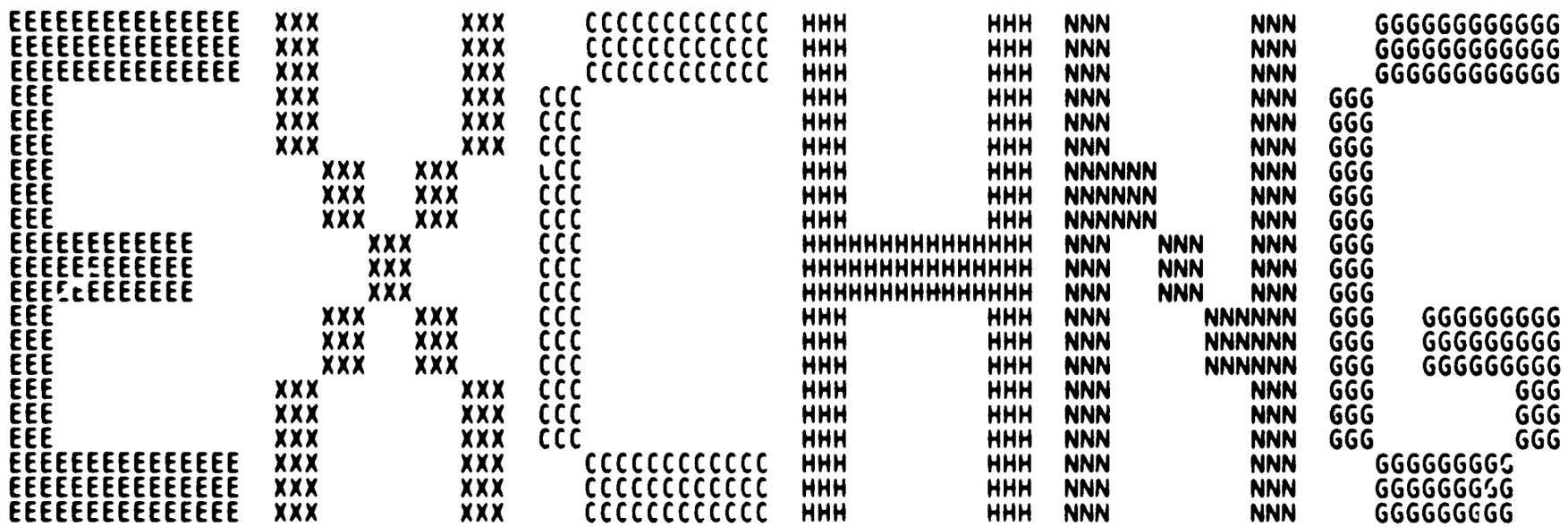
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN 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 GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNN GGG GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNNNNN GGG GGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG GGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGG

FILEID**EXCIO

```
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII      OOOOOO
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII      OOOOOO
EE          XX      XX  CC              II      OO      OO
EE          XX      XX  CC              II      OO      OO
EE            XX  XX    CC              II      OO      OO
EE            XX  XX    CC              II      OO      OO
EEEEEEEE        XX      CC              II      OO      OO
EEEEEEEE        XX      CC              II      OO      OO
EE            XX  XX    CC              II      OO      OO
EE            XX  XX    CC              II      OO      OO
EE          XX      XX  CC              II      OO      OO      ....
EE          XX      XX  CC              II      OO      OO      ....
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII      OOOOOO        ....
EEEEEEEEEE  XX      XX    CCCCCCCC    IIIIII      OOOOOO        ....

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
```

