;
                50            01  D0 001FC 15$:   MOVL    #1, R0                               ; 0498
                                  04 001FF        RET                                          ; 0499

; Routine Size:  512 bytes,    Routine Base:  EXCH$INIT_CODE + 0150
```

```
1 GLOBAL ROUTINE init_foreign_open =      %SBTTL 'init_foreign_open'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Open a foreign device with RMS so that we may initialize it.
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       namb - name block describing the device
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 $dbgtrc_prefix ('init_foreign_open> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock, ! pointer to our work area
2     namb = .init [init$a_namb] : $bblock,               ! Pointer to exchange NAMB structure
2     volb = .init [init$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, .init, init, 571);
2 $block_check (2, namb, namb, 572);
2 $block_check (2, volb, volb, 573);
2
2 ! Get the device information
2 !
3 IF NOT (status = exch$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   ! 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 init_foreign_open ();
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 $debug_print_fao ('volb devnam "!AF", namb device "!AF", namb volid "!AF", concealed !UL',
P   2         .volb [volb$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     $debug_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 $fab_init (
P   2         FAB = fab,                                      ! Volume FAB
P   2         FAC = (BIO,GET,PUT),                            !  Block I/O, read and write
P   2         FNA = volb [volb$t_vol_ident],                  !  Set name addr
P   2         FNS = .volb [volb$l_vol_ident_len],             !  Set name size
P   2         FOP = NFS,                                      !  Non-File Structured
    2         NAM = nam);                                     !  Name block
P   2 $rab_init (
P   2         RAB = rab,                                      ! Volume RAB
P   2         ROP = BIO,                                      !  Block I/O
    2         FAB = fab);                                     !  FAB addr
P   2 $nam_init (
P   2         NAM = nam,                                      ! File name block
P   2         RSA = .volb [volb$a_rsbuf],                     !  Result name addr
P   2         RSS = nam$c_maxrss,                             !  Result name size
P   2         ESA = .volb [volb$a_esbuf],                     !  Expanded name addr
    2         ESS = nam$c_maxrss);                            !  Expanded name size
    2
    2 ! Open and connect to the volume
    2 !
    3 IF NOT (status = $open (fab = fab))
    2 THEN
    2     RETURN exch$util_file_error (exch$_openforeign, .status, fab, .fab [fab$l_stv]);
    2
    2 volb [volb$w_channel] = .fab [fab$l_stv];       ! Save the channel number (NFS ==> user mode channel)
    2
    3 IF NOT (status = $connect (rab = rab))
    2 THEN
    2     RETURN exch$util_file_error (exch$_openforeign, .status, fab, .rab [rab$l_stv]);
    2
    2 ! Set the volume format
    2 !
    2 volb [volb$b_vol_format] = .namb [namb$b_vol_format];
```

```
;  580       0671  2 volb [volb$v_vfmt_explicit] = .namb [namb$v_vfmt_explicit];
;  581       0672  2 volb [volb$v_write] = (BIND devbits = fab [fab$l_dev] : $bblock; (NOT .devbits [dev$v_swl]));   ! Device can
;  582       0673  2
;  583       0674  2 RETURN true;
;  584       0675  1 END;


                                                              .EXTRN  EXCH$_ACCESSFAIL
                                                              .EXTRN  EXCH$_DEVNOTSUIT
                                                              .EXTRN  EXCH$_OPNOTPERF11
                                                              .EXTRN  SYS$OPEN, EXCH$_OPENFOREIGN

                                      0FFC 00000              .ENTRY  INIT_FOREIGN_OPEN, Save R2,R3,R4,R5,R6,R7,-  ; 0500
                                                                      R8,R9,R10,R11                                ;
        50 00000000G  EF          10  C1 00002                ADDL3   #16, EXCH$A_GBL, R0                          ; 0540
                      50          60  D0 0000A                MOVL    (R0), R0                                     ; 0541
                      59          60  D0 0000D                MOVL    (R0), R9                                     ;
                      56      04  A0  D0 00010                MOVL    4(R0), R6                                    ; 0542
                      57      10  A6  D0 00014                MOVL    16(R6), R7                                   ; 0543
                      5A      14  A6  D0 00018                MOVL    20(R6), R10                                  ; 0544
                      58      18  A6  D0 0001C                MOVL    24(R6), R8                                   ; 0545
                      53      40  A9  9E 00020                MOVAB   64(R9), R3                                   ; 0546
                      52 002C00F9 8F  D0 00024                MOVL    #2883833, R2                                 ; 0549
                      51     023B 8F  3C 0002B                MOVZWL  #571, R1                                     ;
                        00000000G EF  16 00030                JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                      52 010A00F7 8F  D0 00036                MOVL    #17432823, R2                                ; 0550
                      51     023C 8F  3C 0003D                MOVZWL  #572, R1                                     ;
                      50          59  D0 00042                MOVL    R9, R0                                       ;
                        00000000G EF  16 00045                JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                      52 041B00F3 8F  D0 0004B                MOVL    #68878579, R2                                ; 0551
                      51     023D 8F  3C 00052                MOVZWL  #573, R1                                     ;
                      50          56  D0 00057                MOVL    R6, R0                                       ;
                        00000000G EF  16 0005A                JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                             0048 8F  BB 00060                PUSHR   #^M<R3,R6>                                   ; 0555
           00000000G  EF          02  FB 00064                CALLS   #2, EXCH$UTIL_VOL_GETDVI                     ;
                      5B          50  D0 0006B                MOVL    R0, STATUS                                   ;
                      17          5B  E8 0006E                BLBS    STATUS, 1$                                   ;
                             0808 8F  BB 00071                PUSHR   #^M<R3,R11>                                  ; 0558
                                  01  DD 00075                PUSHL   #1                                           ;
                        00000000G 8F  DD 00077                PUSHL   #EXCH$_ACCESSFAIL                            ;
           00000000G  00          04  FB 0007D                CALLS   #4, LIB$SIGNAL                               ;
                      50          5B  D0 00084                MOVL    STATUS, R0                                   ; 0559
                                      04 00087                RET                                                  ;
  10          2F      A6          04  E1 00088 1$:            BBC     #4, 47(R6), 2$                               ; 0573
                      50 1C054008 8F  D0 0008D                MOVL    #470106120, MUST_HAVE                        ; 0576
                      51 203220F7 8F  D0 00094                MOVL    #540156151, CANNOT_HAVE                      ; 0577
                                  0E  11 0009B                BRB     3$                                           ; 0573
                      50 0C044020 8F  D0 0009D 2$:            MOVL    #201605152, MUST_HAVE                        ; 0582
                      51 203220C6 8F  D0 000A4                MOVL    #540156102, CANNOT_HAVE                      ; 0583
  52          2C      A6          50  CD 000AB 3$:            XORL3   MUST_HAVE, 44(R6), R2                        ; 0589
                      50          52  D3 000B0                BITL    R2, MUST_HAVE                                ;
                                  06  12 000B3                BNEQ    4$                                           ;
                      51      2C  A6  D3 000B5                BITL    44(R6), CANNOT_HAVE                          ; 0591
                                  09  13 000B9                BEQL    5$                                           ;
                      52 00000000G 8F D0 000BB 4$:            MOVL    #EXCH$_DEVNOTSUIT, TEMP                      ; 0593
                                  27  11 000C2                BRB     8$                                           ;
```

```
                    17       2E    A6              03  E0 000C4 5$:       BBS     #3, 46(R6), 7$                          ; 0598
                                                   53  DD 000C9           PUSHL   R3                                      ; 0601
                                                   56  DD 000CB           PUSHL   R6                                      ;
                         00000000G EF              02  FB 000CD           CALLS   #2, EXCH$MOUN_VMS_MOUNT                 ;
                                   03              50  E8 000D4           BLBS    R0, 6$                                  ;
                                                 00F4  31 000D7           BRW     13$                                     ;
                              FF21 CF              00  FB 000DA 6$:       CALLS   #0, INIT_FOREIGN_OPEN                   ; 0604
                                                       04 000DF           RET                                             ;
                                   18        2F    A6  E8 000E0 7$:       BLBS    47(R6), 9$                              ; 0609
                                   52 00000000G    8F  D0 000E4           MOVL    #EXCH$_OPNOTPERF11, TEMP                ; 0611
                                                   53  DD 000EB 8$:       PUSHL   R3                                      ;
                                                   01  DD 000ED           PUSHL   #1                                      ;
                                                   52  DD 000EF           PUSHL   TEMP                                    ;
                         00000000G 00              03  FB 000F1           CALLS   #3, LIB$SIGNAL                          ;
                                   50              52  D0 000F8           MOVL    TEMP, R0                                ;
                                                       04 000FB           RET                                             ;
                69    A6      008A C9      0080    8F  28 000FC 9$:       MOVC3   #128, 138(R9), 105(R6)                  ; 0623
                             65    A6      0086    C9  D0 00105           MOVL    134(R9), 101(R6)                        ; 0624
      0050   8F           00       6E              00  2C 0010B           MOVC5   #0, (SP), #0, #80, (R7)                 ; 0644
                                                   67     00112
                                   67      5003    8F  B0 00113           MOVW    #20483, (R7)                            ;
                             04    A7 00010000     8F  D0 00118           MOVL    #65536, 4(R7)                           ;
                             16    A7              23  90 00120           MOVB    #35, 22(R7)                             ;
                             1F    A7              02  90 00124           MOVB    #2, 31(R7)                              ;
                             28    A7              58  D0 00128           MOVL    R8, 40(R7)                              ;
                             2C    A7        69    A6  9E 0012C           MOVAB   105(R6), 44(R7)                         ;
                             34    A7        65    A6  90 00131           MOVB    101(R6), 52(R7)                         ;
      0044   8F           00       6E              00  2C 00136           MOVC5   #0, (SP), #0, #68, (R10)                ; 0648
                                                   6A     0013D
                                   6A      4401    8F  B0 0013E           MOVW    #17409, (R10)                           ;
                             04    AA      0800    8F  3C 00143           MOVZWL  #2048, 4(R10)                           ;
                             3C    AA              57  D0 00149           MOVL    R7, 60(R10)                             ;
      0060   8F           00       6E              00  2C 0014D           MOVC5   #0, (SP), #0, #96, (R8)                 ; 0654
                                                   68     00154
                                   68      6002    8F  B0 00155           MOVW    #24578, (R8)                            ;
                             02    A8              01  8E 0015A           MNEGB   #1, 2(R8)                               ;
                             04    A8        20    A6  D0 0015E           MOVL    32(R6), 4(R8)                           ;
                             0A    A8              01  8E 00163           MNEGB   #1, 10(R8)                              ;
                             0C    A8        1C    A6  D0 00167           MOVL    28(R6), 12(R8)                          ;
                                                   57  DD 0016C           PUSHL   R7                                      ; 0658
                         00000000G 00              01  FB 0016E           CALLS   #1, SYS$OPEN                            ;
                                   5B              50  D0 00175           MOVL    R0, STATUS                              ;
                                   05              5B  E8 00178           BLBS    STATUS, 10$                             ;
                                             0C    A7  DD 0017B           PUSHL   12(R7)                                  ; 0660
                                                   17  11 0017E           BRB     11$                                     ;
                             4A    A6        0C    A7  B0 00180 10$:      MOVW    12(R7), 74(R6)                          ; 0662
                                                   5A  DD 00185           PUSHL   R10                                     ; 0664
                         00000000G 00              01  FB 00187           CALLS   #1, SYS$CONNECT                         ;
                                   5B              50  D0 0018E           MOVL    R0, STATUS                              ;
                                   15              5B  E8 00191           BLBS    STATUS, 12$                             ;
                                             0C    AA  DD 00194           PUSHL   12(R10)                                 ; 0666
                                                   57  DD 00197 11$:      PUSHL   R7                                      ;
                                                   5B  DD 00199           PUSHL   STATUS                                  ;
                                      00000000G    8F  DD 0019B           PUSHL   #EXCH$_OPENFOREIGN                      ;
                         00000000G EF              04  FB 001A1           CALLS   #4, EXCH$UTIL_FILE_ERROR                ;
                                                       04 001A8           RET                                             ;
                             58    A6        7A    A9  90 001A9 12$:      MOVB    122(R9), 88(R6)                         ; 0670
```

