G 3

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

J 14

```
**FILE**ID**EXCDOS11
```

```
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      OOOOOO      SSSSSSSS      11          11
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      OOOOOO      SSSSSSSS      11          11
EE          XX      XX  CC          DD      DD  OO      OO  SS            1111        1111
EE          XX      XX  CC          DD      DD  OO      OO  SS            1111        1111
EE            XX  XX    CC          DD      DD  OO      OO  SS              11          11
EE            XX  XX    CC          DD      DD  OO      OO  SS              11          11
EEEEEEEE        XX      CC          DD      DD  OO      OO    SSSSSS        11          11
EEEEEEEE        XX      CC          DD      DD  OO      OO    SSSSSS        11          11
EE            XX  XX    CC          DD      DD  OO      OO          SS      11          11
EE            XX  XX    CC          DD      DD  OO      OO          SS      11          11
EE          XX      XX  CC          DD      DD  OO      OO          SS      11          11           ....
EE          XX      XX  CC          DD      DD  OO      OO          SS      11          11           ....
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      OOOOOO    SSSSSSSS    111111      111111         ....
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      OOOOOO    SSSSSSSS    111111      111111         ....

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

```
0 MODULE  exch$dos11                             %TITLE 'dos11 file and directory routines'
0               (
0               IDENT = 'V04-000',
0               ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:     EXCHANGE - Foreign volume interchange facility
1 !
1 ! ABSTRACT:     DOS-11 volume specific routines
1 !
1 ! ENVIRONMENT:  VAX/VMS User mode
1 !
1 ! AUTHOR:       CW Hobbs                 CREATION DATE: 22-Mar-1983
1 !
1 ! MODIFIED BY:
1 !
1 !         V03-003 CWH3003         CW Hobbs                 12-Apr-1984
1 !                 Make /TRANSFER=BLOCK a global qualifier by looking at
1 !                 both input and output specs.
1 !
1 !         V03-002 CWH3002         CW Hobbs                 6-Feb-1984
1 !                 Change calculation of DOS11 date to use signed INCR to
1 !                 avoid looping in January.
1 !
1 !
1 !--
1
1 ! Include files:
1 !
1 MACRO $module_name_string = 'exch$dos11' %;     ! The require file needs to know our module name
1 REQUIRE 'SRC$:EXCREQ'                           ! Facility-wide require file
1     ;
```

```
1 %SBTTL 'Module table of contents'
1
1 ! Module table of contents:
1 !
1 FORWARD ROUTINE
1     exch$dos11_close_file,                            ! DOS-11 specific file close routine
1     exch$dos11_create_file,                           ! DOS-11 volume file create
1     exch$dos11_directory,                             ! DOS-11 volume directory command
1     exch$dos11_dire_put_item     : NOVALUE,           ! Add an file item to the directory line
1     exch$dos11_exit_handler      : NOVALUE,           ! Write tape marks if an exit occurs with an open output fil
1     exch$dos11_expand_filename   : NOVALUE,           ! Convert tape label to ASCII filename
1     exch$dos11_find_file,                             ! Find a file entry in the temporary
1     exch$dos11_form_cur_date : NOVALUE jsb_r1,        ! Put the current date into a directory entry
1          dos11_match_uic         : jsb_r0r1,          ! Check uic directory matches
1     exch$dos11_mount,                                 ! DOS-11 specific volume mount routine
1     exch$dos11_next_entry,                            ! Retrieve a pointer to the next file name entry
1     exch$dos11_open_file,                             ! DOS-11 specific file open routine
1     exch$dos11_position_tape                          ! Move the tape to the start of a particular file
1     ;
1
1 ! EXCHANGE facility routines
1 !
1 EXTERNAL ROUTINE
1     exch$cmd_fetch_recfmt_implied : NOVALUE,          ! Get or assume the value for /RECORD_FORMAT
1     exch$cmd_match_filename,                          ! Compare expanded file names for match
1     exch$cmd_related_file_parse,                      ! Build an output file name
1     exch$dire_get_columns,                            ! Get number of colums for the output file
1     exch$dire_get_width,                              ! Get the output file width
1     exch$dire_put,                                    ! Put a line to the output file
1     exch$io_dos11_count_blocks,                       ! Count blocks to the next tape mark
1     exch$io_dos11_read,                               ! Read blocks from a sequential device
1     exch$io_dos11_read_label,                         ! Read sequential device file label
1     exch$io_dos11_rewind,                             ! Rewind a sequential device
1     exch$io_dos11_skip_file,                          ! Skip a file on a sequential device
1     exch$io_dos11_skip_record,                        ! Skip a physical record on a sequential device
1     exch$io_dos11_write,                              ! Write blocks to a sequential device
1     exch$io_dos11_write_label,                        ! Write a label on a sequential device
1     exch$io_dos11_write_tape_mark,                    ! Write a tape mark on a sequential device
1     exch$pdp_filter_filename,                         ! Remove invalid characters from a filename
1     exch$pdp_flush_write_buffer,                      ! Flush any records waiting in the write buffer
1     exch$pdp_get,                                     ! Get functions for small PDP record structure
1     exch$pdp_put,                                     ! Put functions for small PDP record structure
1     exch$util_fao_buffer,                             ! Do an FAO conversion
1     exch$util_radix50_from_ascii,                     ! Convert characters to Radix-50 from Ascii
1     exch$util_radix50_to_ascii,                       ! Convert characters from Radix-50 to Ascii
1     exch$util_dos11ctx_allocate,                      ! Get an DOS-11 context block
1     exch$util_dos11ctx_release   : NOVALUE,           ! Give it back
1     exch$util_vm_allocate,                            ! Get some virtual memory
1     exch$util_vm_allocate_zeroed,                     ! Get some virtual memory, initialized to zero
1     exch$util_vm_release                              ! Return some virtual memory
1     ;
1
1 ! Store some constants in read-only own
1 !
1 OWN
1     months : VECTOR [13, LONG] INITIAL
1                 ('Jan-','Feb-','Mar-','Apr-','May-','Jun-','Jul-','Aug-','Sep-','Oct-','Nov-','Dec-','***-')
```

```
1     days_of_months : VECTOR [12, BYTE] PRESET
                 ([0] = 31, [1] = 28, [2] = 31, [3] = 30,  [4] = 31,  [5] = 30,
                  [6] = 31, [7] = 31, [8] = 30, [9] = 31, [10] = 30, [11] = 31),
1     days_of_months_leapyear : VECTOR [12, BYTE] PRESET
                 ([0] = 31, [1] = 29, [2] = 31, [3] = 30,  [4] = 31,  [5] = 30,
                  [6] = 31, [7] = 31, [8] = 30, [9] = 31, [10] = 30, [11] = 31)
1     ;
1
1 ! Equated symbols:
1 !
1 !LITERAL
1 !     ;
1
1 ! Bound declarations:
1 !
1 BIND
1     null_string = %ASCID ''
1     ;
```

```
1 GLOBAL ROUTINE exch$dos11_close_file (filb : $ref_bblock) =     %SBTTL 'exch$dos11_close_file (filb)'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform DOS-11 volume specific close processing
2 !
2 ! INPUT/OUTPUT:
2 !
2 !       filb - pointer to block describing the file
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       filb - receive info pertaining to the file to be closed
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to close the file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('exch$dos11_close_file> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     ctx  = filb [filb$a_context]        : $ref_bblock,
2     namb = filb [filb$a_assoc_namb]     : $ref_bblock,
2     volb = filb [filb$a_assoc_volb]     : $ref_bblock,
2     dosv = volb [volb$a_vfmt_specific]  : $ref_bblock,
2     ent  = dosv [dos11$a_current_entry] : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .filb, filb, 580);                       !?? definitely over-zealous checking
2 $block_check (2, .namb, namb, 581);
2 $block_check (2, .volb, volb, 582);
2 $block_check (2, .dosv, dos11, 442);
2 $block_check (2, .ctx, dos11ctx, 583);
2 $logic_check (4, (.ctx [dos11ctx$a_assoc_filb] EQL .filb), 241);
2 $logic_check (4, (.ctx [dos11ctx$a_assoc_volb] EQL .volb), 248);
```

```
2 ! Output files need some directory tweaks and a buffer flush
2 !
2 IF .ctx [dos11ctx$v_output_file]
2 THEN
3     BEGIN
3
3     ! Flush any blocks that are sitting in the output buffer
3     !
3     status = exch$pdp_flush_write_buffer (.ctx);
3
3     ! Write two tape marks to mark the end of the tape, try even if buffer flush failed
3     !
3     exch$io_dos11_write_tape_mark (.volb);
3     exch$io_dos11_write_tape_mark (.volb);
3
3     ! Cancel the exit handler which was declared to write the tape marks
3     !
3     $canexh (desblk=exch$a_gbl [excg$r_exit_block]);
3     exch$a_gbl [excg$a_exh_routine] = 0;          ! Clear address, means no exit handler is active
3
3     ! If the buffer flush failed, exit now
3     !
3     IF NOT .status
3     THEN
3         RETURN .status;
3
3     ! We have finished this file, and we are sitting on the tape mark ready for the next file
3     !
3     dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;
3     $trace_print_fao ('dosv file number updated to !UL', .dosv [dos11$l_current_file]);
3
3     END
3
3 ! Input file, we just need to see if we got to the tape mark
3 !
2 ELSE
3     BEGIN
3
3     ! If we didn't get to see the tape mark at the end of the file, something happened and we may be lost
3     !
3     IF NOT .dosv [dos11$v_tape_mark]
3     THEN
4         BEGIN
4
4         ! Rewind the tape so that the exact position will be known again
4         !
4         $exch_signal (exch$_dos11_position);      ! This might take a while, warn 'em
4         exch$io_dos11_rewind (.volb);
4         END
4
4     ! We have finished this file, and we are sitting on the tape mark ready for the next file
4     !
3     ELSE
4         BEGIN
4         dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;
4         $trace_print_fao ('dosv file number updated to !UL', .dosv [dos11$l_current_file]);
3         END;
```

```
2     END;
2
2 ! If the block count in the entry is not valid, then update the count from the context and mark it valid
2 !
2 IF NOT .ent [dos11ent$v_blocks_valid]
2 THEN
3     BEGIN
3     REGISTER
3         count;
3
3     ! Update the block count in the entry
3     !
3     count = .ctx [dos11ctx$l_eof_block] + 1;
3     IF .count NEQ 0
3     THEN
4         BEGIN
4         ent [dos11ent$w_blocks] = .count;
4         ent [dos11ent$v_blocks_valid] = true;
3         END;
2     END;
2
2 ! Clear the stream active bit and all other context flags
2 !
2 ctx [dos11ctx$l_flags] = 0;
2
2 RETURN true;
1 END;


                                                                  .TITLE  EXCH$DOS11 dos11 file and directory routines
                                                                  .IDENT  \V04-000\

                                                                  .PSECT  EXCH$DOS11_PLIT,NOWRT,2

                                    2D  6E  61  4A  00000 MONTHS: .ASCII  \Jan-\                                              ;
                                    2D  62  65  46  00004         .ASCII  \Feb-\                                              ;
                                    2D  72  61  4D  00008         .ASCII  \Mar-\                                              ;
                                    2D  72  70  41  0000C         .ASCII  \Apr-\                                              ;
                                    2D  79  61  4D  00010         .ASCII  \May-\                                              ;
                                    2D  6E  75  4A  00014         .ASCII  \Jun-\                                              ;
                                    2D  6C  75  4A  00018         .ASCII  \Jul-\                                              ;
                                    2D  67  75  41  0001C         .ASCII  \Aug-\                                              ;
                                    2D  70  65  53  00020         .ASCII  \Sep-\                                              ;
                                    2D  74  63  4F  00024         .ASCII  \Oct-\                                              ;
                                    2D  76  6F  4E  00028         .ASCII  \Nov-\                                              ;
                                    2D  63  65  44  0002C         .ASCII  \Dec-\                                              ;
                                    2D  2A  2A  2A  00030         .ASCII  \***-\                                              ;
    1F  1E  1F  1E  1F  1F  1E  1F  1E  1F  1C  1F  00034 DAYS_OF_MONTHS:
                                                                  .BYTE   31, 28, 31, 30, 31, 30, 31, 31, 30, 31, -           ;
                                                                          30, 31                                              ;
    1F  1E  1F  1E  1F  1F  1E  1F  1E  1F  1D  1F  00040 DAYS_OF_MONTHS_LEAPYEAR:
                                                                  .BYTE   31, 29, 31, 30, 31, 30, 31, 31, 30, 31, -           ;
                                                                          30, 31                                              ;
                                                  0004C P.AAB:    .BLKB   0                                                   ;
                                        010E0000  0004C P.AAA:    .LONG   17694720                                            ;
                                        00000000' 00050           .ADDRESS P.AAB                                              ;
```

```
                                                     NULL_STRING=         P.AAA
                                                             .EXTRN   EXCH$CMD_FETCH_RECFMT_IMPLIED
                                                             .EXTRN   EXCH$CMD_MATCH_FILENAME
                                                             .EXTRN   EXCH$CMD_RELATED_FILE_PARSE
                                                             .EXTRN   EXCH$DIRE_GET_COLUMNS
                                                             .EXTRN   EXCH$DIRE_GET_WIDTH
                                                             .EXTRN   EXCH$DIRE_PUT, EXCH$IO_DOS11_COUNT_BLOCKS
                                                             .EXTRN   EXCH$IO_DOS11_READ
                                                             .EXTRN   EXCH$IO_DOS11_READ_LABEL
                                                             .EXTRN   EXCH$IO_DOS11_REWIND
                                                             .EXTRN   EXCH$IO_DOS11_SKIP_FILE
                                                             .EXTRN   EXCH$IO_DOS11_SKIP_RECORD
                                                             .EXTRN   EXCH$IO_DOS11_WRITE
                                                             .EXTRN   EXCH$IO_DOS11_WRITE_LABEL
                                                             .EXTRN   EXCH$IO_DOS11_WRITE_TAPE_MARK
                                                             .EXTRN   EXCH$PDP_FILTER_FILENAME
                                                             .EXTRN   EXCH$PDP_FLUSH_WRITE_BUFFER
                                                             .EXTRN   EXCH$PDP_GET, EXCH$PDP_PUT
                                                             .EXTRN   EXCH$UTIL_FAO_BUFFER
                                                             .EXTRN   EXCH$UTIL_RADIX50_FROM_ASCII
                                                             .EXTRN   EXCH$UTIL_RADIX50_TO_ASCII
                                                             .EXTRN   EXCH$UTIL_DOS11CTX_ALLOCATE
                                                             .EXTRN   EXCH$UTIL_DOS11CTX_RELEASE
                                                             .EXTRN   EXCH$UTIL_VM_ALLOCATE
                                                             .EXTRN   EXCH$UTIL_VM_ALLOCATE_ZEROED
                                                             .EXTRN   EXCH$UTIL_VM_RELEASE
                                                             .EXTRN   EXCH$UTIL_BLOCK_CHECK
                                                             .EXTRN   EXCH$A_GBL, SYS$CANEXH
                                                             .EXTRN   EXCH$_DOS11_POSITION

                                                             .PSECT   EXCH$DOS11_CODE,NOWRT,2

                                          03FC 00000         .ENTRY   EXCH$DOS11_CLOSE_FILE, Save R2,R3,R4,R5,R6,-;  0229
                                                                      R7,R8,R9                                    ;
                         59 00000000G  EF   9E 00002         MOVAB    EXCH$A_GBL, R9                              ;
                         58 00000000G  EF   9E 00009         MOVAB    EXCH$IO_DOS11_WRITE_TAPE_MARK, R8           ;
                         57 00000000G  EF   9E 00010         MOVAB    EXCH$UTIL_BLOCK_CHECK, R7                   ;
      56        04       AC            20   C1 00017         ADDL3    #32, FILB, R6                               ;  0269
      54        04       AC            18   C1 0001C         ADDL3    #24, FILB, R4                               ;  0270
      55        04       AC            1C   C1 00021         ADDL3    #28, FILB, R5                               ;  0271
      50                 65 00000054   8F   C1 00026         ADDL3    #84, (R5), R0                               ;  0272
                         53            60   D0 0002E         MOVL     (R0), R3                                    ;  0273
                         52 035B00FA   8F   D0 00031         MOVL     #56295674, R2                               ;  0278
                         51     0244   8F   3C 00038         MOVZWL   #580, R1                                    ;
                         50       04   AC   D0 0003D         MOVL     FILB, R0                                    ;
                                       67   16 00041         JSB      EXCH$UTIL_BLOCK_CHECK                       ;
                         52 010A00F7   8F   D0 00043         MOVL     #17432823, R2                               ;  0279
                         51     0245   8F   3C 0004A         MOVZWL   #581, R1                                    ;
                         50            64   D0 0004F         MOVL     (R4), R0                                    ;
                                       67   16 00052         JSB      EXCH$UTIL_BLOCK_CHECK                       ;
                         52 041B00F3   8F   D0 00054         MOVL     #68878579, R2                               ;  0280
                         51     0246   8F   3C 0005B         MOVZWL   #582, R1                                    ;
                         50            65   D0 00060         MOVL     (R5), R0                                    ;
                                       67   16 00063         JSB      EXCH$UTIL_BLOCK_CHECK                       ;
                         52 003600FD   8F   D0 00065         MOVL     #3539197, R2                                ;  0281
                         51     01BA   8F   3C 0006C         MOVZWL   #442, R1                                    ;
                         50            53   D0 00071         MOVL     R3, R0                                      ;
```

```
                                       67  16 00074          JSB      EXCH$UTIL_BLOCK_CHECK                        ;
                         54            66  D0 00076          MOVL     (R6), R4                                     ; 0282
                         52 008A00FC   8F  D0 00079          MOVL     #9044220, R2                                 ;
                         51      0247  8F  3C 00080          MOVZWL   #583, R1                                     ;
                         50            54  D0 00085          MOVL     R4, R0                                       ;
                                       67  16 00088          JSB      EXCH$UTIL_BLOCK_CHECK                        ;
        2E       28      A4            01  E1 0008A          BBC      #1, 40(R4), 1$                               ; 0287
                                       54  DD 0008F          PUSHL    R4                                           ; 0293
            00000000G    EF            01  FB 00091          CALLS    #1, EXCH$PDP_FLUSH_WRITE_BUFFER              ;
                         52            50  D0 00098          MOVL     R0, STATUS                                   ;
                                       65  DD 0009B          PUSHL    (R5)                                         ; 0297
                         68            01  FB 0009D          CALLS    #1, EXCH$IO_DOS11_WRITE_TAPE_MARK            ;
                                       65  DD 000A0          PUSHL    (R5)                                         ; 0298
                         68            01  FB 000A2          CALLS    #1, EXCH$IO_DOS11_WRITE_TAPE_MARK            ;
        7E               69            2C  C1 000A5          ADDL3    #44, EXCH$A_GBL, -(SP)                       ; 0302
            00000000G    00            01  FB 000A9          CALLS    #1, SYS$CANEXH                               ;
                         50            69  D0 000B0          MOVL     EXCH$A_GBL, R0                               ; 0303
                                 30    A0  D4 000B3          CLRL     48(R0)                                       ;
                         21            52  E8 000B6          BLBS     STATUS, 2$                                   ; 0307
                         50            52  D0 000B9          MOVL     STATUS, R0                                   ; 0309
                                           04 000BC          RET                                                   ;
        18       0C      A3            02  E0 000BD 1$:      BBS      #2, 12(R3), 2$                               ; 0325
                            00000000G  8F  DD 000C2          PUSHL    #EXCH$_DOS11_POSITION                        ; 0331
            00000000G    00            01  FB 000C8          CALLS    #1, LIB$SIGNAL                               ;
                                       65  DD 000CF          PUSHL    (R5)                                         ; 0332
            00000000G    EF            01  FB 000D1          CALLS    #1, EXCH$IO_DOS11_REWIND                     ;
                                       03  11 000D8          BRB      3$                                           ; 0325
                                 0E    A3  D6 000DA 2$:      INCL     14(R3)                                       ; 0339
                         51      1A    A3  D0 000DD 3$:      MOVL     26(R3), R1                                   ; 0346
                         0F      1A    A1  E8 000E1          BLBS     26(R1), 4$                                   ;
        50       20      A4            01  C1 000E5          ADDL3    #1, 32(R4), COUNT                            ; 0354
                                       08  13 000EA          BEQL     4$                                           ; 0355
                 16      A1            50  B0 000EC          MOVW     COUNT, 22(R1)                                ; 0358
                 1A      A1            01  88 000F0          BISB2    #1, 26(R1)                                   ; 0359
                                 28    A4  D4 000F4 4$:      CLRL     40(R4)                                       ; 0365
                         50            01  D0 000F7          MOVL     #1, R0                                       ; 0367
                                           04 000FA          RET                                                   ; 0368

; Routine Size:  251 bytes,    Routine Base:  EXCH$DOS11_CODE + 0000
```

```
1 GLOBAL ROUTINE exch$dos11_create_file = %SBTTL 'exch$dos11_create_file'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform DOS-11 volume specific create processing
2 !
2 ! INPUT:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       copy [copy$a_out_filb] - pointer to the filb for the output file
2 !       copy [copy$a_inp_filb] - pointer to the filb for the input file
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       copy [copy$a_out_filb] - block receives info pertaining to the created file
2
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to create a file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('dos11_create_file> ');
2
2 LOCAL
2     rfp : $bblock [nam$c_bln+nam$c_maxrss],     ! output file parse - an RMS NAM block plus expanded string
2     nam_len,                                    ! temporary to hold length of name
2     typ_len,                                    ! temporary to hold length of type
2     tot_len,                                    ! temporary to hold length of name + type
2     ent : $ref_bblock,                          ! a pointer to the entry we are adding
2     start_block,                                ! the pbn where this file will start
2     temp,
2     status
2     ;
2
2 BIND
2     copy     = exch$a_gbl [excg$a_copy_work]    : $ref_bblock,
2     out_name = copy [copy$q_output_filename]    : $desc_block,
2     inp_filb = copy [copy$a_inp_filb]           : $ref_bblock,
2     inp_namb = inp_filb [filb$a_assoc_namb]     : $ref_bblock,
2     inp_volb = inp_filb [filb$a_assoc_volb]     : $ref_bblock,
2     inp_ctx  = inp_filb [filb$a_context]        : $ref_bblock,
2     out_filb = copy [copy$a_out_filb]           : $ref_bblock,
2     out_namb = out_filb [filb$a_assoc_namb]     : $ref_bblock,
2     out_ctx  = out_filb [filb$a_context]        : $ref_bblock,
2     volb     = out_filb [filb$a_assoc_volb]     : $ref_bblock,
2     dosv     = volb [volb$a_vfmt_specific]      : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .out_filb, filb, 584);
2 $block_check (2, .inp_filb, filb, 585);
2 $block_check (2, .out_namb, namb, 586);
2 $block_check (2, .inp_namb, namb, 587);
2 $block_check (2, .volb, volb, 588);
2 $block_check_if_nonzero (2, .inp_volb, volb, 634);
2 $block_check (2, .dosv, dos11, 595);
2
2
2 ! Make certain that write access is permitted, this should be checked long before we get here
2 !
2 $logic_check (2, (.volb [volb$v_write]), 253);
2
2 ! If the context pointer is null, then allocate and initialize it.
2 !
2 IF .out_ctx EQL 0
2 THEN
2     out_ctx = exch$util_dos11ctx_allocate (.volb, .out_filb)    ! Get an dos11 context block
2
2 ELSE
2     $block_check (2, .out_ctx, dos11ctx, 589);                  ! Make sure that it is what we think it is
2
2 ! Make sure that we haven't crossed signals someplace
2 !
2 $logic_check (4, (.out_ctx [dos11ctx$a_assoc_filb] EQL .out_filb), 254);
2 $logic_check (4, (.out_ctx [dos11ctx$a_assoc_volb] EQL .volb), 255);
2
2 ! Set the rest of the block to nulls, nothing carries over from one output file to the next
2 !
2 CH$FILL (0, dos11ctx$k_end_zero - dos11ctx$k_start_zero,        ! Set rest of block to nulls
2                 .out_ctx + dos11ctx$k_start_zero);
2
2 ! Perform an RMS output file parse on the related name (the result name for the input file) and the
2 ! requested output name from the command line.
2 !
3 IF NOT (status = exch$cmd_related_file_parse (
3                 .out_name [dsc$w_length], .out_name [dsc$a_pointer],                  ! Command line out p
3                 .inp_filb [filb$l_result_name_len], inp_filb [filb$t_result_name],      ! Related name
3                 rfp))                                                                   ! Gets new name
2 THEN
3     BEGIN
3
3     ! Move the raw name to where it is accessible for the outer signal
3     !
3     CH$MOVE (.out_name [dsc$w_length], .out_name [dsc$a_pointer], out_filb [filb$t_result_name]);
3     RETURN .status;
3
2     END;
2
2 ! Create the result file name in the filb
```