EXC
V04

```
;  1     0001  0 MODULE  exch$io                                %TITLE 'IO - Device and File I/O routines'
;  2     0002  0                 (
;  3     0003  0                 IDENT = 'V04-000',
;  4     0004  0                 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
;  5     0005  0                 ) =
;  6     0006  1 BEGIN
;  7     0007  1 !
;  8     0008  1 !*****************************************************************************
;  9     0009  1 !*                                                                           *
; 10     0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 11     0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 12     0012  1 !*  ALL RIGHTS RESERVED.                                                     *
; 13     0013  1 !*                                                                           *
; 14     0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 15     0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16     0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 17     0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 18     0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 19     0019  1 !*  TRANSFERRED.                                                             *
; 20     0020  1 !*                                                                           *
; 21     0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 22     0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 23     0023  1 !*  CORPORATION.                                                             *
; 24     0024  1 !*                                                                           *
; 25     0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 26     0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 27     0027  1 !*                                                                           *
; 28     0028  1 !*                                                                           *
; 29     0029  1 !*****************************************************************************
; 30     0030  1
; 31     0031  1 !++
; 32     0032  1 ! FACILITY:     EXCHANGE - Foreign volume interchange facility
; 33     0033  1 !
; 34     0034  1 ! ABSTRACT:     IO - Device I/O Routines
; 35     0035  1 !
; 36     0036  1 ! ENVIRONMENT:  VAX/VMS User mode
; 37     0037  1 !
; 38     0038  1 ! AUTHOR:       CW Hobbs                 CREATION DATE: 25-Aug-1982
; 39     0039  1 !
; 40     0040  1 ! MODIFIED BY:
; 41     0041  1 !
; 42     0042  1 !       V03-003 CWH3003         CW Hobbs                 21-Jul-1984
; 43     0043  1 !               Remove a debugging message accidentally left in.
; 44     0044  1 !
; 45     0045  1 !       V03-002 CWH3002         CW Hobbs                 12-Nov-1983
; 46     0046  1 !               During read and write checking, make sure to do a $rewind
; 47     0047  1 !               if the block is LBN 0.  Was giving spurious errors during
; 48     0048  1 !               COPY /BOOT operations.
; 49     0049  1 !
; 50     0050  1 !--
; 51     0051  1
; 52     0052  1 ! Include files:
; 53     0053  1 !
; 54     0054  1 MACRO $module_name_string = 'exch$io' %;        ! The require file needs to know our module name
; 55     0055  1 REQUIRE 'SRC$:EXCREQ'                           ! Facility-wide require file
; 56     0056  1     ;
```

```
;    58          0153  1 %SBTTL 'Module table of contents'
;    59          0154  1
;    60          0155  1 ! Module table of contents:
;    61          0156  1 !
;    62          0157  1 FORWARD ROUTINE
;    63          0158  1     exch$io_dos11_count_blocks,             ! Count blocks to the next tape mark
;    64          0159  1     exch$io_dos11_read,                     ! Read a block on a dos-11 device
;    65          0160  1     exch$io_dos11_read_label,               ! Read a label on a dos-11 device
;    66          0161  1     exch$io_dos11_rewind,                   ! Rewind a dos-11 sequential device
;    67          0162  1     exch$io_dos11_set_density,              ! Set the density on a magtape device
;    68          0163  1     exch$io_dos11_skip_file,                ! Skip files on a tape
;    69          0164  1     exch$io_dos11_skip_record,              ! Skip records on a tape
;    70          0165  1     exch$io_dos11_write_label,              ! Write a label on a dos-11 device
;    71          0166  1     exch$io_dos11_write_tape_mark,          ! Write a single tape mark on an dos-11 device
;    72          0167  1     exch$io_rt11_read,                      ! Read blocks from block-addressable device
;    73          0168  1     exch$io_rt11_read_1,                    ! Read blocks one at a time from block-addressable device
;    74          0169  1     exch$io_rt11_write,                     ! Write blocks on block-addressable device
;    75          0170  1     exch$io_rt11_write_1                    ! Write blocks one at a time on block-addressable device
;    76          0171  1     ;
;    77          0172  1
;    78          0173  1 ! EXCHANGE facility routines
;    79          0174  1 !
;    80          0175  1 EXTERNAL ROUTINE
;    81          0176  1     exch$util_file_error                    ! Signal file error
;    82          0177  1     ;
;    83          0178  1
;    84          0179  1 ! Equated symbols:
;    85          0180  1 !
;    86          0181  1 !LITERAL
;    87          0182  1 !    ;
;    88          0183  1
;    89          0184  1 ! Bound declarations:
;    90          0185  1 !
;    91          0186  1 !BIND
;    92          0187  1 !    ;
```

```
1 GLOBAL ROUTINE exch$io_dos11_count_blocks (volb : $ref_bblock) =        %SBTTL 'exch$io_dos11_count_blocks (
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Count blocks until the next tape mark on a dos-11 sequential device
2 !
2 ! INPUTS:
2 !
2 !       volb   - pointer to a volb which contains the active record stream
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       dos11 context block hanging off the volb
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       dos11 context block hanging off the volb
2 !
2 ! ROUTINE VALUE:
2 !
2 !       success or error status
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('io_dos11_count_blocks> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
2     ;
2
2 $debug_print_lit ('entry');
2 $block_check (2, .volb, volb, 611);
2 $block_check (2, .dosv, dos11, 613);
2 $logic_check (2, (.dosv [dos11$v_position_valid]), 269);        ! We should know where we are
2
2 ! Start by assuming no blocks, the skip count includes the tape mark
2 !
2 dosv [dos11$w_block_count] = -1;
2
2 ! Now start looping, we will return from the loop on error or tape mark
2 !
2 WHILE 1
2 DO
3     BEGIN
```

```
;  151          0245  3
;  152          0246  3      ! Issue the io operation to space in the forward direction
;  153          0247  3      !
;  154        P 0248  4      IF (status = $qiow (efn=0,
;  155        P 0249  4                          chan=.volb [volb$w_channel],
;  156        P 0250  4                          func=io$_skiprecord,
;  157        P 0251  4                          iosb=dosv [dos11$q_iosb],
;  158          0252  4                          p1=32767))
;  159          0253  3      THEN
;  160          0254  3          status = .dosv [dos11$w_iosb_status];
;  161          0255  3
;  162          0256  3      ! Finish up based on the status
;  163          0257  3      !
;  164          0258  3      SELECTONE .status OF                                                                                    ; R
;  165          0259  3      SET
;  166          0260  3          ! Normal means we have a very large file, add the skip count to the blocks and continue to loop
;  167          0261  3          !
;  168          0262  3          [ss$_normal] :
;  169          0263  4                          BEGIN
;  170          0264  4
;  171          0265  4                          ! Count the blocks
;  172          0266  4                          !
;  173          0267  4                          dosv [dos11$w_block_count] = .dosv [dos11$w_block_count] + 32767;
;  174          0268  4
;  175          0269  4                          ! Set the bits and pieces to reflect that the tape has moved
;  176          0270  4                          !
;  177          0271  4                          dosv [dos11$v_beg_of_tape] = false;
;  178          0272  4                          dosv [dos11$v_end_of_tape] = false;
;  179          0273  4                          dosv [dos11$v_tape_mark]   = false;
;  180          0274  3                          END;
;  181          0275  3
;  182          0276  3          ! End of file means that we have found the end
;  183          0277  3          !
;  184          0278  3          [ss$_endoffile] :
;  185          0279  4                          BEGIN
;  186          0280  4
;  187          0281  4                          ! Count the blocks, and note that we are at the next file
;  188          0282  4                          !
;  189          0283  4                          dosv [dos11$w_block_count] = .dosv [dos11$w_block_count] + .dosv [dos11$w_iosb_skipc
;  190          0284  4                          dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;
;  191          0285  4
;  192          0286  4                          ! Set the bits and pieces to reflect that the tape has moved
;  193          0287  4                          !
;  194          0288  4                          dosv [dos11$v_beg_of_tape] = false;
;  195          0289  4                          dosv [dos11$v_end_of_tape] = false;
;  196          0290  4                          dosv [dos11$v_tape_mark]   = true;
;  197          0291  4
;  198          0292  4                          RETURN true;
;  199          0293  4
;  200          0294  3                          END;
;  201          0295  3
;  202          0296  3          ! If this is the end, oops
;  203          0297  3          !
;  204          0298  3          [ss$_endoftape] :
;  205          0299  4                          BEGIN
;  206          0300  4                          LOCAL
;  207          0301  4                              stat2;
```

```
;  208     0302  4
;  209     0303  4                          ! Count the blocks, ar  no'  hat we are at the next file
;  210     0304  4                          !
;  211     0305  4                          dosv [dos11$w_block_count     dosv [dos11$w_block_count] + .dosv [dos11$w_iosb_skipc
;  212     0306  4                          !
;  213     0307  4                          ! Signal that we have had    roblem
;  214     0308  4                          !
;  215   P 0309  4                          $exch_signal (exch$_dos11_   error, 2,
;  216     0310  4                                        .volb [vol_$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  217     0311  4                          !
;  218     0312  4                          ! Rewind the tape so that the position will be known again
;  219     0313  4                          !
;  220     0314  4                          $exch_signal (exch$_dos11_position);                    ! This might take a while, w
;  221     0315  4                          dosv [dos11$v_error_rewind] = true;
;  222     0316  5                          IF NOT (stat2 = exch$io_dos11_rewind (.volb))
;  223     0317  4                          THEN
;  224     0318  4                              RETURN .stat2;
;  225     0319  4                          !
;  226     0320  4                          ! Return the ss$_endoftape status
;  227     0321  4                          !
;  228     0322  4                          RETURN .status;
;  229     0323  4                          !
;  230     0324  3                          END;
;  231     0325  3                          !
;  232     0326  3                      ! Any thing else and we'd better crap out
;  233     0327  3                      !
;  234     0328  4                      [OTHERWISE] :   BEGIN
;  235     0329  4                                      dosv [dos11$v_position_v     = false;  ! After an error, we don't know where we are
;  236   P 0330  4                                      $exch_signal_return (exc  dos11_ioerror, 2,
;  237     0331  4                                                    .volb [v  $l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  238     0332  3                                      END;
;  239     0333  3              TES;
;  240     0334  3
;  241     0335  2              END;
;  242     0336  2
;  243     0337  2          RETURN 0;
;  244     0338  1      END;


                                        .TITLE  EXCH$IO IO - Device and File I/O routines
                                        .IDENT  \V04-000\

                                        .EXTRN  EXCH$UTIL_FILE_ERROR
                                        .EXTRN  EXCH$UTIL_BLOCK_CHECK
                                        .EXTRN  EXCH$_BADLOGIC, SYS$QIOW
                                        .EXTRN  EXCH$_DOS11_IOERROR
                                        .EXTRN  EXCH$_DOS11_POSITION

                                        .PSECT  EXCH$IO_CODE,NOWRT,2

                        03FC 00000      .ENTRY  EXCH$IO_DOS11_COUNT_BLOCKS, Save R2,R3,R4,- ; 0188
                                                R5,R6,R7,R8,R9
59 000000000G   EF   9E 00002           MOVAB   EXCH$UTIL_BLOCK_CHECK, R9                    ;
58 000000000G   8F   D0 00009           MOVL    #EXCH$_DOS11_IOERROR, R8                     ;
57 000000000G   00   9E 00010           MOVAB   LIB$SIGNAL, R7                               ;
54         04   AC   D0 00017           MOVL    VOLB, R4                                     ; 0228
52 041B00F3     8F   D0 0001B           MOVL    #68878579, R2                                ; 0232
```

```
                                51      0263    8F  3C 00022          MOVZWL  #611, R1                                        ;
                                50              54  D0 00027          MOVL    R4, R0                                          ;
                                                69  16 0002A          JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                                53        54    A4  D0 0002C          MOVL    84(R4), R3                                      ; 0233
                                52  003600FD     8F  D0 00030          MOVL    #3539197, R2                                    ;
                                51      0265    8F  3C 00037          MOVZWL  #613, R1                                        ;
                                50              53  D0 0003C          MOVL    R3, R0                                          ;
                                                69  16 0003F          JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                                56        0C    A3  9E 00041          MOVAB   12(R3), R6                                      ; 0234
                                14              66  E8 00045          BLBS    (R6), 1$                                        ;
                                7E      010D    8F  3C 00048          MOVZWL  #269, -(SP)                                     ;
                                                01  DD 0004D          PUSHL   #1                                              ;
                                    00000000G   8F  DD 0004F          PUSHL   #EXCH$_BADLOGIC                                 ;
                    00000000G   00              03  FB 00055          CALLS   #3, LIB$STOP                                    ;
                            34  A3              01  AE 0005C  1$:     MNEGW   #1, 52(R3)                                      ; 0238
                                                7E  7C 00060  2$:     CLRQ    -(SP)                                           ; 0252
                                                7E  7C 00062          CLRQ    -(SP)                                           ;
                                                7E  D4 00064          CLRL    -(SP)                                           ;
                                7E      7FFF    8F  3C 00066          MOVZWL  #32767, -(SP)                                   ;
                                                7E  7C 0006B          CLRQ    -(SP)                                           ;
                                          1E    A3  9F 0006D          PUSHAB  30(R3)                                          ;
                                                26  DD 00070          PUSHL   #38                                             ;
                                7E        4A    A4  3C 00072          MOVZWL  74(R4), -(SP)                                   ;
                                                7E  D4 00076          CLRL    -(SP)                                           ;
                    00000000G   00              0C  FB 00078          CALLS   #12, SYS$QIOW                                   ;
                                52              50  D0 0007F          MOVL    R0, STATUS                                      ;
                                04              52  E9 00082          BLBC    STATUS, 3$                                      ;
                                52          1E  A3  3C 00085          MOVZWL  30(R3), STATUS                                  ; 0254
                                01              52  D1 00089  3$:     CMPL    STATUS, #1                                      ; 0262
                                                0B  12 0008C          BNEQ    4$                                              ;
                            34  A3      7FFF    8F  A0 0008E          ADDW2   #32767, 52(R3)                                  ; 0267
                                66              0E  8A 00094          BICB2   #14, (R6)                                       ; 0273
                                                C7  11 00097          BRB     2$                                              ; 0258
                    00000870    8F              52  D1 00099  4$:     CMPL    STATUS, #2160                                   ; 0278
                                                12  12 000A0          BNEQ    5$                                              ;
                            34  A3          20  A3  A0 000A2          ADDW2   32(R3), 52(R3)                                  ; 0283
                                            0E  A3  D6 000A7          INCL    14(R3)                                          ; 0284
                                66              0A  8A 000AA          BICB2   #10, (R6)                                       ; 0289
                                66              04  88 000AD          BISB2   #4, (R6)                                        ; 0290
                                50              01  D0 000B0          MOVL    #1, R0                                          ; 0292
                                                04  000B3          RET                                                     ;
                    00000878    8F              52  D1 000B4  5$:     CMPL    STATUS, #2168                                   ; 0298
                                                2F  12 000BB          BNEQ    6$                                              ;
                            34  A3          20  A3  A0 000BD          ADDW2   32(R3), 52(R3)                                  ; 0305
                                                52  DD 000C2          PUSHL   STATUS                                          ; 0310
                                            69  A4  9F 000C4          PUSHAB  105(R4)                                         ;
                                            65  A4  DD 000C7          PUSHL   101(R4)                                         ;
                                                02  DD 000CA          PUSHL   #2                                              ;
                                                58  DD 000CC          PUSHL   R8                                              ;
                                67                  05  FB 000CE          CALLS   #5, LIB$SIGNAL                              ;
                                    00000000G   8F  DD 000D1          PUSHL   #EXCH$_DOS11_POSITION                           ; 0314
                                67                  01  FB 000D7          CALLS   #1, LIB$SIGNAL                              ;
                                66          40      8F  88 000DA          BISB2   #64, (R6)                                       ; 0315
                                                54  DD 000DE          PUSHL   R4                                              ; 0316
                        0000V   CF                  01  FB 000E0          CALLS   #1, EXCH$IO_DOS11_REWIND                    ;
                                1C                  50  F9 000E5          BLBC    STAT2, 7$                                   ;
                                50                  52  D0 000E8          MOVL    STATUS, R0                                  ; 0322
```

```
                           04 000EB      RET                           ;
       66              01  8A 000EC 6$:  BICB2    #1, (R6)             ; 0329
       55              58  D0 000EF      MOVL     R8, TEMP             ; 0331
                       52  DD 000F2      PUSHL    STATUS               ;
                69     A4  9F 000F4      PUSHAB   105(R4)              ;
                65     A4  DD 000F7      PUSHL    101(R4)              ;
                       02  DD 000FA      PUSHL    #2                   ;
                       55  DD 000FC      PUSHL    TEMP                 ;
       67              05  FB 000FE      CALLS    #5, LIB$SIGNAL       ;
       50              55  D0 00101      MOVL     TEMP, R0             ;
                           04 00104 7$:  RET                           ; 0338

; Routine Size:  261 bytes,    Routine Base:  EXCH$IO_CODE + 0000
```

```
;  246     0339  1 GLOBAL ROUTINE exch$io_dos11_read (volb : $ref_bblock, buffer) =  %SBTTL 'exch$io_dos11_read'
;  247     0340  2 BEGIN
;  248     0341  2 !++
;  249     0342  2 !
;  250     0343  2 ! FUNCTIONAL DESCRIPTION:
;  251     0344  2 !
;  252     0345  2 !       Read one block on a dos-11 sequential device
;  253     0346  2 !
;  254     0347  2 ! INPUTS:
;  255     0348  2 !
;  256     0349  2 !       volb    - pointer to a volb which contains the active record stream
;  257     0350  2 !
;  258     0351  2 ! IMPLICIT INPUTS:
;  259     0352  2 !
;  260     0353  2 !       dos11 context block hanging off the volb
;  261     0354  2 !
;  262     0355  2 ! OUTPUTS:
;  263     0356  2 !
;  264     0357  2 !       buffer - pointer to the buffer to receive the block
;  265     0358  2 !
;  266     0359  2 ! IMPLICIT OUTPUTS:
;  267     0360  2 !
;  268     0361  2 !       dos11 context block hanging off the volb
;  269     0362  2 !
;  270     0363  2 ! ROUTINE VALUE:
;  271     0364  2 !
;  272     0365  2 !       success if label, false if not a label or error status
;  273     0366  2 !
;  274     0367  2 ! SIDE EFFECTS:
;  275     0368  2 !
;  276     0369  2 !       none
;  277     0370  2 !--
;  278     0371  2
;  279     0372  2 $dbgtrc_prefix ('io_dos11_read> ');
;  280     0373  2
;  281     0374  2 LOCAL
;  282     0375  2     func,
;  283     0376  2     status
;  284     0377  2     ;
;  285     0378  2
;  286     0379  2 BIND
;  287     0380  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  288     0381  2     ;
;  289     0382  2
;  290     0383  2 $debug_print_lit ('entry');
;  291     0384  2 $block_check (2, .volb, volb, 625);
;  292     0385  2 $block_check (2, .dosv, dos11, 626);
;  293     0386  2 $check_call (4, exch$dbg_dump_dosv, .volb, (dmpdsv$m_status OR dmpdsv$m_position));
;  294     0387  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 279);         ! We should know where we are
;  295     0388  2 $logic_check (2, (NOT .dosv [dos11$v_tape_mark]), 280);          ! And that should be right after a tape mark
;  296     0389  2 $logic_check (2, (NOT .dosv [dos11$v_end_of_tape]), 281);       ! But not at the end of the tape
```

```
;  298          0390  2 ! Issue the io operation to read the next logical block in the forward direction
;  299          0391  2 !
;  300          0392  2 func = io$_readlblk;
;  301          0393  2 IF .volb [volb$v_read_check] THEN func = .func OR io$m_datacheck;
;  302        P 0394  3 IF (status = $qiow (efn=0,
;  303        P 0395  3                     chan=.volb [volb$w_channel],
;  304        P 0396  3                     func=.func,
;  305        P 0397  3                     iosb=dosv [dos11$q_iosb],
;  306        P 0398  3                     p1=.buffer,
;  307          0399  3                     p2=512))
;  308          0400  2 THEN
;  309          0401  2     status = .dosv [dos11$w_iosb_status];
;  310          0402  2
;  311          0403  2 ! Finish up based on the status
;  312          0404  2 !
;  313          0405  2 SELECTONE .status OF
;  314          0406  2 SET
;  315          0407  2     ! Status is normal
;  316          0408  2     !
;  317          0409  3     [ss$_normal] :      BEGIN
;  318          0410  3
;  319          0411  3                         ! If the block isn't 512 bytes long, pad it with nulls
;  320          0412  3                         !
;  321          0413  3                         IF .dosv [dos11$w_iosb_bytecnt] NEQ 512
;  322          0414  3                         THEN
;  323          0415  4                             BEGIN
;  324          0416  4                             LOCAL
;  325          0417  4                                 adr,
;  326          0418  4                                 len;
;  327          0419  4                             len = 512 - .dosv [dos11$w_iosb_bytecnt];
;  328          0420  4                             adr = .buffer + .dosv [dos11$w_iosb_bytecnt];
;  329          0421  4                             CH$FILL (0, .len, .adr);
;  330          0422  3                             END;
;  331          0423  3
;  332          0424  3                         ! Set the bits and pieces to reflect that the tape has moved
;  333          0425  3                         !
;  334          0426  3                         dosv [dos11$v_end_of_tape] = false;
;  335          0427  3                         dosv [dos11$v_beg_of_tape] = false;
;  336          0428  3                         dosv [dos11$v_tape_mark]   = false;
;  337          0429  2                         END;
;  338          0430  2
;  339          0431  2     ! If the data is longer than the buffer, we have a bad tape format
;  340          0432  2     !
;  341          0433  3     [ss$_dataoverun] :  BEGIN
;  342          0434  3                         dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
;  343        P 0435  3                         $exch_signal_return (exch$_dos11_blocksize, 2,
;  344          0436  3                                              .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  345          0437  2                         END;
;  346          0438  2
;  347          0439  2     ! We saw a tape mark (end of file status)
;  348          0440  2     !
;  349          0441  3     [ss$_endoffile] :   BEGIN
;  350          0442  3                         $trace_print_lit ('got an SS$_ENDOFFILE');
;  351          0443  3                         dosv [dos11$v_tape_mark] = true;
;  352          0444  2                         END;
;  353          0445  2
;  354          0446  2     ! If this is the end, oops
```

```
;  355          0447  2      !
;  356          0448  3      [ss$_endoftape] :   BEGIN
;  357          0449  3                          LOCAL
;  358          0450  3                              stat2;
;  359          0451  3
;  360          0452  3                          ! Signal that we have had a problem
;  361          0453  3                          !
;  362        P 0454  3                          $exch_signal (exch$_dos11_ioerror, 2,
;  363          0455  3                                        .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  364          0456  3
;  365          0457  3                          ! Rewind the tape so that the position will be known again
;  366          0458  3                          !
;  367          0459  3                          $exch_signal (exch$_dos11_position);                    ! This might take a while, w
;  368          0460  3                          dosv [dos11$v_error_rewind] = true;
;  369          0461  4                          IF NOT (stat2 = exch$io_dos11_rewind (.volb))
;  370          0462  3                          THEN
;  371          0463  3                              RETURN .stat2;
;  372          0464  3
;  373          0465  3                          ! Return the ss$_endoftape status
;  374          0466  3                          !
;  375          0467  2                          END;
;  376          0468  2
;  377          0469  2      ! Any thing else and we better crap out
;  378          0470  2      !
;  379          0471  3      [OTHERWISE] :       BEGIN
;  380          0472  3                          dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
;  381        P 0473  3                          $exch_signal_return (exch$_dos11_ioerror, 2,
;  382          0474  3                                        .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  383          0475  2                          END;
;  384          0476  2 TES;
;  385          0477  2
;  386          0478  2 RETURN .status;
;  387          0479  1 END;


                                                .EXTRN  EXCH$_DOS11_BLOCKSIZE

                                 0FFC 00000     .ENTRY  EXCH$IO_DOS11_READ, Save R2,R3,R4,R5,R6,R7,-; 0339
                                                        R8,R9,R10,R11
5B 00000000G   00   9E 00002                    MOVAB   LIB$STOP, R11                               ;
5A 00000000G   8F   D0 00009                    MOVL    #EXCH$_BADLOGIC, R10                        ;
59 00000000G   00   9E 00010                    MOVAB   LIB$SIGNAL, R9                              ;
56       04    AC   D0 00017                    MOVL    VOLB, R6                                    ; 0380
52 041B00F3    8F   D0 0001B                    MOVL    #68878579, R2                               ; 0384
51     0271    8F   3C 00022                    MOVZWL  #625, R1                                    ;
50             56   D0 00027                    MOVL    R6, R0                                      ;
   00000000G   EF   16 0002A                    JSB     EXCH$UTIL_BLOCK_CHECK                       ;
53       54    A6   D0 00030                    MOVL    84(R6), R3                                  ; 0385
52 003600FD    8F   D0 00034                    MOVL    #3539197, R2                                ;
51     0272    8F   3C 0003B                    MOVZWL  #626, R1                                    ;
50             53   D0 00040                    MOVL    R3, R0                                      ;
   00000000G   EF   16 00043                    JSB     EXCH$UTIL_BLOCK_CHECK                       ;
58       0C    A3   9E 00049                    MOVAB   12(R3), R8                                  ; 0387
0C             68   E8 0004D                    BLBS    (R8), 1$                                    ;
7E     0117    8F   3C 00050                    MOVZWL  #279, -(SP)                                 ;
               01   DD 00055                    PUSHL   #1                                          ;
```

```
                                     5A DD 00057          PUSHL   R10                                   ;
                        6B           03 FB 00059          CALLS   #3, LIB$STOP                          ;
          0C            68           02 E1 0005C 1$:      BBC     #2, (R8), 2$                          ; 0388
                        7E     0118  8F 3C 00060          MOVZWL  #280, -(SP)                           ;
                                     01 DD 00065          PUSHL   #1                                    ;
                                     5A DD 00067          PUSHL   R10                                   ;
                        6B           03 FB 00069          CALLS   #3, LIB$STOP                          ;
          0C            68           03 E1 0006C 2$:      BBC     #3, (R8), 3$                          ; 0389
                        7E     0119  8F 3C 00070          MOVZWL  #281, -(SP)                           ;
                                     01 DD 00075          PUSHL   #1                                    ;
                                     5A DD 00077          PUSHL   R10                                   ;
                        6B           03 FB 00079          CALLS   #3, LIB$STOP                          ;
                        50           21 D0 0007C 3$:      MOVL    #33, FUNC                             ; 0392
          05       48   A6           01 E1 0007F          BBC     #1, 72(R6), 4$                        ; 0393
                        50     4000  8F A8 00084          BISW2   #16384, FUNC                          ;
                                     7E 7C 00089 4$:      CLRQ    -(SP)                                 ; 0399
                                     7E 7C 0008B          CLRQ    -(SP)                                 ;
                        7E     0200  8F 3C 0008D          MOVZWL  #512, -(SP)                           ;
                                 08  AC DD 00092          PUSHL   BUFFER                                ;
                                     7E 7C 00095          CLRQ    -(SP)                                 ;
                                 1E  A3 9F 00097          PUSHAB  30(R3)                                ;
                                     50 DD 0009A          PUSHL   FUNC                                  ;
                        7E       4A  A6 3C 0009C          MOVZWL  74(R6), -(SP)                         ;
                                     7E D4 000A0          CLRL    -(SP)                                 ;
             00000000G  00           0C FB 000A2          CALLS   #12, SYS$QIOW                         ;
                        57           50 D0 000A9          MOVL    R0, STATUS                            ;
                        04           57 E9 000AC          BLBC    STATUS, 5$                            ;
                        57       1E  A3 3C 000AF          MOVZWL  30(R3), STATUS                        ; 0401
                        01           57 D1 000B3 5$:      CMPL    STATUS, #1                            ; 0409
                        27           12 000B6             BNEQ    7$                                    ;
                0200    8F       20  A3 B1 000B8          CMPW    32(R3), #512                          ; 0413
                        1A           13 000BE             BEQL    6$                                    ;
                        50       20  A3 3C 000C0          MOVZWL  32(R3), LEN                           ; 0419
          50 00000200   8F           50 C3 000C4          SUBL3   LEN, #512, LEN                        ;
                        51       20  A3 3C 000CC          MOVZWL  32(R3), ADR                           ; 0420
                        51       08  AC C0 000D0          ADDL2   BUFFER, ADR                           ;
 50       00            6E           00 2C 000D4          MOVC5   #0, (SP), #0, LEN, (ADR)              ; 0421
                                     61    000D9                                                        ;
                        68           0E 8A 000DA 6$:      BICB2   #14, (R8)                             ; 0428
                                     2E 11 000DD          BRB     9$                                    ; 0405
             00000838   8F           57 D1 000DF 7$:      CMPL    STATUS, #2104                         ; 0433
                                     19 12 000E6          BNEQ    8$                                    ;
                        68           01 8A 000E8          BICB2   #1, (R8)                              ; 0434
                        52 00000000G 8F D0 000EB          MOVL    #EXCH$_DOS11_BLOCKSIZE, TEMP          ; 0436
                                 69  A6 9F 000F2          PUSHAB  105(R6)                               ;
                                 65  A6 DD 000F5          PUSHL   101(R6)                               ;
                                     02 DD 000F8          PUSHL   #2                                    ;
                                     52 DD 000FA          PUSHL   TEMP                                  ;
                        69           04 FB 000FC          CALLS   #4, LIB$SIGNAL                        ;
                                     5B 11 000FF          BRB     12$                                   ;
             00000870   8F           57 D1 00101 8$:      CMPL    STATUS, #2160                         ; 0441
                                     05 12 00108          BNEQ    10$                                   ;
                        68           04 88 0010A          BISB2   #4, (R8)                              ; 0443
                                     51 11 0010D 9$:      BRB     13$                                   ; 0405
             00000878   8F           57 D1 0010F 10$:     CMPL    STATUS, #2168                         ; 0448
                                     2B 12 00116          BNEQ    11$                                   ;
                                     57 DD 00118          PUSHL   STATUS                                ; 0455
```

```
                     69    A6   9F 0011A          PUSHAB   105(R6`                              ;
                     65    A6   DD 0011D          PUSHL    101(R6)                              ;
                           02   DD 00120          PUSHL    #2                                   ;
              00000000G    8F   DD 00122          PUSHL    #EXCH$_DOS11_IOERROR                 ;
                69         05   FB 00128          CALLS    #5, LIB$SIGNAL                       ;
              00000000G    8F   DD 0012B          PUSHL    #EXCH$_DOS11_POSITION                ;  0459
                69         01   FB 00131          CALLS    #1, LIB$SIGNAL                       ;
                68   40    8F   88 00134          BISB2    #64, (R8)                            ;  0460
                           56   DD 00138          PUSHL    R6                                   ;  0461
        0000V   CF         01   FB 0013A          CALLS    #1, EXCH$IO_DOS11_REWIND             ;
                1E         50   E8 0013F          BLBS     STAT2, 13$                           ;
                                04 00142          RET                                           ;  0463
                68         01   8A 00143  11$:    BICB2    #1, (R8)                             ;  0472
                52 00000000G    8F   D0 00146     MOVL     #EXCH$_DOS11_IOERROR, TEMP           ;  0474
                           57   DD 0014D          PUSHL    STATUS                               ;
                     69    A6   9F 0014F          PUSHAB   105(R6)                              ;
                     65    A6   DD 00152          PUSHL    101(R6)                              ;
                           02   DD 00155          PUSHL    #2                                   ;
                           52   DD 00157          PUSHL    TEMP                                 ;
                69         05   FB 00159          CALLS    #5, LIB$SIGNAL                       ;
                50         52   D0 0015C  12$:    MOVL     TEMP, R0                             ;
                                04 0015F          RET                                           ;
                50         57   D0 00160  13$:    MOVL     STATUS, R0                           ;  0478
                                04 00163          RET                                           ;  0479

; Routine Size:  356 bytes,    Routine Base:  EXCH$IO_CODE + 0105
```

```
;  389          0480  1 GLOBAL ROUTINE exch$io_dos11_read_label (volb : $ref_bblock) =          %SBTTL 'exch$io_dos11_read_label (vo
;  390          0481  2 BEGIN
;  391          0482  2 !++
;  392          0483  2 !
;  393          0484  2 ! FUNCTIONAL DESCRIPTION:
;  394          0485  2 !
;  395          0486  2 !       Read a label on a dos-11 sequential device
;  396          0487  2 !
;  397          0488  2 ! INPUTS:
;  398          0489  2 !
;  399          0490  2 !       volb    - pointer to a volb which contains the active record stream
;  400          0491  2 !
;  401          0492  2 ! IMPLICIT INPUTS:
;  402          0493  2 !
;  403          0494  2 !       dos11 context block hanging off the volb
;  404          0495  2 !
;  405          0496  2 ! OUTPUTS:
;  406          0497  2 !
;  407          0498  2 !       none
;  408          0499  2 !
;  409          0500  2 ! IMPLICIT OUTPUTS:
;  410          0501  2 !
;  411          0502  2 !       dos11 context block hanging off the volb
;  412          0503  2 !
;  413          0504  2 ! ROUTINE VALUE:
;  414          0505  2 !
;  415          0506  2 !       success if label, false if not a label or error status
;  416          0507  2 !
;  417          0508  2 ! SIDE EFFECTS:
;  418          0509  2 !
;  419          0510  2 !       none
;  420          0511  2 !--
;  421          0512  2
;  422          0513  2 $dbgtrc_prefix ('io_dos11_read_label> ');
;  423          0514  2
;  424          0515  2 LOCAL
;  425          0516  2     func,
;  426          0517  2     status
;  427          0518  2     ;
;  428          0519  2
;  429          0520  2 BIND
;  430          0521  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  431          0522  2     ;
;  432          0523  2
;  433          0524  2 $debug_print_lit ('entry');
;  434          0525  2 $block_check (2, .volb, volb, 602);
;  435          0526  2 $block_check (2, .dosv, dos11, 603);
;  436          0527  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 257);        ! We should know where we are
;  437          0528  2 $logic_check (2, (.dosv [dos11$v_tape_mark]), 258);            ! And that should be right after a tape mark
;  438          0529  2 $logic_check (2, (NOT .dosv [dos11$v_end_of_tape]), 268);      ! But not at the end of the tape
```

```
;  440          0530  2 ! Issue the io operation to read the next logical block in the forward direction
;  441          0531  2 !
;  442        L 0532  2 $logic_check (3, (dos11$s_label_buf EQL 14), 267);
; %PRINT:       assumption 267 verified during compilation
;  443          0533  2 func = io$_readlblk;
;  444          0534  2 IF .volb [volb$v_read_check] THEN func = .func OR io$m_datacheck;
;  445        P 0535  3 IF (status = $qiow (efn=0,
;  446        P 0536  3                     chan=.volb [volb$w_channel],
;  447        P 0537  3                     func=.func,
;  448        P 0538  3                     iosb=dosv [dos11$q_iosb],
;  449        P 0539  3                     p1=dosv [dos11$t_label_buf],
;  450          0540  3                     p2=dos11$s_label_buf))
;  451          0541  2 THEN
;  452          0542  2     status = .dosv [dos11$w_iosb_status];
;  453          0543  2
;  454          0544  2 ! Finish up based on the status
;  455          0545  2 !
;  456          0546  2 SELECTONE .status OF
;  457          0547  2 SET
;  458          0548  2     ! Status is normal, make a cursory check that the label is the correct length.  There is little else we
;  459          0549  2     ! do as far as checking, since just about anything that is 14 bytes long could be a valid label.
;  460          0550  2     !
;  461          0551  3     [ss$_normal] :      BEGIN
;  462          0552  3
;  463          0553  3                         ! If the block isn't 14 bytes long, it is not a dos11 label
;  464          0554  3                         !
;  465          0555  3                         IF .dosv [dos11$w_iosb_bytecnt] NEQ 14
;  466          0556  3                         THEN
;  467          0557  4                             BEGIN
;  468          0558  4                             dosv [dos11$v_position_valid] = false;      ! After an error, we don't know wher
;  469        P 0559  4                             $exch_signal_return (exch$_dos11_badlabel, 2,
;  470          0560  4                                     .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  471          0561  3                             END;
;  472          0562  3
;  473          0563  3                         ! Set the bits and pieces to reflect that the tape has moved
;  474          0564  3                         !
;  475          0565  3                         dosv [dos11$v_beg_of_tape] = false;
;  476          0566  3                         dosv [dos11$v_end_of_tape] = false;
;  477          0567  3                         dosv [dos11$v_tape_mark]   = false;
;  478          0568  2                         END;
;  479          0569  2
;  480          0570  2     ! If the data is longer than the buffer, it is obviously not 14 bytes long, therefore we have a bad labe
;  481          0571  2     !
;  482          0572  3     [ss$_dataoverun] :  BEGIN
;  483          0573  3                         dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
;  484        P 0574  3                         $exch_signal_return (exch$_dos11_badlabel, 2,
;  485          0575  3                                 .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  486          0576  2                         END;
;  487          0577  2
;  488          0578  2     ! If this is a freshly initialized dos tape, we should see a tape mark (end of file status)
;  489          0579  2     !
;  490          0580  3     [ss$_endoffile] :   BEGIN
;  491          0581  3                         $debug_print_lit ('got an SS$_ENDOFFILE');
;  492          0582  3                         dosv [dos11$v_end_of_tape] = true;
;  493          0583  2                         END;
```

```
;  495          0584  2     ! If this is the end, oops
;  496          0585  2     !
;  497          0586  3     [ss$_endoftape] :   BEGIN
;  498          0587  3                         LOCAL
;  499          0588  3                             stat2;
;  500          0589  3
;  501          0590  3                         ! Signal that we have had a problem
;  502          0591  3                         !
;  503        P 0592  3                         $exch_signal (exch$_dos11_ioerror, 2,
;  504          0593  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  505          0594  3
;  506          0595  3                         ! Rewind the tape so that the position will be known again
;  507          0596  3                         !
;  508          0597  3                         $exch_signal (exch$_dos11_position);                    ! This might take a while, w
;  509          0598  3                         dosv [dos11$v_error_rewind] = true;
;  510          0599  4                         IF NOT (stat2 = exch$io_dos11_rewind (.volb))
;  511          0600  3                         THEN
;  512          0601  3                             RETURN .stat2;
;  513          0602  3
;  514          0603  3                         ! Return the ss$_endoftape status
;  515          0604  3                         !
;  516          0605  2                         END;
;  517          0606  2
;  518          0607  2     ! Any thing else and we better crap out
;  519          0608  2     !
;  520          0609  3     [OTHERWISE] :       BEGIN
;  521          0610  3                         dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
;  522        P 0611  3                         $exch_signal_return (exch$_dos11_badlabel, 2,
;  523          0612  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  524          0613  2                         END;
;  525          0614  2 TES;
;  526          0615  2
;  527          0616  2 RETURN .status;
;  528          0617  1 END;


                                                .EXTRN  EXCH$_DOS11_BADLABEL

                                03FC 00000      .ENTRY  EXCH$IO_DOS11_READ_LABEL, Save R2,R3,R4,R5,-  ; 0480
                                                        R6,R7,R8,R9
59 00000000G   EF   9E 00002                    MOVAB   EXCH$UTIL_BLOCK_CHECK, R9
58 00000000G   00   9E 00009                    MOVAB   LIB$SIGNAL, R8
57 00000000G   00   9E 00010                    MOVAB   LIB$STOP, R7
56 00000000G   8F   D0 00017                    MOVL    #EXCH$_BADLOGIC, R6
53         04  AC   D0 0001E                    MOVL    VOLB, R3                                    ; 0521
52 041B00F3    8F   D0 00022                    MOVL    #68878579, R2                               ; 0525
51       025A  8F   3C 00029                    MOVZWL  #602, R1
50             53   D0 0002E                    MOVL    R3, R0
               69   16 00031                    JSB     EXCH$UTIL_BLOCK_CHECK
54         54  A3   D0 00033                    MOVL    84(R3), R4                                  ; 0526
52 003600FD    8F   D0 00037                    MOVL    #3539197, R2
51       025B  8F   3C 0003E                    MOVZWL  #603, R1
50             54   D0 00043                    MOVL    R4, R0
               69   16 00046                    JSB     EXCH$UTIL_BLOCK_CHECK
55         0C  A4   9E 00048                    MOVAB   12(R4), R5                                  ; 0527
0C             65   E8 0004C                    BLBS    (R5), 1$
```

```
                          7E      0101    8F  3C 0004F            MOVZWL  #257, -(SP)                                      ;
                                          01  DD 00054            PUSHL   #1                                              ;
                                          56  DD 00056            PUSHL   R6                                              ;
                          67              03  FB 00058            CALLS   #3, LIB$STOP                                    ;
                    0C    65              02  E0 0005B 1$:        BBS     #2, (R5), 2$                                    ; 0528
                          7E      0102    8F  3C 0005F            MOVZWL  #258, -(SP)                                     ;
                                          01  DD 00064            PUSHL   #1                                              ;
                                          56  DD 00066            PUSHL   R6                                              ;
                          67              03  FB 00068            CALLS   #3, LIB$STOP                                    ;
                    0C    65              03  E1 0006B 2$:        BBC     #3, (R5), 3$                                    ; 0529
                          7E      010C    8F  3C 0006F            MOVZWL  #268, -(SP)                                     ;
                                          01  DD 00074            PUSHL   #1                                              ;
                                          56  DD 00076            PUSHL   R6                                              ;
                          67              03  FB 00078            CALLS   #3, LIB$STOP                                    ;
                          50              21  D0 0007B 3$:        MOVL    #33, FUNC                                       ; 0533
                    05    48    A3        01  E1 0007E            BBC     #1, 72(R3), 4$                                  ; 0534
                          50      4000    8F  A8 00083            BISW2   #16384, FUNC                                    ; 0540
                                          7E  7C 00088 4$:        CLRQ    -(SP)                                           ;
                                          7E  7C 0008A            CLRQ    -(SP)                                           ;
                                          0E  DD 0008C            PUSHL   #14                                             ;
                                    26    A4  9F 0008E            PUSHAB  38(R4)                                          ;
                                          7E  7C 00091            CLRQ    -(SP)                                           ;
                                    1E    A4  9F 00093            PUSHAB  30(R4)                                          ;
                                          50  DD 00096            PUSHL   FUNC                                            ;
                          7E        4A    A3  3C 00098            MOVZWL  74(R3), -(SP)                                   ;
                                          7E  D4 0009C            CLRL    -(SP)                                           ;
                00000000G 00              0C  FB 0009E            CALLS   #12, SYS$QIOW                                   ;
                          52              50  D0 000A5            MOVL    R0, STATUS                                      ;
                          04              52  E9 000A8            BLBC    STATUS, 5$                                      ;
                          52        1E    A4  3C 000AB            MOVZWL  30(R4), STATUS                                  ; 0542
                          01              52  D1 000AF 5$:        CMPL    STATUS, #1                                      ; 0551
                                          0B  12 000B2            BNEQ    6$                                              ;
                          0E        20    A4  B1 000B4            CMPW    32(R4), #14                                     ; 0555
                                          50  12 000B8            BNEQ    8$                                              ;
                          65              0E  8A 000BA            BICB2   #14, (R5)                                       ; 0567
                                          68  11 000BD            BRB     9$                                              ; 0546
                00000838  8F              52  D1 000BF 6$:        CMPL    STATUS, #2104                                   ; 0572
                                          42  13 000C6            BEQL    8$                                              ;
                00000870  8F              52  D1 000C8            CMPL    STATUS, #2160                                   ; 0580
                                          05  12 000CF            BNEQ    7$                                              ;
                          65              08  88 000D1            BISB2   #8, (R5)                                        ; 0582
                                          51  11 000D4            BRB     9$                                              ; 0546
                00000878  8F              52  D1 000D6 7$:        CMPL    STATUS, #2168                                   ; 0586
                                          2B  12 000DD            BNEQ    8$                                              ;
                                          52  DD 000DF            PUSHL   STATUS                                          ; 0593
                                    69    A3  9F 000E1            PUSHAB  105(R3)                                         ;
                                    65    A3  DD 000E4            PUSHL   101(R3)                                         ;
                                          02  DD 000E7            PUSHL   #2                                              ;
                      00000000G   8F        DD 000E9            PUSHL   #EXCH$_DOS11_IOERROR                            ;
                          68              05  FB 000EF            CALLS   #5, LIB$SIGNAL                                  ;
                      00000000G   8F        DD 000F2            PUSHL   #EXCH$_DOS11_POSITION                           ; 0597
                          68              01  FB 000F8            CALLS   #1, LIB$SIGNAL                                  ;
                          65        40    8F  88 000FB            BISB2   #64, (R5)                                       ; 0598
                                          53  DD 000FF            PUSHL   R3                                              ; 0599
                    0000V CF              01  FB 00101            CALLS   #1, EXCH$IO_DOS11_REWIND                        ;
                          1E              50  E8 00106            BLBS    STAT2, 9$                                       ;
                                              04 00109            RET                                                     ; 0601
```

```
                                65                01  8A 0010A 8$:      BICB2   #1, (R5)                                        ; 0610
                                54 00000000G      8F  D0 0010D        MOVL    #EXCH$_DOS11_BADLABEL, TEMP                      ; 0612
                                                  52  DD 00114        PUSHL   STATUS                                           ;
                                          69      A3  9F 00116        PUSHAB  105(R3)                                          ;
                                          65      A3  DD 00119        PUSHL   101(R3)                                          ;
                                                  02  DD 0011C        PUSHL   #2                                               ;
                                                  54  DD 0011E        PUSHL   TEMP                                             ;
                                68                05  FB 00120        CALLS   #5, LIB$SIGNAL                                   ;
                                50                54  D0 00123        MOVL    TEMP, R0                                         ;
                                                      04 00126        RET                                                      ;
                                50                52  D0 00127 9$:      MOVL    STATUS, R0                                       ; 0616
                                                      04 0012A        RET                                                      ; 0617

; Routine Size:  299 bytes,    Routine Base:  EXCH$IO_CODE + 0269
```

```
;  530     0618  1 GLOBAL ROUTINE exch$io_dos11_rewind (volb : $ref_bblock) =      %SBTTL 'exch$io_dos11_rewind (volb)'
;  531     0619  2 BEGIN
;  532     0620  2 !++
;  533     0621  2 !
;  534     0622  2 ! FUNCTIONAL DESCRIPTION:
;  535     0623  2 !
;  536     0624  2 !       Rewind a dos-11 sequential device
;  537     0625  2 !
;  538     0626  2 ! INPUTS:
;  539     0627  2 !
;  540     0628  2 !       volb    - pointer to a volb which contains the active record stream
;  541     0629  2 !
;  542     0630  2 ! IMPLICIT INPUTS:
;  543     0631  2 !
;  544     0632  2 !       dos11 context block hanging off the volb
;  545     0633  2 !
;  546     0634  2 ! OUTPUTS:
;  547     0635  2 !
;  548     0636  2 !       volb$v_beg_of_tape is set
;  549     0637  2 !
;  550     0638  2 ! IMPLICIT OUTPUTS:
;  551     0639  2 !
;  552     0640  2 !       dos11 context block hanging off the volb
;  553     0641  2 !
;  554     0642  2 ! ROUTINE VALUE:
;  555     0643  2 !
;  556     0644  2 !       success or error status
;  557     0645  2 !
;  558     0646  2 ! SIDE EFFECTS:
;  559     0647  2 !
;  560     0648  2 !       none
;  561     0649  2 !--
;  562     0650  2
;  563     0651  2 $dbgtrc_prefix ('io_dos11_rewind> ');
;  564     0652  2
;  565     0653  2 LOCAL
;  566     0654  2     status
;  567     0655  2     ;
;  568     0656  2
;  569     0657  2 BIND
;  570     0658  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  571     0659  2     ;
```

```
;  573          0660  2 $debug_print_lit ('entry');
;  574          0661  2 $block_check (2, .volb, volb, 600);
;  575          0662  2 $block_check (2, .dosv, dos11, 601);
;  576          0663  2
;  577          0664  2 ! Issue the io operation to rewind the device
;  578          0665  2 !
;  579        P 0666  3 IF (status = $qiow (efn=0,
;  580        P 0667  3                     chan=.volb [volb$w_channel],
;  581        P 0668  3                     func=io$_rewind,
;  582          0669  3                     iosb=dosv [dos11$q_iosb]))
;  583          0670  2 THEN
;  584          0671  2     status = .dosv [dos11$w_iosb_status];
;  585          0672  2
;  586          0673  2 ! If either the qio or the io operation failed, scream and shout
;  587          0674  2 !
;  588          0675  2 IF NOT .status
;  589          0676  2 THEN
;  590          0677  3     BEGIN
;  591          0678  3     dosv [dos11$v_position_valid] = false;      ! After an error, we don't know where we are
;  592        P 0679  3     $exch_signal_return (exch$_dos11_ioerror, 2,
;  593          0680  3                                .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  594          0681  2     END;
;  595          0682  2
;  596          0683  2 ! Set the bits and pieces to reflect that the tape is now at the beginning
;  597          0684  2 !
;  598          0685  2 dosv [dos11$l_current_file] = 0;                ! BOT is first file (#0)
;  599          0686  2 dosv [dos11$v_position_valid] = true;           ! After a rewind, we know exactly where we are
;  600          0687  2 dosv [dos11$v_beg_of_tape] = true;
;  601          0688  2 dosv [dos11$v_end_of_tape] = false;
;  602          0689  2 dosv [dos11$v_tape_mark]   = true;             ! Make it look like we are sitting right after a tape mark
;  603          0690  2
;  604          0691  2 RETURN .status;
;  605          0692  1 END;


                                 007C 00000     .ENTRY  EXCH$IO_DOS11_REWIND, Save R2,R3,R4,R5,R6    ; 0618
                56 00000000G   EF   9E 00002    MOVAB   EXCH$UTIL_BLOCK_CHECK, R6                    ;
                54        04   AC   D0 00009    MOVL    VOLB, R4                                     ; 0658
                52 041B00 3    8F   D0 0000D    MOVL    #68878579, R2                                ; 0661
                51      0258   8F   3C 00014    MOVZWL  #600, R1                                     ;
                50             54   D0 00019    MOVL    R4, R0                                       ;
                               66   16 0001C    JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                53        54   A4   D0 0001E    MOVL    84(R4), R3                                   ; 0662
                52 00:6000:D   8F   D0 00022    MOVL    #3539197, R2                                 ;
                51      0259   8F   3C 00029    MOVZWL  #601, R1                                     ;
                50             53   D0 0002E    MOVL    R3, R0                                       ;
                               66   16 00031    JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                               7E   7C 00033    CLRQ    -(SP)                                        ; 0669
                               7E   7C 00035    CLRQ    -(SP)                                        ;
                               7E   7C 00037    CLRQ    -(SP)                                        ;
                               7E   7C 00039    CLRQ    -(SP)                                        ;
                         :E    A3   9F 0003B    PUSHAB  30(R3)                                       ;
                               24   DD 0003E    PUSHL   #36                                          ;
                7E        4A   A4   3C 00040    MOVZWL  74(R4), -(SP)                                ;
```

```
                                 7E  D4 00044          CLRL    -(SP)                                                   ;
                    00000000G    00                   0C  FB 00046          CALLS   #12, SYS$QIOW                           ;
                                 55                   50  D0 0004D          MOVL    R0, STATUS                              ;
                                 07                   55  F9 00050          BLBC    STATUS, 1$                              ;
                                 55          1E       A3  3C 00053          MOVZWL  30(R3), STATUS                          ; 0671
                                 22                   55  E8 00057          BLBS    STATUS, 2$                              ; 0675
                           0C    A3                   01  8A 0005A 1$:      BICB2   #1, 12(R3)                              ; 0678
                                 52 00000000G         8F  D0 0005E          MOVL    #EXCH$_DOS11_IOERROR, TEMP              ; 0680
                                                      55  DD 00065          PUSHL   STATUS                                  ;
                                             69       A4  9F 00067          PUSHAB  105(R4)                                 ;
                                             65       A4  DD 0006A          PUSHL   101(R4)                                 ;
                                                      02  DD 0006D          PUSHL   #2                                      ;
                                                      52  DD 0006F          PUSHL   TEMP                                    ;
                    00000000G    00                   05  FB 00071          CALLS   #5, LIB$SIGNAL                          ;
                                 50                   52  D0 00078          MOVL    TEMP, R0                                ;
                                                      04  0007B          RET                                             ;
                                             0E       A3  D4 0007C 2$:      CLRL    14(R3)                                  ; 0685
                           0C    A3                   03  88 0007F          BISB2   #3, 12(R3)                              ; 0687
                           0C    A3                   08  8A 00083          BICB2   #8, 12(R3)                              ; 0688
                           0C    A3                   04  88 00087          BISB2   #4, 12(R3)                              ; 0689
                                 50                   55  D0 0008B          MOVL    STATUS, R0                              ; 0691
                                                      04  0008E          RET                                             ; 0692

; Routine Size:  143 bytes,    Routine Base:  EXCH$IO_CODE + 0394
```

```
;  607     0693  1 GLOBAL ROUTINE exch$io_dos11_set_density (volb : $ref_bblock) =        %SBTTL 'exch$io_dos11_set_density (v
;  608     0694  2 BEGIN
;  609     0695  2 !++
;  610     0696  2 !
;  611     0697  2 ! FUNCTIONAL DESCRIPTION:
;  612     0698  2 !
;  613     0699  2 !       Set the density on a tape device
;  614     0700  2 !
;  615     0701  2 ! INPUTS:
;  616     0702  2 !
;  617     0703  2 !       volb    - pointer to a volb which contains the active record stream
;  618     0704  2 !
;  619     0705  2 ! IMPLICIT INPUTS:
;  620     0706  2 !
;  621     0707  2 !       dos11 context block hanging off the volb
;  622     0708  2 !
;  623     0709  2 ! OUTPUTS:
;  624     0710  2 !
;  625     0711  2 !       none
;  626     0712  2 !
;  627     0713  2 ! IMPLICIT OUTPUTS:
;  628     0714  2 !
;  629     0715  2 !       dos11 context block hanging off the volb
;  630     0716  2 !
;  631     0717  2 ! ROUTINE VALUE:
;  632     0718  2 !
;  633     0719  2 !       success or error status
;  634     0720  2 !
;  635     0721  2 ! SIDE EFFECTS:
;  636     0722  2 !
;  637     0723  2 !       drive density is set to the new value
;  638     0724  2 !--
;  639     0725  2
;  640     0726  2 $dbgtrc_prefix ('io_dos11_set_density> ');
;  641     0727  2
;  642     0728  2 LOCAL
;  643     0729  2     dib : $bblock [12],                         ! First three longwords of dib
;  644     0730  2     dib_desc : VECTOR [2, LONG],                ! A descriptor for the above
;  645     0731  2     status
;  646     0732  2     ;
;  647     0733  2
;  648     0734  2 BIND
;  649     0735  2     mtchr = dib [dib$l_devdepend] : $bblock [4],        ! Characteristics field as structure
;  650     0736  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  651     0737  2     ;
```

```
;  653          0738  2 $debug_print_lit ('entry');
;  654          0739  2 $block_check (2, .volb, volb, 627);
;  655          0740  2 $block_check (2, .dosv, dos11, 628);
;  656          0741  2
;  657          0742  2 ! Get the device information
;  658          0743  2 !
;  659          0744  2 dib_desc [0] = 12;
;  660          0745  2 dib_desc [1] = dib;
;  661          0746  3 IF NOT (status = $getchn (chan=.volb [volb$w_channel], scdbuf=dib_desc))
;  662          0747  2 THEN
;  663          0748  3     BEGIN
;  664          0749  3     LOCAL
;  665          0750  3         desc : VECTOR [2, LONG];
;  666          0751  3     desc [0] = .volb [volb$l_vol_ident_len];
;  667          0752  3     desc [1] = volb [volb$t_vol_ident];
;  668          0753  3     $exch_signal_stop (exch$_accessfail, 1, desc, .status);
;  669          0754  3     END;
;  670        P 0755  2 $debug_print_fao ('devchar !XL, devclass !XB, devtype !XB, devbufsiz !UL, devdepend !XL', .dib [dib$l_devcha
;  671          0756  2                 .dib [dib$b_devclass], .dib [dib$b_devtype], .dib [dib$w_devbufsiz], .dib [dib$l_devdepend])
;  672          0757  2
;  673          0758  2 ! Set the new characteristic buffer
;  674          0759  2 !
;  675          0760  3 mtchr [mt$v_density] = (SELONE cli$_present OF       ! Set the new density
;  676          0761  3                         SET
;  677          0762  3                             [cli$present (%ASCID 'DENSITY.800')] :      mt$k_nrzi_800;
;  678          0763  3                             [cli$present (%ASCID 'DENSITY.1600')] :     mt$k_pe_1600;
;  679          0764  3                             [cli$present (%ASCID 'DENSITY.6250')] :     mt$k_gcr_6250;
;  680          0765  3                             [OTHERWISE] :                               $logic_check (0, (false), 117);
;  681          0766  2                         TES);
;  682          0767  2
;  683          0768  2 ! Issue the setmode io operation to change the density
;  684          0769  2 !
;  685        P 0770  3 IF (status = $qiow (efn=0,
;  686        P 0771  3                     chan=.volb [volb$w_channel],
;  687        P 0772  3                     func=io$_setmode,
;  688        P 0773  3                     iosb=dosv [dos11$q_iosb],
;  689        P 0774  3                     p1=dib [dib$b_devclass],
;  690          0775  3                     p2=8))
;  691          0776  2 THEN
;  692          0777  2     status = .dosv [dos11$w_iosb_status];
;  693          0778  2
;  694          0779  2 ! If either the qio or the io operation failed, scream and shout
;  695          0780  2 !
;  696          0781  2 IF NOT .status
;  697          0782  2 THEN
;  698          0783  3     BEGIN
;  699          0784  3     dosv [dos11$v_position_valid] = false;      ! After an error, we don't know where we are
;  700        P 0785  3     $exch_signal_return (exch$_dos11_ioerror, 2,
;  701          0786  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  702          0787  2     END;
;  703          0788  2
;  704          0789  2 RETURN .status;
;  705          0790  1 END;


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