```
        50     0085    C9             01             02  EF 001AE          EXTZV    #2, #1, 133(R9), R0                 ; 0671
    48  A6             01             06             50  F0 001B5          INSV     R0, #6, #1, 72(R6)                  ;
        50        43   A7             01             01  EF 001BB          EXTZV    #1, #1, 67(R7), R0                  ; 0672
                                      50             50  D2 001C1          MCOML    R0, R0                              ;
    48  A6             01             05             50  F0 001C4          INSV     R0, #5, #1, 72(R6)                  ;
                                      50             01  D0 001CA          MOVL     #1, R0                              ; 0674
                                                         04 001CD          RET                                          ;
                                                     50  D4 001CE  13$:    CLRL     R0                                  ; 0675
                                                         04 001D0          RET                                          ;

; Routine Size:  465 bytes,    Routine Base:  EXCH$INIT_CODE + 0350
```

```
1 GLOBAL ROUTINE init_init : NOVALUE =    %SBTTL 'init_init'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform setups for EXCH$init_initialize
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 init control block
2 !--
2
2 $dbgtrc_prefix ('init_init> ');
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock  ! pointer to our work area
2     ;
2
2
2 ! If our pointer is null, we need to allocate and initialize the work area
2 !
2 IF .init EQL 0
2 THEN
3     BEGIN
3
3     ! Get the right sized chunk of memory, conveniently set to nulls
3     !
3     init = exch$util_vm_allocate_zeroed (exchblk$s_init);
3
3     ! Set the ident fields
3     !
3     $block_init (.init, init);
3
3     ! Set the descriptors up
3     !
3     $dyn_str_desc_init (init [init$q_device]);
```

```
;  643          0733  3     $dyn_str_desc_init (init [init$q_volumeid]);
;  644          0734  3
;  645          0735  2     END;
;  646          0736  2
;  647          0737  2 ! Make sure that our work area is valid
;  648          0738  2 !
;  649          0739  2 $block_check (2, .init, init, 570);
;  650          0740  2
;  651          0741  2 RETURN;
;  652          0742  1 END;


                                                               .EXTRN   EXCH$GQ_DYN_STR_TEMPLATE

                                        001C 00000             .ENTRY   INIT_INIT, Save R2,R3,R4                        ; 0676
                     54 00000000G  EF   9E 00002               MOVAB    TMPL, R4                                        ;
       53 00000000G  EF                 10  C1 00009           ADDL3    #16, EXCH$A_GBL, R3                             ; 0712
                                        63  D5 00011           TSTL     (R3)                                            ; 0718
                                        22  12 00013           BNEQ     1$                                              ;
                                        2C  DD 00015           PUSHL    #44                                             ; 0724
          00000000G  EF                 01  FB 00017           CALLS    #1, EXCH$UTIL_VM_ALLOCATE_ZEROED                ;
                     63                 50  D0 0001E           MOVL     R0, (R3)                                        ;
                 08  A0                 2C  B0 00021           MOVW     #44, 8(R0)                                      ; 0728
                 0A  A0                 07  8E 00025           MNEGB    #7, 10(R0)                                      ;
       50            63                 0C  C1 00029           ADDL3    #12, (R3), R0                                   ; 0732
                     60                 64  7D 0002D           MOVQ     TMPL, (R0)                                      ;
       50            63                 14  C1 00030           ADDL3    #20, (R3), R0                                   ; 0733
                     60                 64  7D 00034           MOVQ     TMPL, (R0)                                      ;
                     52 002C00F9   8F   D0 00037 1$:           MOVL     #2883833, R2                                    ; 0739
                     51      023A  8F   3C 0003E               MOVZWL   #570, R1                                        ;
                     50            63   D0 00043               MOVL     (R3), R0                                        ;
                        00000000G  EF   16 00046               JSB      EXCH$UTIL_BLOCK_CHECK                           ;
                                        04 0004C               RET                                                      ; 0742

; Routine Size:  77 bytes,    Routine Base:  EXCH$INIT_CODE + 0521
```

```
1 GLOBAL ROUTINE exch$init_initialize =   %SBTTL 'exch$init_initialize'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Action routine for the INIT verb, parses and performs main control functions for INIT
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 !       Data is
2 !--
2
2 $dbgtrc_prefix ('init_initialize> ');
2
2 LOCAL
2     message,
2     namb        : $ref_bblock,                  ! Local pointer to a namb
2     volb        : $ref_bblock,                  ! Local pointer to a volb
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock  ! pointer to our work area
2     ;
2
2
2 ! Allocate and/or initialize the work area
2 !
2 init_init ();
2
2 ! Get the individual boolean qualifiers.
2 !
2 init [init$v_q_create] = cli$present (%ASCID 'CREATE');
2
2 ! Set the flag for printing init messages.
2 !
```

```
;  711    0800  2 init [init$v_q_message] = .exch$a_gbl [excg$v_q_message];        ! Default to external state
;  712    0801  2 message = cli$present (%ASCID 'MESSAGE');                         ! Find the flag state for the
;  713    0802  2 IF  .message EQL cli$_present                                     ! Either /MESSAGE or /NOMESSAGE must be spec
;  714    0803  2   OR                                                              !  in order to override the external default
;  715    0804  2     .message EQL cli$_negated
;  716    0805  2 THEN
;  717    0806  2     init [init$v_q_message] = .message;
;  718    0807  2
;  719    0808  2 !\ init [init$v_q_badblocks]      = cli$present (%ASCID 'BADBLOCKS');
;  720    0809  2 !\ init [init$v_q_badblocks_retain] = cli$present (%ASCID 'BADBLOCKS.RETAIN');
;  721    0810  2 !\ init [init$v_q_replace]                = cli$present (%ASCID 'REPLACE');
;  722    0811  2 !\ init [init$v_q_replace_retain]         = cli$present (%ASCID 'REPLACE.RETAIN');
;  723    0812  2
;  724    0813  2 ! Get individual integer-valued qualifiers, routine signals on errors.  If the qualifier is not present, 0 i
;  725    0814  2 ! in the second parameter and -1 (success) is returned as the routine value.  Here we also treat positionals
;  726    0815  2 ! second parameter as globals.
;  727    0816  2 !
;  728    0817  3 IF NOT (status = exch$cmd_cli_get_integer (%ASCID 'ALLOCATION', init [init$l_q_allocation]))
;  729    0818  2 THEN
;  730    0819  2     RETURN .status;
;  731    0820  2
;  732    0821  3 IF NOT (status = exch$cmd_cli_get_integer (%ASCID 'EXTRA_WORDS', init [init$l_q_extra_words]))
;  733    0822  2 THEN
;  734    0823  2     RETURN .status;
;  735    0824  2 IF .init [init$l_q_extra_words] GTRU 119
;  736    0825  2 THEN
;  737    0826  3     BEGIN
;  738    0827  3     $exch_signal (exch$_rt11_extra);
;  739    0828  3     init [init$l_q_ext[illegible]ords] = 119;
;  740    0829  2     END;
;  741    0830  2
;  742    0831  3 IF NOT (status = exch$cmd_cli_get_integer (%ASCID 'SEGMENTS', init [init$l_q_segments]))
;  743    0832  2 THEN
;  744    0833  2     RETURN .status;
;  745    0834  2 IF .init [init$l_q_segments] GTRU 31
;  746    0835  2 THEN
;  747    0836  3     BEGIN
;  748    0837  3     $exch_signal (exch$_rt11_toomanyseg, 1, 31);
;  749    0838  3     init [init$l_q_segments] = 31;
;  750    0839  2     END;
;  751    0840  2
;  752    0841  2 ! Get the volume label
;  753    0842  2 !
;  754    0843  3 IF NOT (status = cli$get_value (%ASCID 'VOLUMELABEL', init [init$q_volumeid]))
;  755    0844  2 THEN
;  756    0845  2     $exch_signal_return (.status);
;  757    0846  2
;  758    0847  2 ! Parse the device name parameter into a newly allocated $NAMB, there are no defaults
;  759    0848  2 !
;  760    0849  2 status = exch$cmd_parse_filespec (%ASCID 'DEVICENAME', 0, 0, init [init$q_device], namb);
;  761    0850  2 init [init$a_namb] = .namb;                          ! Save it in the work area too
;  762    0851  2 IF NOT .status
;  763    0852  2 THEN
;  764    0853  2     $exch_signal_return (exch$_parseerr, 1, init [init$q_device], .status);
;  765    0854  2
;  766    0855  2 ! If a physical init, check the name
;  767    0856  2 !
```