```
2 !
2 out_filb [filb$v_name_change] = false;            ! Assume no name change
2 tot_len = .rfp [nam$b_name];                      ! Remember starting length of name
2 rfp [nam$b_name] = exch$pdp_filter_filename (.rfp [nam$b_name], .rfp [nam$l_name]);     ! Remove invalid cha
2 nam_len = MINU (.rfp [nam$b_name], 9);            ! Maximum name is nine letters
2 IF .tot_len NEQ .nam_len                          ! If length not same as start, then it has changed
2 THEN
2     out_filb [filb$v_name_change] = true;
2
2 tot_len = .rfp [nam$b_type];                      ! Remember starting length of type
2 rfp [nam$b_type] = 1 + exch$pdp_filter_filename (.rfp [nam$b_type] - 1, .rfp [nam$l_type] + 1);
2 typ_len = MINU (.rfp [nam$b_type], 4);            ! Maximum type is three (plus the separating dot)
2 IF .tot_len NEQ .typ_len                          ! If length not same as start, then it has changed
2 THEN
2     out_filb [filb$v_name_change] = true;
2
2 tot_len = .nam_len + .typ_len;                    ! Final length of both
2 out_filb [filb$l_result_name_len] = .volb [volb$l_vol_ident_len] + .tot_len;    ! Length of    me ident
2 $logic_check (2, (.out_filb [filb$l_result_name_len] [EQU filb$s_result_name), 256);
2 CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],                  ! Volume n
2                 .nam_len, .rfp [nam$l_name], .typ_len, .rfp [nam$l_type],
2                 0, filb$s_result_name, out_filb [filb$t_result_name]);
2
2 $debug_print_fao ('Looking for "!AF"', .out_filb [filb$l_result_name_len], out_filb [filb$t_result_name]);
2
2 ! Make sure that the input and output files are not on the same device
2 !
2 IF .volb EQL .inp_volb
2 THEN
2     RETURN exch$_illmtcopy;
2
2 ! Reset the rest of the block to nulls, nothing carries over from before
2 !
2 CH$FILL (0, dos11ctx$k_end_zero - dos11ctx$k_start_zero,         ! Set rest of block to nulls
2                 .out_ctx + dos11ctx$k_start_zero);
2
2 ! If an /ALLOCATION qualifier has been seen, tell user we are ignoring
2 !
2 IF .inp_filb [filb$l_q_allocation] NEQ 0        ! If specified on the input
2  OR
2    .copy [copy$l_q_allocation] NEQ 0            !   or on the output
2 THEN
2     $exch_signal (exch$_noalloc);               !?? also /CONT /BEST ...
2
2 ! Make sure that the record format in the filb is correct
2 !
2 exch$cmd_fetch_recfmt_implied (.out_filb, .rfp [nam$l_type]+1); ! Pass it the type from the parse
2
2 ! Save the addresses of our routines for this volume and record format.
2 !
2 out_filb [filb$a_close_routine] = exch$dos11_close_file;
2 out_filb [filb$a_delete_routine] = exch$dos11_close_file;
2 out_filb [filb$a_get_routine] = 0;
2 out_filb [filb$a_put_routine] = exch$pdp_put;
2
2 ! Carriage control doesn't mean anything for DOS-11 output, tell him we are ignoring
2 !
```

```
2 IF .inp_filb [filb$v_cctl_explicit]
2  OR
2     .out_filb [filb$v_cctl_explicit]
2 THEN
2     $exch_signal (exch$_nocarriage);
2
2 ! For DOS-11 we can treat block transfer mode as fixed 512
2 !
2 IF .out_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
2   OR
2     .inp_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
2 THEN
3     BEGIN
3     out_filb [filb$b_rec_format]  = filb$k_rfmt_fixed;
3     out_filb [filb$l_fixed_len]   = 512;
2     END;
2
2 ! In some circumstances we can do block mode I/O rather than record mode
2 !
3 IF  (.inp_filb [filb$b_transfer_mode] EQL filb$k_xfrm_automatic ! Both input and output must be automatic tr
3      AND .out_filb [filb$b_transfer_mode] EQL filb$k_xfrm_automatic)
2   AND
4     ( NOT (.inp_filb [filb$v_rfmt_explicit]                       ! Both the input and output files must have
3         OR .out_filb [filb$v_rfmt_explicit]))                     !  implied record formats
2   AND
3     (.inp_namb [namb$b_vol_format] EQL volb$k_vfmt_dos11          ! The input must be DOS-11
3      OR .inp_namb [namb$b_vol_format] EQL volb$k_vfmt_rt11)       !  or RT-11
2 THEN
3     BEGIN
3     $debug_print_lit ('using block transfer mode');
3     inp_filb [filb$b_rec_format]  = filb$k_rfmt_fixed;
3     inp_filb [filb$l_fixed_len]   = 512;
3     out_filb [filb$b_rec_format]  = filb$k_rfmt_fixed;
3     out_filb [filb$l_fixed_len]   = 512;
2     END;
2
2 ! Make sure that the directory list is complete
2 !
2 ent = exch$util_dos11ctx_allocate (.volb, 0);           ! Grab a context block for the loop
2 temp = dosnxt$m_rewind OR dosnxt$m_count_blocks;       ! Rewind for the first file
2 WHILE exch$dos11_next_entry (.ent, .temp) NEQ 0
2 DO
2     temp = dosnxt$m_count_blocks;                       ! Just count for later files
2 exch$util_dos11ctx_release (.ent);                      ! Return the context block
2
2 ! Position the tape, if the directory list is empty, we must rewind the tape so that we don't put
2 ! the files after the initial tape mark
2 !
3 IF $queue_empty (dosv [dos11$q_entry_header])
2 THEN
3     BEGIN
3     $trace_print_lit ('empty queue, rewinding');
3     IF NOT exch$io_dos11_rewind (.volb)
3     THEN
3         RETURN 0;
3     END
3
```

```
3 ! Move to the end of the tape
3 !
2 ELSE
3     BEGIN
3
3     ! Our end_of_tape is actually after the second tape mark, so backspace if we are at EOT
3     !
3     IF .dosv [dos11$v_end_of_tape]
3     THEN
4         BEGIN
4         dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1; ! Adjust the file number for the ski
4         dosv [dos11$v_end_of_tape] = false;              ! The skip file routine looks at this bit, we need t
4         temp = -1;                                       ! fact that we are at end-of-tape
4         END
4
4     ! Not at EOT, skip a very, very large number of files to find the end
4     !
3     ELSE
3         temp = 32767;
3
3     ! No do the actual skip
3     !
3     IF NOT exch$io_dos11_skip_file (.volb, .temp)
3     THEN
3         RETURN 0;
2     END;
2
2 ! Create an entry
2 !
2 ent = exch$util_vm_allocate_zeroed (dos11ent$k_length); ! Allocate the memory
2
2 ! Get the date into dos11 format
2 !
2 exch$dos11_form_cur_date (.ent);
2
2 ! Convert the file name to radix 50 and store in the entry
2 !
2 exch$util_radix50_from_ascii (MINU (.nam_len, 6),                  ! First 6 bytes
2                 .rfp [nam$l_name], 6, ent [dos11ent$l_filename_1]);
2 IF .nam_len GTR 6                                                  ! Last 3 bytes
2 THEN
2     exch$util_radix50_from_ascii (.nam_len-6,
2                 .rfp [nam$l_name] + 6, 3, ent [dos11ent$w_filename_2]);
2 exch$util_radix50_from_ascii (.typ_len-1,
2                 .rfp [nam$l_type]+1, dos11ctx$s_exp_type, ent [dos11ent$w_filetype]);
2
2 ! Move the standard protection code
2 !
2 ent [dos11ent$w_protection] = %O '233';
2
2 ! Set the uic to that of the current process, unless the /BY_OWNER qualifier is in effect in which case use
2 ! the uic of the input (Files-11 only) file.  !?? no /by_owner yet
2 !
2 ent [dos11ent$b_member] = MINU (.exch$a_gbl [excg$w_uic_member], 255);
2 ent [dos11ent$b_group]  = MINU (.exch$a_gbl [excg$w_uic_group], 255);
2
2 $trace_print_fao ('dosv current file !UL', .dosv [dos11$l_current_file]);
```

```
2 $trace_print_fao ('     beg_of_tape  !UL', .dosv [dos11$v_beg_of_tape]);
2 $trace_print_fao ('     tape_mark    !UL', .dosv [dos11$v_tape_mark]);
2 $trace_print_fao ('     end_of_tape  !UL', .dosv [dos11$v_end_of_tape]);
2
2 ! Store the file number in the entry
2 !
2 $trace_print_fao ('storing dosv file number !UL into entry', .dosv [dos11$l_current_file]);
2 ent [dos11ent$w_file_number] = .dosv [dos11$l_current_file];
2
2 ! Declare an exit handler to "close" the file if an exit occurs while it is open
2 !
2 $logic_check (1, (.exch$a_gbl [excg$a_exh_routine] EQL 0), 311);
2 $logic_check (2, (NOT (.out_ctx [dos11ctx$v_output_file] OR .out_ctx [dos11ctx$v_stream_active])), 312);
2 exch$a_gbl [excg$a_exh_routine]          = exch$dos11_exit_handler;                  ! Routine to write the tapem
2 exch$a_gbl [excg$l_exh_arg_count]        = 2;                                        ! Status and volb
2 exch$a_gbl [excg$a_exh_status]           = exch$a_gbl [excg$l_exh_condvalu];         ! Address to store status
2 exch$a_gbl [excg$a_exh_volb]             = .volb;                                    ! Pass address of volb
2
3 IF NOT (status = $dclexh (desblk=exch$a_gbl [excg$r_exit_block]))
2 THEN
2     $exch_signal_stop (.status);
2
2 ! Now write the label and put a copy of the label into the context block
2 !
3 IF NOT (status = exch$io_dos11_write_label (.volb, ent [dos11ent$l_filename_1]))
2 THEN
3     BEGIN
3
3     ! If the tape went into the end region
3     !
3     IF .status EQL ss$_endoftape
3     THEN
4         BEGIN
4
4         ! We did actually write the label, so backspace and put a tapemark on top of it and adjust the file
4         !
4         exch$io_dos11_skip_record (.volb, -1);
4         exch$io_dos11_write_tape_mark (.volb);
4 !??     dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] - 1;         ! Adjust the file number
4
4         ! Rewind the tape so that the exact position will be known again
4         !
4 !??     $exch_signal (exch$_dos11_position);    ! This might take a while, warn 'em
4 !??     dosv [dos11$v_error_rewind] = true;     ! Remember that we have had to forcibly reposition the tape
4         dosv [dos11$v_directory] = true;        ! Keep it from shooting off the end to find the directory
4 !??     exch$io_dos11_rewind (.volb);
3         END;
3
3     ! Cancel the exit handler which was declared to write the tape mark
3     !
3     $canexh (desblk=exch$a_gbl [excg$r_exit_block]);
3     exch$a_gbl [excg$a_exh_routine] = 0;        ! Clear address, means no exit handler is active
3
3     RETURN 0;
2     END;
2
2 ! Add the label entry to the end of the list, now that we have written the label
```

```
;  619        0711  2 !
;  620        0712  2 $queue_insert_tail (.ent, dosv [dos11$q_entry_header]); ! Add it to the tail
;  621        0713  2 dosv [dos11$a_current_entry] = .ent;                  ! Make it the current entry (but not the current fil
;  622        0714  2
;  623        0715  2 ! Copy the label entry to the context block
;  624        0716  2 !
;  625        0717  2 CH$MOVE (dos11ctx$s_entry, ent [dos11ent$l_filename_1], out_ctx [dos11ctx$t_entry]);
;  626        0718  2
;  627        0719  2 ! Define a record stream for this file
;  628        0720  2 !
;  629        0721  2 out_ctx [dos11ctx$l_cur_byte]   = 0;                                    ! Context is the first byte in
;  630        0722  2 out_ctx [dos11ctx$l_cur_block]  = 0;                                    !  the first block of the file
;  631        0723  2 out_ctx [dos11ctx$l_eof_block]  = -1;
;  632        0724  2 out_filb [filb$a_record]        = 0;                                    ! No valid record or length
;  633        0725  2 out_filb [filb$l_record_len]    = 0;
;  634        0726  2
;  635        0727  2 ! Expand the radix-50 filename into the standard ascii text fields
;  636        0728  2 !
;  637        0729  2 exch$dos11_expand_filename (.out_ctx);
;  638        0730  2
;  639        0731  2 ! Clear all the flags except the ones we want by writing the masks into the longword
;  640        0732  2 !
;  641        0733  2 out_ctx [dos11ctx$l_flags] =    dos11ctx$m_stream_active        ! A record stream is currently active
;  642        0734  2                              OR dos11ctx$m_output_file;         !  and it is an output file
;  643        0735  2
;  644        0736  2 ! Set up the i/o and record buffer
;  645        0737  2 !
;  646        0738  2 IF .out_ctx [dos11ctx$a_buffer] EQL 0
;  647        0739  2 THEN
;  648        0740  2     out_ctx [dos11ctx$a_buffer] = exch$util_vm_allocate (ctx$k_buffer_length);
;  649        0741  2
;  650        0742  2 ! Set the block pointers to the chunk we are ready to write (i.e. nothing, 'cuz we've done no puts)
;  651        0743  2 !
;  652        0744  2 out_ctx [dos11ctx$l_buf_base_block] = 0;
;  653        0745  2 out_ctx [dos11ctx$l_buf_high_block] = ctx$k_buffer_blocks - 1;
;  654        0746  2
;  655        0747  2 RETURN true;
;  656        0748  1 END;


                                                                       .EXTRN   EXCH$_BADLOGIC, EXCH$_ILLMTCOPY
                                                                       .EXTRN   EXCH$_NOALLOC, EXCH$_NOCARRIAGE
                                                                       .EXTRN   SYS$DCLEXH, LIB$STOP

                                                   0FFC 00000          .ENTRY   EXCH$DOS11_CREATE_FILE, Save R2,R3,R4,R5,-   ; 0369
                                                                                R6,R7,R8,R9,R10,R11                          ;
                             5E       FE78    CE   9E 00002          MOVAB    -392(SP), SP                                 ;
               50 00000000G  EF               04   C1 00007          ADDL3    #4, EXCH$A_GBL, R0                           ; 0418
                             5A               60   D0 0000F          MOVL     (R0), R10                                    ; 0419
                             59         14    AA   9E 00012          MOVAB    20(R10), R9                                  ;
                             57         3C    AA   D0 00016          MOVL     60(R10), R7                                  ; 0421
                     0C      AE         44    AA   9E 0001A          MOVAB    68(R10), 12(SP)                              ; 0424
                             56         0C    BE   D0 0001F          MOVL     @12(SP), R6                                  ; 0425
                     08      AE         20    A6   9E 00023          MOVAB    32(R6), 8(SP)                                ; 0426
                             5B         1C    A6   D0 00028          MOVL     28(R6), R11                                  ; 0428
                     04      AE         54    AB   9E 0002C          MOVAB    84(R11), 4(SP)                               ;
```

```
                                    52 035B00FA   8F   D0 00031         MOVL     #56295674, R2                            ; 0433
                                    51     0248   8F   3C 00038         MOVZWL   #584, R1                                 ;
                                    50            56   D0 0003D         MOVL     R6, R0                                   ;
                                       00000000G  EF   16 00040         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                    52 035B00FA   8F   D0 00046         MOVL     #56295674, R2                            ; 0434
                                    51     0249   8F   3C 0004D         MOVZWL   #585, R1                                 ;
                                    50            57   D0 00052         MOVL     R7, R0                                   ;
                                       00000000G  EF   16 00055         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                    52 010A00F7   8F   D0 0005B         MOVL     #17432823, R2                            ; 0435
                                    51     024A   8F   3C 00062         MOVZWL   #586, R1                                 ;
                                    50       18   A6   D0 00067         MOVL     24(R6), R0                               ;
                                       00000000G  EF   16 0006B         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                    6E       18   A7   D0 00071         MOVL     24(R7), (SP)                             ; 0436
                                    52 010A00F7   8F   D0 00075         MOVL     #17432823, R2                            ;
                                    51     024B   8F   3C 0007C         MOVZWL   #587, R1                                 ;
                                    50            6E   D0 00081         MOVL     (SP), R0                                 ;
                                       00000000G  EF   16 00084         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                    52 041B00F3   8F   D0 0008A         MOVL     #68878579, R2                            ; 0437
                                    51     024C   8F   3C 00091         MOVZWL   #588, R1                                 ;
                                    50            5B   D0 00096         MOVL     R11, R0                                  ;
                                       00000000G  EF   16 00099         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                             1C   A7   D5 0009F         TSTL     28(R7)                                   ; 0438
                                                  16   13 000A2         BEQL     1$                                       ;
                                    52 041B00F3   8F   D0 000A4         MOVL     #68878579, R2                            ;
                                    51     027A   8F   3C 000AB         MOVZWL   #634, R1                                 ;
                                    50       1C   A7   D0 000B0         MOVL     28(R7), R0                               ;
                                       00000000G  EF   16 000B4         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                                    58       04   BE   D0 000BA 1$:     MOVL     @4(SP), R8                               ; 0439
                                    52 003600FD   8F   D0 000BE         MOVL     #3539197, R2                             ;
                                    51     0253   8F   3C 000C5         MOVZWL   #595, R1                                 ;
                                    50            58   D0 000CA         MOVL     R8, R0                                   ;
                                       00000000G  EF   16 000CD         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                      13       48   AB            05   E0 000D3         BBS      #5, 72(R11), 2$                          ; 0444
                                    7E       FD   8F   9A 000D8         MOVZBL   #253, -(SP)                              ;
                                                  01   DD 000DC         PUSHL    #1                                       ;
                                       00000000G  8F   DD 000DE         PUSHL    #EXCH$_BADLOGIC                          ;
                        00000000G   00            03   FB 000E4         CALLS    #3, LIB$STOP                             ;
                                             08   BE   D5 000EB 2$:     TSTL     @8(SP)                                   ; 0448
                                                  11   12 000EE         BNEQ     3$                                       ;
                                                  56   DD 000F0         PUSHL    R6                                       ; 0450
                                                  5B   DD 000F2         PUSHL    R11                                      ;
                        00000000G   EF            02   FB 000F4         CALLS    #2, EXCH$UTIL_DOS11CTX_ALLOCATE          ;
                               08   BE            50   D0 000FB         MOVL     R0, @8(SP)                               ;
                                                  16   11 000FF         BRB      4$                                       ;
                                    52 008A00FC   8F   D0 00101 3$:     MOVL     #9044220, R2                             ; 0453
                                    51     024D   8F   3C 00108         MOVZWL   #589, R1                                 ;
                                    50       08   BE   D0 0010D         MOVL     @8(SP), R0                               ;
                                       00000000G  EF   16 00111         JSB      EXCH$UTIL_BLOCK_CHECK                    ;
                               14   AE       08   BE   D0 00117 4$:     MOVL     @8(SP), 20(SP)                           ; 0463
                      24   AE  14   AE            1C   C1 0011C         ADDL3    #28, 20(SP), 36(SP)                      ;
   006E      8F            00       6E            00   2C 00122         MOVC5    #0, (SP), #0, #110, @36(SP)              ;
                                             24   BE      00129                                                           ;
                                             28   AE   9F 0012B         PUSHAB   RFP                                      ; 0470
                                             5A   A7   9F 0012E         PUSHAB   90(R7)                                   ;
                                             3A   A7   DD 00131         PUSHL    58(R7)                                   ;
                                             04   A9   DD 00134         PUSHL    4(R9)                                    ;
                                    7E            69   3C 00137         MOVZWL   (R9), -(SP)                              ;
```

```
                   00000000G  EF           05  FB 0013A          CALLS   #5, EXCH$CMD_RELATED_FILE_PARSE          ;
                          24  AE           50  D0 00141          MOVL    R0, STATUS                               ;
                          0B         24    AE  E8 00145          BLBS    STATUS, 5$                               ;
             5A    A6     04  B9           69  28 00149          MOVC3   (R9), @4(R9), 90(R6)                     ; 0477
                              50     24    AE  D0 0014F          MOVL    STATUS, R0                               ; 0478
                                           04     00153          RET                                              ;
                          20  AE     2B    A6  9E 00154  5$:     MOVAB   43(R6), 32(SP)                           ; 0484
                          20  BE     80    8F  8A 00159          BICB2   #128, @32(SP)                            ;
                              52     63    AE  9A 0015E          MOVZBL  RFP+59, TOT_LEN                          ; 0485
                                     74    AE  DD 00162          PUSHL   RFP+76                                   ; 0486
                              7E     67    AE  9A 00165          MOVZBL  RFP+59, -(SP)                            ;
                   00000000G  EF           02  FB 00169          CALLS   #2, EXCH$PDP_FILTER_FILENAME             ;
                          63  AE           50  90 00170          MOVB    R0, RFP+59                               ;
                              50     63    AE  9A 00174          MOVZBL  RFP+59, R0                               ; 0487
                              09           50  91 00178          CMPB    R0, #9                                   ;
                                           03  1B 0017B          BLEQU   6$                                       ;
                              50           09  D0 0017D          MOVL    #9, R0                                   ;
                              59           50  D0 00180  6$:     MOVL    R0, NAM_LEN                              ;
                              59           52  D1 00183          CMPL    TOT_LEN, NAM_LEN                         ; 0488
                                           05  13 00186          BEQL    7$                                       ;
                          20  BE     80    8F  88 00188          BISB2   #128, @32(SP)                            ; 0490
                              52     64    AE  9A 0018D  7$:     MOVZBL  RFP+60, TOT_LEN                          ; 0492
                    7E    78  AE           01  C1 00191          ADDL3   #1, RFP+80, -(SP)                        ; 0493
                              7E     68    AE  9A 00196          MOVZBL  RFP+60, -(SP)                            ;
                                           6E  D7 0019A          DECL    (SP)                                     ;
                   00000000G  EF           02  FB 0019C          CALLS   #2, EXCH$PDP_FILTER_FILENAME             ;
             64    AE         50           01  81 001A3          ADDB3   #1, R0, RFP+60                           ;
                              50     64    AE  9A 001A8          MOVZBL  RFP+60, R0                               ; 0494
                              04           50  91 001AC          CMPB    R0, #4                                   ;
                                           03  1B 001AF          BLEQU   8$                                       ;
                              50           04  D0 001B1          MOVL    #4, R0                                   ;
                          1C  AE           50  D0 001B4  8$:     MOVL    R0, TYP_LEN                              ;
                          1C  AE           52  D1 001B8          CMPL    TOT_LEN, TYP_LEN                         ; 0495
                                           05  13 001BC          BEQL    9$                                       ;
                          20  BE     80    8F  88 001BE          BISB2   #128, @32(SP)                            ; 0497
                    52        59     1C    AE  C1 001C3  9$:     ADDL3   TYP_LEN, NAM_LEN, TOT_LEN                ; 0499
                          3A  A6     65  BB42  9E 001C8          MOVAB   @10T(R11)[TOT_LEN], 58(R6)               ; 0500
                   00000100   8F     3A    A6  D1 001CE          CMPL    58(R6), #256                             ; 0501
                                           14  1B 001D6          BLEQU   10$                                      ;
                              7E   0100    8F  3C 001D8          MOVZWL  #256, -(SP)                              ;
                                           01  DD 001DD          PUSHL   #1                                       ;
                                00000000G  8F  DD 001DF          PUSHL   #EXCH$_BADLOGIC                          ;
                   00000000G  00           03  FB 001E5          CALLS   #3, LIB$STOP                             ;
                          18  AE   0100    8F  3C 001EC  10$:    MOVZWL  #256, 24(SP)                             ; 0502
                          10  AE     5A    A6  9E 001F2          MOVAB   90(R6), 16(SP)                           ; 0504
  18    AE          00    69  AB     65    AB  2C 001F7          MOVC5   101(R11), 105(R11), #0, 24(SP), @16(SP)  ;
                                     10    BE     001FF                                                           ;
                                           27  18 00201          BGEQ    11$                                      ;
                          10  AE     65    AB  C0 00203          ADDL2   101(R11), 16(SP)                         ;
                          18  AE     65    AB  C2 00208          SUBL2   101(R11), 24(SP)                         ;
  18    AE          00    74  BE           59  2C 0020D          MOVC5   NAM_LEN, @RFP+76, #0, 24(SP), @16(SP)    ;
                                     10    BE     00214                                                           ;
                                           12  18 00216          BGEQ    11$                                      ;
                          10  AE           59  C0 00218          ADDL2   NAM_LEN, 16(SP)                          ;
                          18  AE           59  C2 0021C          SUBL2   NAM_LEN, 24(SP)                          ;
  18    AE          00    78  BE     1C    AE  2C 00220          MOVC5   TYP_LEN, @RFP+80, #0, 24(SP), @16(SP)    ;
                                     10    BE     00228                                                           ;
```