```
                00 30 30 38 2E 59 54 49 53 4E 45 44  00000 P.AAB:  .ASCII  \DENSITY.800\<0>                    ;
                                                010E000B  0000C P.AAA:  .LONG   17694731                       ;
                                                00000000'  00010        .ADDRESS P.AAB                      ;
                30 30 36 31 2E 59 54 49 53 4E 45 44  00014 P.AAD:  .ASCII  \DENSITY.1600\                   ;
                                                010E000C  00020 P.AAC:  .LONG   17694732                       ;
                                                00000000'  00024        .ADDRESS P.AAD                      ;
                30 35 32 36 2E 59 54 49 53 4E 45 44  00028 P.AAF:  .ASCII  \DENSITY.6250\                   ;
                                                010E000C  00034 P.AAE:  .LONG   17694732                       ;
                                                00000000'  00038        .ADDRESS P.AAF                      ;

                                                                        .EXTRN  SYS$GETCHN, EXCH$_ACCESSFAIL
                                                                        .EXTRN  LIB$STOP, CLI$_PRESENT
                                                                        .EXTRN  CLI$PRESENT

                                                                        .PSECT  EXCH$IO_CODE,NOWRT,2

                                                    01FC 00000          .ENTRY  EXCH$IO_DOS11_SET_DENSITY, Save R2,R3,R4,-  ; 0693
                                                                                R5,R6,R7,R8                         ;
                                58 00000000G   EF   9E 00002           MOVAB   EXCH$UTIL_BLOCK_CHECK, R8           ;
                                57 00000000G   00   9E 00009           MOVAB   LIB$STOP, R7                        ;
                                56 00000000G   00   9E 00010           MOVAB   CLI$PRESENT, R6                     ;
                                5E             1C   C2 00017           SUBL2   #28, SP                             ;
                                53       04    AC   D0 0001A           MOVL    VOLB, R3                            ; 0736
                                52 041B00F3    8F   D0 0001E           MOVL    #68878579, R2                       ; 0739
                                51     0273    8F   3C 00025           MOVZWL  #627, R1                            ;
                                50             53   D0 0002A           MOVL    R3, R0                              ;
                                               68   16 0002D           JSB     EXCH$UTIL_BLOCK_CHECK               ;
                                54       54    A3   D0 0002F           MOVL    84(R3), R4                          ; 0740
                                52 003600FD    8F   D0 00033           MOVL    #3539197, R2                        ;
                                51     0274    8F   3C 0003A           MOVZWL  #628, R1                            ;
                                50             54   D0 0003F           MOVL    R4, R0                              ;
                                               68   16 00042           JSB     EXCH$UTIL_BLOCK_CHECK               ;
                          08    AE             0C   D0 00044           MOVL    #12, DIB_DESC                       ; 0744
                          0C    AE       10    AE   9E 00048           MOVAB   DIB, DIB_DESC+4                     ; 0745
                                         08    AE   9F 0004D           PUSHAB  DIB_DESC                            ; 0746
                                               7E   7C 00050           CLRQ    -(SP)                               ;
                                               7E   D4 00052           CLRL    -(SP)                               ;
                                7E       4A    A3   3C 00054           MOVZWL  74(R3), -(SP)                       ;
                00000000G       00             05   FB 00058           CALLS   #5, SYS$GETCHN                      ;
                                55             50   D0 0005F           MOVL    R0, STATUS                          ;
                                1A             55   E8 00062           BLBS    STATUS, 1$                          ;
                                6E       65    A3   D0 00065           MOVL    101(R3), DESC                       ; 0751
                          04    AE       69    A3   9E 00069           MOVAB   105(R3), DESC+4                     ; 0752
                                               55   DD 0006E           PUSHL   STATUS                              ; 0753
                                         04    AE   9F 00070           PUSHAB  DESC                                ;
                                               01   DD 00073           PUSHL   #1                                  ;
                                00000000G      8F   DD 00075           PUSHL   #EXCH$_ACCESSFAIL                   ;
                                67             04   FB 0007B           CALLS   #4, LIB$STOP                        ;
                                               04   0007E           RET                                         ;
                                52 00000000G   8F   D0 0007F 1$:     MOVL    #CLI$_PRESENT, R2                   ; 0760
                                     0000'     CF   9F 00086           PUSHAB  P.AAA                               ; 0762
                                66             01   FB 0008A           CALLS   #1, CLI$PRESENT                     ;
                                50             52   D1 0008D           CMPL    R2, R0                              ;
                                               05   12 00090           BNEQ    2$                                  ;
                                50             03   D0 00092           MOVL    #3, R0                              ;
                                               31   11 00095           BRB     5$                                  ;
```

```
                                 0000'  CF  9F 00097 2$:    PUSHAB  P.AAC                           ; 0763
                              66        01  FB 0009B        CALLS   #1, CLI$PRESENT                 ;
                              50        52  D1 0009E        CMPL    R2, R0                          ;
                                        05  12 000A1        BNEQ    3$                              ;
                              50        04  D0 000A3        MOVL    #4, R0                          ;
                                        20  11 000A6        BRB     5$                              ;
                                 0000'  CF  9F 000A8 3$:    PUSHAB  P.AAE                           ; 0764
                              66        01  FB 000AC        CALLS   #1, CLI$PRESENT                 ;
                              50        52  D1 000AF        CMPL    R2, R0                          ;
                                        05  12 000B2        BNEQ    4$                              ;
                              50        05  D0 000B4        MOVL    #5, R0                          ;
                                        0F  11 000B7        BRB     5$                              ;
                              7E     75 8F  9A 000B9 4$:    MOVZBL  #117, -(SP)                     ; 0765
                                        01  DD 000BD        PUSHL   #1                              ;
                                 00000000G  8F  DD 000BF    PUSHL   #EXCH$_BADLOGIC                 ;
                              67        03  FB 000C5        CALLS   #3, LIB$STOP                    ;
    19  AE          05        00        50  F0 000C8 5$:    INSV    R0, #0, #5, MTCHR+1             ; 0760
                                        7E  7C 000CE        CLRQ    -(SP)                           ; 0775
                                        7E  7C 000D0        CLRQ    -(SP)                           ;
                                        08  DD 000D2        PUSHL   #8                              ;
                                    28  AE  9F 000D4        PUSHAB  DIB+4                           ;
                                        7E  7C 000D7        CLRQ    -(SP)                           ;
                                    1E  A4  9F 000D9        PUSHAB  30(R4)                          ;
                                        23  DD 000DC        PUSHL   #35                             ;
                              7E        4A  A3  3C 000DE    MOVZWL  74(R3), -(SP)                   ;
                                        7E  D4 000E2        CLRL    -(SP)                           ;
                        00000000G       00  FB 000E4        CALLS   #12, SYS$QIOW                   ;
                              55        50  D0 000EB        MOVL    R0, STATUS                      ;
                              07        55  E9 000EE        BLBC    STATUS, 6$                      ;
                              55        1E  A4  3C 000F1    MOVZWL  30(R4), STATUS                  ; 0777
                              22        55  E8 000F5        BLBS    STATUS, 7$                      ; 0781
                           0C A4                01  8A 000F8 6$:    BICB2   #1, 12(R4)              ; 0784
                              54 00000000G  8F  D0 000FC    MOVL    #EXCH$_DOS11_IOERROR, TEMP      ; 0786
                                        55  DD 00103        PUSHL   STATUS                          ;
                                    69  A3  9F 00105        PUSHAB  105(R3)                         ;
                                    65  A3  DD 00108        PUSHL   101(R3)                         ;
                                        02  DD 0010B        PUSHL   #2                              ;
                                        54  DD 0010D        PUSHL   TEMP                            ;
                        00000000G       00  FB 0010F        CALLS   #5, LIB$SIGNAL                  ;
                              50        54  D0 00116        MOVL    TEMP, R0                        ;
                                        04  00119        RET                                     ;
                              50        55  D0 0011A 7$:    MOVL    STATUS, R0                      ; 0789
                                        04  0011D        RET                                     ; 0790

; Routine Size:  286 bytes,    Routine Base:  EXCH$IO_CODE + 0423
```

```
;  707     0791  1 GLOBAL ROUTINE exch$io_dos11_skip_file (volb : $ref_bblock, files) =    %SBTTL 'exch$io_dos11_skip_file (vol
;  708     0792  2 BEGIN
;  709     0793  2 !++
;  710     0794  2 !
;  711     0795  2 ! FUNCTIONAL DESCRIPTION:
;  712     0796  2 !
;  713     0797  2 !       Skip a given number of files
;  714     0798  2 !
;  715     0799  2 ! INPUTS:
;  716     0800  2 !
;  717     0801  2 !       volb   - pointer to a volb which contains the active record stream
;  718     0802  2 !       files  - number of files to skip (positive - skip forward, negative - skip reverse)
;  719     0803  2 !
;  720     0804  2 ! IMPLICIT INPUTS:
;  721     0805  2 !
;  722     0806  2 !       dos11 context block hanging off the volb
;  723     0807  2 !
;  724     0808  2 ! OUTPUTS:
;  725     0809  2 !
;  726     0810  2 !       none
;  727     0811  2 !
;  728     0812  2 ! IMPLICIT OUTPUTS:
;  729     0813  2 !
;  730     0814  2 !       dos11 context block hanging off the volb
;  731     0815  2 !
;  732     0816  2 ! ROUTINE VALUE:
;  733     0817  2 !
;  734     0818  2 !       success or error status
;  735     0819  2 !
;  736     0820  2 ! SIDE EFFECTS:
;  737     0821  2 !
;  738     0822  2 !       none
;  739     0823  2 !--
;  740     0824  2
;  741     0825  2 $dbgtrc_prefix ('io_dos11_skip_file> ');
;  742     0826  2
;  743     0827  2 LOCAL
;  744     0828  2     skipc,
;  745     0829  2     iosb_skip,
;  746     0830  2     skip_adjust,
;  747     0831  2     status
;  748     0832  2     ;
;  749     0833  2
;  750     0834  2 BIND
;  751     0835  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  752     0836  2     ;
```

```
;  754          0837  2 $debug_print_fao ('initial file count !SL', .files);
;  755          0838  2 $block_check (2, .volb, volb, 614);
;  756          0839  2 $block_check (2, .dosv, dos11, 612);
;  757          0840  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 270);        ! We should know where we are
;  758          0841  2
;  759          0842  2 IF .files EQL 0
;  760          0843  2 THEN
;  761          0844  2     RETURN 1;
;  762          0845  2
;  763          0846  2 ! If the skip count is negative, we will have to skip one extra file and then space forward.  In addition, i
;  764          0847  2 ! we are at the end of tape we need an extra skip to get past the extra tapemark.  We assume that
;  765          0848  2 ! nobody will try to skipfile to file 0 (they should rewind instead).
;  766          0849  2 !
;  767          0850  2 IF (skipc = .files) LSS 0
;  768          0851  2 THEN
;  769          0852  3     BEGIN
;  770          0853  3     skip_adjust = (IF .dosv [dos11$v_end_of_tape] THEN 2 ELSE 1);
;  771          0854  3     skipc = .skipc - .skip_adjust;
;  772          0855  2     END;
;  773          0856  2
;  774          0857  2 ! Issue the io operation to skip to the next file
;  775          0858  2 !
;  776          0859  2 $debug_print_fao ('actual skip count !SL', .skipc);
;  777        P 0860  3 IF (status = $qiow (efn=0,
;  778        P 0861  3                     chan=.volb [volb$w_channel],
;  779        P 0862  3                     func=io$_skipfile,
;  780        P 0863  3                     iosb=dosv [dos11$q_iosb],
;  781          0864  3                     p1=.skipc))
;  782          0865  2 THEN
;  783          0866  2     status = .dosv [dos11$w_iosb_status];
;  784          0867  2 iosb_skip = .dosv [dos11$w_iosb_skipcnt];        ! Save skip count in case we do another I/O
;  785          0868  2
;  786          0869  2 ! If either the qio or the io operation failed, scream and shout
;  787          0870  2 !
;  788          0871  3 IF  (NOT .status)
;  789          0872  2   AND
;  790          0873  3     (.status NEQ ss$_endofvolume)
;  791          0874  2 THEN
;  792          0875  3     BEGIN
;  793          0876  3     dosv [dos11$v_position_valid] = false;      ! After an error, we don't know where we are
;  794        P 0877  3     $exch_signal_return (exch$_dos11_ioerror, 2,
;  795          0878  3                          .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  796          0879  2     END;
;  797          0880  2
;  798          0881  2 ! If the skip was negative, we will be positioned right before the tapemark which precedes the file we want.
;  799          0882  2 ! Therefore, we must read that tapemark so that we will be positioned right before the label.
;  800          0883  2 !
;  801          0884  2 IF .skipc LSS 0
;  802          0885  2 THEN
;  803          0886  3     BEGIN
;  804          0887  3     LOCAL
;  805          0888  3         buf : $bvector [32];
;  806          0889  3
;  807          0890  3     iosb_skip = .iosb_skip - .skip_adjust;      ! Adjust back to the desired skip
;  808          0891  3
;  809          0892  3     ! Issue the io operation to skip over the tapemark
;  810          0893  3     !
```

```
;  811     P 0894  4     IF (status = $qiow (efn=0,
;  812     P 0895  4                         chan=.volb [volb$w_channel],
;  813     P 0896  4                         func=io$_readlblk,
;  814     P 0897  4                         iosb=dosv [dos11$q_iosb],
;  815     P 0898  4                         p1=buf,
;  816       0899  4                         p2=32))
;  817       0900  3     THEN
;  818       0901  3         status = .dosv [dos11$w_iosb_status];
;  819       0902  3
;  820       0903  3     ! If either the qio or the io operation failed, scream and shout
;  821       0904  3     !
;  822       0905  3     IF NOT .status
;  823       0906  3     THEN
;  824       0907  4         BEGIN
;  825       0908  4         IF .status NEQ ss$_endoffile
;  826       0909  4         THEN
;  827       0910  5             BEGIN
;  828       0911  5             dosv [dos11$v_position_valid] = false;      ! After an error, we don't know where we are
;  829     P 0912  5             $exch_signal_return (exch$_dos11_ioerror, 2,
;  830       0913  5                                  .volb [volb$t_vol_ident_len], volb [volb$t_vol_ident], .status);
;  831       0914  5             END
;  832       0915  4         ELSE
;  833       0916  4             status = ss$_normal;                ! Turn endofvolume into normal, return the normal status
;  834       0917  3         END;
;  835       0918  2     END;
;  836       0919  2
;  837       0920  2 ! Set the bits and pieces to reflect that the tape has moved
;  838       0921  2 !
;  839       0922  2 $debug_print_fao ('current file before !UL', .dosv [dos11$l_current_file]);
;  840       0923  2 dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + (IF .skipc GEQ 0 THEN .iosb_skip ELSE -.iosb_sk
;  841       0924  2 $debug_print_fao ('current file after  !UL', .dosv [dos11$l_current_file]);
;  842       0925  2 dosv [dos11$v_beg_of_tape] = false;     !?? might be true
;  843       0926  2 dosv [dos11$v_tape_mark]   = true;
;  844       0927  2
;  845       0928  2 IF .status EQL ss$_endofvolume
;  846       0929  2 THEN
;  847       0930  3     BEGIN
;  848       0931  3     status = true;                              ! Want a success status
;  849       0932  3     dosv [dos11$v_end_of_tape] = true;
;  850       0933  3     END
;  851       0934  2 ELSE
;  852       0935  2     dosv [dos11$v_end_of_tape] = false;
;  853       0936  2
;  854       0937  2 RETURN .status;
;  855       0938  1 END;


                                    0FFC 00000          .ENTRY  EXCH$IO_DOS11_SKIP_FILE, Save R2,R3,R4,R5,- ; 0791
                                                                R6,R7,R8,R9,R10,R11                             ;
                5B 00000000G   EF   9E 00002            MOVAB   EXCH$UTIL_BLOCK_CHECK, R11                  ;
                5A 00000000G   00   9E 00009            MOVAB   SYS$QIOW, R10                               ;
                5E             20   C2 00010            SUBL2   #32, SP                                     ;
                54       04    AC   D0 00013            MOVL    VOLB, R4                                    ; 0835
                52 041B00F3    8F   D0 00017            MOVL    #68878579, R2                               ; 0838
```