```
3 IF NOT (.init [init$v_q_create])
2 THEN
3     BEGIN
3     IF NOT .namb [namb$v_explicit_device]
3     THEN
3         $exch_signal_return (exch$_nodevice, 1, init [init$q_device]);
3     IF .namb [namb$v_explicit_node]
3     THEN
3         $exch_signal_return (exch$_noremote, 1, init [init$q_device]);
3     IF    .namb [namb$v_explicit_directory] OR .namb [namb$v_explicit_name]
3       OR .namb [namb$v_explicit_type] OR .namb [namb$v_explicit_version]
3     THEN
3         $exch_signal (exch$_devonly, 1, init [init$q_device]);
2     END;
2
2 ! If the device is not mounted, attempt to temporarily open a file and perform the operation
2 !
2 volb = .namb [namb$a_assoc_volb];                 ! If it is mounted, we will have a pointer to a volb
3 IF (.volb EQL 0)
2 THEN
3     BEGIN
3
3     ! Allocate a $VOLB to describe the volume
3     !
3     volb = exch$util_volb_allocate ();
3     init [init$a_volb] = .volb;
3
3     ! Temporarily open a channel to the device
3     !
3     IF .init [init$v_q_create]
3     THEN
3         status = init_foreign_create ()
3     ELSE
3         status = init_foreign_open ();
3
3     ! Now do the actual initialize
3     !
3     IF .status
3     THEN
4         BEGIN
4
4         ! The open worked, let's see if we can do the volume-specific part of it
4         !
4         CASE .volb [volb$b_vol_format] FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF
4         SET
5                 [volb$k_vfmt_dos11] :   BEGIN
5                                         status = init_dos11_init ();
5                                         CH$MOVE (6, UPLIT BYTE ('DOS-11'), volb [volb$t_vol_type]);
5                                         volb [volb$l_vol_type_len] = 6;
4                                         END;
5                 [volb$k_vfmt_rt11] :    BEGIN
5                                         status = init_rt11_init ();
5                                         CH$MOVE (5, UPLIT BYTE ('RT-11'), volb [volb$t_vol_type]);
5                                         volb [volb$l_vol_type_len] = 5;
4                                         END;
4 !\              [volb$k_vfmt_rtmt] :    $exch_signal_stop (exch$_notimplement);
4                 [OUTRANGE,INRANGE] :    $logic_check (0, (false), 226);
```

```
  4          TES;
  4
  4          ! Close the volb's file now
  4          !
  4          init_foreign_close ();
  3          END;
  3
  3      ! Release the volb, since we don't plan to mount it
  3      !
  3      exch$util_volb_release (.volb);
  3
  3      END
  3
  3 ! OK, the device has already been mounted
  3 !
  2 ELSE
  3      BEGIN
  3
  3      ! The open worked, let's see if we can do the volume-specific part of it
  3      !
  3      init [init$a_volb] = .volb;
  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 = init_dos11_init ();
  3          [volb$k_vfmt_rt11]       : status = init_rt11_init ();
  3 !\       [volb$k_vfmt_rtmt]       : $exch_signal_stop (exch$_notimplement);
  3          [OUTRANGE,INRANGE]       : $logic_check (0, (false), 307);
  3      TES;
  3
  2      END;
  2
  2 ! Tell them it has been dcne
  2 !
  2 IF .status
  2   AND
  2    .init [init$v_q_message]
  2 THEN
P  2      $exch_signal (exch$_initialized, 4, .volb [volb$l_vol_type_len], volb [volb$t_vol_type],
  2                                          .volb [volb$l_vol_ident_len],  volb [volb$t_vol_ident]);
  2
  2 ! Release the namb we used for the input
  2 !
  2 exch$util_namb_release (.namb);
  2
  2 RETURN .status;
  1 END;


                                                                          .PSECT  EXCH$INIT_PLIT,NOWRT,2

                                                        0001B            .BLKB   1
              00  00  45  54  41  45  52  43            0001C P.AAE:     .ASCII  \CREATE\<0><0>                          ;
                                       010E0006         00024 P.AAD:     .LONG   17694726                                ;
                                       00000000'        00028            .ADDRESS P.AAE                                  ;
              00  45  47  41  53  53  45  4D            0002C P.AAG:     .ASCII  \MESSAGE\<0>                            ;
                                       010E0007         00034 P.AAF:     .LONG   17694727                                ;
```

```
                                                                 F  8
EXCH$INIT          INIT verb dispatch and misc routines          16-Sep-1984 00:59:01    VAX-11 Bliss-32 V4.0-742           Page  27
V04-000            exch$init_initialize                          14-Sep-1984 12:29:05    [EXCHNG.SRC]EXCINIT.B32;1           (8)

                                           00000000' 00038          .ADDRESS P.AAG                                 :
      00  00  4E  4F  49  54  41  43  4F  4C  4C  41  0003C P.AAI:  .ASCII  \ALLOCATION\<0><0>                     :
                                           010E000A  00048 P.AAH:   .LONG   17694730                               :
                                           00000000' 0004C          .ADDRESS P.AAI                                 :
      00  53  44  52  4F  57  5F  41  52  54  58  45  00050 P.AAK:  .ASCII  \EXTRA_WORDS\<0>                        :
                                           010E000B  0005C P.AAJ:   .LONG   17694731                               :
                                           00000000' 00060          .ADDRESS P.AAK                                 :
                      53  54  4E  45  4D  47  45  53  00064 P.AAM:  .ASCII  \SEGMENTS\                             :
                                           010E0008  0006C P.AAL:   .LONG   17694728                               :
                                           00000000' 00070          .ADDRESS P.AAM                                 :
      00  4C  45  42  41  4C  45  4D  55  4C  4F  56  00074 P.AAO:  .ASCII  \VOLUMELABEL\<0>                        :
                                           010E000B  00080 P.AAN:   .LONG   17694731                               :
                                           00000000' 00084          .ADDRESS P.AAO                                 :
      00  00  45  4D  41  4E  45  43  49  56  45  44  00088 P.AAQ:  .ASCII  \DEVICENAME\<0><0>                      :
                                           010E000A  00094 P.AAP:   .LONG   17694730                               :
                                           00000000' 00098          .ADDRESS P.AAQ                                 :
                                  31  31  2D  53  4F  44  0009C P.AAR:  .ASCII  \DOS-11\                           :
                                      31  31  2D  54  52  000A2 P.AAS:  .ASCII  \RT-11\                            :

                                                                    .EXTRN  CLI$_PRESENT, CLI$_NEGATED
                                                                    .EXTRN  EXCH$_RT11_EXTRA
                                                                    .EXTRN  EXCH$_RT11_TOOMANYSEG
                                                                    .EXTRN  CLI$GET_VALUE, EXCH$_PARSEERR
                                                                    .EXTRN  EXCH$_NODEVICE, EXCH$_NOREMOTE
                                                                    .EXTRN  EXCH$_DEVONLY, EXCH$_BADLOGIC
                                                                    .EXTRN  EXCH$_INITIA' 'ZED

                                                                    .PSECT  EXCH$INIT_CO'E,NOWRT,2

                                                  0FFC 00000        .ENTRY  EXCH$INIT_INITIALIZE, Save R2,R3,R4,R5,R6,- ; 0743
                                                                            R7,R8,R9,R10,R11                          :
                                  5B       0000'  CF  9E 00002      MOVAB   P.AAD, R11                                :
                                  5A 00000000G    00  9E 00007      MOVAB   LIB$SIGNAL, R10                           :
                                  5E              04  C2 0000E      SUBL2   #4, SP                                    :
                    52 00000000G  EF              10  C1 00011      ADDL3   #16, EXCH$A_GBL, R2                       ; 0786
                              96  AF              00  FB 00019      CALLS   #0, INIT_INIT                             ; 0792
                                  52              62  D0 0001D      MOVL    (R2), R2                                  ; 0796
                                  59       28     A2  9E 00020      MOVAB   40(R2), R9                                :
                                                  5B  DD 00024      PUSHL   R11                                       :
                       00000000G  00              01  FB 00026      CALLS   #1, CLI$PRESENT                           :
      69            01            00              50  F0 0002D      INSV    R0, #0, #1, (R9)                          :
      50 00000000G  FF            01              02  EF 00032      EXTZV   #2, #1, @EXCH$A_GBL, R0                   ; 0800
      69            01            01              50  F0 0003B      INSV    R0, #1, #1, (R9)                          :
                                           10     AB  9F 00040      PUSHAB  P.AAF                                     ; 0801
                       00000000G  00              01  FB 00043      CALLS   #1, CLI$PRESENT                           :
                       00000000G  8F              50  D1 0004A      CMPL    MESSAGE, #CLI$_PRESENT                    ; 0802
                                                  09  13 00051      BEQL    1$                                        :
                       00000000G  8F              50  D1 00053      CMPL    MESSAGE, #CLI$_NEGATED                    ; 0804
                                                  05  12 0005A      BNEQ    2$                                        :
      69            01            01              50  F0 0005C 1$:  INSV    MESSAGE, #1, #1, (R9)                     ; 0806
                                           1C     A2  9F 00061 2$:  PUSHAB  28(R2)                                    ; 0817
                                           24     AB  9F 00064      PUSHAB  P.AAH                                     :
                       00000000G  EF              02  FB 00067      CALLS   #2, EXCH$CMD_CLI_GET_INTEGER              :
                                  58              50  D0 0006E      MOVL    R0, STATUS                                :
                                  3B              58  E9 00071      BLBC    STATUS, 4$                                :
                                           20     A2  9F 00074      PUSHAB  32(R2)                                    ; 0821
                                           38     AB  9F 00077      PUSHAB  P.AAJ                                     :
```