```
                      1C   A7           5B  D1 0022A 11$:     CMPL     R11, 28(R7)                                  ; 0510
                                        08  12 0022E          BNEQ     12$                                          ;
                           50 00000000G 8F  D0 00230          MOVL     #EXCH$_ILLMTCOPY, R0                         ; 0512
                                            04 00237          RET                                                   ;
                18    AE       14   AE  1C  C1 00238 12$:     ADDL3    #28, 20(SP), 24(SP)                          ; 0517
  006E    8F          00            6E  00  2C 0023E          MOVC5    #0, (SP), #0, #110, @24(SP)                  ;
                                    18  BE     00245                                                            ;
                                    2D  A7  D5 00247          TSTL     45(R7)                                       ; 0521
                                        05  12 0024A          BNEQ     13$                                          ;
                                    24  AA  D5 0024C          TSTL     36(R10)                                      ; 0523
                                        ^D  13 0024F          BEQL     14$                                          ;
                              00000000G 8F  DD 00251 13$:     PUSHL    #EXCH$_NOALLOC                               ; 0525
                00000000G  00           01  FB 00257          CALLS    #1, LIB$SIGNAL                               ;
                54    78   AE           01  C1 0025E 14$:     ADDL3    #1, RFP+80, R4                               ; 0529
                                        54  DD 00263          PUSHL    R4                                           ;
                                        56  DD 00265          PUSHL    R6                                           ;
                00000000G  EF           02  FB 00267          CALLS    #2, EXCH$CMD_FETCH_RECFMT_IMPLIED            ;
                      4A   A6     FC93  CF  9E 0026E          MOVAB    EXCH$DOS11_CLOSE_FILE, 74(R6)                ; 0533
                      4E   A6     FC8D  CF  9E 00274          MOVAB    EXCH$DOS11_CLOSE_FILE, 78(R6)                ; 0534
                                    52  A6  D4 0027A          CLRL     82(R6)                                       ; 0535
                      56   A6 00000000G EF  9E 0027D          MOVAB    EXCH$PDP_PUT, 86(R6)                         ; 0536
                05    2B   A7           01  E0 00285          BBS      #1, 43(R7), 15$                              ; 0540
                0D    20   BE           01  E1 0028A          BBC      #1, @32(SP), 16$                             ; 0542
                              00J00000G 8F  DD 0028F 15$:     PUSHL    #EXCH$_NOCARRIAGE                            ; 0544
                00000000G  00           01  FB J0295          CALLS    #1, LIB$SIGNAL                               ;
                           01       29  A6  91 0029C 16$:     CMPB     41(R6), #1                                   ; 0548
                                        06  13 002A0          BEQL     17$                                          ;
                           01       29  A7  91 002A2          CMPB     41(R7), #1                                   ; 0550
                                        0A  12 002A6          BNEQ     18$                                          ;
                      28   A6           02  90 002A8 17$:     MOVB     #2, 40(R6)                                   ; 0553
                      35   A6     0200  8F  3C 002AC          MOVZWL   #512, 53(R6)                                 ; 0554
                                    29  A7  95 002B2 18$:     TSTB     41'R7)                                       ; 0559
                                        3B  12 002B5          BNEQ     20$                                          ;
                                    29  A6  95 002B7          TSTB     41(R6)                                       ; 0560
                                        36  12 002BA          BNEQ     20$                                          ;
                           32       2B  A7  E8 002BC          BLBS     43(R7), 20$                                  ; 0562
                           2E       20  BE  E8 002C0          BLBS     @32(SP), 20$                                 ; 0563
                50         6E 0000007A  8F  C1 002C4          ADDL3    #122, (SP), R0                               ; 0565
                           01           60  91 002CC          CMPB     (R0), #1                                     ;
                                        0D  13 002CF          BEQL     19$                                          ;
                50         6E 0000007A  8F  C1 002D1          ADDL3    #122, (SP), R0                               ; 0566
                           03           60  91 002D9          CMPB     (R0), #3                                     ;
                                        14  12 002DC          BNEQ     20$                                          ;
                      28   A7           02  90 002DE 19$:     MOVB     #2, 40(R7)                                   ; 0570
                      35   A7     0200  8F  3C 002E2          MOVZWL   #512, 53(R7)                                 ; 0571
                      28   A6           02  90 002E8          MOVB     #2, 40(R6)                                   ; 0572
                      35   A6     0200  8F  3C 002EC          MOVZWL   #512, 53(R6)                                 ; 0573
                                        7E  D4 002F2 20$:     CLRL     -(SP)                                        ; 0578
                                        5B  DD 002F4          PUSHL    R11                                          ;
                00000000G  EF           02  FB 002F6          CALLS    #2, EXCH$UTIL_DOS11CTX_ALLOCATE              ;
                           52           50  D0 002FD          MOVL     R0, ENT                                      ;
                           53           03  D0 00300          MOVL     #3, TEMP                                     ; 0579
                                        0C  BB 00303 21$:     PUSHR    #^M<R2,R3>                                   ; 0580
                     0000V CF           02  FB 00305          CALLS    #2, EXCH$DOS11_NEXT_ENTRY                    ;
                                        50  D5 0030A          TSTL     R0                                           ;
                                        05  13 0030C          BEQL     22$                                          ;
                           53           02  D0 0030E          MOVL     #2, TEMP                                     ; 0582
```

```
                                 F0  11 00311        BRB     21$                                   ;
                                 52  DD 00313 22$:   PUSHL   ENT                                   ; 0583
      00000000G  EF              01  FB 00315        CALLS   #1, EXCH$UTIL_DOS11CTX_RELEASE        ;
                 50       12     A8  9E 0031C        MOVAB   18(R8), R0                            ; 0588
             12  A8              50  D1 00320        CMPL    R0, 18(R8)                            ;
                                 0B  12 00324        BNEQ    23$                                   ;
                                 5B  DD 00326        PUSHL   R11                                   ; 0592
      00000000G  EF              01  FB 00328        CALLS   #1, EXCH$IO_DOS11_REWIND              ;
                                 21  11 0032F        BRB     26$                                   ;
   0C        0C  A8              03  E1 00331 23$:   BBC     #3, 12(R8), 24$                       ; 0604
                          0E     A8  D6 00336        INCL    14(R8)                                ; 0607
             0C  A8              08  8A 00339        BICB2   #8, 12(R8)                            ; 0608
                 53              01  CE 0033D        MNEGL   #1, TEMP                              ; 0609
                                 05  11 00340        BRB     25$                                   ; 0604
                 53     7FFF     8F  3C 00342 24$:   MOVZWL  #32767, TEMP                          ; 0615
                                 53  DD 00347 25$:   PUSHL   TEMP                                  ; 0619
                                 5B  DD 00349        PUSHL   R11                                   ;
      00000000G  EF              02  FB 0034B        CALLS   #2, EXCH$IO_DOS11_SKIP_FILE           ;
                 03              50  E8 00352 26$:   BLBS    R0, 27$                               ;
                               01AD  31 00355        BRW     39$                                   ;
                                 1C  DD 00358 27$:   PUSHL   #28                                   ; 0626
      00000000G  EF              01  FB 0035A        CALLS   #1, EXCH$UTIL_VM_ALLOCATE_ZEROED      ;
                 52              50  D0 00361        MOVL    R0, ENT                               ;
                 51              52  D0 00364        MOVL    ENT, R1                               ; 0630
                              0000V  30 00367        BSBW    EXCH$DOS11_FORM_CUR_DATE              ;
                          08     A2  9F 0036A        PUSHAB  8(ENT)                                ; 0635
                                 06  DD 0036D        PUSHL   #6                                    ;
                          7C     AE  DD 0036F        PUSHL   RFP+76                                ;
                                 59  DD 00372        PUSHL   NAM_LEN                               ; 0634
                 06              6E  D1 00374        CMPL    (SP), #6                              ;
                                 03  1B 00377        BLEQU   28$                                   ;
                 6E              06  D0 00379        MOVL    #6, (SP)                              ;
      00000000G  EF              04  FB 0037C 28$:   CALLS   #4, EXCH$UTIL_RADIX50_FROM_ASCII      ; 0635
                 06              59  D1 00383        CMPL    NAM_LEN, #6                           ; 0636
                                 14  15 00386        BLEQ    29$                                   ;
                          14     A2  9F 00388        PUSHAB  20(ENT)                               ; 0639
                                 03  DD 0038B        PUSHL   #3                                    ;
   7E        7C  AE              06  C1 0038D        ADDL3   #6, RFP+76, -(SP)                     ;
                          FA     A9  9F 00392        PUSHAB  -6(NAM_LEN)                           ; 0638
      00000000G  EF              04  FB 00395        CALLS   #4, EXCH$UTIL_RADIX50_FROM_ASCII      ; 0639
                          0C     A2  9F 0039C 29$:   PUSHAB  12(ENT)                               ; 0641
                                 03  DD 0039F        PUSHL   #3                                    ;
                                 54  DD 003A1        PUSHL   R4                                    ;
   50        28  AE              01  C3 003A3        SUBL3   #1, TYP_LEN, R0                       ; 0640
                                 50  DD 003A8        PUSHL   R0                                    ;
      00000000G  EF              04  FB 003AA        CALLS   #4, EXCH$UTIL_RADIX50_FROM_ASCII      ; 0641
             10  A2       9B     8F  9B 003B1        MOVZBW  #155, 16(ENT)                         ; 0645
                 50 00000000G    EF  D0 003B6        MOVL    EXCH$A_GBL, R0                        ; 0650
                 51       1C     A0  3C 003BD        MOVZWL  28(R0), R1                            ;
           00FF  8F              51  B1 003C1        CMPW    R1, #255                              ;
                                 04  1B 003C6        BLEQU   30$                                   ;
                 51       FF     8F  9A 003C8        MOVZBL  #255, R1                              ;
             0E  A2              51  90 003CC 30$:   MOVB    R1, 14(ENT)                           ;
                 51       1E     A0  3C 003D0        MOVZWL  30(R0), R1                            ; 0651
           00FF  8F              51  B1 003D4        CMPW    R1, #255                              ;
                                 04  1B 003D9        BLEQU   31$                                   ;
                 51       FF     8F  9A 003DB        MOVZBL  #255, R1                              ;
```

```
               0F    A2                51   90 003DF 31$:    MOVB     R1, 15(ENT)                                   ;
               18    A2          0E    A8   B0 003E3          MOVW     14(R8), 24(ENT)                               ; 0661
                                 30    A0   D5 003E8          TSTL     48(R0)                                        ; 0665
                                 14         13 003EB          BEQL     32$                                           ;
                     7E        0137    8F   3C 003ED          MOVZWL   #311, -(SP)                                   ;
                                 01         DD 003F2          PUSHL    #1                                            ;
                          00000000G    8F   DD 003F4          PUSHL    #EXCH$_BADLOGIC                               ;
         00000000G   00                03   FB 003FA          CALLS    #3, LIB$STOP                                  ;
       50       14   AE                28   C1 00401 32$:     ADDL3    #40, 20(SP), R0                               ; 0666
       08            60                01   E0 00406          BBS      #1, (R0), 33$                                 ;
       51       14   AE                28   C1 0040A          ADDL3    #40, 20(SP), R1                               ;
                     14                61   E9 0040F          BLBC     (R1), 34$                                     ;
                     7E        0138    8F   3C 00412 33$:     MOVZWL   #312, -(SP)                                   ;
                                 01         DD 00417          PUSHL    #1                                            ;
                          00000000G    8F   DD 00419          PUSHL    #EXCH$_BADLOGIC                               ;
         00000000G   00                03   FB 0041F          CALLS    #3, LIB$STOP                                  ;
                     50   00000000G    EF   D0 00426 34$:     MOVL     EXCH$A_GBL, R0                                ; 0667
               30    A0       0000V    CF   9E 0042D          MOVAB    EXCH$DOS11_EXIT_HANDLER, 48(R0)               ;
               34    A0                02   D0 00433          MOVL     #2, 52(R0)                                    ; 0668
               38    A0          40    A0   9E 00437          MOVAB    64(R0), 56(R0)                                ; 0669
               3C    A0                5B   D0 0043C          MOVL     R11, 60(R0)                                   ; 0670
                                 2C    A0   9F 00440          PUSHAB   44(R0)                                        ; 0672
         00000000G   00                01   FB 00443          CALLS    #1, SYS$DCLEXH                                ;
               24    AE                50   D0 0044A          MOVL     R0, STATUS                                    ;
                     0B          24    AE   E8 0044E          BLBS     STATUS, 35$                                   ;
                                 24    AE   DD 00452          PUSHL    STATUS                                        ; 0674
         00000000G   00                01   FB 00455          CALLS    #1, LIB$STOP                                  ;
                                            04 0045C          RET                                                    ;
                                 08    A2   9F 0045D 35$:     PUSHAB   8(ENT)                                        ; 0678
                                       5B   DD 00460          PUSHL    R11                                           ;
         00000000G   EF                02   FB 00462          CALLS    #2, EXCH$IO_DOS11_WRITE_LABEL                 ;
               24    AE                50   D0 00469          MOVL     R0, STATUS                                    ;
                     3E          24    AE   E8 0046D          BLBS     STATUS, 37$                                   ;
         00000878    8F          24    AE   D1 00471          CMPL     STATUS, #2168                                 ; 0684
                                 19         12 00479          BNEQ     36$                                           ;
                     7E                01   CE 0047B          MNEGL    #1, -(SP)                                     ; 0690
                                       5B   DD 0047E          PUSHL    R11                                           ;
         00000000G   EF                02   FB 00480          CALLS    #2, EXCH$IO_DOS11_SKIP_RECORD                 ;
                                       5B   DD 00487          PUSHL    R11                                           ; 0691
         00000000G   EF                01   FB 00489          CALLS    #1, EXCH$IO_DOS11_WRITE_TAPE_MARK             ;
               0C    A8                10   88 00490          BISB2    #16, 12(R8)                                   ; 0698
       7E 00000000G  EF                2C   C1 00494 36$:     ADDL3    #44, EXCH$A_GBL, -(SP)                        ; 0704
         00000000G   00                01   FB 0049C          CALLS    #1, SYS$CANEXH                                ;
                     50   00000000G    EF   D0 004A3          MOVL     EXCH$A_GBL, R0                                ; 0705
                                 30    A0   D4 004AA          CLRL     48(R0)                                        ;
                     56                11      004AD          BRB      39$                                           ; 0707
                     50          12    A8   9E 004AF 37$:     MOVAB    18(R8), R0                                    ; 0712
               04    B0                62   0E 004B3          INSQUE   (ENT), @4(R0)                                 ;
                     50          04    BE   D0 004B7          MOVL     @4(SP), R0                                    ; 0713
               1A    A0                52   D0 004BB          MOVL     ENT, 26(R0)                                   ;
                     57          08    BE   D0 004BF          MOVL     @8(SP), R7                                    ; 0717
   3C     A7         08    A2          14   28 004C3          MOVC3    #20, 8(ENT), 60(R7)                           ;
                                 24    A7   D4 004C9          CLRL     36(R7)                                        ; 0721
                                 1C    A7   D4 004CC          CLRL     28(R7)                                        ; 0722
               20    A7                01   CE 004CF          MNEGL    #1, 32(R7)                                    ; 0723
                     50          0C    BE   D0 004D3          MOVL     @12(SP), R0                                   ; 0724
                                 42    A0   7C 004D7          CLRQ     66(R0)                                        ; 0725
```

```
                                          57  DD 004DA          PUSHL   R7                                   ; 0729
                     0000V   CF           01  FB 004DC          CALLS   #1, EXCH$DOS11_EXPAND_FILENAME       ;
                        28   A7           03  D0 004E1          MOVL    #3, 40(R7)                           ; 0734
                                    18    A7  D5 004E5          TSTL    24(R7)                               ; 0738
                                          10  12 004E8          BNEQ    38$                                  ;
                             7E    1800   8F  3C 004EA          MOVZWL  #6144, -(SP)                         ; 0740
                 00000000G   EF           01  FB 004EF          CALLS   #1, EXCH$UTIL_VM_ALLOCATE            ;
                        18   A7           50  D0 004F6          MOVL    R0, 24(R7)                           ;
                                    2C    A7  D4 004FA 38$:     CLRL    44(R7)                               ; 0744
                        30   A7           0B  D0 004FD          MOVL    #11, 48(R7)                          ; 0745
                             50           01  D0 00501          MOVL    #1, R0                               ; 0747
                                              04 00504          RET                                          ;
                                          50  D4 00505 39$:     CLRL    R0                                   ; 0748
                                              04 00507          RET                                          ;

; Routine Size:  1288 bytes,    Routine Base:  EXCH$DOS11_CODE + 00FB
```

```
GLOBAL ROUTINE exch$dos11_directory (volb : $ref_bblock) =      %SBTTL 'exch$dos11_directory'
BEGIN
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       Traverse the DOS-11 directory and display the contents
!
! INPUTS:
!
!       volb - pointer to volb which has been connected to the DOS-11 device
!
! IMPLICIT INPUTS:
!
!       dire - directory work area
!
! OUTPUTS:
!
!       none
!
! IMPLICIT OUTPUTS:
!
!       none
!
! ROUTINE VALUE:
!
!       true if valid, false if not
!
! SIDE EFFECTS:
!
!       error conditions will be signaled
!--

$dbgtrc_prefix ('dos11_directory> ');

LOCAL
    nam : $ref_bblock,                          ! a pointer to the namb block
    ctx : $ref_bblock,                          ! a pointer to an DOS-11 context block
    wid,
    blocks_displayed,
    files_displayed,
    faop : $ref_bblock,                         ! pointer to an fao output
    status,
    flags
    ;

BIND
    dire = exch$a_gbl [excg$a_dire_work]        : $ref_bblock,  ! pointer to work area
    fab  = dire [dire$a_outfab]         : $ref_bblock,  ! pointer to fab for output file
    rab  = dire [dire$a_outrab]         : $ref_bblock,  ! pointer to rab for output file
    dosv = volb [volb$a_vfmt_specific]  : $ref_bblock   ! pointer to dos11 structure
    ;
```

```
  2 $block_check (2, .dire, dire, 617);            ! Check that the directory work area is valid
  2 $block_check (2, .volb, volb, 618);
  2 $block_check (2, .dosv, dos11, 622);
  2
  2 $trace_print_lit ('calling dump_dosv');
  2 $check_call (4, exch$dbg_dump_dosv, .volb);
L 2 %IF switch_check                                ! If checking is on look at all the nambs
  2 %THEN
  2     nam = .volb [volb$a_namb_head];             ! Get the address of the first namb chained to the volb
  2     $logic_check (2, (.nam NEQ 0), 264);
  2     WHILE .nam NEQ 0
  2     DO
  3         BEGIN
  3         $block_check (2, .nam, namb, 597);
  3         nam = .nam [namb$a_next];
  2         END;
  2 %FI
  2
  2 ! Determine the width of a directory item
  2 !
  2 wid = dos11ctx$s_exp_fullname;                                   ! Name and pointer are always there
  2
  2 IF .dire [dire$v_q_owner]                                        ! Is /OWNER present?
  2 THEN
  2     wid = .wid + dos11ctx$s_exp_directory;
  2
  2 IF .dire [dire$v_q_size]                                         ! Is /SIZE present?
  2 THEN
  2     wid = .wid + (IF .dire [dire$v_q_octal] THEN 7 ELSE 5);
  2
  2 IF .dire [dire$v_q_date]                                         !  or /DATE?
  2 THEN
  2     wid = .wid + dos11ctx$s_exp_date + 1;
  2
  2 IF .dire [dire$v_q_full]                                         !  or /FULL? (add protection)
  2 THEN
  2     wid - .wid + 6;                                              ! Protection is always in octal
  2
  2 ! Compute the number of columns for the listing
  2 !
  2 dire [dire$l_item_width] = .wid;                                 ! Put the value in the work area
  2 dire [dire$l_list_width] = .dire [dire$l_list_width] -1;         ! Make a place for the '>' against left colu
  2 exch$dire_get_columns ();                                        ! Fetch the number of columns
  2 dire [dire$l_list_width] = .dire [dire$l_list_width] +1;         ! Restore the original listing width
  2
  2 ! Print the header information
  2 !
  2 exch$dire_put (null_string);                                             ! Print a blank line
  2 faop = exch$util_fao_buffer (%ASCID 'Directory of DOS-11 volume !AF      !%D',  ! Format the header line
  2                 .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], 0);
  2 faop [dsc$w_length] = .faop [dsc$w_length] - 6;                          ! Remove the seconds field f
  2 exch$dire_put (.faop);                                                   ! Print the header line
  2 exch$dire_put (null_string);                                             !  and another blank line
  2
  2 ! Initialize the local variables
  2 !
  2 status = true;                                  ! Assume that we will find some files
```

```
;  768        0858  2 blocks_displayed = 0;
;  769        0859  2 files_displayed = 0;
;  770        0860  2
;  771        0861  2 ! Purchase an DOS11 context block
;  772        0862  2 !
;  773        0863  2 ctx = exch$util_dos11ctx_allocate (.volb, 0);
;  774        0864  2
;  775        0865  2 ! Loop through all the labels in the directory
;  776        0866  2 !
;  777        0867  2 dosv [dos11$v_current_marked] = false;
;  778        0868  2 flags = dosnxt$m_rewind OR dosnxt$m_count_blocks;        ! Rewind for the first file
;  779        0869  2 WHILE exch$dos11_next_entry (.ctx, .flags) NEQ 0
;  780        0870  2 DO
;  781        0871  3     BEGIN
;  782        0872  3
;  783        0873  3     ! If we have aborted, exit the loop
;  784        0874  3     !
;  785        0875  3     IF .exch$a_gbl [excg$v_control_c]
;  786        0876  3     THEN
;  787        0877  3         EXITLOOP;
;  788        0878  3
;  789        0879  3     flags = dosnxt$m_count_blocks;                  ! Just count for later files
;  790        0880  3     nam = .volb [volb$a_namb_head];                 ! Get first namb
;  791        0881  3     WHILE .nam NEQ 0                                !  and loop through them all
;  792        0882  3     DO
;  793        0883  4         BEGIN
;  794        0884  4         BIND                                                ! Need the name and type from the
;  795        0885  4             nam_nam = nam [namb$q_name] : $desc_block,      !  command input because we need
;  796        0886  4             nam_typ = nam [namb$q_type] : $desc_block;      !  to compare this name against
;  797        0887  4                                                             !  any selections in the command
;  798        0888  4         IF exch$cmd_match_filename
;  799        0889  4                 (.ctx [dos11ctx$l_exp_fullname_len], ctx [dos11ctx$t_exp_fullname],
;  800        0890  4                 .nam_nam [dsc$w_length] + .nam_typ [dsc$w_length], .nam_nam [dsc$a_pointer])
;  801        0891  4           AND
;  802        0892  4             dos11_match_uic (.ctx, .nam)
;  803        0893  4         THEN
;  804        0894  5             BEGIN
;  805        0895  5             exch$dos11_dire_put_item (.ctx);
;  806        0896  5             files_displayed = .files_displayed + 1;                     ! update trailer statistics
;  807        0897  5             blocks_displayed = .blocks_displayed + .ctx [dos11ctx$w_blocks];
;  808        0898  5             EXITLOOP;
;  809        0899  4             END;
;  810        0900  4         nam = .nam [namb$a_next];                                       ! to next namb in chain
;  811        0901  3         END;
;  812        0902  2     END;
;  813        0903  2
;  814        0904  2 ! If we have canceled the listing, tell about it
;  815        0905  2 !
;  816        0906  2 IF .exch$a_gbl [excg$v_control_c]
;  817        0907  2 THEN
;  818        0908  2     $exch_signal ($info_stat_copy (exch$_canceled))
;  819        0909  2
;  820        0910  2 ELSE
;  821        0911  3     BEGIN
;  822        0912  3
;  823        0913  3     ! If the tape is at eof and we have printed some files, print the eot marker
;  824        0914  3     !
;  825          0915 4      IF  (NOT .dosv [dos11$v_current_marked])
;  826          0916 3        AND
;  827          0917 4          (.dire [dire$v_items_printed])
;  828          0918 3        AND
;  829          0919 4          (NOT .dosv [dos11$v_error_rewind])
;  830          0920 3      THEN
;  831          0921 3         exch$dos11_dire_put_item (1);            ! 1 means print a ''-> (end-of-tape)''
;  832          0922 3
;  833          0923 3      IF .dire [dire$l_cur_column] NEQ 0          ! If anything is waiting, print it
;  834          0924 3      THEN
;  835          0925 3         exch$dos11_dire_put_item (0);            ! 0 means flush the buffer
;  836          0926 3
;  837          0927 3      ! If we printed some items, then add an extra blank line
;  838          0928 3      !
;  839          0929 3      IF .dire [dire$v_items_printed]
;  840          0930 3      THEN
;  841          0931 3         exch$dire_put (null_string, true)                ! Cancel CTRL/O with second parm
;  842          0932 3
;  843          0933 3      ! Nothing was printed, return a warning status so that command procedures can see whether files were fou
;  844          0934 3      !
;  845          0935 3      ELSE
;  846          0936 3         status = $info_stat_copy (rms$_fnf);    ! Let a guy see that $ EXC DIR CS1:DEFBOO.CMD didn't work
;  847          0937 3
;  848          0938 3      ! Print the statistics line
;  849          0939 3      !
;  850          0940 4      faop = (IF .dire [dire$v_q_octal]
;  851          0941 4              THEN
;  852          0942 4                  %ASCID 'Total of !OW file!%S, !OW block!%S.  (octal)'
;  853          0943 4              ELSE
;  854          0944 3                  %ASCID 'Total of !UL file!%S, !UL block!%S.      (''>'' marks current tape position)');
;  855          0945 3      faop = exch$util_fao_buffer (.faop, .files_displayed, .blocks_displayed);
;  856          0946 3      exch$dire_put (.faop);
;  857          0947 2      END;
;  858          0948 2
;  859          0949 2  ! Free up the context block.
;  860          0950 2  !
;  861          0951 2  exch$util_dos11ctx_release (.ctx);
;  862          0952 2
;  863          0953 2  RETURN .status;
;  864          0954 1  END;