```
                                51      0266   8F  3C 0001E          MOVZWL  #614, R1                                        ;
                                50             54  D0 00023          MOVL    R4, R0                                          ;
                                               6B  16 00026          JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                                53        54   A4  D0 00028          MOVL    84(R4), R3                                      ; 0839
                                52 003600FD   8F  D0 0002C          MOVL    #3539197, R2                                    ;
                                51      0264   8F  3C 00033          MOVZWL  #612, R1                                        ;
                                50             53  D0 00038          MOVL    R3, R0                                          ;
                                               6B  16 0003B          JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                                59        0C   A3  9E 0003D          MOVAB   12(R3), R9                                      ; 0840
                                14             69  E8 00041          BLBS    (R9), 1$                                        ;
                                7E      010E   8F  3C 00044          MOVZWL  #270, -(SP)                                     ;
                                               01  DD 00049          PUSHL   #1                                              ;
                                   00000000G   8F  DD 0004B          PUSHL   #EXCH$_BADLOGIC                                 ;
                     00000000G  00             03  FB 00051          CALLS   #3, LIB$STOP                                    ;
                                               08  AC  D5 00058 1$:   TSTL    FILES                                           ; 0842
                                               04  12 0005B          BNEQ    2$                                              ;
                                50             01  D0 0005D          MOVL    #1, R0                                          ; 0844
                                               04  00060             RET                                                     ;
                                58        08   AC  D0 00061 2$:      MOVL    FILES, SKIPC                                    ; 0850
                                               0F  18 00065          BGEQ    5$                                              ;
                          05    69             03  E1 00067          BBC     #3, (R9), 3$                                    ; 0853
                                57             02  D0 0006B          MOVL    #2, SKIP_ADJUST                                 ;
                                               03  11 0006E          BRB     4$                                              ;
                                57             01  D0 00070 3$:      MOVL    #1, SKIP_ADJUST                                 ;
                                58             57  C2 00073 4$:      SUBL2   SKIP_ADJUST, SKIPC                              ; 0854
                                               7E  7C 00076 5$:      CLRQ    -(SP)                                           ; 0864
                                               7E  7C 00078          CLRQ    -(SP)                                           ;
                                               7E  D4 0007A          CLRL    -(SP)                                           ;
                                               58  DD 0007C          PUSHL   SKIPC                                           ;
                                               7E  7C 0007E          CLRQ    -(SP)                                           ;
                                         1E    A3  9F 00080          PUSHAB  30(R3)                                          ;
                                               25  DD 00083          PUSHL   #37                                             ;
                                7E       4A    A4  3C 00085          MOVZWL  74(R4), -(SP)                                   ;
                                               7E  D4 00089          CLRL    -(SP)                                           ;
                                6A                 0C  FB 0008B          CALLS   #12, SYS$QIOW                                   ;
                                55                 50  D0 0008E          MOVL    R0, STATUS                                      ;
                                04                 55  E9 00091          BLBC    STATUS, 6$                                      ;
                                55       1E    A3  3C 00094          MOVZWL  30(R3), STATUS                                  ; 0866
                                52       20    A3  32 00098 6$:      CVTWL   32(R3), IOSB_SKIP                               ; 0867
                                09                 55  E8 0009C          BLBS    STATUS, 7$                                      ; 0871
                     000009A0   8F                 55  D1 0009F          CMPL    STATUS, #2464                           ; 0873
                                                   36  12 000A6          BNEQ    9$                                              ;
                                                   58  D5 000A8 7$:      TSTL    SKIPC                                           ; 0884
                                                   56  18 000AA          BGEQ    11$                                             ;
                                52                 57  C2 000AC          SUBL2   SKIP_ADJUST, IOSB_SKIP                          ; 0890
                                                   7E  7C 000AF          CLRQ    -(SP)                                           ; 0899
                                                   7E  7C 000B1          CLRQ    -(SP)                                           ;
                                                   20  DD 000B3          PUSHL   #32                                             ;
                                         14    AE  9F 000B5          PUSHAB  BUF                                             ;
                                                   7E  7C 000B8          CLRQ    -(SP)                                           ;
                                         1E    A3  9F 000BA          PUSHAB  30(R3)                                          ;
                                                   21  DD 000BD          PUSHL   #33                                             ;
                                7E       4A    A4  3C 000BF          MOVZWL  74(R4), -(SP)                                   ;
                                               7E  D4 000C3          CLRL    -(SP)                                           ;
                                6A                 0C  FB 000C5          CALLS   #12, SYS$QIOW                                   ;
                                55                 50  D0 000C8          MOVL    R0, STATUS                                      ;
                                07                 55  E9 000CB          BLBC    STATUS, 8$                                      ;
```

```
                         55       1E   A3  3C 000CE           MOVZWL  30(R3), STATUS                          ; 0901
                         2D            55  E8 00CD2           BLBS    STATUS, 11$                             ; 0905
                 00000870 8F            55  D1 000D5 8$:       CMPL    STATUS, #2160                           ; 0908
                                        21  13 000DC           BEQL    10$                                     !
                         69            01  8A 000DE 9$:       BICB2   #1, (R9)                                ; 0911
                         56 00000000G  8F  D0 000E1           MOVL    #EXCH$_DOS11_IOERROR, TEMP              ; 0913
                                       55  DD 000E8           PUSHL   STATUS                                  !
                                    69 A4  9F 000EA           PUSHAB  105(R4)                                 !
                                    65 A4  DD 000ED           PUSHL   101(R4)                                 !
                                       02  DD 000F0           PUSHL   #2                                      !
                                       56  DD 000F2           PUSHL   TEMP                                    !
                 00000000G             00  FB 000F4           CALLS   #5, LIB$SIGNAL                          !
                                       50  D0 000FB           MOVL    TEMP, R0                                !
                                           04 000FE           RET                                             !
                         55            01  D0 000FF 10$:      MOVL    #1, STATUS                              ; 0916
                                       58  D5 00102 11$:      TSTL    SKIPC                                   ; 0923
                                       05  19 00104           BLSS    12$                                     !
                         50            52  D0 00106           MOVL    IOSB_SKIP, R0                           !
                                       03  11 00109           BRB     13$                                     !
                         50            52  CE 0010B 12$:      MNEGL   IOSB_SKIP, R0                           !
                      0E A3            50  C0 0010E 13$:      ADDL2   R0, T4(R3)                              !
                         69            02  8A 00112           BICB2   #2, (R9)                                ; 0925
                         69            04  88 00115           BISB2   #4, (R9)                                ; 0926
                 000009A0 8F            55  D1 00118           CMPL    STATUS, #2464                           ; 0928
                                        08  12 0011F           BNEQ    14$                                     !
                         55            01  D0 00121           MOVL    #1, STATUS                              ; 0931
                         69            08  88 00124           BISB2   #8, (R9)                                ; 0932
                                       03  11 00127           BRB     15$                                     ; 0928
                         69            08  8A 00129 14$:      BICB2   #8, (R9)                                ; 0935
                         50            55  D0 0012C 15$:      MOVL    STATUS, R0                              ; 0937
                                           04 0012F           RET                                             ; 0938

; Routine Size:  304 bytes,    Routine Base:  EXCH$IO_CODE + 0541
```

```
;  857     0939  1 GLOBAL ROUTINE exch$io_dos11_skip_record (volb : $ref_bblock, records) =        %SBTTL 'exch$io_dos11_skip_r
;  858     0940  2 BEGIN
;  859     0941  2 !++
;  860     0942  2 !
;  861     0943  2 ! FUNCTIONAL DESCRIPTION:
;  862     0944  2 !
;  863     0945  2 !       Skip a given number of physcial records on a dos-11 sequential device
;  864     0946  2 !
;  865     0947  2 ! INPUTS:
;  866     0948  2 !
;  867     0949  2 !       volb    - pointer to a volb which contains the active record stream
;  868     0950  2 !       records - number of records to skip (positive => forward, negative => backward)
;  869     0951  2 !
;  870     0952  2 ! IMPLICIT INPUTS:
;  871     0953  2 !
;  872     0954  2 !       dos11 context block hanging off the volb
;  873     0955  2 !
;  874     0956  2 ! OUTPUTS:
;  875     0957  2 !
;  876     0958  2 !       none
;  877     0959  2 !
;  878     0960  2 ! IMPLICIT OUTPUTS:
;  879     0961  2 !
;  880     0962  2 !       dos11 context block hanging off the volb
;  881     0963  2 !
;  882     0964  2 ! ROUTINE VALUE:
;  883     0965  2 !
;  884     0966  2 !       success or error status
;  885     0967  2 !
;  886     0968  2 ! SIDE EFFECTS:
;  887     0969  2 !
;  888     0970  2 !       none
;  889     0971  2 !--
;  890     0972  2
;  891     0973  2 $dbgtrc_prefix ('io_dos11_skip_record> ');
;  892     0974  2
;  893     0975  2 LOCAL
;  894     0976  2     status
;  895     0977  2     ;
;  896     0978  2
;  897     0979  2 BIND
;  898     0980  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;  899     0981  2     ;
;  900     0982  2
;  901     0983  2 $debug_print_lit ('entry');
;  902     0984  2 $block_check (2, .volb, volb, 635);
;  903     0985  2 $block_check (2, .dosv, dos11, 636);
;  904     0986  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 287);        ! We should know where we are
;  905     0987  2
;  906     0988  2 ! Start by assuming no records
;  907     0989  2 !
;  908     0990  2 dosv [dos11$w_block_count] = 0;
;  909     0991  2
;  910     0992  2 ! Issue the io operation to space in the desired direction
;  911     0993  2 !
;  912   P 0994  3 IF (status = $qiow (efn=0,
;  913   P 0995  3                      chan=.volb [volb$w_channel],
```

```
;  914        P 0996  3                    func=io$_skiprecord,
;  915        P 0997  3                    iosb=dosv [dos11$q_iosb],
;  916          0998  3                    p1=.records))
;  917          0999  2 THEN
;  918          1000  2     status = .dosv [dos11$w_iosb_status];
;  919          1001  2
;  920          1002  2 ! Finish up based on the status
;  921          1003  2 !
;  922          1004  2 SELECTONE .status OF
;  923          1005  2 SET
;  924          1006  2     [ss$_normal] :
;  925          1007  3                    BEGIN
;  926          1008  3
;  927          1009  3                    ! Set the bits and pieces to reflect that the tape has moved
;  928          1010  3                    !
;  929          1011  3                    dosv [dos11$v_tape_mark]   = false;
;  930          1012  3                    dosv [dos11$v_end_of_tape] = false;
;  931          1013  3
;  932          1014  2                    END;
;  933          1015  2
;  934          1016  2     ! End of file means that we have found the end
;  935          1017  2     !
;  936          1018  2     [ss$_endoffile] :
;  937          1019  3                    BEGIN
;  938          1020  3
;  939          1021  3                    status = true;              ! Return true instead of endoffile
;  940          1022  3
;  941          1023  3                    ! Set the bits and pieces to reflect that the tape has moved
;  942          1024  3                    !
;  943          1025  3                    dosv [dos11$v_tape_mark]   = true;
;  944          1026  3                    dosv [dos11$v_end_of_tape] = false;
;  945          1027  3                    dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;
;  946          1028  3
;  947          1029  2                    END;
;  948          1030  2
;  949          1031  2     ! If this is the end, oops
;  950          1032  2     !
;  951          1033  2     [ss$_endoftape] :
;  952          1034  3                    BEGIN
;  953          1035  3
;  954          1036  3                    ! Signal that we have had a problem
;  955          1037  3                    !
;  956        P 1038  3                    $exch_signal (exch$_dos11_ioerror, 2,
;  957          1039  3                                  .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  958          1040  3
;  959          1041  3                    ! Set the bits and pieces to reflect that the tape has moved
;  960          1042  3                    !
;  961          1043  3                    dosv [dos11$v_tape_mark]   = false;
;  962          1044  3                    dosv [dos11$v_end_of_tape] = true;
;  963          1045  3                    dosv [dos11$l_current_file] = .dosv [dos11$l_current_file] + 1;
;  964          1046  3
;  965          1047  2                    END;
;  966          1048  2
;  967          1049  2     ! Any thing else and we'd better crap out
;  968          1050  2     !
;  969          1051  3     [OTHERWISE] :   BEGIN
;  970          1052  3                    dosv [dos11$v_position_valid] = false;      ! After an error, we don't know where we are
```

```
;  971        P 1053  3                    $exch_signal_return (exch$_dos11_ioerror, 2,
;  972          1054  3                                        .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  973          1055  2                    END;
;  974          1056  2 TES;
;  975          1057  2
;  976          1058  2 ! Set position info common to all paths
;  977          1059  2 !
;  978          1060  2 dosv [dos11$v_beg_of_tape] = false;
;  979          1061  2
;  980          1062  2 ! Count the blocks
;  981          1063  2 !
;  982          1064  2 dosv [dos11$w_block_count] = .dosv [dos11$w_iosb_skipcnt];
;  983          1065  2
;  984          1066  2 RETURN .status;
;  985          1067  1 END;
```

```
                                03FC 00000          .ENTRY  EXCH$IO_DOS11_SKIP_RECORD, Save R2,R3,R4,-      ; 0939
                                                            R5,R6,R7,R8,R9
                    59 00000000G   EF   9E 00002    MOVAB   EXCH$UTIL_BLOCK_CHECK, R9                       ;
                    58 00000000G   00   9E 00009    MOVAB   LIB$SIGNAL, R8                                  ;
                    57 00000000G   8F   D0 00010    MOVL    #EXCH$_DOS11_IOERROR, R7                        ;
                    54         04  AC   D0 00017    MOVL    VOLB, R4                                        ; 0980
                    52 041B00F3    8F   D0 0001B    MOVL    #68878579, R2                                   ; 0984
                    51     027B    8F   3C 00022    MOVZWL  #635, R1                                        ;
                    50             54   D0 00027    MOVL    R4, R0                                          ;
                                   69   16 0002A    JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                    53         54  A4   D0 0002C    MOVL    84(R4), R3                                      ; 0985
                    52 003600FD    8F   D0 00030    MOVL    #3539197, R2                                    ;
                    51     027C    8F   3C 00037    MOVZWL  #636, R1                                        ;
                    50             53   D0 0003C    MOVL    R3, R0                                          ;
                                   69   16 0003F    JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                    52         0C  A3   9E 00041    MOVAB   12(R3), R2                                      ; 0986
                    14             62   E8 00045    BLBS    (R2), 1$                                        ;
                    7E     011F    8F   3C 00048    MOVZWL  #287, -(SP)                                     ;
                                   01   DD 0004D    PUSHL   #1                                              ;
                        00000000G  8F   DD 0004F    PUSHL   #EXCH$_BADLOGIC                                 ;
00000000G           00             03   FB 00055    CALLS   #3, LIB$STOP                                    ;
                              34   A3   B4 0005C 1$: CLRW    52(R3)                                          ; 0990
                                   7E   7C 0005F    CLRQ    -(SP)                                           ; 0998
                                   7E   7C 00061    CLRQ    -(SP)                                           ;
                                   7E   D4 00063    CLRL    -(SP)                                           ;
                              08   AC   DD 00065    PUSHL   RECORDS                                         ;
                                   7E   7C 00068    CLRQ    -(SP)                                           ;
                              1E   A3   9F 0006A    PUSHAB  30(R3)                                          ;
                                   26   DD 0006D    PUSHL   #38                                             ;
                    7E        4A   A4   3C 0006F    MOVZWL  74(R4), -(SP)                                   ;
                                   7E   D4 00073    CLRL    -(SP)                                           ;
00000000G           00             0C   FB 00075    CALLS   #12, SYS$QIOW                                   ;
                    55             50   D0 0007C    MOVL    R0, STATUS                                      ;
                    04             55   E9 0007F    BLBC    STATUS, 2$                                      ;
                    55         1E  A3   3C 00082    MOVZWL  30(R3), STATUS                                  ; 1000
                    01             55   D1 00086 2$: CMPL    STATUS, #1                                      ; 1006
                                   05   12 00089    BNEQ    3$                                              ;
```

```
                          62            0C  8A 0008B          BICB2   #12, (R2)                                   ; 1012
                          50            11  0008E            BRB     7$                                          ; 1004
                00000870  8F            55  D1 00090 3$:     CMPL    STATUS, #2160                               ; 1018
                                        0B  12 00097          BNEQ    4$
                          55            01  D0 00099          MOVL    #1, STATUS                                  ; 1021
                          62            04  88 0009C          BISB2   #4, (R2)                                    ; 1025
                          62            08  8A 0009F          BICB2   #8, (R2)                                    ; 1026
                                        1E  11 000A2          BRB     5$                                          ; 1027
                00000878  8F            55  D1 000A4 4$:     CMPL    STATUS, #2168                               ; 1033
                                        1A  12 000AB          BNEQ    6$
                                        55  DD 000AD          PUSHL   STATUS                                      ; 1039
                                69      A4  9F 000AF          PUSHAB  105(R4)
                                65      A4  DD 000B2          PUSHL   101(R4)
                                        02  DD 000B5          PUSHL   #2
                                        57  DD 000B7          PUSHL   R7
                          68            05  FB 000B9          CALLS   #5, LIB$SIGNAL
                          62            04  8A 000BC          BICB2   #4, (R2)                                    ; 1043
                          62            08  88 000BF          BISB2   #8, (R2)                                    ; 1044
                                0E      A3  D6 000C2 5$:      INCL    14(R3)                                      ; 1045
                                        19  11 000C5          BRB     7$                                          ; 1004
                          62            01  8A 000C7 6$:     BICB2   #1, (R2)                                    ; 1052
                          56            57  D0 000CA          MOVL    R7, TEMP                                    ; 1054
                                        55  DD 000CD          PUSHL   STATUS
                                69      A4  9F 000CF          PUSHAB  105(R4)
                                65      A4  DD 000D2          PUSHL   101(R4)
                                        02  DD 000D5          PUSHL   #2
                                        56  DD 000D7          PUSHL   TEMP
                          68            05  FB 000D9          CALLS   #5, LIB$SIGNAL
                          50            56  D0 000DC          MOVL    TEMP, R0
                                        04  000DF            RET
                          62            02  8A 000E0 7$:      BICB2   #2, (R2)                                    ; 1060
                       34 A3        20  A3  B0 000E3          MOVW    32(R3), 52(R3)                              ; 1064
                          50            55  D0 000E8          MOVL    STATUS, R0                                  ; 1066
                                        04  000EB            RET                                                 ; 1067

; Routine Size:  236 bytes,    Routine Base:  EXCH$IO_CODE + 0671
```

```
;  987          1068  1 GLOBAL ROUTINE exch$io_dos11_write (volb : $ref_bblock, buffer, len) =  %SBTTL 'exch$io_dos11_write (volb, b
;  988          1069  2 BEGIN
;  989          1070  2 !++
;  990          1071  2 !
;  991          1072  2 ! FUNCTIONAL DESCRIPTION:
;  992          1073  2 !
;  993          1074  2 !       Write one block on a dos-11 sequential device
;  994          1075  2 !
;  995          1076  2 ! INPUTS:
;  996          1077  2 !
;  997          1078  2 !       volb   - pointer to a volb which contains the active record stream
;  998          1079  2 !       buffer - pointer to the buffer containing the block
;  999          1080  2 !       len    - length of the buffer
; 1000          1081  2 !
; 1001          1082  2 ! IMPLICIT INPUTS:
; 1002          1083  2 !
; 1003          1084  2 !       dos11 context block hanging off the volb
; 1004          1085  2 !
; 1005          1086  2 ! OUTPUTS:
; 1006          1087  2 !
; 1007          1088  2 !       none
; 1008          1089  2 !
; 1009          1090  2 ! IMPLICIT OUTPUTS:
; 1010          1091  2 !
; 1011          1092  2 !       dos11 context block hanging off the volb
; 1012          1093  2 !
; 1013          1094  2 ! ROUTINE VALUE:
; 1014          1095  2 !
; 1015          1096  2 !       success if label, false if not a label or error status
; 1016          1097  2 !
; 1017          1098  2 ! SIDE EFFECTS:
; 1018          1099  2 !
; 1019          1100  2 !       none
; 1020          1101  2 !--
; 1021          1102  2
; 1022          1103  2 $dbgtrc_prefix ('io_dos11_write> ');
; 1023          1104  2
; 1024          1105  2 LOCAL
; 1025          1106  2     func,
; 1026          1107  2     io_len,
; 1027          1108  2     status
; 1028          1109  2     ;
; 1029          1110  2
; 1030          1111  2 BIND
; 1031          1112  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
; 1032          1113  2     ;
; 1033          1114  2
; 1034          1115  2 $debug_print_lit ('entry');
; 1035          1116  2 $block_check (2, .volb, volb, 569);
; 1036          1117  2 $block_check (2, .dosv, dos11, 596);
; 1037          1118  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 272);          ! We should know where we are
; 1038          1119  2 $logic_check (2, (NOT .dosv [dos11$v_tape_mark]), 282);           ! And that not should be right after a tape
; 1039          1120  2 $logic_check (2, (.dosv [dos11$v_end_of_tape]), 288);            ! But instead at the end of the tape
```

```
;  1041          1121  2 ! The minimum size io to a tape is 14 bytes.  We assume that any shorter buffers are null padded to at least
;  1042          1122  2 ! 14 bytes.  If the input length is 0 an outer routine has made a mistake.  In addition, all transfers are
;  1043          1123  2 ! even length.
;  1044          1124  2 !
;  1045          1125  2 $logic_check (2, (.len NEQ 0), 290);                ! Make sure it is non-zero
;  1046          1126  2 io_len = MAXU (.len, 14);                           ! Make sure it is at least 14 bytes long
;  1047          1127  2 IF .io_len <0,1,0> THEN io_len = .io_len + 1;       ! If odd, then make it even
;  1048          1128  2 $logic_check (2, (.io_len [EQU 512), 291);         ! Make sure it isn't too long
;  1049          1129  2
;  1050          1130  2 ! Issue the io operation to write the next logical block in the forward direction
;  1051          1131  2 !
;  1052          1132  2 func = io$_writelblk;
;  1053          1133  2 IF .volb [volb$v_read_check] THEN func = .func OR io$m_datacheck;
;  1054        P 1134  3 IF (status = $qiow (efn=0,
;  1055        P 1135  3                     chan=.volb [volb$w_channel],
;  1056        P 1136  3                     func=.func,
;  1057        P 1137  3                     iosb=dosv [dos11$q_iosb],
;  1058        P 1138  3                     p1=.buffer,
;  1059          1139  3                     p2=.io_len))
;  1060          1140  2 THEN
;  1061          1141  2     status = .dosv [dos11$w_iosb_status];
;  1062          1142  2
;  1063          1143  2 ! Finish up based on the status
;  1064          1144  2 !
;  1065          1145  2 SELECTONE .status OF
;  1066          1146  2 SET
;  1067          1147  2     ! Status is normal
;  1068          1148  2     !
;  1069          1149  3     [ss$_normal] :      BEGIN
;  1070          1150  3
;  1071          1151  3                         ! Set the bits and pieces to reflect that the tape has moved
;  1072          1152  3                         !
;  1073          1153  3                         dosv [dos11$v_beg_of_tape] = false;
;  1074          1154  3                         dosv [dos11$v_end_of_tape] = true;
;  1075          1155  3                         dosv [dos11$v_tape_mark]   = false;
;  1076          1156  2                         END;
;  1077          1157  2
;  1078          1158  2     ! If this is the end, oops
;  1079          1159  2     !
;  1080          1160  3     [ss$_endoftape] :   BEGIN
;  1081          1161  3                         LOCAL
;  1082          1162  3                             stat2;
;  1083          1163  3
;  1084          1164  3                         ! Signal that we have had a problem
;  1085          1165  3                         !
;  1086        P 1166  3                         $exch_signal (exch$_dos11_ioerror, 2,
;  1087          1167  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  1088          1168  3
;  1089          1169  3                         ! Set the bits and pieces to reflect that the tape has moved
;  1090          1170  3                         !
;  1091          1171  3                         dosv [dos11$v_beg_of_tape] = false;
;  1092          1172  3                         dosv [dos11$v_end_of_tape] = true;
;  1093          1173  3                         dosv [dos11$v_tape_mark]   = false;
;  1094          1174  3
;  1095          1175  3                         ! Return the ss$_endoftape status
;  1096          1176  3                         !
;  1097          1177  2                         END;
```

```
;  1098          1178  2
;  1099          1179  2      ! Any thing else and we better crap out
;  1100          1180  2      !
;  1101          1181  3      [OTHERWISE] :       BEGIN
;  1102          1182  3                          dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
;  1103        P 1183  3                          $exch_signal_return (exch$_dos11_ioerror, 2,
;  1104          1184  3                                               .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
;  1105          1185  2                          END;
;  1106          1186  2 TES;
;  1107          1187  2
;  1108          1188  2 RETURN .status;
;  1109          1189  1 END;


                                 07FC 00000          .ENTRY  EXCH$IO_DOS11_WRITE, Save R2,R3,R4,R5,R6,-      ; 1068
                                                              R7,R8,R9,R10
                          5A 00000000G   EF  9E 00002        MOVAB   EXCH$UTIL_BLOCK_CHECK, R10                  ;
                          59 00000000G   00  9E 00009        MOVAB   LIB$SIGNAL, R9                              ;
                          58 00000000G   8F  D0 00010        MOVL    #EXCH$_DOS11_IOERROR, R8                    ;
                          57 00000000G   00  9E 00017        MOVAB   LIB$STOP, R7                                ;
                          56 00000000G   8F  D0 0001E        MOVL    #EXCH$_BADLOGIC, R6                         ;
                          53       04    AC  D0 00025        MOVL    VOLB, R3                                    ; 1112
                          52 041B00F3    8F  D0 00029        MOVL    #68878579, R2                               ; 1116
                          51     0239    8F  3C 00030        MOVZWL  #569, R1                                    ;
                          50             53  D0 00035        MOVL    R3, R0                                      ;
                                         6A  16 00038        JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                          54       54    A3  D0 0003A        MOVL    84(R3), R4                                  ; 1117
                          52 003600FD    8F  D0 0003E        MOVL    #3539197, R2                                ;
                          51     0254    8F  3C 00045        MOVZWL  #596, R1                                    ;
                          50             54  D0 0004A        MOVL    R4, R0                                      ;
                                         6A  16 0004D        JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                          55       0C    A4  9E 0004F        MOVAB   12(R4), R5                                  ; 1118
                          0C             65  E8 00053        BLBS    (R5), 1$                                    ;
                          7E     0110    8F  3C 00056        MOVZWL  #272, -(SP)                                 ;
                                         01  DD 0005B        PUSHL   #1                                          ;
                                         56  DD 0005D        PUSHL   R6                                          ;
                                         67  03  FB 0005F        CALLS   #3, LIB$STOP                            ;
                    0C                   65  02  E1 00062 1$:    BBC     #2, (R5), 2$                            ; 1119
                                   7E    011A  8F  3C 00066        MOVZWL  #282, -(SP)                             ;
                                         01  DD 0006B        PUSHL   #1                                          ;
                                         56  DD 0006D        PUSHL   R6                                          ;
                                         67  03  FB 0006F        CALLS   #3, LIB$STOP                            ;
                    0C                   65  03  E0 00072 2$:    BBS     #3, (R5), 3$                            ; 1120
                                   7E    0120  8F  3C 00076        MOVZWL  #288, -(SP)                             ;
                                         01  DD 0007B        PUSHL   #1                                          ;
                                         56  DD 0007D        PUSHL   R6                                          ;
                                         67  03  FB 0007F        CALLS   #3, LIB$STOP                            ;
                                       0C    AC  D5 00082 3$:    TSTL    LEN                                     ; 1125
                                       0C    12  00085           BNEQ    4$                                      ;
                                   7E    0122  8F  3C 00087        MOVZWL  #290, -(SP)                             ;
                                         01  DD 0008C        PUSHL   #1                                          ;
                                         56  DD 0008E        PUSHL   R6                                          ;
                                         67  03  FB 00090        CALLS   #3, LIB$STOP                            ;
                                   50    0C  AC  D0 00093 4$:    MOVL    LEN, R0                                 ; 1126
```