```
          00000000G  EF             02  FB 0007A        CALLS    #2, EXCH$CMD_CLI_GET_INTEGER                       ;
                     58             50  D0 00081        MOVL     R0, STATUS                                         ;
                     28             58  E9 00084        BLBC     STATUS, 4$                                         ;
          00000077   8F       20    A2  D1 00087        CMPL     32(R2), #119                                       ; 0824
                                    0E  1B 0008F        BLEQU    3$                                                 ;
                         00000000G  8F  DD 00091        PUSHL    #EXCH$_RT11_EXTRA                                  ; 0827
                     6A             01  FB 00097        CALLS    #1, LIB$SIGNAL                                     ;
               20    A2       77    8F  9A 0009A        MOVZBL   #119, 32(R2)                                       ; 0828
                              24    A2  9F 0009F  3$:   PUSHAB   36(R2)                                             ; 0831
                              48    AB  9F 000A2        PUSHAB   P.AAL                                              ;
          00000000G  EF             02  FB 000A5        CALLS    #2, EXCH$CMD_CLI_GET_INTEGER                       ;
                     58             50  D0 000AC        MOVL     R0, STATUS                                         ;
                     03             58  E8 000AF  4$:   BLBS     STATUS, 5$                                         ;
                                  0191  31 000B2        BRW      31$                                                ;
                     1F       24    A2  D1 000B5  5$:   CMPL     36(R2), #31                                        ; 0834
                                    11  1B 000B9        BLEQU    6$                                                 ;
                                    1F  DD 000BB        PUSHL    #31                                                ; 0837
                                    01  DD 000BD        PUSHL    #1                                                 ;
                         00000000G  8F  DD 000BF        PUSHL    #EXCH$_RT11_TOOMANYSEG                             ;
                     6A             03  FB 000C5        CALLS    #3, LIB$SIGNAL                                     ;
               24    A2             1F  D0 000C8        MOVL     #31, 36(R2)                                        ; 0838
                              14    A2  9F 000CC  6$:   PUSHAB   20(R2)                                             ; 0843
                              5C    AB  9F 000CF        PUSHAB   P.AAN                                              ;
          00000000G  00             02  FB 000D2        CALLS    #2, CLI$GET_VALUE                                  ;
                     58             50  D0 000D9        MOVL     R0, STATUS                                         ;
                     0A             58  E8 000DC        BLBS     STATUS, 7$                                         ;
                     53             58  D0 000DF        MOVL     STATUS, TEMP                                       ; 0845
                                    53  DD 000E2        PUSHL    TEMP                                               ;
                     6A             01  FB 000E4        CALLS    #1, LIB$SIGNAL                                     ;
                                    32  11 000E7        BRB      8$                                                 ;
                                    5E  DD 000E9  7$:   PUSHL    SP                                                 ; 0849
                     54       0C    A2  9E 000EB        MOVAB    12(R2), R4                                         ;
                                    54  DD 000EF        PUSHL    R4                                                 ;
                                    7E  7C 000F1        CLRQ     -(SP)                                              ;
                              70    AB  9F 000F3        PUSHAB   P.AAP                                              ;
          00000000G  EF             05  FB 000F6        CALLS    #5, EXCH$CMD_PARSE_FILESPEC                        ;
                     58             50  D0 000FD        MOVL     R0, STATUS                                         ;
                     57             6E  D0 00100        MOVL     NAMB, R7                                           ; 0850
                     62             57  D0 00103        MOVL     R7, (R2)                                           ;
                     16             58  E8 00106        BLBS     STATUS, 9$                                         ; 0851
                     53  00000000G  8F  D0 00109        MOVL     #EXCH$_PARSEERR, TEMP                              ; 0853
                              0110  8F  BB 00110        PUSHR    #^M<R4,R8>                                         ;
                                    01  DD 00114        PUSHL    #1                                                 ;
                                    53  DD 00116        PUSHL    TEMP                                               ;
                     6A             04  FB 00118        CALLS    #4, LIB$SIGNAL                                     ;
                     50             53  D0 0011B  8$:   MOVL     TEMP, R0                                           ;
                                        04 0011E        RET                                                         ;
                     46             69  E8 0011F  9$:   BLBS     (R9), 14$                                          ; 0857
                     53       6C    A7  9E 00122        MOVAB    108(R7), R3                                        ; 0860
                                    63  95 00126        TSTB     (R3)                                               ;
                                    09  19 00128        BLSS     10$                                                ;
                     55  00000000G  8F  D0 0012A        MOVL     #EXCH$_NODEVICE, TEMP                              ; 0862
                                    0B  11 00131        BRB      11$                                                ;
     14              63             06  E1 00133  10$:  BBC      #6, (R3), 12$                                      ; 0863
                     55  00000000G  8F  D0 00137        MOVL     #EXCH$_NOREMOTE, TEMP                              ; 0865
                                    54  DD 0013E  11$:  PUSHL    R4                                                 ;
                                    01  DD 00140        PUSHL    #1                                                 ;
```

```
                                       55  DD 00142          PUSHL   TEMP                                    ;
                         6A            03  FB 00144          CALLS   #3, LIB$SIGNAL                          ;
                         50            55  D0 00147          MOVL    TEMP, R0                                ;
                                           04 0014A          RET                                             ;
                         0C       01   A3  E8 0014B 12$:     BLBS    1(R3), 13$                              ; 0866
           08            63            09  E0 0014F          BBS     #9, (R3), 13$                           ;
           04            63            0A  E0 00153          BBS     #10, (R3), 13$                          ; 0867
           0D            63            0B  E1 00157          BBC     #11, (R3), 14$                          ;
                                       54  DD 0015B 13$:     PUSHL   R4                                      ; 0869
                                       01  DD 0015D          PUSHL   #1                                      ;
                            00000000G  8F  DD 0015F          PUSHL   #EXCH$_DEVONLY                          ;
                         6A            03  FB 00165          CALLS   #3, LIB$SIGNAL                          ;
                         56       74   A7  D0 00168 14$:     MOVL    116(R7), VOLB                           ; 0874
                                       7B  12 0016C          BNEQ    23$                                     ; 0875
               00000000G EF            00  FB 0016E          CALLS   #0, EXCH$UTIL_VOLB_ALLOCATE             ; 0881
                         56            50  D0 00175          MOVL    R0, VOLB                                ;
                    04   A2            56  D0 00178          MOVL    VOLB, 4(R2)                             ; 0882
                         07            69  E9 0017C          BLBC    (R9), 15$                               ; 0886
                  FA5E   CF            00  FB 0017F          CALLS   #0, INIT_FOREIGN_CREATE                 ; 0888
                                       05  11 00184          BRB     16$                                     ;
                  FC57   CF            00  FB 00186 15$:     CALLS   #0, INIT_FOREIGN_OPEN                   ; 0890
                         58            50  D0 0018B 16$:     MOVL    R0, STATUS                              ;
                         4D            58  E9 0018E          BLBC    STATUS, 22$                             ; 0894
           03            00       58   A6  8F 00191          CASEB   88(VOLB), #0, #3                        ; 0900
  0031   0008          001D          0008     00196 17$:     .WORD   18$-17$,-                               ;
                                                                     19$-17$,-                               ;
                                                                     18$-17$,-                               ;
                                                                     20$-17$                                 ;
                         7E       E2   8F  9A 0019E 18$:     MOVZBL  #226, -(SP)                             ; 0913
                                       01  DD 001A2          PUSHL   #1                                      ;
                            00000000G  8F  DD 001A4          PUSHL   #EXCH$_BADLOGIC                         ;
               00000000G 00            03  FB 001AA          CALLS   #3, LIB$STOP                            ;
                                       26  11 001B1          BRB     21$                                     ;
                  F8DA   CF            00  FB 001B3 19$:     CALLS   #0, INIT_DOS11_INIT                     ; 0903
                         58            50  D0 001B8          MOVL    R0, STATUS                              ;
       5D    A6     78   AB            06  28 001BB          MOVC3   #6, P.AAR, 93(VOLB)                     ; 0904
                    59   A6            06  D0 001C1          MOVL    #6, 89(VOLB)                            ; 0905
                                       12  11 001C5          BRB     21$                                     ; 0900
                  0000V  CF            00  FB 001C7 20$:     CALLS   #0, INIT_RT11_INIT                      ; 0908
                         58            50  D0 001CC          MOVL    R0, STATUS                              ;
       5D    A6     7E   AB            05  28 001CF          MOVC3   #5, P.AAS, 93(VOLB)                     ; 0909
                    59   A6            C5  D0 001D5          MOVL    #5, 89(VOLB)                            ; 0910
                  F9B7   CF            00  FB 001D9 21$:     CALLS   #0, INIT_FOREIGN_CLOSE                  ; 0918
                                       56  DD 001DE 22$:     PUSHL   VOLB                                    ; 0923
               00000000G EF            01  FB 001E0          CALLS   #1, EXCH$UTIL_VOLB_RELEASE              ;
                                       36  11 001E7          BRB     29$                                     ; 0875
                    04   A2            56  D0 001E9 23$:     MOVL    VOLB, 4(R2)                             ; 0934
           03            00       58   A6  8F 001ED          CASEB   88(VOLB), #0, #3                        ; 0935
  0025   0008          001E          0008     001F2 24$:     .WORD   25$-24$,-                               ;
                                                                     26$-24$,-                               ;
                                                                     25$-24$,-                               ;
                                                                     27$-24$                                 ;
                         7E     0133   8F  3C 001FA 25$:     MOVZWL  #307, -(SP)                             ; 0940
                                       01  DD 001FF          PUSHL   #1                                      ;
                            00000000G  8F  DD 00201          PUSHL   #EXCH$_BADLOGIC                         ;
               00000000G 00            03  FB 00207          CALLS   #3, LIB$STOP                            ;
                                       0F  11 0020E          BRB     29$                                     ;
                    F87D   CF          00  FB 00210 26$:    CALLS    #0, INIT_DOS11_INIT                         ; 0937
                                       05  11 00215         BRB      28$
                   0000V   CF          00  FB 00217 27$:    CALLS    #0, INIT_RT11_INIT                          ; 0938
                           58          50  D0 0021C 28$:    MOVL     R0, STATUS
                           1B          58  E9 0021F 29$:    BLBC     STATUS, 30$                                 ; 0947
           17              69          01  E1 00222         BBC      #1, (R9), 30$                               ; 0949
                                  69   A6  9F 00226         PUSHAB   105(VOLB)                                   ; 0952
                                  65   A6  DD 00229         PUSHL    101(VOLB)
                                  5D   A6  9F 0022C         PUSHAB   93(VOLB)
                                  59   A6  DD 0022F         PUSHL    89(VOLB)
                                       04  DD 00232         PUSHL    #4
                        00000000G      8F  DD 00234         PUSHL    #EXCH$_INITIALIZED
                           6A          06  FB 0023A         CALLS    #6, LIB$SIGNAL
                                       57  DD 0023D 30$:    PUSHL    R7                                          ; 0956
               00000000G   EF          01  FB 0023F         CALLS    #1, EXCH$UTIL_NAMB_RELEASE
                           50          58  D0 00246 31$:    MOVL     STATUS, R0                                  ; 0958
                                       04     00249         RET                                                  ; 0959