                                                                       .PSECT  EXCH$DOS11_PLIT,NOWRT,2

4F  44  20  66  6F  20  79  72  6F  74  63  65  72  69  44  00054 P.AAD:  .ASCII  \Directory of DOS-11 volume !AF      !%D-   ;
46  41  21  20  65  6D  75  6C  6F  76  20  31  31  2D  53  00063                  \<0>                                      ;
                    00  44  25  21  20  20  20  20  20  20  00072                                                            ;
                                            010E0027  0007C P.AAC:  .LONG   17694759                                  ;
                                            00000000' 00080         .ADDRESS P.AAD                                    ;
69  66  20  57  4F  21  20  66  6F  20  6C  61  74  6F  54  00084 P.AAF:  .ASCII  \total of !OW file!%S, !OW block!%S.  (oc\  ;
63  6F  6C  62  20  57  4F  21  20  2C  53  25  21  65  6C  00093                                                            ;
                    63  6F  28  20  20  2E  53  25  21  6B  000A2                                                            ;
                                            29  6C  61  74  000AC         .ASCII  \tal)\                                    ;
                                            010E002C  000B0 P.AAE:  .LONG   17694764                                  ;
                                            00000000' 000B4         .ADDRESS P.AAF                                    ;
69  66  20  4C  55  21  20  66  6F  20  6C  61  74  6F  54  000B8 P.AAH:  .ASCII  \Total of !UL file!%S, !UL block!%S.     \  ;
```

```
63  6F  6C  62  20  4C  55  21  20  2C  53  25  21  65  6C  000C7                                                          ;
                    20  20  20  20  20  2E  53  25  21  6B  000D6                                                          ;
72  75  63  20  73  6B  72  61  6D  20  22  3E  22  28  20  000E0          .ASCII  \ ('">" marks current tape position)\<0>  ;
74  69  73  6F  70  20  65  70  61  74  20  74  6E  65  72  000EF                                                          ;
                                        00  29  6E  6F  69  000FE                                                          ;
                                                        00  00103          .ASCII  <0>                                     ;
                                                  010E004A  00104 P.AAG:   .LONG   17694794                                ;
                                                 00000000'  00108          .ADDRESS P.AAH                                  ;

                                                                           .EXT'   EXCH$_CANCELED

                                                                           .PSECT  EXCH$DOS11_CODE,NOWRT,2

                                                      0FFC  00000          .ENTRY  EXCH$DOS11_DIRECTORY, Save R2,R3,R4,R5,R6,- ; 0749
                                                                                   R7,R8,R9,R10,R11                        ;
                                  5E             0C     C2  00002          SUBL2   #12, SP                                 ;
                     50 00000000G EF             0C     C1  00005          ADDL3   #12, EXCH$A_GBL, R0                     ; 0796
                                  53             60     D0  0000D          MOVL    (R0), R3                                ; 0797
                                  55       04    AC     D0  00010          MOVL    VOLB, R5                                ; 0799
                                  52 024400FE    8F     D0  00014          MOVL    #38011134, R2                           ; 0801
                                  51     0269    8F     3C  0001B          MOVZWL  #617, R1                                ;
                                  50             53     D0  00020          MOVL    R3, R0                                  ;
                                    00000000G    EF     16  00023          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                                  52 041B00F3    8F     D0  00029          MOVL    #68878579, R2                           ; 0802
                                  51     026A    8F     3C  00030          MOVZWL  #618, R1                                ;
                                  50             55     D0  00035          MOVL    R5, R0                                  ;
                                    00000000G    EF     16  00038          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                                  56       54    A5     D0  0003E          MOVL    84(R5), R6                              ; 0803
                                  52 003600FD    8F     D0  00042          MOVL    #3539197, R2                            ;
                                  51     026E    8F     3C  00049          MOVZWL  #622, R1                                ;
                                  50             56     D0  0004E          MOVL    R6, R0                                  ;
                                    00000000G    EF     16  00051          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                                  54       4C    A5     D0  00057          MOVL    76(R5), NAM                             ; 0809
                                                 14     12  0005B          BNEQ    1$                                      ; 0810
                                  7E     0108    8F     3C  0005D          MOVZWL  #264, -(SP)                             ;
                                                 01     DD  00062          PUSHL   #1                                      ;
                                    00000000G    8F     DD  00064          PUSHL   #EXCH$_BADLOGIC                         ;
                        00000000G 00             03     FB  0006A          CALLS   #3, LIB$STOP                            ;
                                                 54     D5  00071 1$:      TSTL    NAM                                     ; 0811
                                                 1B     13  00073          BEQL    2$                                      ;
                                  52 010A00F7    8F     D0  00075          MOVL    #17432823, R2                           ; 0814
                                  51     0255    8F     3C  0007C          MOVZWL  #597, R1                                ;
                                  50             54     D0  00081          MOVL    NAM, R0                                 ;
                                    00000000G    EF     16  00084          JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                                  54       70    A4     D0  0008A          MOVL    112(NAM), NAM                           ; 0815
                                                 E1     11  0008E          BRB     1$                                      ; 0811
                                  51             0D     D0  00090 2$:      MOVL    #13, WID                                ; 0821
                                  59       2C    A3     9E  00093          MOVAB   44(R3), R9                              ; 0823
                       03         69             09     E1  00097          BBC     #9, (R9), 3$                            ;
                                  51             0A     C0  0009B          ADDL2   #10, WID                                ; 0825
                       0F         69             0B     E1  0009E 3$:      BBC     #11, (R9), 6$                           ; 0827
                                                 69     95  000A2          TSTB    (R9)                                    ; 0829
                                                 05     18  000A4          BGEQ    4$                                      ;
                                  50             07     D0  000A6          MOVL    #7, R0                                  ;
                                                 03     11  000A9          BRB     5$                                      ;
                                  50             05     D0  000AB 4$:      MOVL    #5, R0                                  ;
                                  51             50     C0  000AE 5$:      ADDL2   R0, WID                                 ;
```

```
            03         69                03  E1 000B1 6$:      BBC     #3, (R9), 7$                                   ; 0831
                       51                0C  C0 000B5          ADDL2   #12, WID                                       ; 0833
            03         69                06  E1 000B8 7$:      BBC     #6, (R9), 8$                                   ; 0835
                       51                06  C0 000BC          ADDL2   #6, WID                                        ; 0837
               34      A3                51  D0 000BF 8$:      MOVL    WID, 52(R3)                                    ; 0841
                                   38    A3  D7 000C3          DECL    56(R3)                                         ; 0842
         00000000G     EF                00  FB 000C6          CALLS   #0, EXCH$DIRE_GET_COLUMNS                      ; 0843
                                   38    A3  D6 000CD          INCL    56(R3)                                         ; 0844
                                 0000'   CF  9F 000D0          PUSHAB  NULL_STRING                                    ; 0848
         00000000G     EF                01  FB 000D4          CALLS   #1, EXCH$DIRE_PUT
                                         7E  D4 000DB          CLRL    -(SP)                                          ; 0850
                                   69    A5  9F 000DD          PUSHAB  105(R5)                                        ;
                                   65    A5  DD 000E0          PUSHL   101(R5)                                        ;
                                 0000'   CF  9F 000E3          PUSHAB  P.AAC                                          ; 0849
         00000000G     EF                04  FB 000E7          CALLS   #4, EXCH$UTIL_FAO_BUFFER                       ; 0850
                       58                50  D0 000EE          MOVL    R0, FAOP                                       ;
                       68                06  A2 000F1          SUBW2   #6, (FAOP)                                     ; 0851
                                         58  DD 000F4          PUSHL   FAOP                                           ; 0852
         00000000G     EF                01  FB 000F6          CALLS   #1, EXCH$DIRE_PUT                              ;
                                 0000'   CF  9F 000FD          PUSHAB  NULL_STRING                                    ; 0853
         00000000G     EF                01  FB 00101          CALLS   #1, EXCH$DIRE_PUT
                       5B                01  D0 00108          MOVL    #1, STATUS                                     ; 0857
                                   04    AE  7C 0010B          CLRQ    FILES_DISPLAYED                                ; 0859
                                         7E  D4 0010E          CLRL    -(SP)                                          ; 0863
                                         55  DD 00110          PUSHL   R5                                             ;
         00000000G     EF                02  FB 00112          CALLS   #2, EXCH$UTIL_DOS11CTX_ALLOCATE                ;
                       52                50  D0 00119          MOVL    R0, CTX                                        ;
               0C      A6                20  8A 0011C          BICB2   #32, 12(R6)                                    ; 0867
                       57                03  D0 00120          MOVL    #3, FLAGS                                      ; 0868
                       5A          66    A2  9E 00123          MOVAB   102(R2), R10                                   ; 0889
                                 0084    8F  BB 00127 9$:      PUSHR   #^M<R2,R7>                                     ; 0869
              0000V    CF                02  FB 0012B          CALLS   #2, EXCH$DOS11_NEXT_ENTRY                      ;
                                         50  D5 00130          TSTL    R0                                             ;
                                         58  13 00132          BEQL    12$                                            ;
                       58 00000000G      FF  E8 00134          BLBS    @EXCH$A_GBL, 13$                               ; 0875
                       57                02  D0 0013B          MOVL    #2, FLAGS                                      ; 0879
                       54          4C    A5  D0 0013E          MOVL    76(R5), NAM                                    ; 0880
                                         E3  13 00142 10$:     BEQL    9$                                             ; 0881
                       50          50    A4  9E 00144          MOVAB   80(NAM), R0                                    ; 0885
                                   04    A0  DD 00148          PUSHL   4(R0)                                          ; 0890
                       51                60  3C 0014B          MOVZWL  (R0), R1                                       ;
               04      AE          58    A4  3C 0014E          MOVZWL  88(NAM), 4(SP)                                 ;
                                   04 BE41   9F 00153          PUSHAB  @4(SP)[R1]                                     ;
                                         5A  DD 00157          PUSHL   R10                                            ; 0889
                                   50    A2  DD 00159          PUSHL   80(CTX)                                        ;
         00000000G     EF                04  FB 0015C          CALLS   #4, EXCH$CMD_MATCH_FILENAME                    ;
                       20                50  E9 00163          BLBC    R0, 11$                                        ;
                       51                54  D0 00166          MOVL    NAM, R1                                        ; 0892
                       50                52  D0 00169          MOVL    CTX, R0                                        ;
                                      0000V  30 0016C          BSBW    DOS11_MATCH_UIC                                ;
                       14                50  E9 0016F          BLBC    R0, 11$                                        ;
                                         52  DD 00172          PUSHL   CTX                                            ; 0895
              0000V    CF                01  FB 00174          CALLS   #1, EXCH$DOS11_DIRE_PUT_ITEM                   ;
                                   04    AE  D6 00179          INCL    FILES_DISPLAYED                                ; 0896
                       50          4A    A2  3C 0017C          MOVZWL  74(CTX), R0                                    ; 0897
               08      AE                50  C0 00180          ADDL2   R0, BLOCKS_DISPLAYED                           ;
                                         A1  11 00184          BRB     9$                                             ; 0894
```

```
                              54         70  A4  D0 00186 11$:     MOVL    112(NAM), NAM                               ; 0900
                                             B6  11 0018A          BRB     10$                                         ; 0881
                              17 00000000G  FF   E9 0018C 12$:     BLBC    @EXCH$A_GBL, 14$                            ; 0906
                              50 00000000G  8F   D0 00193 13$:     MOVL    #EXCH$_CANCELED, STATUS2                    ; 0908
       50          03         00             03  F0 0019A          INSV    #3, #0, #3, STATUS2                         ;
                                             50  DD 0019F          PUSHL   STATUS2                                     ;
                     00000000G  00           01  FB 001A1          CALLS   #1, LIB$SIGNAL                              ;
                                             70  11 001A8          BRB     21$                                         ;
                   11      0C  A6            05  E0 001AA 14$:     BBS     #5, 12(R6), 15$                             ; 0915
                   0C      30  A3            01  E1 001AF          BBC     #1, 48(R3), 15$                             ; 0917
                   07      0C  A6            06  E0 001B4          BBS     #6, 12(R6), 15$                             ; 0919
                                             01  DD 001B9          PUSHL   #1                                          ; 0921
                         0000V  CF           01  FB 001BB          CALLS   #1, EXCH$DOS11_DIRE_PUT_ITEM                ;
                                         40  A3  D5 001C0 15$:     TSTL    64(R3)                                      ; 0923
                                             07  13 001C3          BEQL    16$                                         ;
                                             7E  D4 001C5          CLRL    -(SP)                                       ; 0925
                         0000V  CF           01  FB 001C7          CALLS   #1, EXCH$DOS11_DIRE_PUT_ITEM                ;
                   0F      30  A3            01  E1 001CC 16$:     BBC     #1, 48(R3), 17$                             ; 0929
                                             01  DD 001D1          PUSHL   #1                                          ; 0931
                                      0000'  CF  9F 001D3          PUSHAB  NULL_STRING                                 ;
                     00000000G  EF           02  FB 001D7          CALLS   #2, EXCH$DIRE_PUT                           ;
                                             0F  11 001DE          BRB     18$                                         ;
                              50 00018292   8F   D0 001E0 17$:     MOVL    #98962, STATUS2                             ; 0936
       50          03         00             03  F0 001E7          INSV    #3, #0, #3, STATUS2                         ;
                              5B             50  D0 001EC          MOVL    STATUS2, STATUS                             ;
                                             69  95 001EF 18$:     TSTB    (R9)                                        ; 0940
                                             07  18 001F1          BGEQ    19$                                         ;
                              58      0000'  CF  9E 001F3          MOVAB   P.AAE, FAOP                                 ; 0941
                                             05  11 001F8          BRB     20$                                         ;
                              58      0000'  CF  9E 001FA 19$:     MOVAB   P.AAG, FAOP                                 ; 0943
                                         08  AE  DD 001FF 20$:     PUSHL   BLOCKS_DISPLAYED                            ; 0945
                                         08  AE  DD 00202          PUSHL   FILES_DISPLAYED                             ;
                                             58  DD 00205          PUSHL   FAOP                                        ;
                     00000000G  EF           03  FB 00207          CALLS   #3, EXCH$UTIL_FAO_BUFFER                    ;
                                58           50  D0 0020E          MOVL    R0, FAOP                                    ;
                                             58  DD 00211          PUSHL   FAOP                                        ; 0946
                     00000000G  EF           01  FB 00213          CALLS   #1, EXCH$DIRE_PUT                           ;
                                             52  DD 0021A 21$:     PUSHL   CTX                                         ; 0951
                     00000000G  EF           01  FB 0021C          CALLS   #1, EXCH$UTIL_DOS11CTX_RELEASE              ;
                                50           5B  D0 00223          MOVL    STATUS, R0                                  ; 0953
                                                 04 00226          RET                                                 ; 0954

; Routine Size:  551 bytes,    Routine Base:  EXCH$DOS11_CODE + 0603
```

```
1 GLOBAL ROUTINE exch$dos11_dire_put_item (ctx : $ref_bblock) : NOVALUE = %SBTTL 'exch$dos11_dire_put_item'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Add a single item to the output listing, writing the record if we hit the column count
2 !
2 ! INPUTS:
2 !
2 !       ctx  - an DOS11 directory entry context block, if address is zero, means write buffer.
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       dire - directory work area
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if put succeeded, false if not
2 !
2 ! SIDE EFFECTS:
2 !
2 !       error conditions will be signaled
2 !--
2
2 $dbgtrc_prefix ('dos11_dire_put_item> ');
2
2 LOCAL
2     desc : VECTOR [2, LONG],
2     status
2     ;
2
2 REGISTER
2     wid                                         ! current listing width
2     ;
2
2 BIND
2     dire = exch$a_gbl [excg$a_dire_work]        : $ref_bblock,                  ! pointer to work area
2     buf = dire [dire$t_list_buffer] : VECTOR [dire$s_list_buffer, BYTE] ! output buffer
2     ;
2
2
2 $block_check (2, .dire, dire, 598);
2
2 ! Perform a couple of sanity checks
2 !
2 $logic_check (2, (.dire [dire$l_cur_column] LSSU .dire [dire$l_q_columns]), 265);
2 $logic_check (2, (.dire [dire$l_cur_width]  LSSU .dire [dire$l_list_width]), 266);
2
```

```
;  2 ! Perform required initializations
;  2 !
;  2 dire [dire$v_items_printed] = true;             ! one or more items has been printed
;  2 wid = .dire [dire$l_cur_width];                 ! a fast register for the current width
;  2
;  2 ! An input address of 0 means that we should flush the buffer
;  2 !
;  3 IF (.ctx EQL 0)
;  2 THEN
;  3     BEGIN
;  3
;  3     IF .wid GTR 0
;  3     THEN
;  4         BEGIN
;  4         desc [0] = .wid;
;  4         desc [1] = buf [0];
;  4         exch$dire_put (desc);
;  3         END;
;  3
;  3     dire [dire$l_cur_column] = 0;
;  3     dire [dire$l_cur_width] = 0;
;  3     RETURN true;
;  2     END;
;  2
;  2 ! Add the separator to the buffer
;  2 !
;  2 IF .dire [dire$l_cur_column] NEQ 0
;  2 THEN
;  3     BEGIN
;  3     CH$FILL (%C ' ', dire$k_item_spacing, buf [.wid]);
;  3     wid = .wid + dire$k_item_spacing;
;  3     END
;  2 ELSE
;  3     BEGIN
;  3     buf [.wid] = %C ' ';
;  3     wid = .wid + 1;
;  2     END;
;  2
;  2 ! If the ctx is a 1, then print a "->(end-of-tape)"
;  2 !
;  2 IF .ctx EQL 1
;  2 THEN
;  3     BEGIN
;  3     CH$MOVE (%CHARCOUNT ('> (end-of-tape)'),  UPLIT BYTE ('> (end-of-tape)'), buf [.wid -1]);
;  3     wid = .wid + %CHARCOUNT ('> (end-of-tape)') -1;
;  3     END
;  3
;  3 ! CTX should be a valid pointer, print the file info
;  3 !
;  2 ELSE
;  3     BEGIN
;  3     $block_check (2, .ctx, dos11ctx, 621);
;  3
;  3      ! We always add the pointer to the directory if this is the current file
;  3      !
;  3      IF .ctx [dos11ctx$v_pointer]
;  3      THEN
```

```
3              buf [.wid -1] = %C '>';
3
3          ! Add the owner directory to the buffer if requested
3          !
3          IF .dire [dire$v_q_owner]
3          THEN
4              BEGIN
4              CH$MOVE (dos11ctx$s_exp_directory, ctx [dos11ctx$t_exp_directory], buf [.wid]);
4              wid = .wid + dos11ctx$s_exp_directory;
3              END;
3
3          ! We always add the filename and directory to the buffer
3          !
3          CH$MOVE (dos11ctx$s_exp_fullname, ctx [dos11ctx$t_exp_fullname], buf [.wid]);
3          wid = .wid + dos11ctx$s_exp_fullname;
3
3          ! Add the file size if requested
3          !
3          IF .dire [dire$v_q_size]
3          THEN
4              BEGIN
4              LOCAL
4                  addwid,
4                  ctrstr;
4              IF .dire [dire$v_q_octal]
4              THEN
5                  BEGIN
5                  addwid = 7;
5                  ctrstr = %ASCID ' !6OW';
5                  END
4              ELSE
5                  BEGIN
5                  addwid = 5;
5                  ctrstr = %ASCID '!5UL';
4                  END;
4              desc [0] = .addwid;
4              desc [1] = buf [.wid];
5              IF NOT (status = $fao (.ctrstr, 0, desc, .ctx [dos11ctx$w_blocks]))
4              THEN
4                  $exch_signal_stop (.status);
4
4              wid = .wid + .addwid;
3              END;
3
3          ! Add the date if requested
3          !
3          IF .dire [dire$v_q_date]
3          THEN
4              BEGIN
4              CH$MOVE (1, UPLIT BYTE (' '), buf [.wid]);
4              wid = .wid + 1;
4              CH$MOVE (dos11ctx$s_exp_date, ctx [dos11ctx$t_exp_date], buf [.wid]);
4              wid = .wid + dos11ctx$s_exp_date;
3              END;
3
3          ! Add the protection if /FULL requested
3          !
```

```
; 1037          1126  3      IF .dire [dire$v_q_full]
; 1038          1127  3      THEN
; 1039          1128  4          BEGIN
; 1040          1129  4          LOCAL
; 1041          1130  4              addwid,
; 1042          1131  4              ctrstr;
; 1043          1132  4          addwid = 6;
; 1044          1133  4          ctrstr = %ASCID ' <!30B>';
; 1045          1134  4          desc [0] = .addwid;
; 1046          1135  4          desc [1] = buf [.wid];
; 1047          1136  5          IF NOT (status = $fao (.ctrstr, 0, desc  .ctx [dos11ctx$w_protection]))
; 1048          1137  4          THEN
; 1049          1138  4              $exch_signal_stop (.status);
; 1050          1139  4
; 1051          1140  4          wid = .wid + .addwid;
; 1052          1141  4
; 1053          1142  3          END;
; 1054          1143  2      END;
; 1055          1144  2
; 1056          1145  2  ! Do we need to flush?
; 1057          1146  2  !
; 1058          1147  2  IF (dire [dire$l_cur_column] = .dire [dire$l_cur_column] + 1) GEQU .dire [dire$l_q_columns]
; 1059          1148  2  THEN
; 1060          1149  3      BEGIN
; 1061          1150  3
; 1062          1151  3      desc [0] = .wid;
; 1063          1152  3      desc [1] = buf [0];
; 1064          1153  3      exch$dire_put (desc);
; 1065          1154  3      dire [dire$l_cur_column] = 0;
; 1066          1155  3      wid = 0;
; 1067          1156  2      END;
; 1068          1157  2
; 1069          1158  2  dire [dire$l_cur_width] = .wid;
; 1070          1159  2
; 1071          1160  2  RETURN;
; 1072          1161  1  END;


                                                                        .PSECT  EXCH$DOS11_PLIT,NOWRT,2

29  65  70  61  74  2D  66  6F  2D  64  6E  65  28  20  3E  0010C P.AAI:   .ASCII  \> (end-of-tape)\                                ;
                                                            0011B          .BLKB   1
                                00  00  00  57  4F  36  21  20  0011C P.AAK:   .ASCII  \ !6OW\<0><0><0>                             ;
                                                010E0005  00124 P.AAJ:   .LONG   17694725                                           ;
                                                00000000' 00128          .ADDRESS P.AAK                                             ;
                                            4C  55  35  21  0012C P.AAM:   .ASCII  \!5UL\                                           ;
                                                010E0004  00130 P.AAL:   .LONG   17694724                                           ;
                                                00000000' 00134          .ADDRESS P.AAM                                             ;
                                                        20  00138 P.AAN:   .ASCII  \ \                                              ;
                                                            00139          .BLKB   3
                                00  3E  42  4F  33  21  3C  20  0013C P.AAP:   .ASCII  \ <!30B>\<0>                                 ;
                                                010E0007  00144 P.AAO:   .LONG   17694727                                           ;
                                                00000000' 00148          .ADDRESS P.AAP                                             ;

                                                                        .EXTRN  SYS$FAO
```

```
                                                               .PSECT  EXCH$DOS11_CODE,NOWRT,2

                                         0FFC 00000            .ENTRY  EXCH$DOS11_DIRE_PUT_ITEM, Save R2,R3,R4,R5,-; 0955
                                                                       R6,R7,R8,R9,R10,R11                          :
                            5E              0C  C2 00002        SUBL2   #12, SP                                     :
           50 00000000G     EF              0C  C1 00005        ADDL3   #12, EXCH$A_GBL, R0                         : 1000
                            58              60  D0 0000D        MOVL    (R0), R8                                    : 1001
                            57        44    A8  9E 00010        MOVAB   68(R8), R7                                  :
                            52 024400FE     8F  D0 00014        MOVL    #38011134, R2                               : 1005
                            51      0256    8F  3C 0001B        MOVZWL  #598, R1                                    :
                            50              58  D0 00020        MOVL    R8, R0                                      :
                               00000000G    EF  16 00023        JSB     EXCH$UTIL_BLOCK_CHECK                       :
                            6E        40    A8  9E 00029        MOVAB   64(R8), (SP)                                : 1009
                   28       A8        00    BE  D1 0002D        CMPL    @0(SP), 40(R8)                              :
                                            14  1F 00032        BLSSU   1$                                          :
                            7E      0109    8F  3C 00034        MOVZWL  #265, -(SP)                                 :
                                            01  DD 00039        PUSHL   #1                                          :
                               00000000G    8F  DD 0003B        PUSHL   #EXCH$_BADLOGIC                             :
              00000000G     00              03  FB 00041        CALLS   #3, LIB$STOP                                :
                   38       A8        3C    A8  D1 00048 1$:    CMPL    60(R8), 56(R8)                              : 1010
                                            14  1F 0004D        BLSSU   2$                                          :
                            7E      010A    8F  3C 0004F        MOVZWL  #266, -(SP)                                 :
                                            01  DD 00054        PUSHL   #1                                          :
                               00000000G    8F  DD 00056        PUSHL   #EXCH$_BADLOGIC                             :
              00000000G     00              03  FB 0005C        CALLS   #3, LIB$STOP                                :
                   30       A8              02  88 00063 2$:    BISB2   #2, 48(R8)                                  : 1014
                            56        3C    A8  D0 00067        MOVL    60(R8), WID                                 : 1015
                            59        04    AC  D0 0006B        MOVL    CTX, R9                                     : 1019
                                            19  12 0006F        BNEQ    4$                                          :
                                            56  D5 00071        TSTL    WID                                         : 1023
                                            0E  15 00073        BLEQ    3$                                          :
                   04       AE              56  7D 00075        MOVQ    WID, DESC                                   : 1026
                                      04    AE  9F 00079        PUSHAB  DESC                                        : 1028
              00000000G     EF              01  FB 0007C        CALLS   #1, EXCH$DIRE_PUT                           :
                                      00    BE  D4 00083 3$:    CLRL    @0(SP)                                      : 1031
                                      3C    A8  D4 00086        CLRL    60(R8)                                      : 1032
                                                04 00089        RET                                                 : 1033
                                      00    BE  D5 0008A 4$:    TSTL    @0(SP)                                      : 1038
                                            0F  13 0008D        BEQL    5$                                          :
  8647              18              00 00202020 8F  F0 0008F    INSV    #2105376, #0, #24, (WID)+[R7]               : 1041
                                    56      02  C0 00099        ADDL2   #2, WID                                     : 1042
                                            04  11 0009C        BRB     6$                                          : 1038
                          8647              20  90 0009E 5$:    MOVB    #32, (WID)+[R7]                             : 1046
                            01              59  D1 000A2 6$:    CMPL    R9, #1                                      : 1052
                                            0E  12 000A5        BNEQ    7$                                          :
          FF A647       0000'       CF      0F  28 000A7        MOVC3   #15, P.AAI, -1(WID)[R7]                     : 1055
                            56              0E  C0 000AF        ADDL2   #14, WID                                    : 1056
                                          00C4  31 000B2        BRW     16$                                         : 1052
                            52 008A00FC     8F  D0 000B5 7$:    MOVL    #9044220, R2                                : 1063
                            51      026D    8F  3C 000BC        MOVZWL  #621, R1                                    :
                            50              59  D0 000C1        MOVL    R9, R0                                      :
                               00000000G    EF  16 000C4        JSB     EXCH$UTIL_BLOCK_CHECK                       :
           05       28      A9              03  E1 000CA        BBC     #3, 40(R9), 8$                              : 1067
                         FF A647            3E  90 000CF        MOVB    #62, -1(WID)[R7]                            : 1069
                            5A        2C    A8  9E 000D4 8$:    MOVAB   44(R8), R10                                 : 1073
           09               6A              09  E1 000D8        BBC     #9, (R10), 9$                               :
         6647       5C      A9              0A  28 000DC        MOVC3   #10, 92(R9), (WID)[R7]                      : 1076
```

```
                       56                  0A  C0 000E2          ADDL2   #10, WID                          ; 1077
     6647        66    A9                  0D  28 000E5  9$:     MOVC3   #13, 102(R9), (WID)[R7]           ; 1082
                       56                  0D  C0 000EB          ADDL2   #13, WID                          ; 1083
       3A              6A                  0B  E1 000EE          BBC     #11, (R10), 12$                   ; 1087
                                           6A  95 000F2          TSTB    (R10)                             ; 1093
                                           0A  18 000F4          BGEQ    10$                               ;
                       52                  07  D0 000F6          MOVL    #7, ADDWID                        ; 1096
                       50         0000'    CF  9E 000F9          MOVAB   P.AAJ, CTRSTR                     ; 1097
                                           08  11 000FE          BRB     11$                               ; 1093
                       52                  05  D0 00100  10$:    MOVL    #5, ADDWID                        ; 1101
                       50         0000'    CF  9E 00103          MOVAB   P.AAL, CTRSTR                     ; 1102
                 04    AE                  52  D0 00108  11$:    MOVL    ADDWID, DESC                      ; 1104
  08   AE              57                  56  C1 0010C          ADDL3   WID, R7, DESC+4                   ; 1105
                       7E           4A     A9  3C 00111          MOVZWL  74(R9), -(SP)                     ; 1106
                                    08     AE  9F 00115          PUSHAB  DESC                              ;
                                           7E  D4 00118          CLRL    -(SP)                             ;
                                           50  DD 0011A          PUSHL   CTRSTR                            ;
           00000000G   00                  04  FB 0011C          CALLS   #4, SYS$FAO                       ;
                       5B                  50  D0 00123          MOVL    R0, STATUS                        ;
                       43                  5B  E9 00126          BLBC    STATUS, 14$                       ;
                       56                  52  C0 00129          ADDL2   ADDWID, WID                       ; 1110
       0F              6A                  03  E1 0012C  12$:    BBC     #3, (R10), 13$                    ; 1115
                     8647         0000'    CF  90 00130          MOVB    P.AAN, (WID)+[R7]                 ; 1118
     6647        7F    A9                  0B  28 00136          MOVC3   #11, 127(R9), (WID)[R7]           ; 1120
                       56                  0B  C0 0013C          ADDL2   #11, WID                          ; 1121
       36              6A                  06  E1 0013F  13$:    BBC     #6, (R10), 16$                    ; 1126
                       52                  06  D0 00143          MOVL    #6, ADDWID                        ; 1132
                       50         0000'    CF  9E 00146          MOVAB   P.AAO, CTRSTR                     ; 1133
                 04    AE                  52  D0 0014B          MOVL    ADDWID, DESC                      ; 1134
  08   AE              57                  56  C1 0014F          ADDL3   WID, R7, DESC+4                   ; 1135
                       7E           44     A9  3C 00154          MOVZWL  68(R9), -(SP)                     ; 1136
                                    08     AE  9F 00158          PUSHAB  DESC                              ;
                                           7E  D4 0015B          CLRL    -(SP)                             ;
                                           50  DD 0015D          PUSHL   CTRSTR                            ;
           00000000G   00                  04  FB 0015F          CALLS   #4, SYS$FAO                       ;
                       5B                  50  D0 00166          MOVL    R0, STATUS                        ;
                       0A                  5B  E8 00169          BLBS    STATUS, 15$                       ;
                                           5B  DD 0016C  14$:    PUSHL   STATUS                            ; 1138
           00000000G   00                  01  FB 0016E          CALLS   #1, LIB$STOP                      ;
                                               04 00175          RET                                       ;
                       56                  52  C0 00176  15$:    ADDL2   ADDWID, WID                       ; 1140
       50        00    BE                  01  C1 00179  16$:    ADDL3   #1, @0(SP), R0                    ; 1147
                 00    BE                  50  D0 0017E          MOVL    R0, @0(SP)                        ;
                 28    A8                  50  D1 00182          CMPL    R0, 40(R8)                        ;
                                           13  1F 00186          BLSSU   17$                               ;
                 04    AE                  56  7D 00188          MOVQ    WID, DESC                         ; 1151
                                    04     AE  9F 0018C          PUSHAB  DESC                              ; 1153
           00000000G   EF                  01  FB 0018F          CALLS   #1, EXCH$DIRE_PUT                 ;
                                    00     BE  D4 00196          CLRL    @0(SP)                            ; 1154
                                           56  D4 00199          CLRL    WID                               ; 1155
                 3C    A8                  56  D0 0019B  17$:    MOVL    WID, 60(R8)                       ; 1158
                                               04 0019F          RET                                       ; 1161

; Routine Size:  416 bytes,    Routine Base:  EXCH$DOS11_CODE + 082A
```

```
  GLOBAL ROUTINE exch$dos11_exit_handler (status, %SBTTL 'exch$dos11_exit_handler (status, volb)'
1                                         volb : $ref_bblock) : NOVALUE =
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Write two tapemarks at the end of the tape
2 !
2 ! INPUTS:
2 !
2 !       status - pointer to status code
2 !       volb - pointer to volb which has been connected to the DOS-11 device
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       none
2 !
2 ! SIDE EFFECTS:
2 !
2 !       any modified directory segments will be written
2 !--
2
2 $dbgtrc_prefix ('dos11_exit_handler> ');
2
2 BIND
2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
2     ;
2
2 $debug_print_lit ('entry');
2 $block_check (2, .volb, volb, 431);
2 $block_check (2, .dosv, dos11, 432);
2
2 $trace_print_fao ('entry - volb !XL', .volb);
2
2 ! If it looks like the tape can be written (i.e. no serious errors have occurred) try to write the tapemarks
2 !
2 IF .dosv [dos11$v_position_valid]               ! We should know where we are
2   AND
2   .dosv [dos11$v_end_of_tape]                   !  and that should be at the end.
2 THEN
3     BEGIN
3
3     ! Write two tape marks to mark the end of the tape
3     !
```

```
; 1131            1219  3      exch$io_dos11_write_tape_mark (.volb);
; 1132            1220  3      exch$io_dos11_write_tape_mark (.volb);
; 1133            1221  3
; 1134            1222  2      END;
; 1135            1223  2
; 1136            1224  2  RETURN;
; 1137            1225  1  END;



                                              003C 00000          .ENTRY   EXCH$DOS11_EXIT_HANDLER, Save R2,R3,R4,R5     ; 1162
                           55 00000000G  EF    9E 00002          MOVAB    EXCH$UTIL_BLOCK_CHECK, R5                      ;
                           54 00000000G  EF    9E 00009          MOVAB    EXCH$IO_DOS11_WRITE_TAPE_MARK, R4              ;
             53       08   AC 00000054   8F    C1 00010          ADDL3    #84, VOLB, R3                                  ; 1200
                           52 041B00F3   8F    D0 00019          MOVL     #68878579, R2                                  ; 1204
                           51     01AF   8F    3C 00020          MOVZWL   #431, R1                                       ;
                           50       08   AC    D0 00025          MOVL     VOLB, R0                                       ;
                                         65    16 00029          JSB      EXCH$UTIL_BLOCK_CHECK                          ;
                           53            63    D0 0002B          MOVL     (R3), R3                                       ; 1205
                           52 003600FD   8F    D0 0002E          MOVL     #3539197, R2                                   ;
                           51     01B0   8F    3C 00035          MOVZWL   #432, R1                                       ;
                           50            53    D0 0003A          MOVL     R3, R0                                         ;
                                         65    16 0003D          JSB      EXCH$UTIL_BLOCK_CHECK                          ;
                           11       0C   A3    E9 0003F          BLBC     12(R3), 1$                                     ; 1211
             0C       0C   A3            03    E1 00043          BBC      #3, 12(R3), 1$                                 ; 1213
                                    08   AC    DD 00048          PUSHL    VOLB                                           ; 1219
                           64            01    FB 0004B          CALLS    #1, EXCH$IO_DOS11_WRITE_TAPE_MARK              ;
                                    08   AC    DD 0004E          PUSHL    VOLB                                           ; 1220
                           64            01    FB 00051          CALLS    #1, EXCH$IO_DOS11_WRITE_TAPE_MARK              ;
                                               04 00054 1$:      RET                                                     ; 1225

; Routine Size:  85 bytes,    Routine Base:  EXCH$DOS11_CODE + 09CA
```

```
1 GLOBAL ROUTINE exch$dos11_expand_filename (ctx : $ref_bblock) : NOVALUE =       %SBTTL 'exch$dos11_expand_fi
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Convert the information in a directory entry to ascii text.  This involves changing the RADIX-50
2 !       filename to ASCII and converting the date to ASCII.
2 !
2 ! INPUTS:
2 !
2 !       ctx - pointer to an dos11ctx structure which contains a copy of the directory entry
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       ctx - receives ASCII filename info
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       success or failure if the entry is invalid
2 !
2 ! SIDE EFFECTS:
2 !
2 !       error conditions will be signaled
2 !--
2
2 $dbgtrc_prefix ('dos11_expand_filename> ');
2
2 LOCAL
2     dayomon : REF VECTOR [12, BYTE],
2     year,
2     mon,
2     day,
2     date_desc : VECTOR [2, LONG],
2     status,
2     ch
2     ;
2
2 BIND
2     volb = ctx [dos11ctx$a_assoc_volb]          : $ref_bblock,
2     dosv = volb [volb$a_vfmt_specific]          : $ref_bblock
2     ;
2
2 $debug_print_fao ('entry - ctx !XL', .ctx);
2 $block_check (2, .ctx, dos11ctx, 599);
2 $block_check (2, .volb, volb, 619);
2 $block_check (2, .dosv, dos11, 620);
2
2 ! Convert the file name from 3 Radix-50 words to 'dos11ctx$s_exp_name' ASCII characters, type from 1 R50 wor
```

```
2 ! 'dos11ctx$s_exp_type' chars
2 !
2 exch$util_radix50_to_ascii (6, ctx [dos11ctx$l_filename_1], ctx [dos11ctx$t_exp_name]);
2 exch$util_radix50_to_ascii (3, ctx [dos11ctx$w_filename_2], ctx [dos11ctx$t_exp_name]+6);
2 ch = CH$FIND_CH (dos11ctx$s_exp_name, ctx [dos11ctx$t_exp_name], ' ');
2 ctx [dos11ctx$l_exp_name_len] = (IF .ch EQL 0 THEN dos11ctx$s_exp_name ELSE .ch - ctx [dos11ctx$t_exp_name])
2
2 exch$util_radix50_to_ascii (dos11ctx$s_exp_type, ctx [dos11ctx$w_filetype], ctx [dos11ctx$t_exp_type]);
2 ch = CH$FIND_CH (dos11ctx$s_exp_type, ctx [dos11ctx$t_exp_type], ' ');
2 ctx [dos11ctx$l_exp_type_len] = (IF .ch EQL 0 THEN dos11ctx$s_exp_type ELSE .ch - ctx [dos11ctx$t_exp_type])
2
2 ! Create a filename in the standard, non-embedded blank format by concatenating the name, a "." and the type
2 !
2 ctx [dos11ctx$l_exp_fullname_len] = .ctx [dos11ctx$l_exp_name_len] + 1 + .ctx [dos11ctx$l_exp_type_len];
2 CH$COPY (.ctx [dos11ctx$l_exp_name_len], ctx [dos11ctx$t_exp_name],              ! the file name
2         1, UPLIT BYTE ('.'),                                                      ! the "." separator
2         .ctx [dos11ctx$l_exp_type_len], ctx [dos11ctx$t_exp_type],                ! the file type
2        %C ' ', dos11ctx$s_exp_fullname, ctx [dos11ctx$t_exp_fullname]);         ! the blank-padded result
2
2 ! Create an ASCII representation of the date
2 !
2 IF .ctx [dos11ctx$w_date] EQL 0
2 THEN
2     CH$MOVE (dos11ctx$s_exp_date, UPLIT BYTE (' < nodate >'), ctx [dos11ctx$t_exp_date])
2 ELSE
3     BEGIN
3     year = 1970 + (.ctx [dos11ctx$w_date]/1000);! 1970 is stored as zero
3     day  = .ctx [dos11ctx$w_date] MOD 1000;     ! Day of year is stored as remainder
4     dayomon =  (IF .year <0,2,0> EQL 0          ! Pick leap year or regular depending
4                THEN days_of_months_leapyear     !  on whether year is divisible by 4
3                ELSE days_of_months);
3     mon = 0;
3     WHILE .day GTRU .dayomon [.mon]              ! Subtract out number of days in each month until we are wit
3         AND                                      !  a month
3           .mon LSSU 12                           ! (Print *** if we didn't find it)
3     DO
4         BEGIN
4         day = .day - .dayomon [.mon];
4         mon = .mon + 1;
3         END;
3     date_desc [0] = dos11ctx$s_exp_date;                     ! set up a descriptor for FAO
3     date_desc [1] = ctx [dos11ctx$t_exp_date];
4     IF NOT (status = $fao (%ASCID '!2UL-!AF!4UL', 0, date_desc, .day, 4, months [.mon], .year))
3     THEN
3         $exch_signal_stop (.status);
2     END;
2
2 ! If the tape is currently at this file, mark for the pointer
2 !
2 IF .ctx [dos11ctx$w_file_number] EQL .dosv [dos11$l_current_file]
2 THEN
3     BEGIN
3     ctx [dos11ctx$v_pointer] = true;
3     dosv [dos11$v_current_marked] = true;
3     END
2 ELSE
2     ctx [dos11ctx$v_pointer] = false;
```

```
; 1253             1340  2
; 1254             1341  2 ! Create an ASCII representation of the UIC format directory
; 1255             1342  2 !
; 1256             1343  2 date_desc [0] = dos11ctx$s_exp_directory;        ! set up a descriptor for FAO
; 1257             1344  2 date_desc [1] = ctx [dos11ctx$t_exp_directory];
; 1258             1345  3 IF NOT (status = $fao (%ASCID '[!3OB,.3OB] ', 0, date_desc, .ctx [dos11ctx$b_group], .ctx [dos11ctx$b_member
; 1259             1346  2 THEN
; 1260             1347  2     $exch_signal_stop (.status);
; 1261             1348  2
; 1262             1349  1 END;


                                                                       .PSECT  EXCH$DOS11_PLIT,NOWRT,2

                                                   2E  0014C P.AAQ:    .ASCII  \.\                                 ;
            3E  20  65  74  61  64  6F  6E  20  3C  20  0014D P.AAR:    .ASCII  \ < nodate >\                       ;
        4C  55  34  21  46  41  21  2D  4C  55  32  21  00158 P.AAT:    .ASCII  \!2UL-!AF!4UL\                      ;
                                            010E000C  00164 P.AAS:    .LONG   17694732                            ;
                                            00000000' 00168           .ADDRESS P.AAT                              ;
        20  5D  42  4F  33  21  2C  42  4F  33  21  5B  0016C P.AAV:    .ASCII  \[!3OB,!3OB] \                      ;
                                            010E000C  00178 P.AAU:    .LONG   17694732                            ;
                                            00000000' 0017C           .ADDRESS P.AAV                              ;


                                                                       .PSECT  EXCH$DOS11_CODE,NOWRT,2

                                                0FFC 00000            .ENTRY  EXCH$DOS11_EXPAND_FILENAME, Save R2,R3,R4,-  ; 1226
                                                                              R5,R6,R7,R8,R9,R10,R11                       ;
                             5B 00000000G  EF  9E 00002            MOVAB   EXCH$UTIL_BLOCK_CHECK, R11                   ;
                             5A 00000000G  EF  9E 00009            MOVAB   EXCH$UTIL_RADIX50_TO_ASCII, R10              ;
                             5E            08  C2 00010            SUBL2   #8, SP                                       ;
                             56        04  AC  D0 00013            MOVL    CTX, R6                                      ; 1273
              53        14   A6 00000054   8F  C1 00017            ADDL3   #84, 20(R6), R3                              ; 1274
                             52 008A00FC   8F  D0 00020            MOVL    #9044220, R2                                 ; 1278
                             51      0257  8F  3C 00027            MOVZWL  #599, R1                                     ;
                             50            56  D0 0002C            MOVL    R6, R0                                       ;
                                           6B  16 0002F            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                             52 041B00F3   8F  D0 00031            MOVL    #68878579, R2                                ; 1279
                             51      026B  8F  3C 00038            MOVZWL  #619, R1                                     ;
                             50        14  A6  D0 0003D            MOVL    20(R6), R0                                   ;
                                           6B  16 00041            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                             59            63  D0 00043            MOVL    (R3), R9                                     ; 1280
                             52 003600FD   8F  D0 00046            MOVL    #3539197, R2                                 ;
                             51      026C  8F  3C 0004D            MOVZWL  #620, R1                                     ;
                             50            59  D0 00052            MOVL    R9, R0                                       ;
                                           6B  16 00055            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                                       73  A6  9F 00057            PUSHAB  115(R6)                                      ; 1285
                                       3C  A6  9F 0005A            PUSHAB  60(R6)                                       ;
                                           06  DD 0005D            PUSHL   #6                                           ;
                             6A            03  FB 0005F            CALLS   #3, EXCH$UTIL_RADIX50_TO_ASCII               ;
                                       79  A6  9F 00062            PUSHAB  121(R6)                                      ; 1286
                                       48  A6  9F 00065            PUSHAB  72(R6)                                       ;
                                           03  DD 00068            PUSHL   #3                                           ;
                             6A            03  FB 0006A            CALLS   #3, EXCH$UTIL_RADIX50_TO_ASCII               ;
         73   A6             09            20  3A 0006D            LOCC    #32, #9, 115(R6)                             ; 1287
```

```
                                        02  12 00072        BNEQ     1$                                          :
                                        51  D4 00074        CLRL     R1                                          :
                              52        51  D0 00076 1$:    MOVL     ^1, CH                                      :
                                        05  12 00079        BNEQ     'S                                          : 1288
                              51        09  D0 0007B        MOVL     #9, R1                                      :
                                        08  11 0007E        BRB      3$                                          :
                              50    73  A6  9E 00080 2$:    MOVAB    115(R6), R0                                 :
                51            52        50  C3 00084        SUBL3    R0, CH, R1                                  :
                         54   A6        51  D0 00088 3$:    MOVL     R1, 84(R6)                                  :
                                    7C  A6  9F 0008C        PUSHAB   124(R6)                                     : 1290
                                    40  A6  9F 0008F        PUSHAB   64(R6)                                      :
                                        03  DD 00092        PUSHL    #3                                          :
                              6A        03  FB 00094        CALLS    #3, EXCH$UTIL_RADIX50_TO_ASCII              :
           7C   A6            03        20  3A 00097        LOCC     #32, #3, 124(R6)                            : 1291
                                        02  12 0009C        BNEQ     4$                                          :
                                        51  D4 0009E        CLRL     R1                                          :
                              52        51  D0 000A0 4$:    MOVL     R1, CH                                      :
                                        05  12 000A3        BNEQ     5$                                          : 1292
                              51        03  D0 000A5        MOVL     #3, R1                                      :
                                        08  11 000A8        BRB      6$                                          :
                              50    7C  A6  9E 000AA 5$:    MOVAB    124(R6), R0                                 :
                51            52        50  C3 000AE        SUBL3    R0, CH, R1                                  :
                         58   A6        51  D0 000B2 6$:    MOVL     R1, 88(R6)                                  :
                50       54   A6    58  A6  C1 000B6        ADDL3    88(R6), 84(R6), R0                          : 1296
                         50   A6    01  A0  9E 000BC        MOVAB    1(R0), 80(R6)                               :
                              58        0D  D0 000C1        MOVL     #13, R8                                     : 1299
                              57    66  A6  9E 000C4        MOVAB    102(R6), R7                                 : 1300
58              20       73   A6    54  A6  2C 000C8        MOVC5    84(R6), 115(R6), #32, R8, (R7)              :
                                        67     000CF                                                         :
                                        1E  18 000D0        BGEQ     7$                                          :
                              57    54  A6  C0 000D2        ADDL2    84(R6), R7                                  :
                              58    54  A6  C2 000D6        SUBL2    84(R6), R8                                  :
58              20    0000'   CF        01  2C 000DA        MOVC5    #1, P.AAQ, #32, R8, (R7)                    :
                                        67     000E1                                                         :
                                        0C  18 000E2        BGEQ     7$                                          :
                                        57  D6 000E4        INCL     R7                                          :
                                        58  D7 000E6        DECL     R8                                          :
58              20       7C   A6    58  A6  2C 000E8        MOVC5    88(R6), 124(R6), #32, R8, (R7)              :
                                        67     000EF                                                         :
                              58    46  A6  3C 000F0 7$:    MOVZWL   70(R6), R8                                  : 1304
                                        09  12 000F4        BNEQ     8$                                          :
           7F   A6    0000'   CF        0B  28 000F6        MOVC3    #11, P.AAR, 127(R6)                         : 1306
                                        6A  11 000FD        BRB      13$                                         :
                50            58 000003E8  8F  C7 000FF 8$:  DIVL3    #1000, R8, R0                               : 1309
                              50     07B2  C0  9E 00107        MOVAB    1970(R0), YEAR                              :
7E              00            58        01  7A 0010C        EMUL     #1, R8, #0, -(SP)                           : 1310
53              53            8E 000003E8  8F  7B 00111        EDIV     #1000, (SP)+, DAY, DAY                      :
                              03        50  93 0011A        BITB     YEAR, #3                                    : 1311
                                        07  12 0011D        BNEQ     9$                                          :
                              52   0000'  CF  9E 0011F        MOVAB    DAYS_OF_MONTHS_LEAPYEAR, DAYOMON            :
                                        05  11 00124        BRB      10$                                         :
                              52   0000'  CF  9E 00126 9$:    MOVAB    DAYS_OF_MONTHS, DAYOMON                     :
                                        51  D4 0012B 10$:   CLRL     MON                                         : 1314
53            6142            08        00  ED 0012D 11$:   CMPZV    #0, #8, (MON)[DAYOMON], DAY                 : 1315
                                        0E  1E 00133        BGEQU    12$                                         :
                              0C        51  D1 00135        CMPL     MON, #12                                    : 1317
                                        09  1E 00138        BGEQU    12$                                         :
```

```
                                      54       8142  9A 0013A          MOVZBL  (MON)+[DAYOMON], R4                          ; 1320
                                      53         54  C2 0013E          SUBL2   R4, DAY
                                                 EA  11 00141          BRB     11$                                          ; 1315
                                      6E         0B  D0 00143 12$:     MOVL    #11, DATE_DESC                               ; 1323
                                04    AE    7F   A6  9E 00146          MOVAB   127(R6), DATE_DESC+4                         ; 1324
                                                 50  DD 0014B          PUSHL   YEAR                                         ; 1325
                                       0000'CF41  DF 0014D          PUSHAL  MONTHS[MON]
                                                 04  DD 00152          PUSHL   #4
                                                 53  DD 00154          PUSHL   DAY
                                            10   AE  9F 00156          PUSHAB  DATE_DESC
                                                 7E  D4 00159          CLRL    -(SP)
                                         0000'   CF  9F 0015B          PUSHAB  P.AAS
                           00000000G  00         07  FB 0015F          CALLS   #7, SYS$FAO
                                      3A         50  E9 00166          BLBC    STATUS, 16$
      0E    A9        4C    A6        10         00  ED 00169 13$:     CMPZV   #0, #16, 76(R6), 14(R9)                      ; 1332
                                                 0A  12 00170          BNEQ    14$
                                28    A6         08  88 00172          BISB2   #8, 40(R6)                                   ; 1335
                                0C    A9         20  88 00176          BISB2   #32, 12(R9)                                  ; 1336
                                                 04  11 0017A          BRB     15$                                          ; 1332
                                28    A6         08  8A 0017C 14$:     BICB2   #8, 40(R6)                                   ; 1339
                                      6E         0A  D0 00180 15$:     MOVL    #10, DATE_DESC                               ; 1343
                                04    AE    5C   A6  9E 00183          MOVAB   92(R6), DATE_DESC+4                          ; 1344
                                      7E    42   A6  9A 00188          MOVZBL  66(R6), -(SP)                                ; 1345
                                      7E    43   A6  9A 0018C          MOVZBL  67(R6), -(SP)
                                            08   AE  9F 00190          PUSHAB  DATE_DESC
                                                 7E  D4 00193          CLRL    -(SP)
                                         0000'   CF  9F 00195          PUSHAB  P.AAU
                           00000000G  00         05  FB 00199          CALLS   #5, SYS$FAO
                                      09         50  E8 001A0          BLBS    STATUS, 17$
                                                 50  DD 001A3 16$:     PUSHL   STATUS                                       ; 1347
                           00000000G  00         01  FB 001A5          CALLS   #1, LIB$STOP
                                                     04 001AC 17$:     RET                                                  ; 1349

; Routine Size:  429 bytes,    Routine Base:  EXCH$DOS11_CODE + 0A1F
```

```
1 GLOBAL ROUTINE exch$dos11_find_file (ctx : $ref_bblock, name, name_len) =        %SBTTL 'exch$dos11_find_file
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Search for a specific file or a DOS-11 magtape
2 !
2 ! INPUT:
2 !
2 !       ctx      - pointer to a DOS11CTX structure
2 !       name     - pointer to the start of the file name for which to search
2 !       name_len - length of the name
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       ctx      - if a file is found, the context block will describe it
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to find the file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2 $dbgtrc_prefix ('dos11_find_file> ');
2
2 LOCAL
2     flags
2     ;
2
2 BIND
2     copy = exch$a_gbl [excg$a_copy_work]        : $ref_bblock,
2     inp_filb = copy [copy$a_inp_filb]           : $ref_bblock,
2     namb     = inp_filb [filb$a_assoc_namb]     : $ref_bblock
2     ;
2
2 ! Loop through all the labels in the directory
2 !
2 flags = (IF .copy [copy$v_q_rewind] THEN dosnxt$m_rewind ELSE 0);       ! Rewind for the first file
2 copy [copy$v_q_rewind] = 0;                                              ! But only for the first file
2
2 WHILE exch$dos11_next_entry (.ctx, .flags) NEQ 0
2 DO
3     BEGIN
3
3     flags = 0;                                  ! Don't rewind for later files
3
```

```
; 1321          1407  3      IF exch$cmd_match_filename
; 1322          1408  3               (.ctx [dos11ctx$l_exp_fullname_len], ctx [dos11ctx$t_exp_fullname], .name_len, .name)
; 1323          1409  3        AND
; 1324          1410  3             dos11_match_uic (.ctx, .namb)
; 1325          1411  3      THEN
; 1326          1412  3         RETURN 1;
; 1327          1413  3
; 1328          1414  2      END;
; 1329          1415  2
; 1330          1416  2 RETURN 0;
; 1331          1417  1 END;



                                      003C 00000          .ENTRY  EXCH$DOS11_FIND_FILE, Save R2,R3,R4,R5           ; 1350
      50 00000000G  EF              04  C1 00002          ADDL3   #4, EXCH$A_GBL, R0                               ; 1391
                    50              60  D0 0000A          MOVL    (R0), R0                                         ; 1392
      55        3C  A0              18  C1 0000D          ADDL3   #24, 60(R0), R5                                  ; 1393
                    05        31    A0  E9 00012          BLBC    49(R0), 1$                                       ; 1398
                    53              01  D0 00016          MOVL    #1, FLAGS                                        ;
                                    02  11 00019          BRB     2$                                               ;
                                    53  D4 0001B 1$:      CLRL    FLAGS                                            ;
                31  A0              01  8A 0001D 2$:      BICB2   #1, 49(R0)                                       ; 1399
      54        04  AC 00000066     8F  C1 00021          ADDL3   #102, CTX, R4                                    ; 1408
                    52        04    AC  D0 0002A          MOVL    CTX, R2                                          ;
                                    53  DD 0002E 3$:      PUSHL   FLAGS                                            ; 1401
                              04    AC  DD 00030          PUSHL   CTX                                              ;
             0000V  CF              02  FB 00033          CALLS   #2, EXCH$DOS11_NEXT_ENTRY                        ;
                                    50  D5 00038          TSTL    R0                                               ;
                                    28  13 0003A          BEQL    4$                                               ;
                                    53  D4 0003C          CLRL    FLAGS                                            ; 1405
                              08    AC  DD 0003E          PUSHL   NAME                                             ; 1408
                              0C    AC  DD 00041          PUSHL   NAME_LEN                                         ;
                                    54  DD 00044          PUSHL   R4                                               ;
                              50    A2  DD 00046          PUSHL   80(R2)                                           ;
         00000000G  EF              04  FB 00049          CALLS   #4, EXCH$CMD_MATCH_FILENAME                      ;
                    DB              50  E9 00050          BLBC    R0, 3$                                           ;
                    51              65  D0 00053          MOVL    (R5), R1                                         ; 1410
                    50        04    AC  D0 00056          MOVL    CTX, R0                                          ;
                                 0000V  30 0005A          BSBW    DOS11_MATCH_UIC                                  ;
                    CE              50  E9 0005D          BLBC    R0, 3$                                           ;
                    50              01  D0 00060          MOVL    #1, R0                                           ; 1412
                                        04 00063          RET                                                      ;
                                    50  D4 00064 4$:      CLRL    R0                                               ; 1416
                                        04 00066          RET                                                      ; 1417

; Routine Size:  103 bytes,    Routine Base:  EXCH$DOS11_CODE + 0BCC
```

```
1 GLOBAL ROUTINE exch$dos11_form_cur_date (ent : $ref_bblock) : NOVALUE jsb_r1 =  %SBTTL 'exch$dos11_form_cur_
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Format the current date, placing it into the date field of an DOS-11 directory entry
2 !
2 ! INPUT:
2 !
2 !       ent - pointer to the directory entry
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       none
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('dos11_form_cur_date> ');
2
2 LOCAL
2     date : WORD,
2     dayomon : REF VECTOR [12, BYTE],
2     timbuf : VECTOR [7, WORD]
2     ;
2
2 BIND
2     year = timbuf [0] : WORD, month = timbuf [1] : WORD, day = timbuf [2] : WORD;
2
2 $numtim (timbuf=timbuf);
2
2 date = (.year - 1970) * 1000;                       ! Multiply years since 1970 times 100
2
3 dayomon =  (IF .year <0,2,0> EQL 0                  ! Pick leap year or regular depending
3              THEN days_of_months_leapyear           !  on whether year is divisible by 4
2              ELSE days_of_months);
2
2 INCR mon FROM 0 TO .month-2                         ! Subtract 1 to convert 1-12 to 0-11 and one because we do n
2 DO                                                  !  current month, add days of previous months to the year
2     date = .date + .dayomon [.mon];
2
2 date = .date + .day;                                ! Add days of this month
```

; R

```
; 1390           1475  2
; 1391           1476  2 ent [dos11ent$w_date] = .date;                 ! Store the final DOS-11 date
; 1392           1477  2
; 1393           1478  2 RETURN;
; 1394           1479  1 END;



                                                                  .EXTRN  SYS$NUMTIM

                               5E            20  C2 00000 EXCH$DOS11_FORM_CUR_DATE::
                                                                  SUBL2   #32, SP                                   ; 1418
                     08   AE                 51  D0 00003         MOVL    R1, ENT                                   ;
                                             7E  D4 00007         CLRL    -(SP)                                     ; 1462
                                      14     AE  9F 00009         PUSHAB  TIMBUF                                    ;
                  00000000G  00              02  FB 0000C         CALLS   #2, SYS$NUMTIM                            ;
                               50     10     AE  3C 00013         MOVZWL  YEAR, R0                                  ; 1464
                               50   F84E     C0  9E 00017         MOVAB   -1970(R0), R0                             ;
            04    AE           50   03E8     8F  A5 0001C         MULW3   #1000, R0, DATE                           ;
                               03     10     AE  93 00023         BITB    YEAR, #3                                  ; 1466
                                             07  12 00027         BNEQ    1$                                        ;
                               50   0000'    CF  9E 00029         MOVAB   DAYS_OF_MONTHS_LEAPYEAR, DAYOMON          ;
                                             05  11 0002E         BRB     2$                                        ;
                               50   0000'    CF  9E 00030 1$:     MOVAB   DAYS_OF_MONTHS, DAYOMON                   ;
                     0C   AE          12     AE  3C 00035 2$:     MOVZWL  MONTH, 12(SP)                             ; 1470
                     0C   AE                 02  C2 0003A         SUBL2   #2, 12(SP)                                ;
                               51            01  CE 0003E         MNEGL   #1, MON                                   ; 1472
                                             08  11 00041         BRB     4$                                        ;
                               6E          6140  9A 00043 3$:     MOVZBL  (MON)[DAYOMON], (SP)                      ;
                     04   AE                 6E  A0 00047         ADDW2   (SP), DATE                                ;
                F3             51     0C     AE  F3 0004B 4$:     AOBLEQ  12(SP), MON, 3$                           ;
                     04   AE          14     AE  A0 00050         ADDW2   DAY, DATE                                 ; 1474
                50   08   AE                 12  C1 00055         ADDL3   #18, ENT, R0                              ; 1476
                               60     04     AE  B0 0005A         MOVW    DATE, (R0)                                ;
                               5E            20  C0 0005E         ADDL2   #32, SP                                   ; 1479
                                                 05 00061         RSB                                               ;

; Routine Size:  98 bytes,    Routine Base:  EXCH$DOS11_CODE + 0C33
; 1396          1480  1 GLOBAL ROUTINE dos11_match_uic (ctx : $ref_bblock, namb : $ref_bblock) : jsb_r0r1 =     %SBTTL 'dos11_match_
; 1397          1481  2 BEGIN
; 1398          1482  2 !++
; 1399          1483  2 !
; 1400          1484  2 ! FUNCTIONAL DESCRIPTION:
; 1401          1485  2 !
; 1402          1486  2 !       Check to see if the uic info from the nam block match those of the file
; 1403          1487  2 !
; 1404          1488  2 ! INPUTS:
; 1405          1489  2 !
; 1406          1490  2 !       ctx  - pointer to dos11ctx block
; 1407          1491  2 !       namb - pointer to namb block
; 1408          1492  2 !
; 1409          1493  2 ! ROUTINE VALUE:
; 1410          1494  2 !
; 1411          1495  2 !       true if the file is selected by the namb, false if not
; 1412          1496  2 !
; 1413          1497  2 ! SIDE EFFECTS:
; 1414          1498  2 !
; 1415          1499  2 !       none
; 1416          1500  2 !--
; 1417          1501  2
; 1418          1502  2 $dbgtrc_prefix ('dos11_match_uic> ');
; 1419          1503  2
; 1420          1504  2 IF NOT .namb [namb$v_uic_directory]             ! If no directory specified, it matches
; 1421          1505  2 THEN
; 1422          1506  2     RETURN 1;
; 1423          1507  2
; 1424          1508  2 IF NOT .namb [namb$v_wild_group]                 ! If group not wild, we must check group
; 1425          1509  2 THEN
; 1426          1510  2     IF .namb [namb$b_uic_group] NEQ .ctx [dos11ctx$b_group]
; 1427          1511  2     THEN
; 1428          1512  2         RETURN 0;
; 1429          1513  2
; 1430          1514  2 IF NOT .namb [namb$v_wild_member]                ! If member not wild, we must check member
; 1431          1515  2 THEN
; 1432          1516  2     IF .namb [namb$b_uic_member] NEQ .ctx [dos11ctx$b_member]
; 1433          1517  2     THEN
; 1434          1518  2         RETURN 0;
; 1435          1519  2
; 1436          1520  2 RETURN 1;
; 1437          1521  1 END;



                     1A      6D   A1            06  E1 00000 DOS11_MATCH_UIC::
                                                                  BBC     #6, 109(NAMB), 2$                        ; 1504
                     08      6C   A1            04  E0 00005      BBS     #4, 108(NAMB), 1$                        ; 1508
                             43   A0     0084   C1  91 0000A      CMPB    132(NAMB), 67(CTX)                       ; 1510
                                                11  12 00010      BNEQ    3$                                       ;
                     08      6C   A1            05  E0 00012 1$:  BBS     #5, 108(NAMB), 2$                        ; 1514
                             42   A0     0083   C1  91 00017      CMPB    131(NAMB), 66(CTX)                       ; 1516
                                                04  12 0001D      BNEQ    3$                                       ;
                                  50            01  D0 0001F 2$:  MOVL    #1, R0                                   ; 1520
                                                    05 00022      RSB                                              ;
```

```
                                                                    F  2
EXCH$DOS11      dos11 file and directory routines                   16-Sep-1984 00:52:08    VAX-11 Bliss-32 V4.0-742            Page  47
V04-000         dos11_match_uic                                     14-Sep-1984 12:29:04    [EXCHNG.SRC]EXCDOS11.B32;1              (13)

                                               50  D4 00023 3$:      CLRL    R0                                                ; 1521
                                                   05 00025          RSB                                                       ;

; Routine Size:  38 bytes,    Routine Base:  EXCH$DOS11_CODE + 0C95
```

```
GLOBAL ROUTINE exch$dos11_mount (volb : $ref_bblock) =  %SBTTL 'exch$dos11_mount (volb)'
BEGIN
!++
!
! FUNCTIONAL DESCRIPTION:
!
!       Perform DOS-11 volume specific mount processing
!
! INPUTS:
!
!       volb - pointer to volb which has been connected to the DOS-11 device
!
! IMPLICIT INPUTS:
!
!       none
!
! OUTPUTS:
!
!       none
!
! IMPLICIT OUTPUTS:
!
!       none
!
! ROUTINE VALUE:
!
!       none
!
! SIDE EFFECTS:
!
!       none
!--

$dbgtrc_prefix ('dos11_mount> ');

LOCAL
    dosv : $ref_bblock,                         ! a pointer to the dos11 volb extension
    status
    ;

$debug_print_lit ('entry');

$block_check (2, .volb, volb, 590);
$logic_check (3, (.volb [volb$a_vfmt_specific] EQL 0), 259);

! Allocate and initialize our volb extension
!
dosv = exch$util_vm_allocate (exchblk$s_dos11);
CH$FILL (0, dos11$k_end_zero - dos11$k_start_zero, .dosv + dos11$k_start_zero); ! Set part of block to nulls

volb [volb$a_vfmt_specific] = .dosv;                    ! Store the address of the extension

$block_init (.dosv, dos11);                             ! Set the block type and size fields

$queue_initialize (dosv [dos11$q_entry_header]);        ! Init the queue of directory entries
```

```
; 1495          1577  2 ! Rewind the magtape, since that is the only way we can know what the initial position is
; 1496          1578  2 !
; 1497          1579  3 IF NOT (status = exch$io_dos11_rewind (.volb))
; 1498          1580  2 THEN
; 1499          1581  2     RETURN .status;
; 1500          1582  2
; 1501          1583  2 ! Read the magtape label as a sanity check, we should have either a label or end of tape
; 1502          1584  2 !
; 1503          1585  3 IF NOT (sta'us = exch$io_dos11_read_label (.volb))
; 1504          1586  2 THEN
; 1505          1587  2     IF NOT .dosv [dos11$v_end_of_tape]
; 1506          1588  2     THEN
; 1507          1589  2         RETURN .status;
; 1508          1590  2
; 1509          1591  2 ! Rewind the magtape again, since the beginning is a good place to start
; 1510          1592  2 !
; 1511          1593  3 IF NOT (status = exch$io_dos11_rewind (.volb))
; 1512          1594  2 THEN
; 1513          1595  2     RETURN .status;
; 1514          1596  2
; 1515          1597  2 ! Set the volume type field
; 1516          1598  2 !
; 1517          1599  2 CH$MOVE (6, UPLIT BYTE ('DOS-11'), volb [volb$t_vol_type]);
; 1518          1600  2 volb [volb$l_vol_type_len] = 6;
; 1519          1601  2
; 1520          1602  2 RETURN 1;
; 1521          1603  1 END;


                                                                .PSECT  EXCH$DOS11_PLIT,NOWRT,2

                          31  31  2D  53  4F  44  00180 P.AAW:  .ASCII  \DOS-11\                                  ;


                                                                .PSECT  EXCH$DOS11_CODE,NOWRT,2

                                            01FC 00000          .ENTRY  EXCH$DOS11_MOUNT, Save R2,R3,R4,R5,R6,R7,R8 ; 1522
                        58 00000000G  EF  9E 00002          MOVAB   EXCH$IO_DOS11_REWIND, R8                 ;
                        57        04  AC  D0 00009          MOVL    VOLB, R7                                 ; 1564
                        52 041B00F3   8F  D0 0000D          MOVL    #68878579, R2                            ;
                        51      024E  8F  3C 00014          MOVZWL  #590, R1                                 ;
                        50            57  D0 00019          MOVL    R7, R0                                   ;
                           00000000G  EF  16 0001C          JSB     EXCH$UTIL_BLOCK_CHECK                    ;
                                      36  DD 00022          PUSHL   #54                                      ; 1569
              00000000G  EF           01  FB 00024          CALLS   #1, EXCH$UTIL_VM_ALLOCATE                ;
                         56           50  D0 0002B          MOVL    R0, DOSV                                 ;
     2A            00    6E           00  2C 0002E          MOVC5   #0, (SP), #0, #42, 12(DOSV)              ; 1570
                               0C     A6     00033
                    54   A7           56  D0 00035          MOVL    DOSV, 84(R7)                             ; 1572
                    08   A6           36  B0 00039          MOVW    #54, 8(DOSV)                             ; 1574
                    0A   A6           03  8E 0003D          MNEGB   #3, 10(DOSV)                             ;
                         50     12    A6  9E 00041          MOVAB   18(DOSV), R0                             ; 1576
                         60           50  D0 00045          MOVL    R0, (R0)                                 ;
                    04   A0           50  D0 00048          MOVL    R0, 4(R0)                                ;
                                      57  DD 0004C          PUSHL   R7                                       ; 1579
                         68      01  FB 0004E        CALLS   #1, EXCH$IO_DOS11_REWIND                   ;
                         27      50  E9 00051        BLBC    STATUS, 2$                                 ;
                         57      DD  00054           PUSHL   R7                                         ; 1585
          00000000G      EF      01  FB 00056        CALLS   #1, EXCH$IO_DOS11_READ_LABEL               ;
                         05      50  E8 0005D        BLBS    STATUS, 1$                                 ;
     16         0C       A6      03  E1 00060        BBC     #3, 12(DOSV), 2$                           ; 1587
                         57      DD  00065 1$:       PUSHL   R7                                         ; 1593
                         68      01  FB 00067        CALLS   #1, EXCH$IO_DOS11_REWIND                   ;
                         0E      50  E9 0006A        BLBC    STATUS, 2$                                 ;
   5D    A7     0000'    CF      06  28 0006D        MOVC3   #6, P.AAW, 93(R7)                          ; 1599
                59       A7      06  D0 00074        MOVL    #6, 89(R7)                                 ; 1600
                         50      01  D0 00078        MOVL    #1, R0                                     ; 1602
                                 04  0007B 2$:       RET                                                ; 1603

; Routine Size:  124 bytes,    Routine Base:  EXCH$DOS11_CODE + 0CBB
```

```
1 GLOBAL ROUTINE exch$dos11_next_entry (ctx : $ref_bblock, flags : $bblock) =     %SBTTL 'exch$dos11_next_entr
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Return the next entry from an DOS-11 directory
2 !
2 ! INPUT:
2 !
2 !       ctx   - pointer to an dos11ctx structure
2 !       flags - a structure of bits with the following meanings:
2 !
2 !                       flags [dosnxt$v_rewind]       - start at the beginning of the tape
2 !                       flags [dosnxt$v_count_blocks] - make sure that the block count is valid
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       anything we can find hanging off the context block
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       the context block is updated with context necessary for wildcard processing
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if next entry is now inside ctx, 0 if no more entries
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('dos11_next_entry> ');
2
2 LOCAL
2     ent : $ref_bblock,
2     status
2     ;
2
2 BIND
2     volb = ctx [dos11ctx$a_assoc_volb]  : $ref_bblock,
2     dosv = volb [volb$a_vfmt_specific]  : $ref_bblock
2     ;
2
2 $debug_print_fao ('entry - ctx !XL', .ctx);
2 $block_check (2, .ctx, dos11ctx, 608);
2 $block_check (2, .volb, volb, 609);
2 $block_check (2, .dosv, dos11, 610);
```

; F

```
2 ! If the rewind flag is set, then point the current entry at the queue header
2 !
2 IF .flags [dosnxt$v_rewind]
2 THEN
3     BEGIN
3     $debug_print_lit ('"rewinding"');
3     dosv [dos11$a_current_entry] = dosv [dos11$q_entry_header];
3     dosv [dos11$v_error_rewind]  = false;
2     END;
2
2 ! Move to the next entry, by moving to the entry after the current entry
2 !
2 $debug_print_fao ('entry before moving !XL', .dosv [dos11$a_current_entry]);
2 ent = ..dosv [dos11$a_current_entry];
2 $debug_print_fao ('entry after  moving !XL', .ent);
2
2 ! If the pointer points to the queue header, we have reached the end of the list
2 !
2 IF .ent EQL dosv [dos11$q_entry_header]
2 THEN
3     BEGIN
3
3     ! If there has been a rewind due to error, we want to fake the directory as being complete
3     !
3     IF .dosv [dos11$v_error_rewind]
3     THEN
3         dosv [dos11$v_directory] = true;
3
3     ! If the directory bit is set, then the list contains the entire directory and we are done
3     !
3     IF .dosv [dos11$v_directory]
3     THEN
4         BEGIN
4         dosv [dos11$a_current_entry] = .ent;    ! Save the pointer, effectively rewounds to beginning of tap
4         RETURN 0;                               ! We return 0, which signals that there is no next entry
4         END
4
4     ! Directory bit is not set, we need to read an additional label and create an entry
4     !
3     ELSE
4         BEGIN
4         LOCAL
4             last : $ref_bblock;                 ! pointer to last entry
4
4         ! Make sure that the tape is correctly positioned after the last known file
4         !
4         last = .dosv [dos11$a_entry_blink];                ! Get a pointer to the last entry in the queue
4         IF .last NEQ dosv [dos11$q_entry_header]           ! If there are no entries, skip the position
4         THEN
5             BEGIN
5
5             ! If the tape position has been lost, rewind the tape so that the position will be known again
5             !
5             IF NOT .dosv [dos11$v_position_valid]
5             THEN
6                 BEGIN
6                 $exch_signal (exch$_dos11_position);    ! This might take a while, warn 'em about it
```

```
6          IF NOT exch$io_dos11_rewind (.volb)
6          THEN
6              RETURN 0;
5          END;
5
5        ! Check to see if the current tape position is right after the last file
5        !
6        IF NOT (
7            (( last [dos11ent$w_file_number] + 1) EQL .dosv [dos11$l_current_file]) ! Is current one pas
6            AND
7            (.dosv [dos11$v_tape_mark])                                               ! And are we at the
6              )
5        THEN
6            BEGIN
6
6            ! Position to the last known entry, then skip to the next file
6            !
6            IF NOT exch$dos11_position_tape (.volb, .dosv [dos11$a_entry_blink])
6            THEN
6                RETURN 0;
6            IF NOT exch$io_dos11_skip_file (.volb, 1)
6            THEN
6                RETURN 0;
5            END;
4        END;
4
4    ! Read the label
4    !
4    status = exch$io_dos11_read_label (.volb);
4    IF NOT .status
4    THEN
5        BEGIN
5
5        ! If we found the end of the tape, record this fact and "rewind" the directory
5        !
5        IF .dosv [dos11$v_end_of_tape]      ! Normal double tape mark end
5          OR
5           .status EQL ss$_endoftape        ! Hit the physical end of tape in QIO
5        THEN
6            BEGIN
6            dosv [dos11$v_directory] = true;                                   ! Directory is complete
6            dosv [dos11$a_current_entry] = dosv [dos11$q_entry_header];        ! Rewind the directory
5            END;
5
5        RETURN 0;
4        END;
4
4    ! Create an entry and add it to the end of the list
4    !
4    ent = exch$util_vm_allocate_zeroed (dos11ent$k_length); ! Allocate the memory
4    $queue_insert_tail (.ent, dosv [dos11$q_entry_header]); ! Add it to the tail
4
4    ! Copy the label information to the entry
4    !
4    CH$MOVE (dos11$s_label_buf, dosv [dos11$t_label_buf], ent [dos11ent$l_filename_1]);
4
4    ! Fill in the file number for this entry
```

```
4           !
4           $trace_print_fao ('storing dosv file number !UL into entry', .dosv [dos11$l_current_file]);
4           ent [dos11ent$w_file_number] = .dosv [dos11$l_current_file];
4
4           ! If requested, fill in the size of the file
4           !
4           IF .flags [dosnxt$v_count_blocks]
4           THEN
5               BEGIN
5               LOCAL
5                   stat2;
5
5               ! Count the blocks in the file, and store the count
5               !
5               stat2 = exch$io_dos11_count_blocks (.volb);            ! Count tne blocks
5               ent [dos11ent$w_blocks] = .dosv [dos11$w_block_count];      ! Save the blocks even if an error,
5               ent [dos11ent$v_blocks_valid] = true;                      !  we don't want to try later (we wo
5                                                                         !  be able to do any better then)
5               $trace_print_fao ('count stat2 !XL', .stat2);
5               IF NOT .stat2
5               THEN
6                   BEGIN
6
6                   ! If we hit the physical end of tape, treat it like a double tape mark and flag the director
6                   ! as being complete
6                   !
6                   IF .stat2 EQL ss$_endoftape
6                   THEN
7                       BEGIN
7                       $trace_print_lit ('ss$_endoftape, setting end of directory');
7                       dosv [dos11$v_directory] = true;                          ! Directory is complete
7                       dosv [dos11$a_current_entry] = dosv [dos11$q_entry_header]; ! Rewind the directory
7                       END
6                   ELSE
6                       RETURN 0;                       ! Return failure
5                   END;
4               END;
3           END;
2       END;
```

```
; 1731            1810  2 !+
; 1732            1811  2 !
; 1733            1812  2 !  ENT now points to the directory entry
; 1734            1813  2 !
; 1735            1814  2 !-
; 1736            1815  2
; 1737            1816  2 ! If we need the block count, but we don't have it, we will need to locate the file and count the blocks
; 1738            1817  2 !
; 1739            1818  2 IF .flags [dosnxt$v_count_blocks]
; 1740            1819  2 THEN
; 1741            1820  2     IF NOT .ent [dos11ent$v_blocks_valid]
; 1742            1821  2     THEN
; 1743            1822  3         BEGIN
; 1744            1823  3
; 1745            1824  3         ! Position to the last known entry, then skip to the next file
; 1746            1825  3         !
; 1747            1826  3         IF NOT exch$dos11_position_tape (.volb, .ent)
; 1748            1827  3         THEN
; 1749            1828  3             RETURN 0;
; 1750            1829  3
; 1751            1830  3         ! Count the number of blocks to the next tape mark, value will be stored in dosv structure
; 1752            1831  3         !
; 1753            1832  3         IF NOT exch$io_dos11_count_blocks (.volb)
; 1754            1833  3         THEN
; 1755            1834  3             RETURN 0;
; 1756            1835  3
; 1757            1836  3         ! Save the values in the entry
; 1758            1837  3         !
; 1759            1838  3         ent [dos11ent$w_blocks] = .dosv [dos11$w_block_count];
; 1760            1839  3         ent [dos11ent$v_blocks_valid] = true;
; 1761            1840  3
; 1762            1841  2         END;
; 1763            1842  2
; 1764            1843  2 ! Copy the entry to the context block
; 1765            1844  2 !
; 1766          L 1845  2 $logic_check (3, ($BYTEOFFSET(dos11ctx$t_entry) EQL $BYTEOFFSET(dos11ctx$l_filename_1)), 252);
; %PRINT:         assumption 252 verified during compilation
; 1767            1846  2 CH$MOVE (dos11ctx$s_entry, ent [dos11ent$l_filename_1], ctx [dos11ctx$t_entry]);
; 1768            1847  2
; 1769            1848  2 ! Expand the directory entry filename information into the context block
; 1770            1849  2 !
; 1771            1850  2 exch$dos11_expand_filename (.ctx);
; 1772            1851  2
; 1773            1852  2 ! Make this the current entry
; 1774            1853  2 !
; 1775            1854  2 dosv [dos11$a_current_entry] = .ent;
; 1776            1855  2
; 1777          P 1856  2 $debug_print_fao ('Found "!AF!AF", File number !UL, entry !XL',
; 1778          P 1857  2                 .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
; 1779          P 1858  2                 .ctx [dos11ctx$l_exp_fullname_len] + dos11ctx$s_exp_directory, ctx [dos11ctx$t_exp_directory
; 1780            1859  2                 .ctx [dos11ctx$w_file_number], .ent);
; 1781            1860  2
; 1782            1861  2 RETURN 1;
; 1783            1862  1 END;
```

```
                              0FFC 00000         .ENTRY  EXCH$DOS11_NEXT_ENTRY, Save R2,R3,R4,R5,R6,-; 1604
                                                         R7,R8,R9,R10,R11                            ;
               5B 00000000G    EF 9E 00002       MOVAB   EXCH$IO_DOS11_COUNT_BLOCKS, R11             ;
               5A 00000000G    EF 9E 00009       MOVAB   EXCH$UTIL_BLOCK_CHECK, R10                  ;
58         04  AC              14 C1 00010       ADDL3   #20, CTX, R8                                ; 1649
59             68  00000054    8F C1 00015       ADDL3   #84, (R8), R9                               ; 1650
               52  008A00FC    8F D0 0001D       MOVL    #9044220, R2                                ; 1654
               51      0260    8F 3C 00024       MOVZWL  #608, R1                                    ;
               50        04    AC D0 00029       MOVL    CTX, R0                                     ;
                               6A 16 0002D       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
               52  041B00F3    8F D0 0002F       MOVL    #68878579, R2                               ; 1655
               51      0261    8F 3C 00036       MOVZWL  #609, R1                                    ;
               50              68 D0 0003B       MOVL    (R8), R0                                    ;
                               6A 16 0003E       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
               53              69 D0 00040       MOVL    (R9), R3                                    ; 1656
               52  003600FD    8F D0 00043       MOVL    #3539197, R2                                ;
               51      0262    8F 3C 0004A       MOVZWL  #610, R1                                    ;
               50              53 D0 0004F       MOVL    R3, R0                                      ;
                               6A 16 00052       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
               0A        08    AC E9 00054       BLBC    FLAGS, 1$                                   ; 1659
           1A  A3        12    A3 9E 00058       MOVAB   18(R3), 26(R3)                              ; 1663
           0C  A3        40    8F 8A 0005D       BICB2   #64, 12(R3)                                 ; 1664
               57        1A    B3 D0 00062 1$:   MOVL    @26(R3), ENT                                ; 1670
               55        12    A3 9E 00066       MOVAB   18(R3), R5                                  ; 1675
               55              57 D1 0006A       CMPL    ENT, R5                                     ;
                               03 13 0006D       BEQL    2$                                          ;
                             00CE 31 0006F       BRW     12$                                         ;
               52        0C    A3 9E 00072 2$:   MOVAB   12(R3), R2                                  ; 1681
03             62              06 E1 00076       BBC     #6, (R2), 3$                                ;
               62              10 88 0007A       BISB2   #16, (R2)                                   ; 1683
06             62              04 E1 0007D 3$:   BBC     #4, (R2), 4$                                ; 1687
           1A  A3              57 D0 00081       MOVL    ENT, 26(R3)                                 ; 1690
                               70 11 00085       BRB     9$                                          ; 1691
               54        16    A3 D0 00087 4$:   MOVL    22(R3), LAST                                ; 1703
               55              54 D1 0008B       CMPL    LAST, R5                                    ; 1704
                               47 13 0008E       BEQL    7$                                          ;
               19              62 E8 00090       BLBS    (R2), 5$                                    ; 1710
                  00000000G    8F DD 00093       PUSHL   #EXCH$_DOS11_POSITION                       ; 1713
    00000000G  00              01 FB 00099       CALLS   #1, LIB$SIGNAL                              ;
                               68 DD 000A0       PUSHL   (R8)                                        ; 1714
    00000000G  EF              01 FB 000A2       CALLS   #1, EXCH$IO_DOS11_REWIND                    ;
               4B              50 E9 000A9       BLBC    R0, 9$                                      ;
               50        18    A4 3C 000AC 5$:   MOVZWL  24(LAST), R0                                ; 1722
                               50 D6 000B0       INCL    R0                                          ;
           0E  A3              50 D1 000B2       CMPL    R0, 14(R3)                                  ;
                               04 12 000B6       BNEQ    6$                                          ;
1B             62              02 E0 000B8       BBS     #2, (R2), 7$                                ; 1724
                         16    A3 DD 000BC 6$:   PUSHL   22(R3)                                      ; 1731
                               68 DD 000BF       PUSHL   (R8)                                        ;
        0000V  CF              02 FB 000C1       CALLS   #2, EXCH$DOS11_POSITION_TAPE                ;
               2E              50 E9 000C6       BLBC    R0, 9$                                      ;
                               01 DD 000C9       PUSHL   #1                                          ; 1734
                               68 DD 000CB       PUSHL   (R8)                                        ;
    00000000G  EF              02 FB 000CD       CALLS   #2, EXCH$IO_DOS11_SKIP_FILE                 ;
               7B              50 E9 000D4       BLBC    R0, 13$                                     ;
```

```
                                                                C  3
EXCH$DOS11        dos11 file and directory routines             16-Sep-1984 00:52:08    VAX-11 Bliss-32 V4.0-742            Page  57
V04-000           exch$dos11_next_entry (ctx, flags)            14-Sep-1984 12:29:04    [EXCHNG.SRC]EXCDOS11.B32;1              (18)

                                           68  DD 000D7 7$:    PUSHL   (R8)                                    ; 1742
                 00000000G  EF             01  FB 000D9        CALLS   #1, EXCH$IO_DOS11_READ_LABEL            ;
                            17             50  E8 000E0        BLBS    STATUS, 10$                             ; 1743
            09              62             03  E0 000E3        BBS     #3, (R2), 8$                            ; 1749
                 00000878   8F             50  D1 000E7        CMPL    STATUS, #2168                           ; 1751
                                           45  12 000EE        BNEQ    11$                                     ;
                            62             10  88 000F0 8$:    BISB2   #16, (R2)                               ; 1754
                       1A   A3             55  D0 000F3        MOVL    R5, 26(R3)                              ; 1755
                                         008B  31 000F7 9$:    BRW     15$                                     ; 1758
                                           1C  DD 000FA 10$:   PUSHL   #28                                     ; 1763
                 00000000G  EF             01  FB 000FC        CALLS   #1, EXCH$UTIL_VM_ALLOCATE_ZEROED        ;
                            57             50  D0 00103        MOVL    R0, ENT                                 ;
                       04   B5             67  0E 00106        INSQUE  (ENT), @4(R5)                           ; 1764
                            56             69  D0 0010A        MOVL    (R9), R6                                ; 1768
      08    A7         26   A6             0E  28 0010D        MOVC3   #14, 38(R6), 8(ENT)                     ;
                       18   A7       0E    A6  B0 00113        MOVW    14(R6), 24(ENT)                         ; 1773
            4C         08   AC             01  E1 00118        BBC     #1, FLAGS, 14$                          ; 1777
                                           68  DD 0011D        PUSHL   (R8)                                    ; 1785
                            6B             01  FB 0011F        CALLS   #1, EXCH$IO_DOS11_COUNT_BLOCKS          ;
                       16   A7       34    A6  B0 00122        MOVW    52(R6), 22(ENT)                         ; 1786
                       1A   A7             01  88 00127        BISB2   #1, 26(ENT)                             ; 1787
                            12             50  E8 0012B        BLBS    STAT2, 12$                              ; 1790
                 00000878   8F             50  D1 0012E        CMPL    STAT2, #2168                            ; 1797
                                           4E  12 00135 11$:   BNEQ    15$                                     ;
                       0C   A6             10  88 00137        BISB2   #16, 12(R6)                             ; 1801
                       1A   A6       12    A6  9E 0013B        MOVAB   18(R6), 26(R6)                          ; 1802
            24         08   AC             01  E1 00140 12$:   BBC     #1, FLAGS, 14$                          ; 1818
                       20            1A    A7  E8 00145        BLBS    26(ENT), 14$                            ; 1820
                                           57  DD 00149        PUSHL   ENT                                     ; 1826
                                           68  DD 0014B        PUSHL   (R8)                                    ;
                     0000V  CF             02  FB 0014D        CALLS   #2, EXCH$DOS11_POSITION_TAPE            ;
                            30             50  E9 00152 13$:   BLBC    R0, 15$                                 ;
                                           68  DD 00155        PUSHL   (R8)                                    ; 1832
                            6B             01  FB 00157        CALLS   #1, EXCH$IO_DOS11_COUNT_BLOCKS          ;
                            28             50  E9 0015A        BLBC    R0, 15$                                 ;
                            50             69  D0 0015D        MOVL    (R9), R0                                ; 1838
                       16   A7       34    A0  B0 00160        MOVW    52(R0), 22(ENT)                         ;
                       1A   A7             01  88 00165        BISB2   #1, 26(ENT)                             ; 1839
                            58       04    AC  D0 00169 14$:   MOVL    CTX, R8                                 ; 1846
      3C    A8         08   A7             14  28 0016D        MOVC3   #20, 8(ENT), 60(R8)                     ;
                                           58  DD 00173        PUSHL   R8                                      ; 1850
                      FB6E  CF             01  FB 00175        CALLS   #1, EXCH$DOS11_EXPAND_FILENAME          ;
                            50             69  D0 0017A        MOVL    (R9), R0                                ; 1854
                       1A   A0             57  D0 0017D        MOVL    ENT, 26(R0)                             ;
                            50             01  D0 00181        MOVL    #1, R0                                  ; 1861
                                               04 00184        RET                                             ;
                                           50  D4 00185 15$:   CLRL    R0                                      ; 1862
                                               04 00187        RET                                             ;

; Routine Size:  392 bytes,    Routine Base:  EXCH$DOS11_CODE + 0D37
```

```
1 GLOBAL ROUTINE exch$dos11_open_file =   %SBTTL 'exch$dos11_open_file'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform DOS-11 volume specific open processing
2 !
2 ! INPUT:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       copy work area
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       filb - receive info pertaining to the open file
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to open a file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('dos11_open_file> ');
2
2 LOCAL
2     out_filb    : $ref_bblock,
2     status
2     ;
2
2 BIND
2     copy     = exch$a_gbl [excg$a_copy_work]    : $ref_bblock,
2     inp_filb = copy [copy$a_inp_filb]           : $ref_bblock,
2     ctx      = inp_filb [filb$a_context]        : $ref_bblock,
2     namb     = inp_filb [filb$a_assoc_namb]     : $ref_bblock,
2     nam_nam  = namb [namb$q_name]               : $desc_block,          ! Get name and type components from
2     nam_typ  = namb [namb$q_type]               : $desc_block,          !  the namb
2     volb     = inp_filb [filb$a_assoc_volb]     : $ref_bblock,
2     inp_namb = copy [copy$a_inp_namb]           : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .inp_filb, filb, 591);
2 $block_check (2, .namb, namb, 592);
2 $block_check (2, .volb, volb, 593);
2
```

```
; 1842          1920  2 ! Make sure that the output filb points to a valid filb
; 1843          1921  2 !
; 1844          1922  2 IF (out_filb = .copy [copy$a_out_filb]) EQL 0
; 1845          1923  2 THEN
; 1846          1924  2     out_filb = .inp_filb;
; 1847          1925  2 $block_check (2, .out_filb, filb, 473);
```

```
2 ! If the context pointer is null, then allocate and initialize it.
2 !
2 IF .ctx EQL 0
2 THEN
2     ctx = exch$util_dos11ctx_allocate (.volb, .inp_filb)        ! Create an dos11 context block
2
2 ! If non-null, we are doing a subsequent lookup in a wildcard search
2 !
2 ELSE
3     BEGIN
3     LITERAL
3         wilds = (namb$m_wild_name OR namb$m_wild_type OR namb$m_wild_group OR namb$m_wild_member);
3
3     ! If not wildcard, then we must be done
3     !
3     IF (.namb [namb$l_nameflags] AND wilds) EQL 0
3     THEN
3         RETURN false;
3
3     $block_check (2, .ctx, dos11ctx, 594);
3
2     END;
2
2 ! Make sure that we haven't crossed signals someplace
2 !
2 $logic_check (4, (.ctx [dos11ctx$a_assoc_filb] EQL .inp_filb), 260);
2 $logic_check (4, (.ctx [dos11ctx$a_assoc_volb] EQL .volb), 261);
2
2 ! We assume that the file name and type fields in the namb are adjacent.  If they aren't, the next call to
2 ! exch$dos11_find_file will choke.
2 !
2 $logic_check (3, (.nam_typ [dsc$a_pointer] EQL (.nam_nam [dsc$w_length] + .nam_nam [dsc$a_pointer])), 262);
```

```
3 IF (
3     IF .copy [copy$v_reopen_input]              ! If we are retrying, then reuse the context block
3     THEN
3         1
3     ELSE                                        ! Otherwise skip to the next file
3         exch$dos11_find_file (.ctx, .nam_nam [dsc$a_pointer], .nam_nam [dsc$w_length] + .nam_typ [dsc$w_leng
3     )
2 THEN
3     BEGIN
3
3     ! Create the result name string in the filb
3     !
3     inp_filb [filb$l_result_name_len] = .volb [volb$l_vol_ident_len]     ! Length of volume ident
3                                 + .ctx [dos11ctx$l_exp_fullname_len];    !  plus dos fullname
3     $logic_check (3, (.inp_filb [filb$l_result_name_len] LEQU filb$s_result_name), 263);
3     CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],      ! Volume name
3             .ctx [dos11ctx$l_exp_fullname_len], ctx [dos11ctx$t_exp_fullname],
3             0, filb$s_result_name, inp_filb [filb$t_result_name]);
3
3     $trace_print_fao ('Found "!AF"', .inp_filb [filb$l_result_name_len], inp_filb [filb$t_result_name]);
3
3     ! Physically position the tape to this file, since the routine expects the address of a dos11ent
3     ! structure, we alter the address so that the offsets of the label fields will match
3     !
3     IF NOT exch$dos11_position_tape (.volb, (ctx [dos11ctx$t_entry] - $byteoffset (dos11ent$l_filename_1)))
3     THEN
3         RETURN 0;
3
3     ! Define a record stream for this file
3     !
3     ctx [dos11ctx$l_cur_byte]     = 0;                                   ! Context is the first byte in
3     ctx [dos11ctx$l_cur_block]    = 0;                                   !  the first block of the file
3     ctx [dos11ctx$l_eof_block]    = -1;                                  ! Pick a huge number
3     inp_filb [filb$l_block_count] = .ctx [dos11ctx$w_blocks];            ! Put the size in the filb (if we kn
3     inp_filb [filb$a_record]      = 0;                                   ! No valid record or length
3     inp_filb [filb$l_record_len]  = 0;
3
3     ! Make sure that the record format in the filb is correct
3     !
3     exch$cmd_fetch_recfmt_implied (.inp_filb, ctx [dos11ctx$t_exp_type]);
3
3     ! For DOS-11 we can treat block transfer mode as fixed 512
3     !
3     IF .out_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
3       OR
3        .inp_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
3     THEN
4         BEGIN
4         inp_filb [filb$b_rec_format]  = filb$k_rfmt_fixed;
4         inp_filb [filb$l_fixed_len]   = 512;
3         END;
3
3     ! Clear all the flags except the ones we want by writing the masks into the longword
3     !
3     ctx [dos11ctx$l_flags] = dos11ctx$m_stream_active;  ! A record stream is currently active
3     inp_filb [filb$v_files_found] = true;              ! One or more files have been opened
3
```

```
; 1939          2015  3      ! Set up the i/o and record buffer
; 1940          2016  3      !
; 1941          2017  3      IF .ctx [dos11ctx$a_buffer] EQL 0           ! If doing wildcards buffer might be there
; 1942          2018  3      THEN
; 1943          2019  3          ctx [dos11ctx$a_buffer] = exch$util_vm_allocate (ctx$k_buffer_length);
; 1944          2020  3
; 1945          2021  3      ! Set the block pointers so that we will advance the buffer on the first get
; 1946          2022  3      !
; 1947          2023  3      ctx [dos11ctx$l_buf_base_block] = 0;
; 1948          2024  3      ctx [dos11ctx$l_buf_high_block] = -1;
; 1949          2025  3
; 1950          2026  3      ! Save the addresses of our routines for this volume and record format.
; 1951          2027  3      !
; 1952          2028  3      inp_filb [filb$a_close_routine] = exch$dos11_close_file;
; 1953          2029  3      inp_filb [filb$a_put_routine] = 0;
; 1954          2030  3      inp_filb [filb$a_get_routine] = exch$pdp_get;
; 1955          2031  3
; 1956          2032  3      RETURN true;                                ! True means its open
; 1957          2033  2      END;
; 1958          2034  2
; 1959          2035  2  ! If no files were found, then scream and shout
; 1960          2036  2  !
; 1961          2037  2  IF NOT .inp_filb [filb$v_files_found]
; 1962          2038  2  THEN
; 1963          2039  3      BEGIN
; 1964          2040  3      REGISTER
; 1965          2041  3          fao_desc = 0 : $ref_bblock;
; 1966          2042  3
; 1967          2043  3      ! Concatenate the volume name to the file name and type fields
; 1968          2044  3      !
; 1969          2045  3      fao_desc = exch$util_fao_buffer (%ASCID '!AF!AF', .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
; 1970          2046  3                      .nam_nam [dsc$w_length] + .nam_typ [dsc$w_length], .nam_nam [dsc$a_pointer]);
; 1971          2047  3      $exch_signal (exch$_filenotfound, 1, .fao_desc);
; 1972          2048  3      RETURN rms$_fnf;
; 1973          2049  3
; 1974          2050  2      END;
; 1975          2051  2
; 1976          2052  2  ! Normal exit, return a 0
; 1977          2053  2  !
; 1978          2054  2  RETURN 0;
; 1979          2055  1  END;


                                                                   .PSECT  EXCH$DOS11_PLIT,NOWRT,2

                                                00186              .BLKB   2
                      00  00  46  41  21  46  41  21  00188 P.AAY:  .ASCII  \!AF!AF\<0><0>                                        ;
                                       010E0006  00190 P.AAX:  .LONG   17694726                                              ;
                                       00000000' 00194         .ADDRESS P.AAY                                                ;

                                                                   .EXTRN  EXCH$_FILENOTFOUND

                                                                   .PSECT  EXCH$DOS11_CODE,NOWRT,2

                                            0FFC 00000             .ENTRY  EXCH$DOS11_OPEN_FILE, Save R2,R3,R4,R5,R6,-   ; 1863
                                                                           R7,R8,R9,R10,R11                              ;
```

```
                               5E            04  C2 00002        SUBL2    #4, SP                                   :
                 50 00000000G  EF            04  C1 00005        ADDL3    #4, EXCH$A_GBL, R0                       : 1904
                               53            60  D0 0000D        MOVL     (R0), R3                                 : 1905
                               57        3C  A3  D0 00010        MOVL     60(R3), R7                               : 1906
                               59        18  A7  D0 00014        MOVL     24(R7), R9                               : 1908
                                         50  A9  9F 00018        PUSHAB   80(R9)                                   :
                               52 035B00FA  8F  D0 0001B        MOVL     #56295674, R2                            : 1916
                               51      024F  8F  3C 00022        MOVZWL   #591, R1                                 :
                               50            57  D0 00027        MOVL     R7, R0                                   :
                                  00000000G  EF  16 0002A        JSB      EXCH$UTIL_BLOCK_CHECK                    :
                               52 010A00F7  8F  D0 00030        MOVL     #17432823, R2                            : 1917
                               51      0250  8F  3C 00037        MOVZWL   #592, R1                                 :
                               50            59  D0 0003C        MOVL     R9, R0                                   :
                                  00000000G  EF  16 0003F        JSB      EXCH$UTIL_BLOCK_CHECK                    :
                               58        1C  A7  D0 00045        MOVL     28(R7), R8                               : 1918
                               52 041B00F3  8F  D0 00049        MOVL     #68878579, R2                            :
                               51      0251  3F  3C 00050        MOVZWL   #593, R1                                 :
                               50            58  D0 00055        MOVL     R8, R0                                   :
                                  00000000G  EF  16 00058        JSB      EXCH$UTIL_BLOCK_CHECK                    :
                               5B        44  A3  D0 0005E        MOVL     68(R3), OUT_FILB                         : 1922
                                             03  12 00062        BNEQ     1$                                       :
                               5B            57  D0 00064        MOVL     R7, OUT_FILB                             : 1924
                               52 035B00FA  8F  D0 00067 1$:    MOVL     #56295674, R2                            : 1925
                               51      01D9  8F  3C 0006E        MOVZWL   #473, R1                                 :
                               50            5B  D0 00073        MOVL     OUT_FILB, R0                             :
                                  00000000G  EF  16 00076        JSB      EXCH$UTIL_BLOCK_CHECK                    :
                                         20  A7  D5 0007C        TSTL     32(R7)                                   : 1928
                                             11  12 0007F        BNEQ     2$                                       :
                                             57  DD 00081        PUSHL    R7                                       : 1930
                                             58  DD 00083        PUSHL    R8                                       :
                    00000000G  EF            02  FB 00085        CALLS    #2, EXCH$UTIL_DOS11CTX_ALLOCATE          :
                           20  A7            50  D0 0008C        MOVL     R0, 32(R7)                               :
                                             1F  11 00090        BRB      5$                                       :
                               36        6C  A9  93 00092 2$:    BITB     108(R9), #54                             : 1941
                                             03  12 00096        BNEQ     4$                                       :
                                           0127  31 00098 3$:    BRW      12$                                      :
                               52 008A00FC  8F  D0 0009B 4$:    MOVL     #9044220, R2                             : 1945
                               51      0252  8F  3C 000A2        MOVZWL   #594, R1                                 :
                               50        20  A7  D0 000A7        MOVL     32(R7), R0                               :
                                  00000000G  EF  16 000AB        JSB      EXCH$UTIL_BLOCK_CHECK                    :
                 20        34  A3            02  E0 000B1 5$:    BBS      #2, 52(R3), 6$                           : 1959
                               50        00  BE  3C 000B6        MOVZWL   @0(SP), R0                               : 1963
                               51        58  A9  3C 000BA        MOVZWL   88(R9), R1                               :
                                           6140  9F 000BE        PUSHAB   (R1)[R0]                                 :
                 52        04  AE            04  C1 000C1        ADDL3    #4, 4(SP), R2                            :
                                             62  DD 000C6        PUSHL    (R2)                                     :
                                         20  A7  DD 000C8        PUSHL    32(R7)                                   :
                         FC3D  CF            03  FB 000CB        CALLS    #3, EXCH$DOS11_FIND_FILE                 :
                               03            50  E8 000D0        BLBS     R0, 6$                                   :
                                           00AC  31 000D3        BRW      11$                                      :
                               56        20  A7  D0 000D6 6$:    MOVL     32(R7), R6                               : 1971
            3A   A7        65  A8        50  A6  C1 000DA        ADDL3    80(R6), 101(R8), 58(R7)                  :
                           04  AE      0100  8F  3C 000E1        MOVZWL   #256, 4(SP)                              : 1974
                               5A        5A  A7  9E 000E7        MOVAB    90(R7), R10                              : 1975
   04    AE      00        69  A8        65  A8  2C 000EB        MOVC5    101(R8), 105(R8), #0, 4(SP), (R10)       :
                                             6A     000F3                                                          :
                                             12  18 000F4        BGEQ     7$                                       :
```

```
                                5A        65     A8  C0 000F6          ADDL2    101(R8), R10                                     ;
                         04     AE        65     A8  C2 000FA          SUBL2    101(R8), 4(SP)                                   ;
   04   AE          00   66     A6        50     A6  2C 000FF          MOVC5    80(R6), 102(R6), #0, 4(SP), (R10)                ;
                                                 6A     00107                                                                    ;
                                          34     A6  9F 00108 7$:      PUSHAB   52(R6)                                           ; 1982
                                                 58  DD 0010B          PUSHL    R8                                               ;
                      0000V     CF               02  FB 0010D          CALLS    #2, EXCH$DOS11_POSITION_TAPE                     ;
                                83               50  E9 00112          BLBC     R0, 3$                                           ;
                                          24     A6  D4 00115          CLRL     36(R6)                                           ; 1988
                                          1C     A6  D4 00118          CLRL     28(R6)                                           ; 1989
                         20     A6               01  CE 0011B          MNEGL    #1, 32(R6)                                       ; 1990
                         3E     A7        4A     A6  3C 0011F          MOVZWL   74(R6), 62(R7)                                   ; 1991
                                          42     A7  7C 00124          CLRQ     66(R7)                                           ; 1993
                                          7C     A6  9F 00127          PUSHAB   124(R6)                                          ; 1997
                                                 57  DD 0012A          PUSHL    R7                                               ;
                  00000000G     EF               02  FB 0012C          CALLS    #2, EXCH$CMD_FETCH_RECFMT_IMPLIED                ;
                                01        29     AB  91 00133          CMPB     41(OUT_FILB), #1                                 ; 2001
                                                 06  13 00137          BEQL     8$                                               ;
                                01        29     A7  91 00139          CMPB     41(R7), #1                                       ; 2003
                                                 0A  12 0013D          BNEQ     9$                                               ;
                         28     A7               02  90 0013F 8$:      MOVB     #2, 40(R7)                                       ; 2006
                         35     A7      0200     8F  3C 00143          MOVZWL   #512, 53(R7)                                     ; 2007
                         28     A6               01  D0 00149 9$:      MOVL     #1, 40(R6)                                       ; 2012
                         2B     A7               08  88 0014D          BISB2    #8, 43(R7)                                       ; 2013
                                          18     A6  D5 00151          TSTL     24(R6)                                           ; 2017
                                                 10  12 00154          BNEQ     10$                                              ;
                                7E      1800     8F  3C 00156          MOVZWL   #6144, -(SP)                                     ; 2019
                  00000000G     EF               01  FB 0015B          CALLS    #1, EXCH$UTIL_VM_ALLOCATE                        ;
                         18     A6               50  D0 00162          MOVL     R0, 24(R6)                                       ;
                                          2C     A6  D4 00166 10$:     CLRL     44(R6)                                           ; 2023
                         30     A6               01  CE 00169          MNEGL    #1, 48(R6)                                       ; 2024
                         4A     A7      EFD0     CF  9E 0016D          MOVAB    EXCH$DOS11_CLOSE_FILE, 74(R7)                    ; 2028
                                          56     A7  D4 00173          CLRL     86(R7)                                           ; 2029
                         52     A7 00000000G     EF  9E 00176          MOVAB    EXCH$PDP_GET, 82(R7)                             ; 2030
                                50               01  D0 0017E          MOVL     #1, R0                                           ; 2032
                                                 04     00181          RET                                                       ;
                    3B   2B     A7               03  E0 00182 11$:     BBS      #3, 43(R7), 12$                                  ; 2037
                    52          6E               04  C1 00187          ADDL3    #4, (SP), R2                                     ; 2046
                                                 62  DD 0018B          PUSHL    (R2)                                             ;
                                50        04     BE  3C 0018D          MOVZWL   @4(SP), R0                                       ;
                                51        58     A9  3C 00191          MOVZWL   88(R9), R1                                       ;
                                        6140     9F     00195          PUSHAB   (R1)[R0]                                         ;
                                          69     A8  9F 00198          PUSHAB   105(R8)                                          ; 2045
                                          65     A8  DD 0019B          PUSHL    101(R8)                                          ;
                                       0000'     CF  9F 0019E          PUSHAB   P.AAX                                            ;
                  00000000G     EF               05  FB 001A2          CALLS    #5, EXCH$UTIL_FAO_BUFFER                         ;
                                                 50  DD 001A9          PUSHL    FAO_DESC                                         ; 2047
                                                 01  DD 001AB          PUSHL    #1                                               ;
                                   00000000G     8F  DD 001AD          PUSHL    #EXCH$_FILENOTFOUND                              ;
                  00000000G     00               03  FB 001B3          CALLS    #3, LIB$SIGNAL                                   ;
                                50 00018292      8F  D0 001BA          MOVL     #98962, R0                                       ; 2048
                                                 04     001C1          RET                                                       ;
                                                 50  D4 001C2 12$:     CLRL     R0                                               ; 2055
                                                 04     001C4          RET                                                       ;

; Routine Size:  453 bytes,    Routine Base:  EXCH$DOS11_CODE + 0EBF
```

```
  1 GLOBAL ROUTINE exch$dos11_position_tape (volb : $ref_bblock, ent : $ref_bblock) =      %SBTTL 'exch$dos11_p
  2 BEGIN
  2 !++
  2 !
  2 ! FUNCTIONAL DESCRIPTION:
  2 !
  2 !       Position the tape to the first data block of the file described by ent.
  2 !
  2 ! INPUT:
  2 !
  2 !       volb    - the volume block for the device
  2 !       ent     - a pointer to the entry in the directory queue
  2 !
  2 ! IMPLICIT INPUTS:
  2 !
  2 !       anything we can find hanging off the volb
  2 !
  2 ! OUTPUTS:
  2 !
  2 !       none
  2 !
  2 ! IMPLICIT OUTPUTS:
  2 !
  2 !       the volb (and dos11 extension) are updated
  2 !
  2 ! ROUTINE VALUE:
  2 !
  2 !       true if able to position, 0 if not
  2 !
  2 ! SIDE EFFECTS:
  2 !
  2 !       the tape may be moved
  2 !--
  2
  2 $dbgtrc_prefix ('dos11_position_tape> ');
  2
  2 LOCAL
  2     skip,
  2     status
  2     ;
  2
  2 BIND
  2     dosv = volb [volb$a_vfmt_specific]  : $ref_bblock
  2     ;
  2
  2 $debug_print_fao ('entry');
  2 $block_check (2, .volb, volb, 615);
  2 $block_check (2, .dosv, dos11, 616);
  2
P 2 $trace_print_fao ('entry file number !UL, name_1,2,type !XL,!XW,!XW',
P 2                 .ent [dos11ent$w_file_number], .ent [dos11ent$l_filename_1],
  2                 .ent [dos11ent$w_filename_2], .ent [dos11ent$w_filetype]);
  2 $trace_print_lit ('dosv at entry');
  2 $check_call (4, exch$dbg_dump_dosv, .volb, (dmpdsv$m_status OR dmpdsv$m_position));
  2
  2 ! If the tape position has been lost, rewind the tape so that the position will be known again
  2 !
```

```
2 IF NOT .dosv [dos11$v_position_valid]
2 THEN
3     BEGIN
3     $exch_signal (exch$_dos11_position);        ! This might take a while, warn 'em about it
3     IF NOT exch$io_dos11_rewind (.volb)
3     THEN
3         RETURN 0;
2     END;
2
2 ! If the desired file is the first file, then rewind.  Otherwise, use the skipfile routine to go there
2 !
2 IF .ent [dos11ent$w_file_number] EQL 0
2 THEN
3     BEGIN
3
3     ! Rewind the tape so that we will know the position
3     !
3     IF NOT exch$io_dos11_rewind (.volb)
3     THEN
3         RETURN 0;
2     END;
2
2 ! Determine how far we are from where we should be
2 !
2 skip = .ent [dos11ent$w_file_number] - .dosv [dos11$l_current_file];
2 $trace_print_fao ('skip file count !SL', .skip);
2
2 ! Move the tape that number of files, routine simply returns if skip count is zero
2 !
2 IF NOT exch$io_dos11_skip_file (.volb, .skip)
2 THEN
2     RETURN 0;
2
2 $trace_print_lit ('after skip ->');
2 $check_call (4, exch$dbg_dump_dosv, .volb, (dmpdsv$m_status OR dmpdsv$m_position OR dmpdsv$m_entries));
2
2 IF .dosv [dos11$v_end_of_tape]
2 THEN
3     BEGIN
3     $trace_print_lit ('forcing -1 skip ****************************** dosv file number !UL', .dosv [dos11$l_c
3     dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;     ! Adjust the file number for the ski
3     IF NOT exch$io_dos11_skip_file (.volb, -1)
3     THEN
3         RETURN 0;
3     $trace_print_lit ('dosv file number !UL after skip', .dosv [dos11$l_current_file]);
2     END;
2
2 ! Read the label if we are at a tape mark
2 !
2 IF .dosv [dos11$v_tape_mark]
2 THEN
3     BEGIN
3     status = exch$io_dos11_read_label (.volb);
3     IF NOT .status
3     THEN
3         RETURN 0;
2     END;
```

```
; 2095         2170  2
; 2096         2171  2 ! Make sure that the correct file has been read
; 2097         2172  2 !
; 2098         2173  2 IF CH$NEQ (dos11$s_label_buf, dosv [dos11$t_label_buf], dos11$s_label_buf, ent [dos11ent$l_filename_1], %C '
; 2099         2174  2 THEN
; 2100         2175  3     BEGIN
; 2101         2176  3     BIND
; 2102         2177  3         got = dosv [dos11$t_label_buf] : VECTOR [,BYTE],
; 2103         2178  3         want = ent [dos11ent$l_filename_1] : VECTOR [,BYTE];
; 2104         2179  3
; 2105       P 2180  3     $trace_print_fao ('wanted  !14(3XB)', .want [0], .want [1], .want [2], .want [3], .want [4], .want [5],
; 2106         2181  3                 .want [6], .want [7], .want [8], .want [9], .want [10], .want [11], .want [12], .want [13]);
; 2107         2182  3
; 2108       P 2183  3     $trace_print_fao ('got     !14(3XB)', .got [0], .got [1], .got [2], .got [3], .got [4], .got [5],
; 2109         2184  3                 .got [6], .got [7], .got [8], .got [9], .got [10], .got [11], .got [12], .got [13]);
; 2110         2185  3
; 2111         2186  3     ! Tell them we have to rewind
; 2112         2187  3     !
; 2113         2188  3     $exch_signal (exch$_dos11_position);
; 2114         2189  3
; 2115         2190  3     ! Rewind the tape so that we will know the position
; 2116         2191  3     !
; 2117         2192  3     IF NOT exch$io_dos11_rewind (.volb)
; 2118         2193  3     THEN
; 2119         2194  3         RETURN 0;
; 2120         2195  3
; 2121         2196  3     ! Call ourselves again now that the position is known
; 2122         2197  3     !
; 2123         2198  3     RETURN exch$dos11_position_tape (.volb, .ent);
; 2124         2199  2     END;
; 2125         2200  2
; 2126         2201  2 $trace_print_lit ('exiting with position');
; 2127         2202  2 $check_call (4, exch$dbg_dump_dosv, .volb, (dmpdsv$m_status OR dmpdsv$m_position));
; 2128         2203  2
; 2129         2204  2 RETURN 1;
; 2130         2205  1 END;



                                           0FFC 00000         .ENTRY  EXCH$DOS11_POSITION_TAPE, Save R2,R3,R4,R5,-; 2056
                                                                      R6,R7,R8,R9,R10,R11                         ;
                        5B 00000000G  EF   9E 00002           MOVAB   EXCH$UTIL_BLOCK_CHECK, R11                  ;
                        5A 00000000G  EF   9E 00009           MOVAB   EXCH$IO_DOS11_SKIP_FILE, R10                ;
                        59 00000000G  00   9E 00010           MOVAB   LIB$SIGNAL, R9                              ;
                        58 00000000G  8F   D0 00017           MOVL    #EXCH$_DOS11_POSITION, R8                   ;
                        57 00000000G  EF   9E 0001E           MOVAB   EXCH$IO_DOS11_REWIND, R7                    ;
                        56        04  AC   D0 00025           MOVL    VOLB, R6                                    ; 2098
                        52 041B00F3   8F   D0 00029           MOVL    #68878579, R2                               ; 2102
                        51      0267  8F   3C 00030           MOVZWL  #615, R1                                    ;
                        50            56   D0 00035           MOVL    R6, R0                                      ;
                                      6B   16 00038           JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                        54        54  A6   D0 0003A           MOVL    84(R6), R4                                  ; 2103
                        52 003600FD   8F   D0 0003E           MOVL    #3539197, R2                                ;
                        51      0268  8F   3C 00045           MOVZWL  #616, R1                                    ;
                        50            54   D0 0004A           MOVL    R4, R0                                      ;
```

```
                                       6B  16 0004D             JSB     EXCH$UTIL_BLOCK_CHECK                  ;
                        0D       0C    A4  E8 0004F             BLBS    12(R4), 1$                             ; 2113
                                       58  DD 00053             PUSHL   R8                                     ; 2116
                        69             01  FB 00055             CALLS   #1, LIB$SIGNAL                         ;
                                       56  DD 00058             PUSHL   R6                                     ; 2117
                        67             01  FB 0005A             CALLS   #1, EXCH$IO_DOS11_REWIND               ;
                        6A             50  E9 0005D             BLBC    R0, 6$                                 ;
                        55       08    AC  D0 00060  1$:        MOVL    ENT, R5                                ; 2124
                                 18    A5  B5 00064             TSTW    24(R5)                                 ;
                                       08  12 00067             BNEQ    2$                                     ;
                                       56  DD 00069             PUSHL   R6                                     ; 2130
                        67             01  FB 0006B             CALLS   #1, EXCH$IO_DOS11_REWIND               ;
                        59             50  E9 0006E             BLBC    R0, 6$                                 ;
                        50       18    A5  3C 00071  2$:        MOVZWL  24(R5), SKIP                           ; 2137
                        50       0E    A4  C2 00075             SUBL2   14(R4), SKIP                           ;
                                       50  DD 00079             PUSHL   SKIP                                   ; 2142
                                       56  DD 0007B             PUSHL   R6                                     ;
                        6A             02  FB 0007D             CALLS   #2, EXCH$IO_DOS11_SKIP_FILE            ;
                        47             50  E9 00080             BLBC    R0, 6$                                 ;
         0E       0C    A4             03  E1 00083             BBC     #3, 12(R4), 3$                         ; 2149
                                 0E    A4  D6 00088             INCL    14(R4)                                 ; 2153
                        7E             01  CE 0008B             MNEGL   #1, -(SP)                              ; 2154
                                       56  DD 0008E             PUSHL   R6                                     ;
                        6A             02  FB 00090             CALLS   #2, EXCH$IO_DOS11_SKIP_FILE            ;
                        34             50  E9 00093             BLBC    R0, 6$                                 ;
         0C       0C    A4             02  E1 00096  3$:        BBC     #2, 12(R4), 4$                         ; 2162
                                       56  DD 0009B             PUSHL   R6                                     ; 2165
              00000000G EF             01  FB 0009D             CALLS   #1, EXCH$IO_DOS11_READ_LABEL           ;
                        23             50  E9 000A4             BLBC    STATUS, 6$                             ; 2166
    08   A5       26    A4             0E  29 000A7  4$:        CMPC3   #14, 38(R4), 8(R5)                     ; 2173
                                       17  13 000AD             BEQL    5$                                     ;
                                       58  DD 000AF             PUSHL   R8                                     ; 2188
                        69             01  FB 000B1             CALLS   #1, LIB$SIGNAL                         ;
                                       56  DD 000B4             PUSHL   R6                                     ; 2192
                        67             01  FB 000B6             CALLS   #1, EXCH$IO_DOS11_REWIND               ;
                        0E             50  E9 000B9             BLBC    R0, 6$                                 ;
                                       55  DD 000BC             PUSHL   R5                                     ; 2198
                                       56  DD 000BE             PUSHL   R6                                     ;
                   FF3B CF             02  FB 000C0             CALLS   #2, EXCH$DOS11_POSITION_TAPE           ;
                                           04 000C5             RET                                            ;
                        50             01  D0 000C6  5$:        MOVL    #1, R0                                 ; 2204
                                           04 000C9             RET                                            ;
                                       50  D4 000CA  6$:        CLRL    R0                                     ; 2205
                                           04 000CC             RET                                            ;

; Routine Size:  205 bytes,    Routine Base:  EXCH$DOS11_CODE + 1084
; 2132           2206  1 END
; 2133           2207  0 ELUDOM



                                                                            .EXTRN  LIB$SIGNAL, LIB$STOP

;                                          PSECT SUMMARY
;
;       Name                           Bytes                            Attributes
;
;   EXCH$DOS11_PLIT                       408  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   EXCH$DOS11_CODE                      4433  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        21         0        1000         00:01.9
;   _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1             1151       202        17          79         00:01.4



;                                          COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:EXCDOS11/OBJ=OBJ$:EXCDOS11 MSRC$:EXCDOS11/UPDATE=(ENH$:EXCDOS11)

; Size:          4433 code + 408 data bytes
; Run Time:         01:22.9
; Elapsed Time:     04:08.1
; Lines/CPU Min:     1596
; Lexemes/CPU-Min: 21205
; Memory Used:  356 pages
; Compilation Complete
```

0160 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY

EXCDEFS
LIS

[illegible]

EXCDIRE
LIS

[illegible]

EXCDOS11
LIS

[illegible]

EXCDBG
LIS

[illegible]

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXCFIL11
LIS

EXCINIT
LIS

EXCLIB
LIS

EXCIO
LIS