```
                       0E          50  D1 00097        CMPL    R0, #14                    :
                                   03  1E 0009A        BGEQU   5$                         :
                       50          0E  D0 0009C        MOVL    #14, R0                    :
                       52          50  D0 0009F 5$:    MOVL    R0, IO_LEN                 :
                       02          52  E9 000A2        BLBC    IO_LEN, 6$                 : 1127
                                   52  D6 000A5        INCL    IO_LEN                     :
             00000200  8F          52  D1 000A7 6$:    CMPL    IO_LEN, #512               : 1128
                                   0C  1B 000AE        BLEQU   7$                         :
                       7E    0123  8F  3C 000B0        MOVZWL  #291, -(SP)                :
                                   01  DD 000B5        PUSHL   #1                         :
                                   56  DD 000B7        PUSHL   R6                         :
                       67          03  FB 000B9        CALLS   #3, LIB$STOP               :
                       50          20  D0 000BC 7$:    MOVL    #32, FUNC                  : 1132
        05       48    A3          01  E1 000BF        BBC     #1, 72(R3), 8$             : 1133
                       50    4000  8F  A8 000C4        BISW2   #16384, FUNC               :
                                   7E  7C 000C9 8$:    CLRQ    -(SP)                      : 1139
                                   7E  7C 000CB        CLRQ    -(SP)                      :
                                   52  DD 000CD        PUSHL   IO_LEN                     :
                             08    AC  DD 000CF        PUSHL   BUFFER                     :
                                   7E  7C 000D2        CLRQ    -(SP)                      :
                             1E    A4  9F 000D4        PUSHAB  30(R4)                     :
                                   50  DD 000D7        PUSHL   FUNC                       :
                       7E    4A    A3  3C 000D9        MOVZWL  74(R3), -(SP)              :
                                   7E  D4 000DD        CLRL    -(SP)                      :
             00000000G 00          0C  FB 000DF        CALLS   #12, SYS$QIOW              :
                       52          50  D0 000E6        MOVL    R0, STATUS                 :
                       04          52  E9 000E9        BLBC    STATUS, 9$                 :
                       52    1E    A4  3C 000EC        MOVZWL  30(R4), STATUS             : 1141
                       01          52  D1 000F0 9$:    CMPL    STATUS, #1                 : 1149
                                   18  13 000F3        BEQL    10$                        :
             00000878  8F          52  D1 000F5        CMPL    STATUS, #2168              : 1160
                                   1A  12 000FC        BNEQ    11$                        :
                                   52  DD 000FE        PUSHL   STATUS                     : 1167
                             69    A3  9F 00100        PUSHAB  105(R3)                    :
                             65    A3  DD 00103        PUSHL   101(R3)                    :
                                   02  DD 00106        PUSHL   #2                         :
                                   58  DD 00108        PUSHL   R8                         :
                       69          05  FB 0010A        CALLS   #5, LIB$SIGNAL             :
                       65          02  8A 0010D 10$:   BICB2   #2, (R5)                   : 1171
                       65          08  88 00110        BISB2   #8, (R5)                   : 1172
                       65          04  8A 00113        BICB2   #4, (R5)                   : 1173
                                   19  11 00116        BRB     12$                        : 1145
                       65          01  8A 00118 11$:   BICB2   #1, (R5)                   : 1182
                       54          58  D0 0011B        MOVL    R8, TEMP                   : 1184
                                   52  DD 0011         PUSHL   STATUS                     :
                             69    A3  9F 00120        PUSHAB  105(R3)                    :
                             65    A3  DD 00123        PUSHL   101(R3)                    :
                                   02  DD 00126        PUSHL   #2                         :
                                   54  DD 00128        PUSHL   TEMP                       :
                       69          05  FB 0012A        CALLS   #5, LIB$SIGNAL             :
                       50          54  D0 0012D        MOVL    TEMP, R0                   :
                                       04 00130        RET                                :
                       50          52  D0 00131 12$:   MOVL    STATUS, R0                 : 1188
                                       04 00134        RET                                : 1189

; Routine Size:  309 bytes,    Routine Base:  EXCH$IO_CODE + 075D
```

```
;1111   1190  1 GLOBAL ROUTINE exch$io_dos11_write_label (volb : $ref_bblock, label_buf) =      %SBTTL 'exch$io_dos11_write_
;1112   1191  2 BEGIN
;1113   1192  2 !++
;1114   1193  2 !
;1115   1194  2 ! FUNCTIONAL DESCRIPTION:
;1116   1195  2 !
;1117   1196  2 !       Write a label on a dos-11 sequential device
;1118   1197  2 !
;1119   1198  2 ! INPUTS:
;1120   1199  2 !
;1121   1200  2 !       volb    - pointer to a volb which contains the active record stream
;1122   1201  2 !       label_buf  - pointer to the label to be written
;1123   1202  2 !
;1124   1203  2 ! IMPLICIT INPUTS:
;1125   1204  2 !
;1126   1205  2 !       dos11 context block hanging off the volb
;1127   1206  2 !
;1128   1207  2 ! OUTPUTS:
;1129   1208  2 !
;1130   1209  2 !       none
;1131   1210  2 !
;1132   1211  2 ! IMPLICIT OUTPUTS:
;1133   1212  2 !
;1134   1213  2 !       dos11 context block hanging off the volb
;1135   1214  2 !
;1136   1215  2 ! ROUTINE VALUE:
;1137   1216  2 !
;1138   1217  2 !       success if able to write, false if error status
;1139   1218  2 !
;1140   1219  2 ! SIDE EFFECTS:
;1141   1220  2 !
;1142   1221  2 !       none
;1143   1222  2 !--
;1144   1223  2
;1145   1224  2 $dbgtrc_prefix ('io_dos11_write_label> ');
;1146   1225  2
;1147   1226  2 LOCAL
;1148   1227  2     func,
;1149   1228  2     status
;1150   1229  2     ;
;1151   1230  2
;1152   1231  2 BIND
;1153   1232  2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
;1154   1233  2     ;
;1155   1234  2
;1156   1235  2 $debug_print_lit ('entry');
;1157   1236  2 $block_check (2, .volb, volb, 469);
;1158   1237  2 $block_check (2, .dosv, dos11, 564);
;1159   1238  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 284);        ! We should know where we are
;1160   1239  2 $logic_check (2, (.dosv [dos11$v_tape_mark]), 285);            ! And that should be at a tape mark
```

```
;                                                N 12
; 1162          1240  2 ! Issue the io operation to write a logical block in the forward direction
; 1163          1241  2 !
; 1164        L 1242  2 $logic_check (3, (dos11$s_label_buf EQL 14), 286);
; %PRINT:       assumption 286 verified during compilation
; 1165          1243  2 func = io$_writelblk;
; 1166          1244  2 IF .volb [volb$v_read_check] THEN func = .func OR io$m_datacheck;
; 1167        P 1245  3 IF (status = $qiow (efn=0,
; 1168        P 1246  3                     chan=.volb [volb$w_channel],
; 1169        P 1247  3                     func=.func,
; 1170        P 1248  3                     iosb=dosv [dos11$q_iosb],
; 1171        P 1249  3                     p1=.label_buf,
; 1172          1250  3                     p2=dos11$s_label_buf))
; 1173          1251  2 THEN
; 1174          1252  2     status = .dosv [dos11$w_iosb_status];
; 1175          1253  2
; 1176          1254  2 ! Finish up based on the status
; 1177          1255  2 !
; 1178          1256  2 SELECTONE .status OF
; 1179          1257  2 SET
; 1180          1258  2     ! Status is normal
; 1181          1259  2     !
; 1182          1260  3     [ss$_normal] :      BEGIN
; 1183          1261  3
; 1184          1262  3                         ! Set the bits and pieces to reflect that the tape has moved
; 1185          1263  3                         !
; 1186          1264  3                         dosv [dos11$v_beg_of_tape] = false;
; 1187          1265  3                         dosv [dos11$v_end_of_tape] = true;
; 1188          1266  3                         dosv [dos11$v_tape_mark]   = false;
; 1189          1267  2                         END;
; 1190          1268  2
; 1191          1269  2     ! If this is the end, oops
; 1192          1270  2     !
; 1193          1271  3     [ss$_endoftape] :   BEGIN
; 1194          1272  3                         LOCAL
; 1195          1273  3                             stat2;
; 1196          1274  3
; 1197          1275  3                         ! Signal that we have had a problem
; 1198          1276  3                         !
; 1199        P 1277  3                         $exch_signal (exch$_dos11_ioerror, 2,
; 1200          1278  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
; 1201          1279  3
; 1202          1280  3                         ! Set the bits and pieces to reflect that the tape has moved
; 1203          1281  3                         !
; 1204          1282  3                         dosv [dos11$v_beg_of_tape] = false;
; 1205          1283  3                         dosv [dos11$v_end_of_tape] = true;
; 1206          1284  3                         dosv [dos11$v_tape_mark]   = false;
; 1207          1285  3
; 1208          1286  3                         ! Return the ss$_endoftape status
; 1209          1287  3                         !
; 1210          1288  2                         END;
; 1211          1289  2
; 1212          1290  2     ! Any thing else and we better crap out
; 1213          1291  2     !
; 1214          1292  3     [OTHERWISE] :       BEGIN
; 1215          1293  3                         dosv [dos11$v_position_valid] = false;  ! After an error, we don't know where we are
; 1216        P 1294  3                         $exch_signal_return (exch$_dos11_ioerror, 2,
; 1217          1295  3                                       .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], .status);
```

```
;  1218          1296  2                          END;
;  1219          1297  2 TES;
;  1220          1298  2
;  1221          1299  2 RETURN .status;
;  1222          1300  1 END;


                                07FC 00000      .ENTRY  EXCH$IO_DOS11_WRITE_LABEL, Save R2,R3,R4,-      ; 1190
                                                        R5,R6,R7,R8,R9,R10
                5A 00000000G    EF   9E 00002   MOVAB   EXCH$UTIL_BLOCK_CHECK, R10
                59 00000000G    00   9E 00009   MOVAB   LIB$SIGNAL, R9
                58 00000000G    8F   D0 00010   MOVL    #EXCH$_DOS11_ICERROR, R8
                57 00000000G    00   9E 00017   MOVAB   LIB$STOP, R7
                56 00000000G    8F   D0 0001E   MOVL    #EXCH$_BADLOGIC, R6
                53          04  AC   D0 00025   MOVL    VOLB, R3                                        ; 1232
                52 041B00F3     8F   D0 00029   MOVL    #68878579, R2                                   ; 1236
                51       01D5   8F   3C 00030   MOVZWL  #469, R1
                50              53   D0 00035   MOVL    R3, R0
                                6A   16 00038   JSB     EXCH$UTIL_BLOCK_CHECK
                54          54  A3   D0 0003A   MOVL    84(R3), R4                                      ; 1237
                52 003600FD     8F   D0 0003E   MOVL    #3539197, R2
                51       0234   8F   3C 00045   MOVZWL  #564, R1
                50              54   D0 0004A   MOVL    R4, R0
                                6A   16 0004D   JSB     EXCH$UTIL_BLOCK_CHECK
                55          0C  A4   9E 0004F   MOVAB   12(R4), R5                                       ; 1238
                0C              65   E8 00053   BLBS    (R5), 1$
                7E       011C   8F   3C 00056   MOVZWL  #284, -(SP)
                                01   DD 0005B   PUSHL   #1
                                56   DD 0005D   PUSHL   R6
                            67                  CALLS   #3, LIB$STOP
                0C              65              BBS     #2, (R5), 2$                                    ; 1239
                            7E      011D        MOVZWL  #285, -(SP)
                                                PUSHL   #1
                                                PUSHL   R6
                            67                  CALLS   #3, LIB$STOP
                            50                  MOVL    #32, FUNC                                       ; 1243
                05      48  A3                  BBC     #1, 72(R3), 3$                                  ; 1244
                            50      4000        BISW2   #16384, FUNC
                            7E              3$: CLRQ    -(SP)                                           ; 1250
                            7E                  CLRQ    -(SP)
                            0E                  PUSHL   #14
                                08              PUSHL   LABEL_BUF
                            7E                  CLRQ    -(SP)
                                1E              PUSHAB  30(R4)
                                                PUSHL   FUNC
                            7E      4A          MOVZWL  74(R3), -(SP)
                            7E                  CLRL    -(SP)
                00000000G   00                  CALLS   #12, SYS$QIOW
                            52                  MOVL    R0, STATUS
                            04                  BLBC    STATUS, 4$
                            52      1E          MOVZWL  30(R4), STATUS                                  ; 1252
                            01                  CMPL    STATUS, #1                                      ; 1260
                                                BEQL    5$
                00000878    8F                  CMPL    STATUS, #2168                                   ; 1271
```

```
                                 1A  12 000B2          BNEQ    6$                                                      ;
                                 52  DD 000B4          PUSHL   STATUS                                                  ; 1278
                              69 A3  9F 000B6          PUSHAB  105(R3)                                                 ;
                              65 A3  DD 000B9          PUSHL   101(R3)                                                 ;
                                 02  DD 000BC          PUSHL   #2                                                      ;
                                 58  DD 000BE          PUSHL   R8                                                      ;
                           69    05  FB 000C0          CALLS   #5, LIB$SIGNAL                                          ;
                           65    02  8A 000C3 5$:      BICB2   #2, (R5)                                                ; 1282
                           65    08  88 000C6          BISB2   #8, (R5)                                                ; 1283
                           65    04  8A 000C9          BICB2   #4, (R5)                                                ; 1284
                                 19  11 000CC          BRB     7$                                                      ; 1256
                           65    01  8A 000CE 6$:      BICB2   #1, (R5)                                                ; 1293
                           54    58  D0 000D1          MOVL    R8, TEMP                                                ; 1295
                                 52  DD 000D4          PUSHL   STATUS                                                  ;
                              69 A3  9F 000D6          PUSHAB  105(R3)                                                 ;
                              65 A3  DD 000D9          PUSHL   101(R3)                                                 ;
                                 02  DD 000DC          PUSHL   #2                                                      ;
                                 54  DD 000DE          PUSHL   TEMP                                                    ;
                           69    05  FB 000E0          CALLS   #5, LIB$SIGNAL                                          ;
                           50    54  D0 000E3          MOVL    TEMP, R0                                                ;
                                 04  000E6          RET                                                             ;
                           50    52  D0 000E7 7$:      MOVL    STATUS, R0                                              ; 1299
                                 04  000EA          RET                                                             ; 1300

; Routine Size:  235 bytes,    Routine Base:  EXCH$IO_CODE + 0892
```

```
1 GLOBAL ROUTINE exch$io_dos11_write_tape_mark (volb : $ref_bblock) =     %SBTTL 'exch$io_dos11_write_tape_mar
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Write a single tape mark on a dos-11 sequential device
2 !
2 ! INPUTS:
2 !
2 !       volb   - pointer to a volb which contains the active record stream
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       dos11 context block hanging off the volb
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       dos11 context block hanging off the volb
2 !
2 ! ROUTINE VALUE:
2 !
2 !       success or error status
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('io_dos11_write_tape_mark> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock    ! Format specific block, contains iosb
2     ;
```

```
;  1267          1343  2 $debug_print_lit ('entry');
;  1268          1344  2 $block_check (2, .volb, volb, 606);
;  1269          1345  2 $block_check (2, .dosv, dos11, 607);
;  1270          1346  2 $logic_check (2, (.dosv [dos11$v_position_valid]), 271);        ! We should know where we are
;  1271          1347  2
;  1272          1348  2 ! Issue the io operation to write the tape mark
;  1273          1349  2 !
;  1274        P 1350  3 IF (status = $qiow (efn=0,
;  1275        P 1351  3                     chan=.volb [volb$w_channel],
;  1276        P 1352  3                     func=io$_writeof,
;  1277          1353  3                     iosb=dosv [dos11$q_iosb]))
;  1278          1354  2 THEN
;  1279          1355  2     status = .dosv [dos11$w_iosb_status];
;  1280          1356  2
;  1281          1357  2 ! If either the qio or the io operation failed, scream and shout
;  1282          1358  2 !
;  1283          1359  2 IF NOT .status
;  1284          1360  2   AND
;  1285          1361  2     .status NEQ ss$_endoftape                    ! End of tape can be ignored
;  1286          1362  2 THEN
;  1287          1363  3     BEGIN
;  1288          1364  3     dosv [dos11$v_position_valid] = false;       ! After an error, we don't know where we are
;  1289        P 1365  3     $exch_signal_return (exch$_dos11_ioerror, 2,
;  1290          1366  3                                .volb [volb$t_vol_ident_len], volb [volb$t_vol_ident], .status);
;  1291          1367  2     END;
;  1292          1368  2
;  1293          1369  2 ! Set the bits and pieces to reflect what we did to the tape
;  1294          1370  2 !
;  1295          1371  2 dosv [dos11$v_end_of_tape] = true;
;  1296          1372  2 dosv [dos11$v_tape_mark]   = true;
;  1297          1373  2
;  1298          1374  2 RETURN .status;
;  1299          1375  1 END;


                                    007C 00000          .ENTRY  EXCH$IO_DOS11_WRITE_TAPE_MARK, Save R2,R3,- ; 1301
                                                                R4,R5,R6
                          56 00000000G  EF   9E 00002   MOVAB   EXCH$UTIL_BLOCK_CHECK, R6
                          54        04  AC   D0 00009   MOVL    VOLB, R4                                    ; 1341
                          52 041B00F3   8F   D0 0000D   MOVL    #68878579, R2                               ; 1344
                          51      025E  8F   3C 00014   MOVZWL  #606, R1
                          50            54   D0 00019   MOVL    R4, R0
                                        66   16 0001C   JSB     EXCH$UTIL_BLOCK_CHECK
                          53        54  A4   D0 0001E   MOVL    84(R4), R3                                  ; 1345
                          52 003600FD   8F   D0 00022   MOVL    #3539197, R2
                          51      025F  8F   3C 00029   MOVZWL  #607, R1
                          50            53   D0 0002E   MOVL    R3, R0
                                        66   16 00031   JSB     EXCH$UTIL_BLOCK_CHECK
                          14        0C  A3   E8 00033   BLBS    12(R3), 1$                                  ; 1346
                          7E      010F  8F   3C 00037   MOVZWL  #271, -(SP)
                                        01   DD 0003C   PUSHL   #1
                              00000000G 8F   DD 0003E   PUSHL   #EXCH$_BADLOGIC
                00000000G     00        03   FB 00044   CALLS   #3, LIB$STOP
                                        7E   7C 0004B 1$: CLRQ    -(SP)                                       ; 1353
```

```
                                                7E  7C 0004D          CLRQ    -(SP)                                                  ;
                                                7E  7C 0004F          CLRQ    -(SP)                                                  ;
                                                7E  7C 00051          CLRQ    -(SP)                                                  ;
                                          1E    A3  9F 00053          PUSHAB  30(R3)                                                 ;
                                                28  DD 00056          PUSHL   #40                                                    ;
                                    7E    4A    A4  3C 00058          MOVZWL  74(R4), -(SP)                                          ;
                                                7E  D4 0005C          CLRL    -(SP)                                                  ;
                          00000000G   00        0C  FB 0005E          CALLS   #12, SYS$QIOW                                          ;
                                      55        50  D0 00065          MOVL    R0, STATUS                                             ;
                                      07        55  E9 00068          BLBC    STATUS, 2$                                             ;
                                      55    1E  A3  3C 0006B          MOVZWL  30(R3), STATUS                                         ; 1355
                                      2B        55  E8 0006F          BLBS    STATUS, 3$                                             ; 1359
                          00000878    8F        55  D1 00072 2$:      CMPL    STATUS, #2168                                          ; 1361
                                                22  13 00079          BEQL    3$                                                     ;
                                0C    A3        01  8A 0007B          BICB2   #1, 12(R3)                                             ; 1364
                                      52  00000000G 8F  D0 0007F      MOVL    #EXCH$_DOS11_IOERROR, TEMP                             ; 1366
                                                55  DD 00086          PUSHL   STATUS                                                 ;
                                          69    A4  9F 00088          PUSHAB  105(R4)                                                ;
                                          65    A4  DD 0008B          PUSHL   101(R4)                                                ;
                                                02  DD 0008E          PUSHL   #2                                                     ;
                                                52  DD 00090          PUSHL   TEMP                                                   ;
                          00000000G   00        05  FB 00092          CALLS   #5, LIB$SIGNAL                                         ;
                                      50        52  D0 00099          MOVL    TEMP, R0                                               ;
                                                04  00009C           RET                                                            ;
                                0C    A3        0C  88 0009D 3$:      BISB2   #12, 12(R3)                                            ; 1372
                                      50        55  D0 000A1          MOVL    STATUS, R0                                             ; 1374
                                                04  000A4           RET                                                            ; 1375

; Routine Size:  165 bytes,    Routine Base:  EXCH$IO_CODE + 097D
```

```
;1301     1376  1 GLOBAL ROUTINE exch$io_rt11_read (volb : $ref_bblock,   %SBTTL 'exch$io_rt11_read (volb, pbn, cnt, adr)'
;1302     1377  1                                   blkpbn, blkcnt, bufadr : $ref_bvector) =
;1303     1378  2 BEGIN
;1304     1379  2 !++
;1305     1380  2 !
;1306     1381  2 ! FUNCTIONAL DESCRIPTION:
;1307     1382  2 !
;1308     1383  2 !       Read a range of "physical" blocks from the device.  Block
;1309     1384  2 !       numbers are in the range 0 to MaxBlocks-1.
;1310     1385  2 !
;1311     1386  2 ! INPUTS:
;1312     1387  2 !
;1313     1388  2 !       volb   - pointer to a volb which contains the active record stream
;1314     1389  2 !       blkpbn - starting physical block number on device
;1315     1390  2 !       blkcnt - number of blocks to read
;1316     1391  2 !
;1317     1392  2 ! IMPLICIT INPUTS:
;1318     1393  2 !
;1319     1394  2 !       none
;1320     1395  2 !
;1321     1396  2 ! OUTPUTS:
;1322     1397  2 !
;1323     1398  2 !       bufadr - pointer to buf'er to receive the data
;1324     1399  2 !
;1325     1400  2 ! IMPLICIT OUTPUTS:
;1326     1401  2 !
;1327     1402  2 !       none
;1328     1403  2 !
;1329     1404  2 ! ROUTINE VALUE:
;1330     1405  2 !
;1331     1406  2 !       success or error status
;1332     1407  2 !
;1333     1408  2 ! SIDE EFFECTS:
;1334     1409  2 !
;1335     1410  2 !       none
;1336     1411  2 !--
;1337     1412  2
;1338     1413  2 $dbgtrc_prefix ('io_rt11_read> ');
;1339     1414  2
;1340     1415  2 LOCAL
;1341     1416  2     check_buf : $bvector [ctx$k_buffer_length],
;1342     1417  2     bytecnt,
;1343     1418  2     transferred,
;1344     1419  2     lbn,
;1345     1420  2     status
;1346     1421  2     ;
;1347     1422  2
;1348     1423  2 BIND
;1349     1424  2     fab = .volb [volb$a_fab] : $bblock, ! New name for File Access Block
;1350     1425  2     rab = .volb [volb$a_rab] : $bblock, ! New name for Record Access Block
;1351     1426  2     nam = .volb [volb$a_nam] : $bblock  ! New name for NAM block
;1352     1427  2     ;
;1353     1428  2
;1354     1429  2 $block_check (2, .volb, volb, 527);
;1355     1430  2 $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
```