; Routine Size:  586 bytes,    Routine Base:  EXCH$INIT_CODE + 056E
```

```
1 GLOBAL ROUTINE init_rt11_init = %SBTTL 'init_rt11_init'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform RT11 volume specific init actions
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       work area for INIT
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 !       RT11 directory will be initialized
2 !--
2
2 $dbgtrc_prefix ('init_rt11_init> ');
2
2 LOCAL
2     ent : $ref_bblock,                          ! the first entry in the block
2     hdr : $ref_bblock,                          ! pointer to the rt11 directory block
2     hom : $ref_bblock,                          ! pointer to the rt11 home block
2     rtv : $ref_bblock,                          ! rt11 volume extension
2     bnum,                                       ! number of blocks on device
2     snum,                                       ! number of segments in directory
2     start,                                      ! start block for files
2     hdrbuf : $bvector [rt11$k_dirseglen],       ! actual buffer
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock, ! pointer to our work area
2     volb = init [init$a_volb]           : $ref_bblock  ! pointer to exchange VOLB structure
2     ;
```

```
2 ! Boot program. The following PDP-11 program will type out the attached message when the volume is booted on
2 ! PDP-11, informing the user that this is not a system disk.  (Thanks to <INIT.SRC>ININDX.B32)
2 !
2 BIND
2     boot_program = UPLIT WORD (
2
2                          %O'000240',                  !  BOOTBK: NOP                          ; NOP IDENTIFIES BOO
2                          %O'012706',  %O'001000',     !          MOV     #1000,SP             ; SET TEMP STACK
2                          %O'010700',                  !          MOV     PC,R0                ; SET ADDRESS
2                          %O'062700',  %O'000036',     !          ADD     #BOTMSG-.,R0         ; OF MESSAGE
2                          %O'112001',                  !  10$:    MOVB    (R0)+,R1             ; GET NEXT CHARACTER
2                          %O'001403',                  !          BEQ     20$                  ; END
2                          %O'004767',  %O'000006',     !          CALL    TYPIT                ; NO, PRINT IT
2                          %O'000773',                  !          BR      10$                  ; LOOP FOR NEXT CHAR
2                          %O'000005',                  !  20$:    RESET                        ;
2                          %O'000000',                  !          HALT                         ; HALT
2                                                       !
2                                                       !
2                          %O'110137',  %O'177566',     !  TYPIT:  MOVB    R1,@#TPB             ; PRINT CHARACTER
2                          %O'105737',  %O'177564',     !  10$:    TSTB    @#TPS                ; DONE?
2                          %O'100375',                  !          BPL     10$                  ; NO, WAIT
2                          %O'000207'                   !          RETURN                       ;
2                                                       !
2                                                       !
2                                                       !  BOTMSG:
2                          ),
2
2     ! Boot message, we will add the volume id a little later
2     !
2     boot_message = UPLIT BYTE (
2                          7, 13, 10, 10, 7,
2                          'The volume labeled "          " is not a system volume.',
2                          7, 13, 10, 10, 7, 0
2                          );
2
2 LITERAL
2         boot_prog_len   = 38,                   ! boot program is 38 bytes long
2         boot_mesg_len   = 68,                   ! message is 68 bytes long
2         boot_volname    = boot_prog_len+25;     ! volume label offset in boot block message
```

```
  2 $block_check (2, .init, init, 574);
  2 $block_check (2, .volb, volb, 576);
  2
  2 ! Make sure that we can do it
  2 !
  2 IF NOT .volb [volb$v_write]
  2 THEN
P 2     $exch_signal_return ($warning_stat_copy (exch$_writelock), 2,
  2                                 .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
  2
  2 ! Get a zeroed buffer for the block and a pointer to the first entry
  2 !
  2 hdr = hdrbuf;
  2 hom = hdrbuf + 512;
  2 CH$FILL (0, rt11$k_dirseglen, hdrbuf);
  2 ent = .hdr + rt11hdr$k_length;
  2
  2 ! Determine the number of device blocks
  2 !
  4 bnum = (BEGIN
  4         LOCAL
  4             bmax;
  4         bmax = MINU (65535, .volb [volb$l_devmaxblock]);
  4         IF .volb [volb$v_virtual]
  4         THEN
  5             BEGIN
  5             IF .init [init$l_q_allocation] NEQ 0
  5               AND
  5                NOT .init [init$v_q_create]
  5             THEN
  5                 $exch_signal (exch$_virtnochange);
  5             .bmax
  5             END
  4         ELSE IF .init [init$l_q_allocation] NEQ 0
  4         THEN
  5             BEGIN
  5             IF .init [init$l_q_allocation] GTRU .bmax
  5             THEN
  6                 BEGIN
  6                 $exch_signal (exch$_rt11_toomanyblk, 1, .bmax);
  6                 .bmax
  6                 END
  5             ELSE
  5                 .init [init$l_q_allocation]
  5             END
  4         ELSE
  4             .bmax
  2         END);
  2 bnum = MAXU (40, .bnum);
```

```
2 ! Determine the number of directory segments
2 !
3 snum = (SELECTONE true OF
3         SET
3
3         ! If a /SEGMENTS was given, use that value
3         !
3         [.init [init$l_q_segments] NEQ 0] :     .init [init$l_q_segments];
3
3         ! If no /SEGMENTS, use a default based on device size (ala RT-11 DUP)
3         !
3         [.bnum LEQU 512] :                        1;
3         [.bnum LEQU 2048] :                       4;
3         [.bnum LEQU 12288] :                     16;
3         [OTHERWISE] :                            31;
3
2         TES);
2
2 ! Determine the start block for files
2 !
2 start = rt11$k_root_block + (2 * .snum);
2 IF .start+32 GTRU .bnum                      ! If room for fewer than 32 blocks for files
2 THEN
3     BEGIN
3     snum = 1;                                ! Reduce to one segment
3     start = rt11$k_root_block + 2;           ! Start at a given block
3     $exch_signal (exch$_rt11_toomanyseg, 1, 1); ! And tell the world
2     END;
```

```
! Set up the boot and home blocks
!
$logic_check (0, (rt11hom$s_owner_name EQL excg$s_username), 310);
%PRINT:   assumption 310 verified during compilation
CH$MOVE (rt11hom$s_owner_name, exch$a_gbl [excg$t_username], hom [rt11hom$t_owner_name]);
CH$MOVE (rt11hom$s_system_id,  UPLIT BYTE ('DECVMSEXCHNG'),  hom [rt11hom$t_system_id]);
CH$MOVE (boot_prog_len + boot_mesg_len, boot_program, hdrbuf [0]);
(BEGIN
 BIND
     desc = init [init$q_volumeid] : $desc_block;
 CH$COPY (.desc [dsc$w_length], .desc [dsc$a_pointer], %C ' ', rt11hom$s_volume_id, hom [rt11hom$t_volume_id
 CH$COPY (.desc [dsc$w_length], .desc [dsc$a_pointer], %C ' ', rt11hom$s_volume_id, hdrbuf [boot_volname]);
 END);
hom [rt11hom$w_system_vers] = %RAD50_11 'V40';
hom [rt11hom$w_cluster]     = 1;
hom [rt11hom$w_first_seg]   = rt11$k_root_block;

! Write the boot and home blocks.
!
IF NOT (status = exch$io_rt11_write (.volb, 0, 2, .hdr))
THEN
    RETURN .status;

! If the volume format extension exists, overwrite the cached home block
!
rtv = .volb [volb$a_vfmt_specific];
IF .rtv NEQ 0
THEN
    BEGIN
    $block_check (2, .rtv, rt11, 629);                                  ! If not an rtv we are hopelessly co
    CH$MOVE (512, .hom, rtv [rt11$t_block_1]);                          ! Copy the home block to cache
    END;

! We will zero the disk to the end of the directory area.
!
CH$FILL (0, rt11$k_dirseglen, hdrbuf);          ! Set it back to zeroes
INCR p FROM 2 TO .start-1 BY 2
DO
    IF NOT (status = exch$io_rt11_write (.volb, .p, 2, .hdr))
    THEN
        RETURN .status;

! Since Files-11 writes a large number of home blocks on a device, make sure that we zero most of them so th
! don't see strange things happening during a foreign mount.
!
IF NOT (status = init_zero_home_blocks (.start, .hdr))          ! Pass # of first unzeroed block and zeroed
THEN
    RETURN .status;

! Now set up the header of the first segment
!
hdr [rt11hdr$w_num_segs] = .snum;
hdr [rt11hdr$w_next_seg] = 0;                   ! Only one segment in the directory
hdr [rt11hdr$w_high_seg] = 1;
hdr [rt11hdr$w_extra_bytes] = 2 * .init [init$l_q_extra_words];
hdr [rt11hdr$w_start_block] = .start;

```

```
; 1099       1183  2 ! Make the empty entry followed by end of segment marker
; 1100       1184  2 !
; 1101       1185  2 ent [rt11ent$b_type_byte] = rt11ent$m_typ_empty;
; 1102       1186  2 ent [rt11ent$l_filename] = r50_empty;          ! Name is simple "EMPTY.FIL"
; 1103       1187  2 ent [rt11ent$w_filetype] = r50_fil;
; 1104       1188  2 exch$rt11_format_current_date (.ent);
; 1105       1189  2 ent [rt11ent$w_blocks] = .bnum - .hdr [rt11hdr$w_start_block];
; 1106       1190  2 ent = .ent + rt11ent$k_length + .hdr [rt11hdr$w_extra_bytes];
; 1107       1191  2 $logic_check (2, (.ent LSSU .hdr + 510), 247);
; 1108       1192  2 ent [rt11ent$b_type_byte] = rt11ent$m_typ_end_segment;
; 1109       1193  2
; 1110       1194  2 ! If the volume format extension exists, overwrite the cached directory
; 1111       1195  2 !
; 1112       1196  2 IF .rtv NEQ 0
; 1113       1197  2 THEN
; 1114       1198  3     BEGIN
; 1115       1199  3     CH$MOVE (512, .hdr, rtv [rt11$t_dire_segments]);                  ! Copy the new directory to cache
; 1116       1200  3     $logic_check (2, (exch$rtacp_verify_directory (.volb)), 249);     ! Make sure the directory is still o
; 1117       1201  2     END;
; 1118       1202  2
; 1119       1203  2 ! Write out the new root directory, only the first block necessary
; 1120       1204  2 !
; 1121       1205  2 status = exch$io_rt11_write (.volb, rt11$k_root_block, 1, .hdr);
; 1122       1206  2
; 1123       1207  2 RETURN .status;
; 1124       1208  1 END;


                                                                                          .PSECT  EXCH$INIT_PLIT,NOWRT,2

                                                                          000A7           .BLKB   1
0006  09F7  0303  9401  001E  65C0  11C0  0200  15C6  00A0  000A8 P.AAT:  .WORD   160, 5574, 512, 4544, 26048, 30, -27647, -  ;
      0087  80FD  FF74  8BDF  FF76  905F  0000  0005  01FB  000BC                 771, 2551, 6, 507, 5, 0, -28577, -138, -   ;
                                                                                  -29729, -140, -32515, 135                  ;
                                         07  0A  0A  0D  07  000CE P.AAU:  .BYTE   7, 13, 10, 10, 7                           ;
65  62  61  6C  20  65  6D  75  6C  6F  76  20  65  68  54  000D3          .ASCII  \The volume labeled "              " is not\ ;
20  20  20  20  20  20  20  20  20  20  22  20  64  65  6C  000E2                                                     ;
                    74  6F  6E  20  73  69  20  22  20  20  000F1                                                     ;
6D  75  6C  6F  76  20  6D  65  74  73  79  73  20  61  20  000FB          .ASCII  \ a system volume.\                ;
                                                        2E  65  0010A                                                 ;
                                         00  07  0A  0A  0D  07  0010C          .BYTE   7, 13, 10, 10, 7, 0                        ;
            47  4E  48  43  58  45  53  4C  56  43  45  44  00112 P.AAV:  .ASCII  \DECVMSEXCHNG\                     ;

                                                                  BOOT_PROGRAM=          P.AAT
                                                                  BOOT_MESSAGE=          P.AAU
                                                                          .EXTRN  EXCH$_VIRTNOCHANGE

                                                                          .PSECT  EXCH$INIT_CODE,NOWRT,2

                                          0FFC 00000              .ENTRY  INIT_RT11_INIT, Save R2,R3,R4,R5,R6,R7,R8,- ; 0960
                                                                          R9,R10,R11                                 ;
                      5E        FBEC    CE  9E 00002              MOVAB   -1044(SP), SP                              ;
           50 00000000G  EF             10  C1 00007              ADDL3   #16, EXCH$A_GBL, R0                        ; 1008
                      5A                60  D0 0000F              MOVL    (R0), R10                                  ; 1009
                      52 002C00F9       8F  D0 00012              MOVL    #2883833, R2                               ; 1050
                      51        023E    8F  3C 00019              MOVZWL  #574, R1                                   ;
```

```
                               50          5A  D0 0001E          MOVL    R10, R0                                 ;
                                 00000000G  EF  16 00021          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                               6E       04  AA  D0 00027          MOVL    4(R10), (SP)                            ; 1051
                               52 041B00F3  8F  D0 0002B          MOVL    #68878579, R2                           ;
                               51     0240  8F  3C 00032          MOVZWL  #576, R1                                ;
                               50           6E  D0 00037          MOVL    (SP), R0                                ;
                                 00000000G  EF  16 0003A          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                50             6E 00000048  8F  C1 00040          ADDL3   #72, (SP), R0                           ; 1055
                31             60           05  E0 00048          BBS     #5, (R0), 1$                            ;
                               50 00000000G 8F  D0 0004C          MOVL    #EXCH$_WRITELOCK, STATUS2               ; 1058
                               50           07  8A 00053          BICB2   #7, STATUS2                             ;
                               52           50  D0 00056          MOVL    STATUS2, TEMP                           ;
                50             6E 00000069  8F  C1 00059          ADDL3   #105, (SP), R0                          ;
                                            50  DD 00061          PUSHL   R0                                      ;
                53       04    AE 00000065  8F  C1 00063          ADDL3   #101, 4(SP), R3                         ;
                                            63  DD 0006C          PUSHL   (R3)                                    ;
                                            02  DD 0006E          PUSHL   #2                                      ;
                                            52  DD 00070          PUSHL   TEMP                                    ;
                     00000000G  00          04  FB 00072          CALLS   #4, LIB$SIGNAL                          ;
                               50           52  D0 00079          MOVL    TEMP, R0                                ;
                                                04 0007C          RET                                             ;
                               58       14  AE  9E 0007D 1$:      MOVAB   HDRBUF, HDR                             ; 1062
                               5B     FE00  CD  9E 00081          MOVAB   HDRBUF+512, HOM                         ; 1063
    0400    8F  00             6E           00  2C 00086          MOVC5   #0, (SP), #0, #1024, HDRBUF             ; 1064
                                        14  AE     0008D
                               59       0A  A8  9E 0008F          MOVAB   10(R8), ENT                             ; 1065
                51             6E 00000040  8F  C1 00093          ADDL3   #64, (SP), R1                           ; 1072
                               50           61  DC 0009B          MOVL    (R1), R0                                ;
                     0000FFFF  8F           50  D1 0009E          CMPL    R0, #65535                              ;
                                            05  1B 000A5          BLEQU   2$                                      ;
                               50     FFFF  8F  3C 000A7          MOVZWL  #65535, R0                              ;
                               52           50  D0 000AC 2$       MOVL    R0, BMAX                                ;
                53             6E 00000048  8F  C1 000AF          ADDL3   #72, (SP), R3                           ; 1073
                18             63           04  E1 000B7          BBC     #4, (R3), 3$                            ;
                                        1C  AA  D5 000BB          TSTL    28(R10)                                 ; 1076
                                            35  13 000BE          BEQL    5$                                      ;
                               31       28  AA  E8 000C0          BLBS    40(R10), 5$                             ; 1078
                                 00000000G  8F  DD 000C4          PUSHL   #EXCH$_VIRTNOCHANGE                     ; 1080
                     00000000G  00          01  FB 000CA          CALLS   #1, LIB$SIGNAL                          ;
                                            22  11 000D1          BRB     5$                                      ; 1081
                               50       1C  AA  D0 000D3 3$:      MOVL    28(R10), R0                             ; 1083
                                            1C  13 000D7          BEQL    5$                                      ;
                               52           50  D1 000D9          CMPL    R0, BMAX                                ; 1086
                                            14  1B 000DC          BLEQU   4$                                      ;
                                            52  DD 000DE          PUSHL   BMAX                                    ; 1089
                                            01  DD 000E0          PUSHL   #1                                      ;
                                 00000000G  8F  DD 000E2          PUSHL   #EXCH$_RT11_TOOMANYBLK                  ;
                     00000000G  00          03  FB 000E8          CALLS   #3, LIB$SIGNAL                          ;
                               50           52  D0 000EF          MOVL    BMAX, R0                                ; 1090
                               52           50  D0 000F2 4$:      MOVL    R0, R2                                  ; 1085
                               56           52  D0 000F5 5$:      MOVL    R2, BNUM                                ; 1083
                               28           52  D1 000F8          CMPL    R2, #40                                 ; 1098
                                            03  1E 000FB          BGEQU   6$                                      ;
                               52           28  D0 000FD          MOVL    #40, R2                                 ;
                               56           52  D0 00100 6$:      MOVL    R2, BNUM                                ;
                                        24  AA  D5 00103          TSTL    36(R10)                                 ; 1106
                                            07  13 00106          BEQL    7$                                      ;
```

```
                           04   AE         24  AA  D0 00108          MOVL    36(R10), SNUM                          :
                                               31  11 0010D          BRB     11$                                    :
                     00000200   8F             56  D1 0010F 7$:      CMPL    BNUM, #512                             : 1110
                                               06  1A 00116          BGTRU   8$                                     :
                           04   AE             01  D0 00118          MOVL    #1, SNUM                               :
                                               22  11 0011C          BRB     11$                                    :
                     00000800   8F             56  D1 0011E 8$:      CMPL    BNUM, #2048                            : 1111
                                               06  1A 00125          BGTRU   9$                                     :
                           04   AE             04  D0 00127          MOVL    #4, SNUM                               :
                                               13  11 0012B          BRB     11$                                    :
                     00003000   8F             56  D1 0012D 9$:      CMPL    BNUM, #12288                           : 1112
                                               06  1A 00134          BGTRU   10$                                    :
                           04   AE             10  D0 00136          MOVL    #16, SNUM                              :
                                               04  11 0013A          BRB     11$                                    :
                           04   AE             1F  D0 0013C 10$:     MOVL    #31, SNUM                              : 1113
              0C   AE      04   AE             01  78 00140 11$:     ASHL    #1, SNUM, START                        : 1119
                           0C   AE             06  C0 00146          ADDL2   #6, START                              :
                   50      0C   AE             20  C1 0014A          ADDL3   #32, START, R0                         : 1120
                                56             50  D1 0014F          CMPL    R0, BNUM                               :
                                               19  1B 00152          BLEQU   12$                                    :
                           04   AE             01  D0 00154          MOVL    #1, SNUM                               : 1123
                           0C   AE             08  D0 00158          MOVL    #8, START                              : 1124
                                               01  DD 0015C          PUSHL   #1                                     : 1125
                                               01  DD 0015E          PUSHL   #1                                     :
                                    00000000G  8F  DD 00160          PUSHL   #EXCH$_RT11_TOOMANYSEG                 :
                     00000000G  00             03  FB 00166          CALLS   #3, LIB$SIGNAL                         :
                                50  00000000G  EF  D0 0016D 12$:     MOVL    EXCH$A_GBL, R0                         : 1130
           01E4    CB      20   A0             0C  28 00174          MOVC3   #12, 32(R0), 484(HOM)                  :
           01F0    CB    0000'  CF             0C  28 0017B          MOVC3   #12, P.AAV, 496(HOM)                   : 1131
             14    AE    0000'  CF       006A  8F  28 00183          MOVC3   #106, BOOT_PROGRAM, HDRBUF             : 1132
                                57         14  AA  9E 0018C          MOVAB   20(R10), R7                            : 1135
      0C           20      04   B7             67  2C 00190          MOVC5   (R7), @4(R7), #32, #12, 472(HOM)       : 1136
                                         01D8  CB     00196                                                         :
      0C           20      04   B7             67  2C 00199          MOVC5   (R7), @4(R7), #32, #12, HDRBUF+63      : 1137
                                           53  AE     0019F                                                         :
                         01D2   CB             01  B0 001A1          MOVW    #1, 466(HOM)                           : 1140
                         01D4   CB   8EEE0006  8F  D0 001A6          MOVL    #-1897005050, 468(HOM)                 : 1141
                                               58  DD 001AF          PUSHL   HDR                                    : 1145
                                               02  DD 001B1          PUSHL   #2                                     :
                                               7E  D4 001B3          CLRL    -(SP)                                  :
                                           0C  AE  DD 001B5          PUSHL   12(SP)                                 :
                     00000000G  EF             04  FB 001B8          CALLS   #4, EXCH$IO_RT11_WRITE                 :
                           08   AE             50  D0 001BF          MOVL    R0, STATUS                             :
                                70         08  AE  E9 001C3          BLBC    STATUS, 16$                            :
                   50           6E   00000054  8F  C1 001C7          ADDL3   #84, (SP), R0                          : 1151
                                57             60  D0 001CF          MOVL    (R0), RTV                              :
                                           10  AE  D4 001D2          CLRL    16(SP)                                 : 1152
                                               57  D5 001D5          TSTL    RTV                                    :
                                               20  13 001D7          BEQL    13$                                    :
                                           10  AE  D6 001D9          INCL    16(SP)                                 :
                                52   880E00F5  8F  D0 001DC          MOVL    #-2012348171, R2                       : 1155
                                51       0275  8F  3C 001E3          MOVZWL  #629, R1                               :
                                50             57  D0 001E8          MOVL    RTV, R0                                :
                                    00000000G  EF  16 001EB          JSB     EXCH$UTIL_BLOCK_CHECK                  :
           020E    C7           6B       0200  8F  28 001F1          MOVC3   #512, (HOM), 526(RTV)                  : 1156
  0400   8F        00           6E             00  2C 001F9 13$:     MOVC5   #0, (SP), #0, #1024, HDRBUF            : 1161
                                           14  AE     00200                                                         :
```