```
; 1357     1431  2 ! Adjust the externally useful block count to the RMS required byte count
; 1358     1432  2 !
; 1359     1433  2 $logic_check (2, ((.blkcnt GTRU 0) AND (.blkcnt LEQU 127)), 140);
; 1360     1434  2 bytecnt = .blkcnt * 512;
; 1361     1435  2
; 1362     1436  2 ! Return an end of file error if the request is past the end of the device
; 1363     1437  2 !
; 1364     1438  2 IF .blkpbn GEQU .volb [volb$l_volmaxblock]
; 1365     1439  2 THEN
; 1366     1440  2     RETURN exch$util_file_error (exch$_readerr, rms$_eof, fab, 0);
; 1367     1441  2
; 1368     1442  2 ! Adjust the externally useful "physical" block number to the RMS required
; 1369     1443  2 ! logical block number
; 1370     1444  2 !
; 1371     1445  2 lbn = .blkpbn;                                  ! Assume volume relative lbn.
; 1372     1446  2 IF .volb [volb$v_virtual]
; 1373     1447  2 THEN
; 1374     1448  2     lbn = .lbn + 1;                             ! File-relative virtual block numbers
; 1375     1449  2
; 1376     1450  2 ! See if the request is for block 0
; 1377     1451  2 !
; 1378     1452  2 IF .lbn EQL 0
; 1379     1453  2 THEN
; 1380     1454  3     BEGIN
; 1381     1455  3
; 1382     1456  3     ! Since BKT = 0 means sequential read, we must trick it with the
; 1383     1457  3     ! documented technique of REWIND, then set BKT=0.
; 1384     1458  3     !
; 1385     1459  4     IF NOT (status = $rewind (rab = rab))
; 1386     1460  3     THEN
; 1387     1461  4         BEGIN
; 1388     1462  4         $debug_print_lit ('rewind error');
; 1389     1463  4         RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
; 1390     1464  3         END;
; 1391     1465  3
; 1392     1466  2     END;
; 1393     1467  2
; 1394     1468  2 ! Set the RMS record stream buffer and block parameters
; 1395     1469  2 !
; 1396     1470  2 rab [rab$l_ubf] = .bufadr;                      ! User buffer address
; 1397     1471  2 rab [rab$w_usz] = .bytecnt;                     ! User buffer size
; 1398     1472  2 rab [rab$l_bkt] = .lbn;
; 1399     1473  2
; 1400     1474  2 ! Read the chunk
; 1401     1475  2 !
; 1402     1476  3 IF NOT (status = $read (rab = rab))
; 1403     1477  2 THEN
; 1404     1478  3     BEGIN
; 1405     1479  3     $debug_print_fao ('error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz] / 512);
; 1406     1480  3     $check_call (4, exch$util_file_error, exch$_readerr, .status, fab, .rab [rab$l_stv]);
; 1407     1481  3     IF .status EQL rms$_rer
; 1408     1482  3     THEN
; 1409     1483  4         BEGIN
; 1410     1484  4         LOCAL
; 1411     1485  4             stv,
; 1412     1486  4             stat_1;
; 1413     1487  4         stv = .rab [rab$l_stv];
```

```
;  1414     1488  4          stat_1 = exch$io_rt11_read_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;  1415     1489  4          IF .stat_1
;  1416     1490  4          THEN
;  1417   P 1491  4              $exch_signal (exch$_readerrrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
;  1418     1492  4                                                 .status, .stv);
;  1419     1493  4          RETURN .stat_1;
;  1420     1494  4          END
;  1421     1495  3      ELSE
;  1422     1496  3          RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;  1423     1497  2      END;
;  1424     1498  2
;  1425     1499  2  transferred = .rab [rab$w_rsz];                ! Save actual byte count
;  1426     1500  2  IF .transferred NEQ .bytecnt            !??shouldn't happen, but has been
;  1427     1501  2  THEN
;  1428   P 1502  2      $print_fao ('transfer count error, requested !UL bytes, received !UL bytes, status !XL',
;  1429     1503  2                  bytecnt, .transferred, .status);
;  1430     1504  2
;  1431     1505  2  ! If read checking, reread the chunk and compare
;  1432     1506  2  !
;  1433     1507  2  IF .volb [volb$v_read_check]
;  1434     1508  2  THEN
;  1435     1509  3      BEGIN
;  1436     1510  3      LOCAL
;  1437     1511  3          len;
;  1438     1512  3
;  1439     1513  3      ! See if the request is for block 0, if so rewind
;  1440     1514  3      !
;  1441     1515  3      IF .lbn EQL 0
;  1442     1516  3      THEN
;  1443     1517  4          BEGIN
;  1444     1518  5          IF NOT (status = $rewind (rab = rab))
;  1445     1519  4          THEN
;  1446     1520  5              BEGIN
;  1447     1521  5              $debug_print_lit ('rewind error');
;  1448     1522  5              RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;  1449     1523  4              END;
;  1450     1524  3          END;
;  1451     1525  3
;  1452     1526  3      len = .bytecnt;
;  1453     1527  3
;  1454     1528  3      rab [rab$l_ubf] = check_buf;                ! User buffer address
;  1455     1529  3      rab [rab$w_usz] = .len;                     ! User buffer size
;  1456     1530  3      rab [rab$l_bkt] = .lbn;
;  1457     1531  3
;  1458     1532  3      ! Read the chunk again
;  1459     1533  3      !
;  1460     1534  4      IF NOT (status = $read (rab = rab))
;  1461     1535  3      THEN
;  1462     1536  4          BEGIN
;  1463     1537  4          $trace_print_fao ('reread error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz] / 51
;  1464     1538  4          $check_call (4, exch$util_file_error, exch$_readcheck, .status, fab, .rab [rab$l_stv]);
;  1465     1539  4          IF .status EQL rms$_rer
;  1466     1540  4          THEN
;  1467     1541  5              BEGIN
;  1468     1542  5              LOCAL
;  1469     1543  5                  stv,
;  1470     1544  5                  stat_1;
```

```
;  1471          1545  5              stv = .rab [rab$l_stv];
;  1472          1546  5              stat_1 = exch$io_rt11_read_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;  1473          1547  5              IF .stat_1
;  1474          1548  5              THEN
;  1475        P 1549  5                  $exch_signal (exch$_readcheckrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
;  1476          1550  5                                                  .status, .stv);
;  1477          1551  5              RETURN .stat_1;
;  1478          1552  5              END
;  1479          1553  4          ELSE
;  1480          1554  4              RETURN exch$util_file_error (exch$_readcheck, .status, fab, .rab [rab$l_stv]);
;  1481          1555  3          END;
;  1482          1556  3
;  1483          1557  3      IF CH$NEQ (.len, check_buf, .len, .bufadr, 0)
;  1484          1558  3      THEN
;  1485          1559  4          BEGIN
;  1486        L 1560  4          %IF switch_trace
;  1487        U 1561  4          %THEN
;  1488        U 1562  4              LITERAL
;  1489        U 1563  4                  chunk = 100;
;  1490        U 1564  4              LOCAL
;  1491        U 1565  4                  c,              ! count of loops
;  1492        U 1566  4                  l,              ! length of string remaining
;  1493        U 1567  4                  a1,             ! address of remainder of string 1
;  1494        U 1568  4                  a2;             ! address of remainder of string 2
;  1495        U 1569  4              c = 0;
;  1496        U 1570  4              l = .len;
;  1497        U 1571  4              a1 = .bufadr;
;  1498        U 1572  4              a2 = check_buf;
;  1499        U 1573  4              $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
;  1500        U 1574  4              WHILE .l GTR 0
;  1501        U 1575  4              DO
;  1502        U 1576  4                  BEGIN
;  1503        U 1577  4                  LOCAL
;  1504        U 1578  4                      ll;         ! length for this loop
;  1505        U 1579  4                  c = .c + 1;
;  1506        U 1580  4                  ll = MINU (.l, chunk);
;  1507        U 1581  4                  IF CH$NEQ (.ll, .a1, .ll, .a2, 0)
;  1508        U 1582  4                  THEN
;  1509        U 1583  4                      BEGIN
;  1510        U 1584  4                      $trace_print_fao ('!4UL  "!AF"', .c, .ll, .a1);
;  1511        U 1585  4                      $trace_print_fao ('      "!AF"', .ll, .a2);
;  1512        U 1586  4                      END;
;  1513        U 1587  4                  l = .l - chunk;
;  1514        U 1588  4                  a1 = .a1 + chunk;
;  1515        U 1589  4                  a2 = .a2 + chunk;
;  1516        U 1590  4                  END;
;  1517          1591  4          %FI
;  1518          1592  5          BEGIN
;  1519          1593  5          LOCAL
;  1520          1594  5              stat_1;
;  1521          1595  5          stat_1 = exch$io_rt11_read_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;  1522          1596  5          IF .stat_1
;  1523          1597  5          THEN
;  1524          1598  5              $exch_signal (exch$_readcheckrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  1525          1599  5          RETURN .stat_1;
;  1526          1600  4          END;
;  1527          1601  3          END;
```

```
;  1528          1602  2     END;
;  1529          1603  2
;  1530          1604  2 ! Check to make sure that the correct number of bytes was found
;  1531          1605  2 !
;  1532          1606  2 IF .transferred NEQ .bytecnt
;  1533          1607  2 THEN
;  1534          1608  2     $logic_check (0, (false), 315);     ! This error should not occur
;  1535          1609  2
;  1536          1610  2 RETURN .status;
;  1537          1611  1 END;


                                                                .PSECT  EXCH$IO_PLIT,NOWRT,2

20 74 6E 75 6F 63 20 72 65 66 73 6E 61 72 74 0003C P.AAH:   .ASCII  \transfer count error, requested !UL byte\  ;
65 74 73 65 75 71 65 72 20 2C 72 6F 72 72 65 0004B                                                            ;
                  65 74 79 62 20 4C 55 21 20 64 0005A                                                         ;
4C 55 21 20 64 65 76 69 65 63 65 72 20 2C 73 00064          .ASCII  \s, received !UL bytes, status !XL\<0><0>  ;
20 73 75 74 61 74 73 20 2C 73 65 74 79 62 20 00073                                                            ;
                              00 00 4C 58 21 00082                                                            ;
                                          00  00087          .ASCII  <0>                                       ;
                                    010E0049  00088 P.AAG:   .LONG   17694793                                  ;
                                    00000000' 0008C          .ADDRESS P.AAH                                    ;

                                                                .EXTRN  SYS$REWIND, SYS$READ
                                                                .EXTRN  EXCH$_READERRREC
                                                                .EXTRN  EXCH$UTIL_FAO_BUFFER
                                                                .EXTRN  LIB$PUT_OUTPUT, EXCH$_READCHECK
                                                                .EXTRN  EXCH$_READCHECKREC

                                                                .PSECT  EXCH$IO_CODE,NOWRT,2

                                        0FFC 00000              .ENTRY  EXCH$IO_RT11_READ, Save R2,R3,R4,R5,R6,R7,- ; 1376
                                                                        R8,R9,R10,R11
                    5E      E7FC    CE  9E 00002                MOVAB   -6148(SP), SP
                    54        04    AC  D0 00007                MOVL    VOLB, R4                                    ; 1424
                              10    A4  DD 0000B                PUSHL   16(R4)
                    53        14    A4  D0 0000E                MOVL    20(R4), R3                                  ; 1425
                    52  041B00F3    8F  D0 00012                MOVL    #68878579, R2                               ; 1429
                    51      020F    8F  3C 00019                MOVZWL  #527, R1
                    50            54  D0 0001E                MOVL    R4, R0
                        00000000G   EF  16 00021                JSB     EXCH$UTIL_BLOCK_CHECK
                    5A        0C    AC  D0 00027                MOVL    BLKCNT, R10                                 ; 1433
                                  09  13 0002B                BEQL    1$
              0000007F  8F            5A  D1 0002D            CMPL    R10, #127
                                      13  1B 00034            BLEQU   2$
                    7E        8C    8F  9A 00036 1$:        MOVZBL  #140, -(SP)
                                      01  DD 0003A            PUSHL   #1
                        00000000G   8F  DD 0003C            PUSHL   #EXCH$_BADLOGIC
              00000000G 00            03  FB 00042            CALLS   #3, LIB$STOP
59                    5A            09  78 00049 2$:        ASHL    #9, R10, BYTECNT                            ; 1434
                    5B        08    AC  D0 0004D            MOVL    BLKPBN, R11                                 ; 1438
                 44   A4            5B  D1 00051            CMPL    R11, 68(R4)
                                      0E  1F 00055            BLSSU   3$
                                      7E  D4 00057            CLRL    -(SP)                                       ; 1440
                              04    AE  DD 00059            PUSHL   4(SP)
```

```
                                0001827A   8F   DD 0005C          PUSHL   #98938                                    ;
                                     00CA  31   00062            BRW     11$                                      ;
                                        58 5B   D0 00065 3$:     MOVL    R11, LBN                                 ; 1445
                              02     48 A4      04   E1 00068    BBC     #4, 72(R4), 4$                           ; 1446
                                        58      D6   00070      INCL    LBN                                      ; 1448
                                    04  AE      D4   00072 4$:   CLRL    4(SP)                                    ; 1452
                                        58      D5   00074      TSTL    LBN                                      ;
                                        15      12   00076      BNEQ    5$                                       ;
                                    04  AE      D6   00078      INCL    4(SP)                                    ;
                                        53      DD   0007B      PUSHL   R3                                       ; 1459
                          00000000G     00      01   FB 0007D   CALLS   #1, SYS$REWIND                           ;
                                        55      50   D0 00084   MOVL    R0, STATUS                               ;
                                        03      55   E8 00087   BLBS    STATUS, 5$                               ;
                                         009C   31   00088      BRW     10$                                      ;
                                        56  10  AC   D0 0008B 5$:  MOVL    BUFADR, R6                               ; 1470
                                     24 A3      56   D0 0008F   MOVL    R6, 36(R3)                               ;
                                     20 A3      59   B0 00093   MOVW    BYTECNT, 32(R3)                          ; 1471
                                     38 A3      58   D0 00097   MOVL    LBN, 56(R3)                              ; 1472
                                        53      DD   0009B      PUSHL   R3                                       ; 1476
                          00000000G     00      01   FB 0009D   CALLS   #1, SYS$READ                            ;
                                        55      50   D0 000A4   MOVL    R0, STATUS                               ;
                                        3D      55   E8 000A7   BLBS    STATUS, 7$                               ;
                                0001C0F4  8F      55   D1 000AA   CMPL    STATUS, #114932                          ; 1481
                                        74      12   000B1      BNEQ    10$                                      ;
                                        52  0C  A3   D0 000B3   MOVL    12(R3), STV                              ; 1487
                                        56      DD   000B7      PUSHL   R6                                       ; 1488
                                        5A      DD   000B9      PUSHL   R10                                      ;
                                     0810  8F      BB   000BB   PUSHR   #^M<R4,R11>                              ;
                                0000V   CF         04   FB 000BF   CALLS   #4, EXCH$IO_RT11_ EAD_1                ;
                                        57         50   D0 000C4   MOVL    R0, STAT_1                           ;
                                        19         57   E9 000C7   BLBC    STAT_1, 6$                           ; 1489
                                                   52   DD 000CA   PUSHL   STV                                  ; 1492
                                                   55   DD 000CC   PUSHL   STATUS                               ;
                                                69 A4   9F 000CE   PUSHAB  105(R4)                               ;
                                                65 A4   DD 000D1   PUSHL   101(R4)                               ;
                                                   02   DD 000D4   PUSHL   #2                                    ;
                                          00000000G  8F   DD 000D6   PUSHL   #EXCH$_READERRREC                     ;
                                00000000G     00      06   FB 000DC   CALLS   #6, LIB$SIGNAL                      ;
                                              50      57   D0 000E3 6$:  MOVL    STAT_1, R0                       ; 1496
                                                      04   000E6      RET                                         ;
                                              50  22  A3   3C 000E7 7$:  MOVZWL  34(R3), TRANSFERRED             ; 1499
                                                      57   D4 000EB   CLRL    R7                                  ; 1500
                                              59      50   D1 000ED   CMPL    TRANSFERRED, BYTECNT                ;
                                                      1A   13 000F0   BEQL    8$                                  ;
                                                      57   D6 000F2   INCL    R7                                  ;
                                                      21   BB 000F4   PUSHR   #^M<R0,R5>                         ; 1503
                                                      59   DD 000F6   PUSHL   BYTECNT                             ;
                                                0000'  CF   9F 000F8   PUSHAB  P.AAG                               ;
                                00000000G     EF      04   FB 000FC   CALLS   #4, EXCH$UTIL_FAO_BUFFER           ;
                                                      50   DD 00103   PUSHL   R0                                  ;
                                00000000G     00      01   FB 00105   CALLS   #1, LIB$PUT_OUTPUT                  ;
                             03       48  A4         01   E0 0010C 8$:  BBS     #1, 72(R4), 9$                   ; 1507
                                                   00C8  31   00111   BRW     18$                                 ;
                                              1F      04   AE E9 00114 9$:  BLBC    4(SP), 12$                   ; 1515
                                                      53   DD 00118   PUSHL   R3                                  ; 1518
                                00000000G     00      01   FB 0011A   CALLS   #1, SYS$REWIND                      ;
                                              55      50   D0 00121   MOVL    R0, STATUS                          ;
```

```
                        10            55  E8 00124        BLBS    STATUS, 12$                             ;
                                0C    A3  DD 00127  10$:  PUSHL   12(R3)                                  ; 1522
                                04    AE  DD 0012A        PUSHL   4(SP)                                   ;
                                      55  DD 0012D        PUSHL   STATUS                                  ;
                          00F810B0    8F  DD 0012F  11$:  PUSHL   #16257200                               ;
                                      6A  11 00135        BRB     15$                                     ;
                        52            59  D0 00137  12$:  MOVL    BYTECNT, LEN                            ; 1526
                  24    A3      08    AE  9E 0013A        MOVAB   CHECK_BUF, 36(R3)                       ; 1528
                  20    A3            52  B0 0013F        MOVW    LEN, 32(R3)                             ; 1529
                  38    A3            58  D0 00143        MOVL    LBN, 56(R3)                             ; 1530
                                      53  DD 00147        PUSHL   R3                                      ; 1534
           00000000G    00            01  FB 00149        CALLS   #1, SYS$READ                            ;
                        55            50  D0 00150        MOVL    R0, STATUS                              ;
                        53            55  E8 00153        BLBS    STATUS, 16$                             ;
            0001C0F4    8F            55  D1 00156        CMPL    STATUS, #114932                         ; 1539
                                      34  12 0015D        BNEQ    14$                                     ;
                        53      0C    A3  D0 0015F        MOVL    12(R3), STV                             ; 1545
                                      56  DD 00163        PUSHL   R6                                      ; 1546
                                      5A  DD 00165        PUSHL   R10                                     ;
                              0810    8F  BB 00167        PUSHR   #^M<R4,R11>                             ;
               0000V    CF            04  FB 0016B        CALLS   #4, EXCH$IO_RT11_READ_1                 ;
                        59            50  D0 00170        MOVL    R0, STAT_1                              ;
                        19            59  E9 00173        BLBC    STAT_1, 13$                             ; 1547
                                      53  DD 00176        PUSHL   STV                                     ; 1550
                                      55  DD 00178        PUSHL   STATUS                                  ;
                                69    A4  9F 0017A        PUSHAB  105(R4)                                 ;
                                65    A4  DD 0017D        PUSHL   101(R4)                                 ;
                                      02  DD 00180        PUSHL   #2                                      ;
                         00000000G    8F  DD 00182        PUSHL   #EXCH$_READCHECKREC                     ;
           00000000G    00            06  FB 00188        CALLS   #6, LIB$SIGNAL                          ;
                        50            59  D0 0018F  13$:  MOVL    STAT_1, R0                              ; 1554
                                         04 00192        RET                                             ;
                                0C    A3  DD 00193  14$:  PUSHL   12(R3)                                  ;
                                04    AE  DD 00196        PUSHL   4(SP)                                   ;
                                      55  DD 00199        PUSHL   STATUS                                  ;
                         00000000G    8F  DD 0019B        PUSHL   #EXCH$_READCHECK                        ;
           00000000G    EF            04  FB 001A1  15$:  CALLS   #4, EXCH$UTIL_FILE_ERROR                ;
                                         04 001A8        RET                                             ;
      66        08      AE            52  29 001A9  16$:  CMPC3   LEN, CHECK_BUF, (R6)                    ; 1557
                                      2C  13 001AE        BEQL    18$                                     ;
                                      56  DD 001B0        PUSHL   R6                                      ; 1595
                                      5A  DD 001B2        PUSHL   R10                                     ;
                              0810    8F  BB 001B4        PUSHR   #^M<R4,R11>                             ;
               0000V    CF            04  FB 001B8        CALLS   #4, EXCH$IO_RT11_READ_1                 ;
                        52            50  D0 001BD        MOVL    R0, STAT_1                              ;
                        15            52  E9 001C0        BLBC    STAT_1, 17$                             ; 1596
                                69    A4  9F 001C3        PUSHAB  105(R4)                                 ; 1598
                                65    A4  DD 001C6        PUSHL   101(R4)                                 ;
                                      02  DD 001C9        PUSHL   #2                                      ;
                         00000000G    8F  DD 001CB        PUSHL   #EXCH$_READCHECKREC                     ;
           00000000G    00            04  FB 001D1        CALLS   #4, LIB$SIGNAL                          ;
                        50            52  D0 001D8  17$:  MOVL    STAT_1, R0                              ; 1599
                                         04 001DB        RET                                             ;
                        14            57  E9 001DC  18$:  BLBC    R7, 19$                                 ; 1606
                        7E    013B    8F  3C 001DF        MOVZWL  #315, -(SP)                             ; 1608
                                      01  DD 001E4        PUSHL   #1                                      ;
                         00000000G    8F  DD 001E6        PUSHL   #EXCH$_BADLOGIC                         ;
```

```
                 000000000G  00        03  FB 001EC          CALLS   #3, LIB$STOP                                    ;
                             50        55  D0 001F3 19$:      MOVL    STATUS, R0                                      ; 1610
                                           04 001F6           RET                                                     ; 1611

; Routine Size:  503 bytes,    Routine Base:  EXCH$IO_CODE + 0A22
```

```
;1539     1612  1 GLOBAL ROUTINE exch$io_rt11_read_1 (volb : $ref_bblock,        %SBTTL 'exch$io_rt11_read_1 (volb, pbn, cnt,
;1540     1613  1                                     blkpbn, blkcnt, bufadr : $ref_bvector) =
;1541     1614  2 BEGIN
;1542     1615  2 !++
;1543     1616  2 !
;1544     1617  2 ! FUNCTIONAL DESCRIPTION:
;1545     1618  2 !
;1546     1619  2 !       Read a range of "physical" blocks from the device.  Block numbers are in the range 0 to MaxBlocks-1.
;1547     1620  2 !       Blocks are read one at a time to try to locate bad blocks
;1548     1621  2 !
;1549     1622  2 ! INPUTS:
;1550     1623  2 !
;1551     1624  2 !       volb   - pointer to a volb which contains the active record stream
;1552     1625  2 !       blkpbn - starting physical block number on device
;1553     1626  2 !       blkcnt - number of blocks to read
;1554     1627  2 !
;1555     1628  2 ! IMPLICIT INPUTS:
;1556     1629  2 !
;1557     1630  2 !       none
;1558     1631  2 !
;1559     1632  2 ! OUTPUTS:
;1560     1633  2 !
;1561     1634  2 !       bufadr - pointer to buffer to receive the data
;1562     1635  2 !
;1563     1636  2 ! IMPLICIT OUTPUTS:
;1564     1637  2 !
;1565     1638  2 !       none
;1566     1639  2 !
;1567     1640  2 ! ROUTINE VALUE:
;1568     1641  2 !
;1569     1642  2 !       success or error status
;1570     1643  2 !
;1571     1644  2 ! SIDE EFFECTS:
;1572     1645  2 !
;1573     1646  2 !       none
;1574     1647  2 !--
;1575     1648  2
;1576     1649  2 $dbgtrc_prefix ('io_rt11_read_1> ');
;1577     1650  2
;1578     1651  2 LOCAL
;1579     1652  2     check_buf : $bvector [512],
;1580     1653  2     adr,
;1581     1654  2     lbn,
;1582     1655  2     status
;1583     1656  2     ;
;1584     1657  2
;1585     1658  2 BIND
;1586     1659  2     fab = .volb [volb$a_fab] : $bblock, ! New name for File Access Block
;1587     1660  2     rab = .volb [volb$a_rab] : $bblock, ! New name for Record Access Block
;1588     1661  2     nam = .volb [volb$a_nam] : $bblock  ! New name for NAM block
;1589     1662  2     ;
;1590     1663  2
;1591     1664  2 $block_check (2, .volb, volb, 563);
;1592     1665  2 $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
```

```
;1594     1666  2 $logic_check (2, ((.blkcnt GTRU 0) AND (.blkcnt LEQU 127)), 205);
;1595     1667  2
;1596     1668  2 ! Adjust the externally useful "physical" block number to the RMS required
;1597     1669  2 ! logical block number
;1598     1670  2 !
;1599     1671  2 $logic_check (2, (.blkpbn LSSU .volb [volb$l_volmaxtlock]), 213);
;1600     1672  2 lbn = .blkpbn;                                  ! Assume volume relative lbn.
;1601     1673  2 IF .volb [volb$v_virtual]
;1602     1674  2 THEN
;1603     1675  2     lbn = .lbn + 1;                             ! File-relative virtual block numbers
;1604     1676  2
;1605     1677  2 ! See if the request is for block 0
;1606     1678  2 !
;1607     1679  2 IF .lbn EQL 0
;1608     1680  2 THEN
;1609     1681  3     BEGIN
;1610     1682  3
;1611     1683  3     ! Since BKT = 0 means sequential read, we must trick it with the
;1612     1684  3     ! documented technique of REWIND, then set BKT=0.
;1613     1685  3     !
;1614     1686  4     IF NOT (status = $rewind (rab = rab))
;1615     1687  3     THEN
;1616     1688  4         BEGIN
;1617     1689  4         $debug_print_lit ('rewind error');
;1618     1690  4         RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;1619     1691  3         END;
;1620     1692  3
;1621     1693  2     END;
;1622     1694  2
;1623     1695  2 ! Prepare for the loop
;1624     1696  2 !
;1625     1697  2 rab [rab$w_usz] = 512;                          ! User buffer size
;1626     1698  2 adr = .bufadr;
;1627     1699  2
;1628     1700  2 DECR cnt FROM .blkcnt-1 TO 0
;1629     1701  2 DO
;1630     1702  3     BEGIN
;1631     1703  3
;1632     1704  3     ! Set the RMS record stream buffer and block parameters
;1633     1705  3     !
;1634     1706  3     rab [rab$l_ubf] = .adr;                     ! User buffer address
;1635     1707  3     rab [rab$l_bkt] = .lbn;
;1636     1708  3
;1637     1709  3     ! Read the chunk
;1638     1710  3     !
;1639     1711  4     IF NOT (status = $read (rab = rab))
;1640     1712  3     THEN
;1641     1713  4         BEGIN
;1642     1714  4         $trace_print_fao ('error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz] / 512);
;1643     1715  4         RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;1644     1716  3         END;
;1645     1717  3
;1646     1718  3     ! If read checking, reread the chunk and compare
;1647     1719  3     !
;1648     1720  3     IF .volb [volb$v_read_check]
;1649     1721  3     THEN
;1650     1722  4         BEGIN
```

```
;  1651          1723  4          LOCAL
;  1652          1724  4              tmp_desc : VECTOR [2, LONG],
;  1653          1725  4              len;
;  1654          1726  4
;  1655          1727  4          ! See if the request is for block 0, if so rewind
;  1656          1728  4          !
;  1657          1729  4          IF .lbn EQL 0
;  1658          1730  4          THEN
;  1659          1731  5              BEGIN
;  1660          1732  6              IF NOT (status = $rewind (rab = rab))
;  1661          1733  5              THEN
;  1662          1734  6                  BEGIN
;  1663          1735  6                  $debug_print_lit ('rewind error');
;  1664          1736  6                  RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;  1665          1737  5                  END;
;  1666          1738  4              END;
;  1667          1739  4
;  1668          1740  4          len = 512;
;  1669          1741  4
;  1670          1742  4          rab [rab$l_ubf] = check_buf;            ! User buffer address
;  1671          1743  4          rab [rab$w_usz] = .len;                 ! User buffer size
;  1672          1744  4          rab [rab$l_bkt] = .lbn;
;  1673          1745  4
;  1674          1746  4          ! Read the chunk again
;  1675          1747  4          !
;  1676          1748  5          IF NOT (status = $read (rab = rab))
;  1677          1749  4          THEN
;  1678          1750  5              BEGIN
;  1679          1751  5              $trace_print_fao ('reread error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz]
;  1680          1752  5              RETURN exch$util_file_error (exch$_readcheck, .status, fab, .rab [rab$l_stv]);
;  1681          1753  4              END;
;  1682          1754  4
;  1683          1755  4          IF CH$NEQ (.len, check_buf, .len, .adr, 0)
;  1684          1756  4          THEN
;  1685          1757  5              BEGIN
;  1686        L 1758  5              %IF switch_trace
;  1687        U 1759  5              %THEN
;  1688        U 1760  5                  LITERAL
;  1689        U 1761  5                      chunk = 100;
;  1690        U 1762  5                  LOCAL
;  1691        U 1763  5                      c,          ! count of loops
;  1692        U 1764  5                      l,          ! length of string remaining
;  1693        U 1765  5                      a1,         ! address of remainder of string 1
;  1694        U 1766  5                      a2;         ! address of remainder of string 2
;  1695        U 1767  5                  c = 0;
;  1696        U 1768  5                  l = .len;
;  1697        U 1769  5                  a1 = .adr;
;  1698        U 1770  5                  a2 = check_buf;
;  1699        U 1771  5                  $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .adr, .volb);
;  1700        U 1772  5                  WHILE .l GTR 0
;  1701        U 1773  5                  DO
;  1702        U 1774  5                      BEGIN
;  1703        U 1775  5                      LOCAL
;  1704        U 1776  5                          ll;             ! length for this loop
;  1705        U 1777  5                      c = .c + 1;
;  1706        U 1778  5                      ll = MINU (.l, chunk);
;  1707        U 1779  5                      IF CH$NEQ (.ll, .a1, .ll, .a2, 0)
```

```
;  1708        U 1780  5                       THEN
;  1709        U 1781  5                           BEGIN
;  1710        U 1782  5                           $trace_print_fao ('!4UL  "!AF"', .c, .ll, .a1);
;  1711        U 1783  5                           $trace_print_fao ('      "!AF"', .ll, .a2);
;  1712        U 1784  5                           END;
;  1713        U 1785  5                       l = .l - chunk;
;  1714        U 1786  5                       a1 = .a1 + chunk;
;  1715        U 1787  5                       a2 = .a2 + chunk;
;  1716        U 1788  5                       END;
;  1717          1789  5                 %FI
;  1718          1790  5                 tmp_desc [0] = .volb [volb$l_vol_ident_len];
;  1719          1791  5                 tmp_desc [1] = volb [volb$t_vol_ident];
;  1720          1792  5                 $exch_signal_return (exch$_readcheck, 1, tmp_desc);
;  1721          1793  4                 END;
;  1722          1794  4
;  1723          1795  3             END;
;  1724          1796  3
;  1725          1797  3         ! Move to the next block
;  1726          1798  3         !
;  1727          1799  3         adr = .adr + 512;
;  1728          1800  3         lbn = .lbn + 1;
;  1729          1801  3
;  1730          1802  2         END;
;  1731          1803  2
;  1732          1804  2 RETURN .status;
;  1733          1805  1 END;


                                                OFFC 00000          .ENTRY  EXCH$IO_RT11_READ_1, Save R2,R3,R4,R5,R6,-      ; 1612
                                                                            R7,R8,R9,R10,R11
                                5B 00000000G    8F   D0 00002       MOVL    #EXCH$_BADLOGIC, R11
                                5E       FDF8   CE   9E 00009       MOVAB   -520(SP), SP
                                56         04   AC   D0 0000E       MOVL    VOLB, R6                                        ; 1659
                                5A         10   A6   D0 00012       MOVL    16(R6), R10
                                55         14   A6   D0 00016       MOVL    20(R6), R5                                      ; 1660
                                52 041B00F3     8F   D0 0001A       MOVL    #68878579, R2                                   ; 1664
                                51       0233   8F   3C 00021       MOVZWL  #563, R1
                                50              56   D0 00026       MOVL    R6, R0
                                   00000000G    EF   16 00029       JSB     EXCH$UTIL_BLOCK_CHECK
                                54         0C   AC   D0 0002F       MOVL    BLKCNT, R4                                      ; 1666
                                                09   13 00033       BEQL    1$
0000007F                        8F              54   D1 00035       CMPL    R4, #127
                                                0F   1B 0003C       BLEQU   2$
                                7E         CD   8F   9A 0003E 1$:   MOVZBL  #205, -(SP)
                                                01   DD 00042       PUSHL   #1
                                                5B   DD 00044       PUSHL   R11
00000000G                       00              03   FB 00046       CALLS   #3, LIB$STOP
       44                       A6         08   AC   D1 0004D 2$:   CMPL    BLKPBN, 68(R6)                                  ; 1671
                                                05   1F 00052       BLSSU   3$
                                7E         D5   8F   9A 00054       MOVZBL  #213, -(SP)
                                                01   DD 00058       PUSHL   #1
                                                5B   DD 0005A       PUSHL   R11
00000000G                       00              03   FB 0005C       CALLS   #3, LIB$STOP
                                58         08   AC   D0 00063 3$:   MOVL    BLKPBN, LBN                                     ; 1672
```

```
                02      48   A6              04  E1 00067          BBC     #4, 72(R6), 4$                          ; 1673
                                             58  D6 0006C          INCL    LBN                                     ; 1675
                                             0F  12 0006E 4$:      BNEQ    5$                                      ; 1679
                                             55  DD 00070          PUSHL   R5                                      ; 1686
                    00000000G   00           01  FB 00072          CALLS   #1, SYS$REWIND                          ;
                                57           50  D0 00079          MOVL    R0, STATUS                              ;
                                3F           57  E9 0007C          BLBC    STATUS, 8$                              ;
                          20    A5      0200 8F  B0 0007F 5$:      MOVW    #512, 32(R5)                            ; 1697
                                59        10 AC  D0 00085          MOVL    BUFADR, ADR                             ; 1698
                                         00A6 31  00089            BRW     13$                                     ; 1700
                          24    A5           59  D0 0008C 6$:      MOVL    ADR, 36(R5)                             ; 1706
                          38    A5           58  D0 00090          MOVL    LBN, 56(R5)                             ; 1707
                                             55  DD 00094          PUSHL   R5                                      ; 1711
                    00000000G   00           01  FB 00096          CALLS   #1, SYS$READ                            ;
                                57           50  D0 0009D          MOVL    R0, STATUS                              ;
                                1B           57  E9 000A0          BLBC    STATUS, 8$                              ;
                03      48   A6              01  E0 000A3          BBS     #1, 72(R6), 7$                          ; 1720
                                         0080 31  000A8            BRW     12$                                     ;
                                             58  D5 000AB 7$:      TSTL    LBN                                     ; 1729
                                             1E  12 000AD          BNEQ    9$                                      ;
                                             55  DD 000AF          PUSHL   R5                                      ; 1732
                    00000000G   00           01  FB 000B1          CALLS   #1, SYS$REWIND                          ;
                                57           50  D0 000B8          MOVL    R0, STATUS                              ;
                                0F           57  E8 000BB          BLBS    STATUS, 9$                              ;
                                      0C     A5  DD 000BE 8$:      PUSHL   12(R5)                                  ; 1736
                                    0480     8F  BB 000C1          PUSHR   #^M<R7,R10>                             ;
                                00F810B0     8F  DD 000C5          PUSHL   #16257200                               ;
                                             2E  11 000CB          BRB     10$                                     ;
                          52        0200     8F  3C 000CD 9$:      MOVZWL  #512, LEN                               ; 1740
                          24    A5    08     AE  9E 000D2          MOVAB   CHECK_BUF, 36(R5)                       ; 1742
                          20    A5           52  B0 000D7          MOVW    LEN, 32(R5)                             ; 1743
                          38    A5           58  D0 000DB          MOVL    LBN, 56(R5)                             ; 1744
                                             55  DD 000DF          PUSHL   R5                                      ; 1748
                    00000000G   00           01  FB 000E1          CALLS   #1, SYS$READ                            ;
                                57           50  D0 000E8          MOVL    R0, STATUS                              ;
                                15           57  E8 000EB          BLBS    STATUS, 11$                             ;
                                      0C     A5  DD 000EE          PUSHL   12(R5)                                  ; 1752
                                    0480     8F  BB 000F1          PUSHR   #^M<R7,R10>                             ;
                                00000000G    8F  DD 000F5          PUSHL   #EXCH$_READCHECK                        ;
                    00000000G   EF           04  FB 000FB 10$:     CALLS   #4, EXCH$UTIL_FILE_ERROR                ;
                                             04  00102            RET                                             ;
             69       08    AE              52  29 00103 11$:     CMPC3   LEN, CHECK_BUF, (ADR)                   ; 1755
                                             21  13 00108          BEQL    12$                                     ;
                                6E        65 A6  D0 0010A          MOVL    101(R6), TMP_DESC                       ; 1790
                          04    AE        69 A6  9E 0010E          MOVAB   105(R6), TMP_DESC+4                     ; 1791
                                52 00000000G 8F  D0 00113          MOVL    #EXCH$_READCHECK, TEMP                  ; 1792
                                             5E  DD 0011A          PUSHL   SP                                      ;
                                             01  DD 0011C          PUSHL   #1                                      ;
                                             52  DD 0011E          PUSHL   TEMP                                    ;
                    00000000G   00           03  FB 00120          CALLS   #3, LIB$SIGNAL                          ;
                                50           52  D0 00127          MOVL    TEMP, R0                                ;
                                             04  0012A            RET                                             ;
                                59      0200 C9  9E 0012B 12$:     MOVAB   512(R9), ADR                            ; 1799
                                             58  D6 00130          INCL    LBN                                     ; 1800
                                02           54  F4 00132 13$:     SOBGEQ  CNT, 14$                                ; 1700
                                             03  11 00135          BRB     15$                                     ;
                                         FF52 31  00137 14$:     BRW     6$                                      ;
```

```
                                50          57  D0 0013A 15$:    MOVL    STATUS, R0                      ; 1804
                                                04 0013D         RET                                     ; 1805

; Routine Size:  318 bytes,    Routine Base:  EXCH$IO_CODE + 0C19
```

```
;  1735     1806  1 GLOBAL ROUTINE exch$io_rt11_write (volb : $ref_bblock,  %SBTTL 'exch$io_rt11_write (volb, pbn, cnt, adr)'
;  1736     1807  1                                    blkpbn, blkcnt, bufadr : $ref_bvector) =
;  1737     1808  2 BEGIN
;  1738     1809  2 !++
;  1739     1810  2 !
;  1740     1811  2 ! FUNCTIONAL DESCRIPTION:
;  1741     1812  2 !
;  1742     1813  2 !       Write a range of "physical" blocks to a random access device.  Block
;  1743     1814  2 !       numbers are in the range 0 to MaxBlocks-1.
;  1744     1815  2 !
;  1745     1816  2 ! INPUTS:
;  1746     1817  2 !
;  1747     1818  2 !       volb   - pointer to a volb which contains the active record stream
;  1748     1819  2 !       blkpbn - starting physical block number on device
;  1749     1820  2 !       blkcnt - number of blocks to write
;  1750     1821  2 !       bufadr - pointer to buffer containing the data
;  1751     1822  2 !
;  1752     1823  2 ! IMPLICIT INPUTS:
;  1753     1824  2 !
;  1754     1825  2 !       none
;  1755     1826  2 !
;  1756     1827  2 ! OUTPUTS:
;  1757     1828  2 !
;  1758     1829  2 !       none
;  1759     1830  2 !
;  1760     1831  2 ! IMPLICIT OUTPUTS:
;  1761     1832  2 !
;  1762     1833  2 !       none
;  1763     1834  2 !
;  1764     1835  2 ! ROUTINE VALUE:
;  1765     1836  2 !
;  1766     1837  2 !       success or error status
;  1767     1838  2 !
;  1768     1839  2 ! SIDE EFFECTS:
;  1769     1840  2 !
;  1770     1841  2 !       none
;  1771     1842  2 !--
;  1772     1843  2
;  1773     1844  2 $dbgtrc_prefix ('io_rt11_write> ');
;  1774     1845  2
;  1775     1846  2 LOCAL
;  1776     1847  2     check_buf : $bvector [ctx$k_buffer_length],
;  1777     1848  2     bytecnt,
;  1778     1849  2     lbn,
;  1779     1850  2     status
;  1780     1851  2     ;
;  1781     1852  2
;  1782     1853  2 BIND
;  1783     1854  2     fab = .volb [volb$a_fab] : $bblock, ! New name for File Access Block
;  1784     1855  2     rab = .volb [volb$a_rab] : $bblock, ! New name for Record Access Block
;  1785     1856  2     nam = .volb [volb$a_nam] : $bblock  ! New name for NAM block
;  1786     1857  2     ;
;  1787     1858  2
;  1788     1859  2 $block_check (2, .volb, volb, 428);
;  1789     1860  2 $logic_check (2, (.volb [volb$v_write]), 145);  ! We shouldn't get this far if we aren't supposed to write t
;  1790     1861  2
;  1791     1862  2 $trace_print_fao ('  pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
```

```
;  1793     1863  2 ! Adjust the externally useful block count to the RMS required byte count
;  1794     1864  2 !
;  1795     1865  2 $logic_check (2, ((.blkcnt GTRU 0) AND (.blkcnt LEQU 127)), 112);
;  1796     1866  2 bytecnt = .blkcnt * 512;
;  1797     1867  2
;  1798     1868  2 ! Adjust the externally useful "physical" block number to the RMS required
;  1799     1869  2 ! logical block number
;  1800     1870  2 !
;  1801     1871  2 $logic_check (2, (.blkpbn LSSU .volb [volb$l_volmaxblock]), 113);
;  1802     1872  2 lbn = .blkpbn;                                  ! Assume volume relative lbn.
;  1803     1873  2 IF .volb [volb$v_virtual]
;  1804     1874  2 THEN
;  1805     1875  2     lbn = .lbn + 1;                             ! File-relative virtual block numbers
;  1806     1876  2
;  1807     1877  2 ! See if the request is for block 0
;  1808     1878  2 !
;  1809     1879  2 IF .lbn EQL 0
;  1810     1880  2 THEN
;  1811     1881  3     BEGIN
;  1812     1882  3
;  1813     1883  3     ! Since BKT = 0 means sequential read, we must trick it with the
;  1814     1884  3     ! documented technique of REWIND, then set BKT=0.
;  1815     1885  3     !
;  1816     1886  4     IF NOT (status = $rewind (rab = rab))
;  1817     1887  3     THEN
;  1818     1888  4         BEGIN
;  1819     1889  4         $debug_print_lit ('rewind error');
;  1820     1890  4         RETURN exch$util_file_error (exch$_writeerr, .status, fab, .rab [rab$l_stv]);
;  1821     1891  3         END;
;  1822     1892  3
;  1823     1893  2     END;
;  1824     1894  2
;  1825     1895  2 ! Set the RMS record stream buffer and block parameters
;  1826     1896  2 !
;  1827     1897  2 rab [rab$l_rbf] = .bufadr;                      ! Record buffer address
;  1828     1898  2 rab [rab$w_rsz] = .bytecnt;                     ! Record buffer size
;  1829     1899  2 rab [rab$l_bkt] = .lbn;
;  1830     1900  2
;  1831     1901  2 ! Write the chunk
;  1832     1902  2 !
;  1833     1903  3 IF NOT (status = $write (rab = rab))
;  1834     1904  2 THEN
;  1835     1905  3     BEGIN
;  1836     1906  3     $trace_print_fao ('error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_rsz] / 512);
;  1837     1907  3     $check_call (4, exch$util_file_error, exch$_writeerr, .status, fab, .rab [rab$l_stv]);
;  1838     1908  3
;  1839     1909  3     ! If the error is a simple WER write error, try to rewrite it using single block I/O
;  1840     1910  3     !
;  1841     1911  3     IF .status EQL rms$_wer
;  1842     1912  3     THEN
;  1843     1913  4         BEGIN
;  1844     1914  4         LOCAL
;  1845     1915  4             stv,
;  1846     1916  4             stat_1;
;  1847     1917  4         stv = .rab [rab$l_stv];
;  1848     1918  4         stat_1 = exch$io_rt11_write_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;  1849     1919  4         IF .stat_1
```

```
;1850        1920  4          THEN
;1851      P 1921  4              $exch_signal (exch$_writeerrrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
;1852        1922  4                                                  .status, .stv);
;1853        1923  4          RETURN .stat_1;
;1854        1924  4          END
;1855        1925  4
;1856        1926  4      ! If the error is anything else, signal it and return
;1857        1927  4      !
;1858        1928  3      ELSE
;1859        1929  3          RETURN exch$util_file_error (exch$_writeerr, .status, fab, .rab [rab$l_stv]);
;1860        1930  2      END;
;1861        1931  2
;1862        1932  2  ! If write checking, reread the chunk and compare
;1863        1933  2  !
;1864        1934  2  IF .volb [volb$v_write_check]
;1865        1935  2  THEN
;1866        1936  3      BEGIN
;1867        1937  3      LOCAL
;1868        1938  3          len;
;1869        1939  3
;1870        1940  3      ! See if the request is for block 0, if so rewind
;1871        1941  3      !
;1872        1942  3      IF .lbn EQL 0
;1873        1943  3      THEN
;1874        1944  4          BEGIN
;1875        1945  5          IF NOT (status = $rewind (rab = rab))
;1876        1946  4          THEN
;1877        1947  5              BEGIN
;1878        1948  5              $debug_print_lit ('rewind error');
;1879        1949  5              RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;1880        1950  4              END;
;1881        1951  3          END;
;1882        1952  3
;1883        1953  3      len = .bytecnt;
;1884        1954  3
;1885        1955  3      rab [rab$l_ubf] = check_buf;                ! User buffer address
;1886        1956  3      rab [rab$w_usz] = .len;
;1887        1957  3      rab [rab$l_bkt] = .lbn;
;1888        1958  3
;1889        1959  3      ! Read the chunk again
;1890        1960  3      !
;1891        1961  4      IF NOT (status = $read (rab = rab))
;1892        1962  3      THEN
;1893        1963  4          BEGIN
;1894        1964  4          $trace_print_fao ('reread error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz] / 51
;1895        1965  4          $check_call (4, exch$util_file_error, exch$_writecheck, .status, fab, .rab [rab$l_stv]);
;1896        1966  4          IF .status EQL rms$_rer
;1897        1967  4          THEN
;1898        1968  5              BEGIN
;1899        1969  5              LOCAL
;1900        1970  5                  stv,
;1901        1971  5                  stat_1;
;1902        1972  5              stv = .rab [rab$l_stv];
;1903        1973  5              stat_1 = exch$io_rt11_write_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;1904        1974  5              IF .stat_1
;1905        1975  5              THEN
;1906      P 1976  5                  $exch_signal (exch$_writecheckrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],
```

```
;  1907          1977  5                                                    .status, .stv);
;  1908          1978  5                RETURN .stat_1;
;  1909          1979  5                END
;  1910          1980  4            ELSE
;  1911          1981  4                RETURN exch$util_file_error (exch$_writecheck, .status, fab, .rab [rab$l_stv]);
;  1912          1982  3            END;
;  1913          1983  3
;  1914          1984  3        IF CH$NEQ (.len, check_buf, .len, .bufadr, 0)
;  1915          1985  3        THEN
;  1916          1986  4            BEGIN
;  1917        L 1987  4            %IF switch_trace
;  1918        U 1988  4            %THEN
;  1919        U 1989  4                LITERAL
;  1920        U 1990  4                    chunk = 100;
;  1921        U 1991  4                LOCAL
;  1922        U 1992  4                    c,                  ! count of loops
;  1923        U 1993  4                    l,                  ! length of string remaining
;  1924        U 1994  4                    a1,                 ! address of remainder of string 1
;  1925        U 1995  4                    a2;                 ! address of remainder of string 2
;  1926        U 1996  4                c = 0;
;  1927        U 1997  4                l = .len;
;  1928        U 1998  4                a1 = .bufadr;
;  1929        U 1999  4                a2 = check_buf;
;  1930        U 2000  4                $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
;  1931        U 2001  4                WHILE .l GTR 0
;  1932        U 2002  4                DO
;  1933        U 2003  4                    BEGIN
;  1934        U 2004  4                    LOCAL
;  1935        U 2005  4                        ll;             ! length for this loop
;  1936        U 2006  4                    c = .c + 1;
;  1937        U 2007  4                    ll = MINU (.l, chunk);
;  1938        U 2008  4                    IF CH$NEQ (.ll, .a1, .ll, .a2, 0)
;  1939        U 2009  4                    THEN
;  1940        U 2010  4                        BEGIN
;  1941        U 2011  4                        $trace_print_fao ('!4UL  "!AF"', .c, .ll, .a1);
;  1942        U 2012  4                        $trace_print_fao ('      "!AF"', .ll, .a2);
;  1943        U 2013  4                        END;
;  1944        U 2014  4                    l = .l - chunk;
;  1945        U 2015  4                    a1 = .a1 + chunk;
;  1946        U 2016  4                    a2 = .a2 + chunk;
;  1947        U 2017  4                    END;
;  1948          2018  4            %FI
;  1949          2019  5            BEGIN
;  1950          2020  5            LOCAL
;  1951          2021  5                stat_1;
;  1952          2022  5            stat_1 = exch$io_rt11_write_1 (.volb, .blkpbn, .blkcnt, .bufadr);
;  1953          2023  5            IF .stat_1
;  1954          2024  5            THEN
;  1955          2025  5                $exch_signal (exch$_writecheckrec, 2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  1956          2026  5            RETURN .stat_1;
;  1957          2027  4            END;
;  1958          2028  3            END;
;  1959          2029  2        END;
;  1960          2030  2
;  1961          2031  2 RETURN .status;
;  1962          2032  1 END;
```

```
                                                        .EXTRN  SYS$WRITE, EXCH$_WRITEERRREC
                                                        .EXTRN  EXCH$_WRITECHECK
                                                        .EXTRN  EXCH$_WRITECHECKREC

                                 0FFC 00000             .ENTRY  EXCH$IO_RT11_WRITE, Save R2,R3,R4,R5,R6,R7,-;  1806
                                                                R8,R9,R10,R11
                          5E      E7FC  CE  9E 00002            MOVAB   -6148(SP), SP
                          54        04  AC  D0 00007            MOVL    VOLB, R4                                1854
                                    10  A4  DD 0000B            PUSHL   16(R4)
                          53        14  A4  D0 0000E            MOVL    20(R4), R3                              1855
                          52  041B00F3  8F  D0 00012            MOVL    #68878579, R2                           1859
                          51      01AC  8F  3C 00019            MOVZWL  #428, R1
                          50            54  D0 0001E            MOVL    R4, R0
                              00000000G EF  16 00021            JSB     EXCH$UTIL_BLOCK_CHECK
                    13        48  A4        05  E0 00027        BBS     #5, 72(R4), 1$                          1860
                              7E        91  8F  9A 0002C            MOVZBL  #145, -(SP)
                                            01  DD 00030            PUSHL   #1
                              00000000G     8F  DD 00032            PUSHL   #EXCH$_BADLOGIC
                 00000000G    00            03  FB 00038            CALLS   #3, LIB$STOP
                              59        0C  AC  D0 0003F  1$:       MOVL    BLKCNT, R9                      1865
                                            09  13 00043            BEQL    2$
                 0000007F     8F            59  D1 00045            CMPL    R9, #127
                                            13  1B 0004C            BLEQU   3$
                              7E        70  8F  9A 0004E  2$:       MOVZBL  #112, -(SP)
                                            01  DD 00052            PUSHL   #1
                              00000000G     8F  DD 00054            PUSHL   #EXCH$_BADLOGIC
                 00000000G    00            03  FB 0005A            CALLS   #3, LIB$STOP
              04    AE                    59  09  78 00061  3$:       ASHL    #9, R9, BYTECNT                 1866
                              5B        08  AC  D0 00066            MOVL    BLKPBN, R11                     1871
                                    44  A4  5B  D1 0006A            CMPL    R11, 68(R4)
                                            13  1F 0006E            BLSSU   4$
                              7E        71  8F  9A 00070            MOVZBL  #113, -(SP)
                                            01  DD 00074            PUSHL   #1
                              00000000G     8F  DD 00076            PUSHL   #EXCH$_BADLOGIC
                 00000000G    00            03  FB 0007C            CALLS   #3, LIB$STOP
                              58            5B  D0 00083  4$:       MOVL    R11, LBN                        1872
                          02        48  A4        04  E1 00086        BBC     #4, 72(R4), 5$                  1873
                                            58  D6 0008B            INCL    LBN                             1875
                                            5A  D4 0008D  5$:       CLRL    R10                             1879
                                            58  D5 0008F            TSTL    LBN
                                            11  12 00091            BNEQ    6$
                                            5A  D6 00093            INCL    R10
                                            53  DD 00095            PUSHL   R3                              1886
                 00000000G    00            01  FB 00097            CALLS   #1, SYS$REWIND
                              55            50  D0 0009E            MOVL    R0, STATUS
                              5D            55  E9 000A1            BLBC    STATUS, 8$
                              56        10  AC  D0 000A4  6$:       MOVL    BUFADR, R6                      1897
                                    28  A3  56  D0 000A8            MOVL    R6, 40(R3)
                                    22  A3  04  AE  B0 000AC        MOVW    BYTECNT, 34(R3)                 1898
                                    38  A3  58  D0 000B1            MOVL    LBN, 56(R3)                     1899
                                            53  DD 000B5            PUSHL   R3                              1903
                 00000000G    00            01  FB 000B7            CALLS   #1, SYS$WRITE
                              55            50  D0 000BE            MOVL    R0, STATUS
                              4D            55  E8 000C1            BLBS    STATUS, 9$
                 0001C114     8F            55  D1 000C4            CMPL    STATUS, #114964                 1911
```

```
                                   34  12 000CB          BNEQ    8$                             ;
                  52          0C   A3  D0 000CD          MOVL    12(R3), STV                    ; 1917
                                   56  DD 000D1          PUSHL   R6                             ; 1918
                                   59  DD 000D3          PUSHL   R9                             ;
                            0810   8F  BB 000D5          PUSHR   #^M<R4,R11>                    ;
          0000V   CF               04  FB 000D9          CALLS   #4, EXCH$IO_RT11_WRITE_1       ;
                  57               50  D0 000DE          MOVL    R0, STAT_1                     ;
                  19               57  E9 000E1          BLBC    STAT_1, 7$                     ; 1919
                                   52  DD 000E4          PUSHL   STV                            ; 1922
                                   55  DD 000E6          PUSHL   STATUS                         ;
                              69   A4  9F 000E8          PUSHAB  105(R4)                        ;
                              65   A4  DD 000EB          PUSHL   101(R4)                        ;
                                   02  DD 000EE          PUSHL   #2                             ;
                       00000000G   8F  DD 000F0          PUSHL   #EXCH$_WRITEERRREC             ;
       00000000G  00               06  FB 000F6          CALLS   #6, LIB$SIGNAL                 ;
                  50               57  D0 000FD 7$:      MOVL    STAT_1, R0                     ; 1929
                                       04 00100          RET                                    ;
                              0C   A3  DD 00101 8$:      PUSHL   12(R3)                         ;
                              04   AE  DD 00104          PUSHL   4(SP)                          ;
                                   55  DD 00107          PUSHL   STATUS                         ;
                       00F810D0    8F  DD 00109          PUSHL   #16257232                      ;
                                   28  11 0010F          BRB     11$                            ;
   03         48  A4               02  E0 00111 9$:      BBS     #2, 72(R4), 10$                ; 1934
                                 00C6  31 00116          BRW     17$                            ;
                  1F               5A  E9 00119 10$:     BLBC    R10, 12$                       ; 1942
                                   53  DD 0011C          PUSHL   R3                             ; 1945
       00000000G  00               01  FB 0011E          CALLS   #1, SYS$REWIND                 ;
                  55               50  D0 00125          MOVL    R0, STATUS                     ;
                  10               55  E8 00128          BLBS    STATUS, 12$                    ;
                              0C   A3  DD 0012B          PUSHL   12(R3)                         ; 1949
                              04   AE  DD 0012E          PUSHL   4(SP)                          ;
                                   55  DD 00131          PUSHL   STATUS                         ;
                       00F810B0    8F  DD 00133          PUSHL   #16257200                      ;
                                   69  11 00139 11$:     BRB     14$                            ;
                  57          04   AE  D0 0013B 12$:     MOVL    BYTECNT, LEN                   ; 1953
              24  A3          08   AE  9E 0013F          MOVAB   CHECK_BUF, 36(R3)              ; 1955
              20  A3               57  B0 00144          MOVW    LEN, 32(R3)                    ; 1956
              38  A3               58  D0 00148          MOVL    LBN, 56(R3)                    ; 1957
                                   53  DD 0014C          PUSHL   R3                             ; 1961
       00000000G  00               01  FB 0014E          CALLS   #1, SYS$READ                   ;
                  55               50  D0 00155          MOVL    R0, STATUS                     ;
                  51               55  E8 00158          BLBS    STATUS, 15$                    ;
       0001C0F4   8F               55  D1 0015B          CMPL    STATUS, #114932                ; 1966
                                   32  12 00162          BNEQ    13$                            ;
                  53          0C   A3  D0 00164          MOVL    12(R3), STV                    ; 1972
                                   56  DD 00168          PUSHL   R6                             ; 1973
                                   59  DD 0016A          PUSHL   R9                             ;
                            0810   8F  BB 0016C          PUSHR   #^M<R4,R11>                    ;
          0000V   CF               04  FB 00170          CALLS   #4, EXCH$IO_RT11_WRITE_1       ;
                  52               50  D0 00175          MOVL    R0, STAT_1                     ;
                  60               52  E9 00178          BLBC    STAT_1, 16$                    ; 1974
                                   53  DD 0017B          PUSHL   STV                            ; 1977
                                   55  DD 0017D          PUSHL   STATUS                         ;
                              69   A4  9F 0017F          PUSHAB  105(R4)                        ;
                              65   A4  DD 00182          PUSHL   101(R4)                        ;
                                   02  DD 00185          PUSHL   #2                             ;
                       00000000G   8F  DD 00187          PUSHL   #EXCH$_WRITECHECKREC           ;
```

```
                00000000G  00              06  FB 0018D              CALLS   #6, LIB$SIGNAL                                   ;
                                            45  11 00194              BRB     16$                                             ; 1981
                                       0C   A3  DD 00196 13$:        PUSHL   12(R3)                                          ;
                                       04   AE  DD 00199             PUSHL   4(SP)                                           ;
                                            55  DD 0019C             PUSHL   STATUS                                          ;
                                 00000000G  8F  DD 0019E             PUSHL   #EXCH$_WRITECHECK                               ;
                00000000G  EF              04  FB 001A4 14$:        CALLS   #4, EXCH$UTIL_FILE_ERROR                        ;
                                            04  001AB                RET                                                     ;
                     66    08   AE          57  29 001AC 15$:        CMPC3   LEN, CHECK_BUF, (R6)                            ; 1984
                                            2C  13 001B1             BEQL    17$                                             ;
                                            56  DD 001B3             PUSHL   R6                                              ; 2022
                                            59  DD 001B5             PUSHL   R9                                              ;
                                      0810  8F  BB 001B7             PUSHR   #^M<R4,R11>                                     ;
                         0000V      CF          04  FB 001BB         CALLS   #4, EXCH$IO_RT11_WRITE_1                        ;
                                    52          50  D0 001C0         MOVL    R0, STAT_1                                      ;
                                    15          52  E9 001C3         BLBC    STAT_1, 16$                                     ; 2023
                                          69    A4  9F 001C6         PUSHAB  105(R4)                                         ; 2025
                                          65    A4  DD 001C9         PUSHL   101(R4)                                         ;
                                                02  DD 001CC         PUSHL   #2                                              ;
                                 00000000G  8F  DD 001CE             PUSHL   #EXCH$_WRITECHECKREC                            ;
                00000000G  00              04  FB 001D4             CALLS   #4, LIB$SIGNAL                                  ;
                                    50          52  D0 001DB 16$:        MOVL    STAT_1, R0                                      ; 2026
                                                04  001DE                RET                                                     ;
                                    50          55  D0 001DF 17$:        MOVL    STATUS, R0                                      ; 2031
                                                04  001E2                RET                                                     ; 2032

; Routine Size:  483 bytes,    Routine Base:  EXCH$IO_CODE + 0D57
```

```
;1964        2033  1 GLOBAL ROUTINE exch$io_rt11_write_1 (volb : $ref_bblock,        %SBTTL 'exch$io_rt11_write_1 (volb  pbn, cnt
;1965        2034  1                                      blkpbn, blkcnt, bufadr : $ref_bvector) =
;1966        2035  2 BEGIN
;1967        2036  2 !++
;1968        2037  2 !
;1969        2038  2 ! FUNCTIONAL DESCRIPTION:
;1970        2039  2 !
;1971        2040  2 !       Write a range of "physical" blocks to a random access device.  Block
;1972        2041  2 !       numbers are in the range 0 to MaxBlocks-1.  Blocks are written one at
;1973        2042  2 !       a time to find exact pbns of bad blocks
;1974        2043  2 !
;1975        2044  2 ! INPUTS:
;1976        2045  2 !
;1977        2046  2 !       volb   - pointer to a volb which contains the active record stream
;1978        2047  2 !       blkpbn - starting physical block number on device
;1979        2048  2 !       blkcnt - number of blocks to write
;1980        2049  2 !       bufadr - pointer to buffer containing the data
;1981        2050  2 !
;1982        2051  2 ! IMPLICIT INPUTS:
;1983        2052  2 !
;1984        2053  2 !       none
;1985        2054  2 !
;1986        2055  2 ! OUTPUTS:
;1987        2056  2 !
;1988        2057  2 !       none
;1989        2058  2 !
;1990        2059  2 ! IMPLICIT OUTPUTS:
;1991        2060  2 !
;1992        2061  2 !       none
;1993        2062  2 !
;1994        2063  2 ! ROUTINE VALUE:
;1995        2064  2 !
;1996        2065  2 !       success or error status
;1997        2066  2 !
;1998        2067  2 ! SIDE EFFECTS:
;1999        2068  2 !
;2000        2069  2 !       none
;2001        2070  2 !--
;2002        2071  2
;2003        2072  2 $dbgtrc_prefix ('io_rt11_write_1> ');
;2004        2073  2
;2005        2074  2 LOCAL
;2006        2075  2     check_buf : $bvector [512],
;2007        2076  2     adr,
;2008        2077  2     lbn,
;2009        2078  2     status
;2010        2079  2     ;
;2011        2080  2
;2012        2081  2 BIND
;2013        2082  2     fab = .volb [volb$a_fab] : $bblock, ! New name for File Access Block
;2014        2083  2     rab = .volb [volb$a_rab] : $bblock, ! New name for Record Access Block
;2015        2084  2     nam = .volb [volb$a_nam] : $bblock  ! New name for NAM block
;2016        2085  2     ;
;2017        2086  2
;2018        2087  2 $block_check (2, .volb, volb, 562);
;2019        2088  2 $logic_check (2, (.volb [volb$v_write]), 170);  ! We shouldn't get this far if we aren't supposed to write t
;2020        2089  2
```

```
; 2021          2090  2 $trace_print_fao ('  pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .bufadr, .volb);
```

```
;  2023     2091  2 $logic_check (2, ((.blkcnt GTRU 0) AND (.blkcnt LEQU 127)), 185);
;  2024     2092  2
;  2025     2093  2 ! Adjust the externally useful "physical" block number to the RMS required
;  2026     2094  2 ! logical block number
;  2027     2095  2 !
;  2028     2096  2 $logic_check (2, (.blkpbn LSSU .volb [volb$l_volmaxblock]), 193);
;  2029     2097  2 lbn = .blkpbn;                                  ! Assume volume relative lbn.
;  2030     2098  2 IF .volb [volb$v_virtual]
;  2031     2099  2 THEN
;  2032     2100  2     lbn = .lbn + 1;                             ! File-relative virtual block numbers
;  2033     2101  2
;  2034     2102  2 ! See if the request is for block 0
;  2035     2103  2 !
;  2036     2104  2 IF .lbn EQL 0
;  2037     2105  2 THEN
;  2038     2106  3     BEGIN
;  2039     2107  3
;  2040     2108  3     ! Since BKT = 0 means sequential read, we must trick it with the
;  2041     2109  3     ! documented technique of REWIND, then set BKT=0.
;  2042     2110  3     !
;  2043     2111  4     IF NOT (status = $rewind (rab = rab))
;  2044     2112  3     THEN
;  2045     2113  4         BEGIN
;  2046     2114  4         $debug_print_lit ('rewind error');
;  2047     2115  4         RETURN exch$util_file_error (exch$_writeerr, .status, fab, .rab [rab$l_stv]);
;  2048     2116  3         END;
;  2049     2117  3
;  2050     2118  2     END;
;  2051     2119  2
;  2052     2120  2 ! Prepare for the loop
;  2053     2121  2 !
;  2054     2122  2 rab [rab$w_rsz] = 512;                          ! Record buffer size
;  2055     2123  2 adr = .bufadr;
;  2056     2124  2
;  2057     2125  2 DECR cnt FROM .blkcnt-1 TO 0
;  2058     2126  2 DO
;  2059     2127  3     BEGIN
;  2060     2128  3
;  2061     2129  3     ! Set the RMS record stream buffer and block parameters
;  2062     2130  3     !
;  2063     2131  3     rab [rab$l_rbf] = .adr;                     ! Record buffer address
;  2064     2132  3     rab [rab$l_bkt] = .lbn;
;  2065     2133  3
;  2066     2134  3     ! Write the chunk
;  2067     2135  3     !
;  2068     2136  4     IF NOT (status = $write (rab = rab))
;  2069     2137  3     THEN
;  2070     2138  4         BEGIN
;  2071     2139  4         $trace_print_fao ('error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_rsz] / 512);
;  2072     2140  4         RETURN exch$util_file_error (exch$_writeerr, .status, fab, .rab [rab$l_stv]);
;  2073     2141  3         END;
;  2074     2142  3
;  2075     2143  3     ! If write checking, reread the chunk and compare
;  2076     2144  3     !
;  2077     2145  3     IF .volb [volb$v_write_check]
;  2078     2146  3     THEN
;  2079     2147  4         BEGIN
```

```
;  2080          2148  4          LOCAL
;  2081          2149  4              tmp_desc : VECTOR [2, LONG],
;  2082          2150  4              len;
;  2083          2151  4
;  2084          2152  4          ! See if the request is for block 0, if so rewind
;  2085          2153  4          !
;  2086          2154  4          IF .lbn EQL 0
;  2087          2155  4          THEN
;  2088          2156  5              BEGIN
;  2089          2157  6              IF NOT (status = $rewind (rab = rab))
;  2090          2158  5              THEN
;  2091          2159  6                  BEGIN
;  2092          2160  6                  $debug_print_lit ('rewind error');
;  2093          2161  6                  RETURN exch$util_file_error (exch$_readerr, .status, fab, .rab [rab$l_stv]);
;  2094          2162  5                  END;
;  2095          2163  4              END;
;  2096          2164  4
;  2097          2165  4          len = 512;
;  2098          2166  4
;  2099          2167  4          rab [rab$l_ubf] = check_buf;            ! User buffer address
;  2100          2168  4          rab [rab$w_usz] = .len;
;  2101          2169  4          rab [rab$l_bkt] = .lbn;
;  2102          2170  4
;  2103          2171  4          ! Read the chunk again
;  2104          2172  4          !
;  2105          2173  5          IF NOT (status = $read (rab = rab))
;  2106          2174  4          THEN
;  2107          2175  5              BEGIN
;  2108          2176  5              $trace_print_fao ('reread error at lbn !UL     blk_cnt !UL', .rab [rab$l_bkt], .rab [rab$w_usz]
;  2109          2177  5              RETURN exch$util_file_error (exch$_writecheck, .status, fab, .rab [rab$l_stv]);
;  2110          2178  4              END;
;  2111          2179  4
;  2112          2180  4          IF CH$NEQ (.len, check_buf, .len, .adr, 0)
;  2113          2181  4          THEN
;  2114          2182  5              BEGIN
;  2115        L 2183  5              %IF switch_trace
;  2116        U 2184  5              %THEN
;  2117        U 2185  5                  LITERAL
;  2118        U 2186  5                      chunk = 100;
;  2119        U 2187  5                  LOCAL
;  2120        U 2188  5                      c,          ! count of loops
;  2121        U 2189  5                      l,          ! length of string remaining
;  2122        U 2190  5                      a1,         ! address of remainder of string 1
;  2123        U 2191  5                      a2;         ! address of remainder of string 2
;  2124        U 2192  5                  c = 0;
;  2125        U 2193  5                  l = .len;
;  2126        U 2194  5                  a1 = .adr;
;  2127        U 2195  5                  a2 = check_buf;
;  2128        U 2196  5                  $trace_print_fao ('pbn!5UL,  cnt!3UL,  adr !XL,  volb !XL', .blkpbn, .blkcnt, .adr, .volb);
;  2129        U 2197  5                  WHILE .l GTR 0
;  2130        U 2198  5                  DO
;  2131        U 2199  5                      BEGIN
;  2132        U 2200  5                      LOCAL
;  2133        U 2201  5                          ll;             ! length for this loop
;  2134        U 2202  5                      c = .c + 1;
;  2135        U 2203  5                      ll = MINU (.l, chunk);
;  2136        U 2204  5                      IF CH$NEQ (.ll, .a1, .ll, .a2, 0)
```

```
;  2137          U 2205  5                       THEN
;  2138          U 2206  5                           BEGIN
;  2139          U 2207  5                           $trace_print_fao ('!4UL  "!AF"', .c, .ll, .a1);
;  2140          U 2208  5                           $trace_print_fao ('      "!AF"', .ll, .a2);
;  2141          U 2209  5                           END;
;  2142          U 2210  5                       l = .l - chunk;
;  2143          U 2211  5                       a1 = .a1 + chunk;
;  2144          U 2212  5                       a2 = .a2 + chunk;
;  2145          U 2213  5                       END;
;  2146            2214  5                 %FI
;  2147            2215  5                 tmp_desc [0] = .volb [volb$l_vol_ident_len];
;  2148            2216  5                 tmp_desc [1] = volb [volb$t_vol_ident];
;  2149            2217  5                 $exch_signal (exch$_writecheck, 1, tmp_desc);
;  2150            2218  5                 RETURN 1;
;  2151            2219  4                 END;
;  2152            2220  4
;  2153            2221  3             END;
;  2154            2222  3
;  2155            2223  3         ! Move to the next block
;  2156            2224  3         !
;  2157            2225  3         adr = .adr + 512;
;  2158            2226  3         lbn = .lbn + 1;
;  2159            2227  3
;  2160            2228  2         END;
;  2161            2229  2
;  2162            2230  2 RETURN .status;
;  2163            2231  1 END;


                                                OFFC 00000          .ENTRY  EXCH$IO_RT11_WRITE_1, Save R2,R3,R4,R5,R6,- ; 2033
                                                                            R7,R8,R9,R10,R11
                                5B 00000000G    8F   D0 00002          MOVL    #EXCH$_BADLOGIC, R11
                                5E       FDF8   CE   9E 00009          MOVAB   -520(SP), SP
                                56         04   AC   D0 0000E          MOVL    VOLB, R6                                    ; 2082
                                5A         10   A6   D0 00012          MOVL    16(R6), R10
                                55         14   A6   D0 00016          MOVL    20(R6), R5                                  ; 2083
                                52 041B00F3     8F   D0 0001A          MOVL    #68878579, R2                               ; 2087
                                51       0232   8F   3C 00021          MOVZWL  #562, R1
                                50              56   D0 00026          MOVL    R6, R0
                                   00000000G    EF   16 00029          JSB     EXCH$UTIL_BLOCK_CHECK
                          0F         48   A6        05 E0 0002F          BBS     #5, 72(R6), 1$                              ; 2088
                                7E         AA   8F   9A 00034          MOVZBL  #170, -(SP)
                                                01   DD 00038          PUSHL   #1
                                                5B   DD 0003A          PUSHL   R11
                             00000000G     00        03   FB 0003C     CALLS   #3, LIB$STOP
                                54         0C   AC   D0 00043 1$:     MOVL    BLKCNT, R4                                  ; 2091
                                                09   13 00047          BEQL    2$
                             0000007F      8F        54   D1 00049     CMPL    R4, #127
                                                0F   1B 00050          BLEQU   3$
                                7E         B9   8F   9A 00052 2$:     MOVZBL  #185, -(SP)
                                                01   DD 00056          PUSHL   #1
                                                5B   DD 00058          PUSHL   R11
                             00000000G     00        03   FB 0005A     CALLS   #3, LIB$STOP
                                   44      A6   08   AC   D1 00061 3$:     CMPL    BLKPBN, 68(R6)                          ; 2096
```

```
                                 0F  1F 00066          BLSSU   4$
                          7E      C1 8F  9A 00068          MOVZBL  #193, -(SP)
                                     01  DD 0006C          PUSHL   #1
                                     5B  DD 0006E          PUSHL   R11
                00000000G            00  FB 00070          CALLS   #3, LIB$STOP
                          58      08 AC  D0 00077 4$:      MOVL    BLKPBN, LBN                     ;2097
                02        48    A6     04  E1 0007B          BBC     #4, 72(R6), 5$                  ;2098
                                     58  D6 00080          INCL    LBN                             ;2100
                                     0F  12 00082 5$:      BNEQ    6$                              ;2104
                                     55  DD 00C84          PUSHL   R5                              ;2111
                00000000G            00  FB 00086          CALLS   #1, SYS$REWIND
                          57            50  D0 0008D          MOVL    R0, STATUS
                          24            57  E9 00090          BLBC    STATUS, 8$
                          22    A5  0200 8F  B0 00093 6$:      MOVW    #512, 34(R5)                    ;2122
                          59      10 AC  D0 00099          MOVL    BUFADR, ADR                     ;2123
                                 00AF 31  00 0009D          BRW     14$                             ;2125
                          28    A5     59  D0 000A0 7$:      MOVL    ADR, 40(R5)                     ;2131
                          38    A5     58  D0 000A4          MOVL    LBN, 56(R5)                     ;2132
                                     55  DD 000A8          PUSHL   R5                              ;2136
                00000000G            00  FB 000AA          CALLS   #1, SYS$WRITE
                          57            50  D0 000B1          MOVL    R0, STATUS
                          0F            57  E8 000B4          BLBS    STATUS, 9$
                             0C    A5     DD 000B7 8$:      PUSHL   12(R5)                          ;2140
                           0480    8F     BB 000BA          PUSHR   #^M<R7,R10>
                       00F810D0    8F     DD 000BE          PUSHL   #16257232
                                     55  11 000C4          BRB     11$
                7D        48    A6     02  E1 000C6 9$:      BBC     #2, 72(R6), 13$                 ;2145
                                     58  D5 000CB          TSTL    LBN                             ;2154
                                     1E  12 000CD          BNEQ    10$
                                     55  DD 000CF          PUSHL   R5                              ;2157
                00000000G            00  FB 000D1          CALLS   #1, SYS$REWIND
                          57            50  D0 000D8          MOVL    R0, STATUS
                          0F            57  E8 000DB          BLBS    STATUS, 10$
                             0C    A5     DD 000DE          PUSHL   12(R5)                          ;2161
                           0480    8F     BB 000E1          PUSHR   #^M<R7,R10>
                       00F810B0    8F     DD 000E5          PUSHL   #16257200
                                     2E  11 000EB          BRB     11$
                          52      0200 8F  3C 000ED 10$:     MOVZWL  #512, LEN                       ;2165
                             24    A5      08 AE  9E 000F2          MOVAB   CHECK_BUF, 36(R5)       ;2167
                             20    A5     52  B0 000F7          MOVW    LEN, 32(R5)                     ;2168
                             38    A5     58  D0 000FB          MOVL    LBN, 56(R5)                     ;2169
                                     55  DD 000FF          PUSHL   R5                              ;2173
                00000000G            00  FB 00101          CALLS   #1, SYS$READ
                          57            50  D0 00108          MOVL    R0, STATUS
                          15            57  E8 0010B          BLBS    STATUS, 12$
                             0C    A5     DD 0010E          PUSHL   12(R5)                          ;2177
                           0480    8F     BB 00111          PUSHR   #^M<R7,R10>
                       00000000G    8F     DD 00115          PUSHL   #EXCH$_WRITECHECK
                00000000G            EF  04 FB 00118 11$:     CALLS   #4, EXCH$UTIL_FILE_ERROR
                                     04  00122          RET
                69        08    AE     52  29 00123 12$:     CMPC3   LEN, CHECK_BUF, (ADR)           ;2180
                                     1E  13 00128          BEQL    13$
                          6E      65 A6  D0 0012A          MOVL    101(R6), TMP_DESC               ;2215
                             04    AE      69 A6  9E 0012E          MOVAB   105(R6), TMP_DESC+4     ;2216
                                     5E  DD 00133          PUSHL   SP                              ;2217
                                     01  DD 00135          PUSHL   #1
                       00000000G    8F     DD 00137          PUSHL   #EXCH$_WRITECHECK
```

```
                    00000000G  00            03  FB 0013D          CALLS   #3, LIB$SIGNAL                                ;
                               50            01  D0 00144          MOVL    #1, R0                                       ; 2218
                                                 04 00147          RET                                                  ;
                               59      0200  C9  9E 00148 13$:     MOVAB   512(R9), ADR                                 ; 2225
                                             58  D6 0014D          INCL    LBN                                          ; 2226
                               02            54  F4 0014F 14$:     SOBGEQ  CNT, 15$                                     ; 2125
                                             03  11 00152          BRB     16$                                          ;
                                           FF49  31 00154 15$:     BRW     7$                                           ;
                               50            57  D0 00157 16$:     MOVL    STATUS, R0                                   ; 2230
                                                 04 0015A          RET                                                  ; 2231

; Routine Size:  347 bytes,    Routine Base:  EXCH$IO_CODE + 0F3A
```

```
; 2165          2232  1 END
; 2166          2233  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;   EXCH$IO_CODE                 4245  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   EXCH$IO_PLIT                  144  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         40         0        1000        00:01.9
;   _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1  1151         67         5          79        00:01.4



;                                   COMMAND QUALIFIERS

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

; Size:          4245 code + 144 data bytes
; Run Time:         01:18.0
; Elapsed Time:     03:56.8
; Lines/CPU Min.     1718
; Lexemes/CPU-Min: 25001
; Memory Used:  200 pages
; Compilation Complete
```

EXCFIL11
LIS

EXCINIT
LIS

EXCLIB
LIS

EXCIO
LIS