```
          53      0C  AE             01  C3 00202          SUBL3   #1, START, R3                       ; 1162
                                     52  D4 00207          CLRL    P                                   ;
                                     18  11 00209          BRB     15$                                 ;
                                     58  DD 0020B  14$:    PUSHL   HDR                                 ; 1164
                                     02  DD 0020D          PUSHL   #2                                  ;
                                     52  DD 0020F          PUSHL   P                                   ;
                              0C     AE  DD 00211          PUSHL   12(SP)                              ;
                 00000000G  EF       04  FB 00214          CALLS   #4, EXCH$IO_RT11_WRITE              ;
                        08  AE       50  D0 0021B          MOVL    R0, STATUS                          ;
                        14      08   AE  E9 0021F          BLBC    STATUS, 16$                         ;
     FFE2         52        02       53  F1 00223  15$:    ACBL    R3, #2, P, 14$                      ;
                                     58  DD 00229          PUSHL   HDR                                 ; 1171
                              10     AE  DD 0022B          PUSHL   START                               ;
                     0000V  CF       02  FB 0022E          CALLS   #2, INIT_ZERO_HOME_BLOCKS           ;
                        08  AE       50  D0 00233          MOVL    R0, STATUS                          ;
                            03  08   AE  E8 00237  16$:    BLBS    STATUS, 17$                         ;
                                  009D  31 0023B          BRW     20$                                 ;
                            68  04   AE  3C 0023E  17$:    MOVZWL  SNUM, (HDR)                         ; 1177
                        04  A8       01  B0 00242          MOVW    #1, 4(HDR)                          ; 1179
          06  A8        20  AA       02  A5 00246          MULW3   #2, 32(R10), 6(HDR)                 ; 1180
                        08  A8  0C   AE  B0 0024C          MOVW    START, 8(HDR)                       ; 1181
                        01  A9       02  90 00251          MOVB    #2, 1(ENT)                          ; 1185
                        02  A9 80E82158 8F  D0 00255       MOVL    #-2132270760, 2(ENT)                ; 1186
                        06  A9    26F4  8F  B0 0025D       MOVW    #9972, 6(ENT)                       ; 1187
                            51       59  D0 00263          MOVL    ENT, R1                             ; 1188
                         00000000G   EF  16 00266          JSB     EXCH$RT11_FORMAT_CURRENT_DATE       ;
          08  A9            56  08   A8  A3 0026C          SUBW3   8(HDR), BNUM, 8(ENT)                ; 1189
                            50  06   A8  3C 00272          MOVZWL  6(HDR), R0                          ; 1190
                            59  0E A049  9E 00276          MOVAB   14(R0)[ENT], ENT                    ;
                            51 01FE  C8  9E 0027B          MOVAB   510(R8), R1                         ; 1191
                            51       59  D1 00280          CMPL    ENT, R1                             ;
                                     13  1F 00283          BLSSU   18$                                 ;
                            7E  F7   8F  9A 00285          MOVZBL  #247, -(SP)                         ;
                                     01  DD 00289          PUSHL   #1                                  ;
                         00000000G   8F  DD 0028B          PUSHL   #EXCH$_BADLOGIC                     ;
                 00000000G  00       03  FB 00291          CALLS   #3, LIB$STOP                        ;
                        01  A9       08  90 00298  18$:    MOVB    #8, 1(ENT)                          ; 1192
                            27  10   AE  E9 0029C          BLBC    16(SP), 19$                         ; 1196
      0C0E  C7              68 0200  8F  28 002A0          MOVC3   #512, (HDR), 3086(RTV)              ; 1199
                                     6E  DD 002A8          PUSHL   (SP)                                ; 1200
                 00000000G  EF       01  FB 002AA          CALLS   #1, EXCH$RTACP_VERIFY_DIRECTORY     ;
                            13       50  E8 002B1          BLBS    R0, 19$                             ;
                            7E  F9   8F  9A 002B4          MOVZBL  #249, -(SP)                         ;
                                     01  DD 002B8          PUSHL   #1                                  ;
                         00000000G   8F  DD 002BA          PUSHL   #EXCH$_BADLOGIC                     ;
                 00000000G  00       03  FB 002C0          CALLS   #3, LIB$STOP                        ;
                                     58  DD 002C7  19$:    PUSHL   HDR                                 ; 1205
                                     01  DD 002C9          PUSHL   #1                                  ;
                                     06  DD 002CB          PUSHL   #6                                  ;
                              0C     AE  DD 002CD          PUSHL   12(SP)                              ;
                 00000000G  EF       04  FB 002D0          CALLS   #4, EXCH$IO_RT11_WRITE              ;
                        08  AE       50  D0 002D7          MOVL    R0, STATUS                          ;
                            50  08   AE  D0 002DB  20$:    MOVL    STATUS, R0                          ; 1207
                                         04 002DF          RET                                         ; 1208

; Routine Size:  736 bytes,    Routine Base:  EXCH$INIT_CODE + 07B8
```

```
1 GLOBAL ROUTINE init_zero_home_blocks (start, buf) =     %SBTTL 'init_zero_home_blocks (start, buf)'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Zero any possible Files-11 home blocks on the volume to prevent extraneous privilege problems with
2 !       future mounts.
2 !
2 ! INPUTS:
2 !
2 !       start - the pbn of the first uninitialized block on the volume
2 !       buf   - the address of a 1024-byte buffer which has been set to zeroes
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       work area for INIT
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Success or worst error
2 !
2 ! SIDE EFFECTS:
2 !
2 !       disk blocks may be zeroed
2 !--
2 $dbgtrc_prefix ('init_zero_home_blocks> ');
2
2 LOCAL
2     blockfact,                                              ! device blocking factor
2     delta,                                                  ! home block search delta
2     device_char : $bblock [dib$k_length],                   ! block for device characteristics
2     devchar_desc : VECTOR [2, LONG],                        ! desc for above
2     pbn,                                                    ! physical block number to check
2     status
2     ;
2
2 BIND
2     init = exch$a_gbl [excg$a_init_work] : $ref_bblock,     ! pointer to our work area
2     volb = init [init$a_volb]            : $ref_bblock      ! pointer to exchange VOLB structure
2     ;
```

```
2 ! For virtual volumes we cannot perform a normal home block scan, since the home block search sequence depen
2 ! the physical device geometry.  This is unfortunate, since a virtual volume might be a copy of (and be copi
2 ! back to) a physical device.  Usually, this copy will only be between a small disk (i.e. floppy or TU58) an
2 ! virtual volume.  We will use our knowlege of these disks to perform ad hoc home block zeroing.
2 !
2 IF .volb [volb$v_virtual]
2 THEN
3     BEGIN
3     status = true;                                  ! Assume success
3
3     SELECTONE .volb [volb$l_volmaxblock] OF
3     SET
3         [494] :         IF .start LEQU 8        ! Single density floppy puts alternate home on pbn 8
3                         THEN
3                             status = exch$io_rt11_write (.volb, 8, 1, .buf);
3
3         [988] :         IF .start LEQU 15       ! Double density floppy puts alternate home on pbn 15
3                         THEN
3                             status = exch$io_rt11_write (.volb, 15, 1, .buf);
3
3         [OTHERWISE] :   ;                       ! Ignore large disks, TU58 puts home blocks on pbn 1 and 2 w
3                                                 !  we know that we have already hit
3     TES;
3
3     RETURN .status;                             ! All done with virtual volumes
3     END;
2
2 ! Read the device characteristics
2 !
2 devchar_desc [0] = dib$k_length;               ! Init length of char buffer
2 devchar_desc [1] = device_char;                !  and address of buffer
2
3 IF NOT (status = $getchn (chan=.volb [volb$w_channel], pribuf=devchar_desc))
2 THEN
2     $exch_signal_stop (.status);
```

```
2 ! (Home block geometry calculations borrowed from <INIT.SRC>RDHOME.B32)
2 !
2 ! Compute the home block search delta from the volume geometry in the device table. This is done according t
2 ! following rules, where volume geometry is expressed in the order sectors, tracks, cylinders:
2 !
2 !         n x 1 x 1:      1
2 !         1 x n x 1:      1
2 !         1 x 1 x n:      1
2 !
2 !         n x m x 1:      n+1
2 !         n x 1 x m:      n+1
2 !         1 x n x m:      n+1
2 !
2 !         s x t x c:      (t+1)*s+1
2 !
3 blockfact = (.device_char [dib$b_sectors]
3            * .device_char [dib$b_tracks]
3            * .device_char [dib$w_cylinders])
2            / .device_char [dib$l_maxblock];
2
2 delta = 1;
2 IF  .device_char [dib$w_cylinders] GTR 1
2   AND
2     .device_char [dib$b_tracks] GTR 1
2 THEN
2     delta = .delta + .device_char [dib$b_tracks];
2
2 IF  .device_char [dib$b_sectors] GTR 1
2   AND
3     (.device_char [dib$w_cylinders] GTR 1
3      OR
3      .device_char [dib$b_tracks] GTR 1)
2 THEN
2     delta = (.delta * .device_char [dib$b_sectors] + .blockfact) / .blockfact;
2
2 IF .delta EQL 0
2   OR
2     .delta GTRU .device_char [dib$l_maxblock] / 10
2 THEN
2     delta = 1;
2 $trace_print_fao ('block factor is !UL, delta is !UL', .blockfact, .delta);
2
2 ! Search for the home blocks to zero.  To save time, we will just zap the first five possible positions for
2 ! home blocks.  Note the potential hole:  Disks with the home block far into the disk might not be completel
2 ! zeroed and might have protection anomalies.  C'est la vie.
2 !
2 pbn = 1;                                        ! Start search at pbn 1
2 DECR j FROM 4 TO 0
2 DO
3     BEGIN
3     $trace_print_fao ('index !UL, pbn !UL', .j, .pbn);
3     IF .start LEQU .pbn
3     THEN
4         IF NOT (status = exch$io_rt11_write (.volb, .pbn, 1, .buf))
3         THEN
3             RETURN .status;
3     pbn = .pbn + .delta;
```

```
; 1269          1350  2      END;
; 1270          1351  2
; 1271          1352  2 RETURN .status;
; 1272          1353  1 END;



                                                                   .EXTRN   SYS$GETCHN, LIB$STOP

                                              007C 00000           .ENTRY   INIT_ZERO_HOME_BLOCKS, Save R2,R3,R4,R5,R6   ; 1209
                        56 000000^0G  EF   9E 00002                MOVAB    EXCH$IO_RT11_WRITE, R6                       ;
                        5E        84  AE   9E 00009                MOVAB    -124(SP), SP                                 ;
             50 00000000G EF          10   C1 0000D                ADDL3    #16, EXCH$A_GBL, R0                          ; 1255
             50           6^          04   C1 00015                ADDL3    #4, (R0), R0                                 ; 1256
                          53          60   D0 00019                MOVL     (R0), R3                                     ; 1263
             40       48  A3          04   E1 0001C                BBC      #4, 72(R3), 4$                               ;
                          51          01   D0 00021                MOVL     #1, STATUS                                   ; 1266
                          50      44  A3   D0 00024                MOVL     68(R3), R0                                   ; 1268
                000001EE  8F          50   D1 00028                CMPL     R0, #494                                     ; 1270
                                      0F   12 0002F                BNEQ     1$                                           ;
                          08      04  AC   D1 00031                CMPL     START, #8                                    ;
                                      27   1A 00035                BGTRU    3$                                           ;
                                  08  AC   DD 00037                PUSHL    BUF                                          ; 1272
                                      01   DD 0003A                PUSHL    #1                                           ;
                                      08   DD 0003C                PUSHL    #8                                           ;
                                      16   11 0003E                BRB      2$                                           ;
                000003DC  8F          50   D1 00040 1$:            CMPL     R0, #988                                     ; 1274
                                      15   12 00047                BNEQ     3$                                           ;
                          0F      04  AC   D1 00049                CMPL     START, #15                                   ;
                                      0F   1A 0004D                BGTRU    3$                                           ;
                                  08  AC   DD 0004F                PUSHL    BUF                                          ; 1276
                                      01   DD 00052                PUSHL    #1                                           ;
                                      0F   DD 00054                PUSHL    #15                                          ;
                                      53   DD 00056 2$:            PUSHL    R3                                           ;
                          66          04   FB 00058                CALLS    #4, EXCH$IO_RT11_WRITE                       ;
                          51          50   D0 0005B                MOVL     R0, STATUS                                   ;
                                    00AC   31 0005E 3$:            BRW      13$                                          ; 1282
                          6E      74  8F   9A 00061 4$:            MOVZBL   #116, DEVCHAR_DESC                           ; 1287
                     04   AE      08  AE   9E 00065                MOVAB    DEVICE_CHAR, DEVCHAR_DESC+4                  ; 1288
                                      7E   7C 0006A                CLRQ     -(SP)                                        ; 1290
                                  08  AE   9F 0006C                PUSHAB   DEVCHAR_DESC                                 ;
                                      7E   D4 0006F                CLRL     -(SP)                                        ;
                          7E      4A  A3   3C 00071                MOVZWL   74(R3), -(SP)                                ;
                00000000G 00          05   FB 00075                CALLS    #5, SYS$GETCHN                               ;
                          51          50   D0 0007C                MOVL     R0, STATUS                                   ;
                          0A          51   E8 0007F                BLBS     STATUS, 5$                                   ;
                                      51   DD 00082                PUSHL    STATUS                                       ; 1292
                00000000G 00          01   FB 00084                CALLS    #1, LIB$STOP                                 ;
                                           04 0008B                RET                                                   ;
                          50      10  AE   9A 0008C 5$:            MOVZBL   DEVICE_CHAR+8, R0                            ; 1309
                          52      11  AE   9A 00090                MOVZBL   DEVICE_CHAR+9, R2                            ;
                          50          52   C4 00094                MULL2    R2, R0                                       ;
                          54      12  AE   3C 00097                MOVZWL   DEVICE_CHAR+10, R4                           ; 1310
                          50          54   C4 0009B                MULL2    R4, R0                                       ;
             54           50      78  AE   C7 0009E                DIVL3    DEVICE_CHAR+112, R0, BLOCKFACT               ; 1311
                          52          01   D0 000A3                MOVL     #1, DELTA                                    ; 1313
                                      50   D4 000A6                CLRL     R0                                           ; 1314
```

```
                 01      12   AE  B1 000A8        CMPW    DEVICE_CHAR+10, #1                         ;
                              0F  1B 000AC        BLEQU   6$                                         ;
                              50  D6 000AE        INCL    R0                                         ;
                 01      11   AE  91 000B0        CMPB    DEVICE_CHAR+9, #1                          ; 1316
                              07  1B 000B4        BLEQU   6$                                         ;
                 55      11   AE  9A 000B6        MOVZBL  DEVICE_CHAR+9, R5                          ; 1318
                 52           55  C0 000BA        ADDL2   R5, DELTA                                  ;
                 01      10   AE  91 000BD 6$:    CMPB    DEVICE_CHAR+8, #1                          ; 1320
                              17  1B 000C1        BLEQU   8$                                         ;
                 06           50  E8 000C3        BLBS    R0, 7$                                     ; 1322
                 01      11   AE  91 000C6        CMPB    DEVICE_CHAR+9, #1                          ; 1324
                              0E  1B 000CA        BLEQU   8$                                         ;
                 50      10   AE  9A 000CC 7$:    MOVZBL  DEVICE_CHAR+8, R0                          ; 1326
                 50           52  C4 000D0        MULL2   DELTA, R0                                  ;
                 50           54  C0 000D3        ADDL2   BLOCKFACT, R0                              ;
         52      50           54  C7 000D6        DIVL3   BLOCKFACT, R0, DELTA                       ;
                              52  D5 000DA 8$:    TSTL    DELTA                                      ; 1328
                              0A  13 000DC        BEQL    9$                                         ;
         50      78      AE   0A  C7 000DE        DIVL3   #10, DEVICE_CHAR+112, R0                   ; 1330
                         50   52  D1 000E3        CMPL    DELTA, R0                                  ;
                              03  1B 000E6        BLEQU   10$                                        ;
                 52           01  D0 000E8 9$:    MOVL    #1, DELTA                                  ; 1332
                 54           01  D0 000EB 10$:   MOVL    #1, PBN                                    ; 1339
                 55           04  D0 000EE        MOVL    #4, J                                      ; 1349
                 54      04   AC  D1 000F1 11$:   CMPL    START, PBN                                 ; 1344
                              10  1A 000F5        BGTRU   12$                                        ;
                         08   AC  DD 000F7        PUSHL   BUF                                        ; 1346
                              01  DD 000FA        PUSHL   #1                                         ;
                              18  BB 000FC        PUSHR   #^M<R3,R4>                                 ;
                 66           04  FB 000FE        CALLS   #4, EXCH$IO_RT11_WRITE                     ;
                 51           50  D0 00101        MOVL    R0, STATUS                                 ;
                 06           51  E9 00104        BLBC    STATUS, 13$                                ;
                 54           52  C0 00107 12$:   ADDL2   DELTA, PBN                                 ; 1349
                 E4           55  F4 0010A        SOBGEQ  J, 11$                                     ; 1340
                 50           51  D0 0010D 13$:   MOVL    STATUS, R0                                 ; 1352
                                  04 00110        RET                                                ; 1353

; Routine Size:  273 bytes,    Routine Base:  EXCH$INIT_CODE + 0A98
; 1274           1354  1 END
; 1275           1355  0 ELUDOM


                                                                   .EXTRN  LIB$SIGNAL, LIB$STOP

;                                        PSECT SUMMARY
;
;       Name                          Bytes                      Attributes
;
;   EXCH$INIT_PLIT                      286  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   EXCH$INIT_CODE                     2985  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       122         0        1000        00:01.8
;   _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1          1151       142        12          79        00:01.3



;                                       COMMAND QUALIFIERS

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

; Size:          2985 code + 286 data bytes
; Run Time:         00:55.5
; Elapsed Time:     03:18.7
; Lines/CPU Min:     1465
; Lexemes/CPU-Min: 25197
; Memory Used:  279 pages
; Compilation Complete
```

0161 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY