```
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGGGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN       NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN       NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNNNNN       NNN  GGG
EEEEEEEEEEEE           XXX        CCC              HHHHHHHHHHHHHHH  NNN    NNN   NNN  GGG
EEEEEEEEEEEE           XXX        CCC              HHHHHHHHHHHHHHH  NNN    NNN   NNN  GGG
EEEEEEEEEEEE           XXX        CCC              HHHHHHHHHHHHHHH  NNN    NNN   NNN  GGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN       NNNNNN  GGG   GGGGGGGGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN       NNNNNN  GGG   GGGGGGGGG
EEE                 XXX   XXX     CCC              HHH         HHH  NNN       NNNNNN  GGG   GGGGGGGGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG         GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG         GGG
EEE              XXX         XXX  CCC              HHH         HHH  NNN          NNN  GGG         GGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGG
EEEEEEEEEEEEEEE  XXX         XXX     CCCCCCCCCCC   HHH         HHH  NNN          NNN     GGGGGGGGG
```

```
**FILE**ID**EXCFIL11                                   D  4


EEEEEEEEEE  XX      XX    CCCCCCCC  FFFFFFFFFF    IIIIII    LL              11          11
EEEEEEEEEE  XX      XX    CCCCCCCC  FFFFFFFFFF    IIIIII    LL              11          11
EE          XX      XX  CC          FF              II      LL            1111        1111
EE          XX      XX  CC          FF              II      LL            1111        1111
EE            XX  XX    CC          FF              II      LL              11          11
EE            XX  XX    CC          FF              II      LL              11          11
EEEEEEEE        XX      CC          FFFFFFFF        II      LL              11          11
EEEEEEEE        XX      CC          FFFFFFFF        II      LL              11          11
EE            XX  XX    CC          FF              II      LL              11          11
EE            XX  XX    CC          FF              II      LL              11          11
EE          XX      XX  CC          FF              II      LL              11          11
EE          XX      XX  CC          FF              II      LL              11          11
EEEEEEEEEE  XX      XX    CCCCCCCC  FF            IIIIII    LLLLLLLLLL    111111      111111        ::::
EEEEEEEEEE  XX      XX    CCCCCCCC  FF            IIIIII    LLLLLLLLLL    111111      111111        ::::


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

EXCI
V04-

```
0001 0 MODULE  exch$fil11                                      %TITLE 'Files-11 volume specific routines'
0002 0                  (
0003 0                  IDENT = 'V04-000',
0004 0                  ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
0005 0                  ) =
0006 1 BEGIN
0007 1 !
0008 1 !****************************************************************************
0009 1 !*                                                                          *
0010 1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0011 1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0012 1 !*  ALL RIGHTS RESERVED.                                                    *
0013 1 !*                                                                          *
0014 1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0015 1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0016 1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0017 1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0018 1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0019 1 !*  TRANSFERRED.                                                            *
0020 1 !*                                                                          *
0021 1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0022 1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0023 1 !*  CORPORATION.                                                            *
0024 1 !*                                                                          *
0025 1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0026 1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0027 1 !*                                                                          *
0028 1 !*                                                                          *
0029 1 !****************************************************************************
0030 1
0031 1 !++
0032 1 ! FACILITY:      EXCHANGE - Foreign volume interchange facility
0033 1 !
0034 1 ! ABSTRACT:      Files-11 specific routines
0035 1 !
0036 1 ! ENVIRONMENT:   VAX/VMS User mode
0037 1 !
0038 1 ! AUTHOR:        CW Hobbs                  CREATION DATE: 26-Aug-1982
0039 1 !
0040 1 ! MODIFIED BY:
0041 1 !
0042 1 !       V03-002 CWH3002         CW Hobbs                      12-Apr-1984
0043 1 !               Supply the close routine as the delete routine for files-11
0044 1 !               to prevent leaving files open after random errors.
0045 1 !
0046 1 !--
0047 1
0048 1 ! Include files:
0049 1 !
0050 1 MACRO $module_name_string = 'exch$fil11' %;      ! The require file needs to know our module name
0051 1 REQUIRE 'SRC$:EXCREQ'                            ! Facility-wide require file
0052 1     ;
```

```
1 %SBTTL 'Module table of contents'
1
1 ! Module table of contents:
1 !
1 FORWARD ROUTINE
1     exch$fil11_close_file,                    ! Files-11 specific file close routine
1     exch$fil11_create_file,                   ! Files-11 specific file create routine
1     exch$fil11_get,                           ! Get record
1     exch$fil11_open_file,                     ! Files-11 specific file open routine
1     exch$fil11_put                            ! Put record
1     ;
1
1 ! EXCHANGE facility routines
1 !
1 EXTERNAL ROUTINE
1     exch$cmd_related_file_parse,              ! Perform an RMS output file parse
1     exch$util_file_error,                     ! Signal an RMS error
1     exch$util_rmsb_allocate                   ! Get an RMSB
1     ;
1
1 ! Equated symbols:
1 !
1 !LITERAL
1 !   ;
1
1 ! Bound declarations:
1 !
1 !BIND
1 !   ;
```

EXCH
V04-

; Ro

```
1 GLOBAL ROUTINE exch$fil11_close_file (filb : $ref_bblock) =         %SBTTL 'exch$fil11_close_file (filb)'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform Files-11 volume specific close processing
2 !
2 ! INPUT/OUTPUT:
2 !
2 !       filb - pointer to block describing the file
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       filb - receive info pertaining to the file to be closed
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to close the file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('fil11_close_file> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     namb = filb [filb$a_assoc_namb]      : $ref_bblock,
2     ctx  = filb [filb$a_context]         : $ref_bblock,
2     fab  = ctx [rmsb$a_fab]              : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .filb, filb, 497);                            !?? definitely over-zealous checking
2 $block_check (2, .namb, namb, 498);
2 $block_check (2, .ctx, rmsb, 499);
2
2
2 ! Close the file
2 !
2 $trace_print_fao ('closing, fab=!XL', .fab);
3 IF NOT (status = $close (fab = .fab))
2 THEN
```

```
;  141       0235  2       exch$util_file_error (exch$_closeerr, .status, .fab, .fab [fab$l_stv]);
;  142       0236  2
;  143       0237  2  RETURN .status;
;  144       0238  1  END;


                                                                   .TITLE  EXCH$FIL11 Files-11 volume specific routines
                                                                   .IDENT  \V04-000\

                                                                   .EXTRN  EXCH$CMD_RELATED_FILE_PARSE
                                                                   .EXTRN  EXCH$UTIL_FILE_ERROR
                                                                   .EXTRN  EXCH$UTIL_RMSB_ALLOCATE
                                                                   .EXTRN  EXCH$UTIL_BLOCK_CHECK
                                                                   .EXTRN  SYS$CLOSE, EXCH$_CLOSEERR

                                                                   .PSECT  EXCH$FIL11_CODE,NOWRT,2

                                          007C 00000               .ENTRY  EXCH$FIL11_CLOSE_FILE, Save R2,R3,R4,R5,R6   ; 0178
                        56 00000000G   EF   9E 00002               MOVAB   EXCH$UTIL_BLOCK_CHECK, R6                    ;
          53       04   AC             18   C1 00009               ADDL3   #24, FILB, R3                                ; 0218
          54       04   AC             20   C1 0000E               ADDL3   #32, FILB, R4                                ; 0219
          55            64             10   C1 00013               ADDL3   #16, (R4), R5                                ; 0220
                        52 035B00FA    8F   D0 00017               MOVL    #56295674, R2                                ; 0225
                        51       01F1  8F   3C 0001E               MOVZWL  #497, R1                                     ;
                        50         04  AC   D0 00023               MOVL    FILB, R0                                     ;
                                       66   16 00027               JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                        52 010A00F7    8F   D0 00029               MOVL    #17432823, R2                                ; 0226
                        51       01F2  8F   3C 00030               MOVZWL  #498, R1                                     ;
                        50             63   D0 00035               MOVL    (R3), R0                                     ;
                                       66   16 00038               JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                        52 031600F6    8F   D0 0003A               MOVL    #51773686, R2                                ; 0227
                        51       01F3  8F   3C 00041               MOVZWL  #499, R1                                     ;
                        50             64   D0 00046               MOVL    (R4), R0                                     ;
                                       66   16 00049               JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                        52             65   D0 0004B               MOVL    (R5), R2                                     ; 0233
                                       52   DD 0004E               PUSHL   R2                                           ;
               00000000G   00          01   FB 00050               CALLS   #1, SYS$CLOSE                                ;
                           53          50   D0 00057               MOVL    R0, STATUS                                   ;
                           14          53   E8 0005A               BLBS    STATUS, 1$                                   ;
                                 0C    A2   DD 0005D               PUSHL   12(R2)                                       ; 0235
                                       52   DD 00060               PUSHL   R2                                           ;
                                       53   DD 00062               PUSHL   STATUS                                       ;
                              00000000G 8F  DD 00064               PUSHL   #EXCH$_CLOSEERR                              ;
               00000000G   EF          04   FB 0006A               CALLS   #4, EXCH$UTIL_FILE_ERROR                     ;
                           50          53   D0 00071  1$:          MOVL    STATUS, R0                                   ; 0237
                                            04 00074               RET                                                  ; 0238

; Routine Size:  117 bytes,    Routine Base:  EXCH$FIL11_CODE + 0000
```

```
1 GLOBAL ROUTINE exch$fil11_create_file = %SBTTL 'exch$fil11_create_file'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform Files-11 volume specific create processing
2 !
2 ! INPUT:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       copy [copy$a_out_filb] - pointer to filb for the output file
2 !       copy [copy$a_inp_filb] - pointer to filb for the input file
2 !
2 ! OUTPUTS:
2 !
2 !       out_filb - receive info pertaining to the created file
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to create a file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       none
2 !--
2
2 $dbgtrc_prefix ('fil11_create_file> ');
2
2 LOCAL
2     rfp : $bblock [nam$c_bln+nam$c_maxrss],     ! An RMS NAM block plus the expanded string buffer for outpu
2     status
2     ;
2
2 BIND
2     copy     = exch$a_gbl [excg$a_copy_work]     : $ref_bblock,
2     out_name = copy [copy$q_output_filename]     : $desc_block,
2     inp_filb = copy [copy$a_inp_filb]            : $ref_bblock,
2     out_filb = copy [copy$a_out_filb]            : $ref_bblock,
2     ctx      = out_filb [filb$a_context]         : $ref_bblock,
2     out_namb = out_filb [filb$a_assoc_namb]      : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .out_filb, filb, 511);
2 $block_check (2, .inp_filb, filb, 525);
2 $block_check (2, .out_namb, namb, 512);
2 $logic_check (2, (.out_filb [filb$a_assoc_volb] EQL 0), 138);
2
```

```
;   203          0296  2 ! If the context block is null, then allocate an RMSB
;   204          0297  2 !
;   205          0298  2 IF .ctx EQL 0
;   206          0299  2 THEN
;   207          0300  2     ctx = exch$util_rmsb_allocate ()             ! Get a fresh one
;   208          0301  2 ELSE
;   209          0302  2     $block_check (2, .ctx, rmsb, 513);          ! Check the old one
;   210          0303  2
;   211          0304  3 BEGIN
;   212          0305  3 BIND
;   213          0306  3     fab = ctx [rmsb$a_fab] : $ref_bblock,
;   214          0307  3     rab = ctx [rmsb$a_rab] : $ref_bblock,
;   215          0308  3     nam = ctx [rmsb$a_nam] : $ref_bblock;
;   216          0309  3
;   217          0310  3 ! Create a name string in the out_filb for the "NOTCOPIED" message, just in case we exit with an error
;   218          0311  3 !
;   219          0312  3 out_filb [filb$l_result_name_len] = .out_name [dsc$w_length];
;   220          0313  3 CH$COPY (.out_name [dsc$w_length], .out_name [dsc$a_pointer], 0,
;   221          0314  3                 filb$s_result_name, out_filb [filb$t_result_name]);
;   222          0315  3
;   223          0316  3 ! Perform an RMS output file parse on the related name (the result name for the input file) and the
;   224          0317  3 ! requested output name from the command line.
;   225          0318  3 !
;   226          0319  4 IF NOT (status = exch$cmd_related_file_parse (
;   227          0320  4                 .out_name [dsc$w_length], .out_name [dsc$a_pointer],                  ! Command line out p
;   228          0321  4                 .inp_filb [filb$l_result_name_len], inp_filb [filb$t_result_name],    ! Related name
;   229          0322  4                 rfp))                                                                 ! Gets new name
;   230          0323  3 THEN
;   231          0324  3     $exch_signal_return (exch$_openout, 1, out_name, .status);
;   232          0325  3
;   233          0326  3 $trace_print_fao ('trying to create "!AF"', .rfp [nam$b_esl], .rfp [nam$l_esa]);
;   234          0327  3
;   235          0328  3 ! Initialize the RMS structures
;   236          0329  3 !
;   237        P 0330  3 $fab_init (
;   238        P 0331  3         FAB = .fab,                             ! Output file FAB
;   239        P 0332  3         FAC = (BRO,PUT),                        !  Put only, block I/O in case we can do things faster that
;   240        P 0333  3         FNA = .rfp [nam$l_esa],                 !  Set name addr
;   241        P 0334  3         FNS = .rfp [nam$b_esl],                 !  Set name size
;   242        P 0335  3         FOP = SQO,                              !  Sequential only
;   243        P 0336  3         NAM = .nam,                             !  Name block
;   244        P 0337  3         RAT = CR,                               !  Carriage-return carriage control
;   245        P 0338  3         RFM = VAR,                              !  Variable-length records
;   246          0339  3         SHR = (GET,PUT,UPI));                   !  Allow other readers/writers
;   247        P 0340  3 $rab_init (
;   248        P 0341  3         RAB = .rab,                             ! Output file RAB
;   249        P 0342  3         MBF = 2,                                !  Multi-buffer count (MBC from process or system default)
;   250        P 0343  3         RAC = SEQ,                              !  Sequential only
;   251        P 0344  3         ROP = WBH,                              !  Write behind
;   252          0345  3         FAB = .fab);                            !  FAB addr
;   253        P 0346  3 $nam_init (
;   254        P 0347  3         NAM = .nam,                             ! File name block
;   255        P 0348  3         RSA = .ctx [rmsb$a_rsbuf],              !  Result name addr
;   256        P 0349  3         RSS = nam$c_maxrss,                     !  Result name size
;   257        P 0350  3         ESA = .ctx [rmsb$a_esbuf],              !  Expanded name addr
;   258          0351  3         ESS = nam$c_maxrss);                    !  Expanded name size
;   259          0352  3
```

```
3  ! Set the desired file attributes
3  !
3  fab [fab$v_mxv] = NOT .out_filb [filb$v_explicit_version];         ! Use explicit version if given, otherwise m
3
3  ! We allow several Files-11 "output" qualifiers to be placed on the input parameter.  We interpret "output"
3  ! qualifiers on the output spec (or the verb) as applying to all output files.  "Output" qualifiers on the
3  ! input specs apply to files created for that input spec.  If on both, use the one from the output (or verb)
3  !
4  fab [fab$l_alq] = (IF .copy [copy$l_q_allocation] NEQ 0             ! If specified on the output
4                     THEN                                              !   then
4                         .copy [copy$l_q_allocation]                   !   use that quantity
4                     ELSE IF .inp_filb [filb$l_q_allocation] NEQ 0! otherwise if /ALLOCATION was on the input
4                     THEN                                              !   then
4                         .inp_filb [filb$l_q_allocation]               !   use the /ALLOC quantity from the input f
4                     ELSE                                              ! otherwise
3                         .inp_filb [filb$l_block_count]);              !   use the size of the input file.
3
4  fab [fab$w_deq] = (IF .copy [copy$l_q_extension] NEQ 0
4                     THEN
4                         .copy [copy$l_q_extension]
4                     ELSE
3                         .inp_filb [filb$l_q_extension]);
3
4  fab [fab$v_cbt] = (IF .copy [copy$v_q_best_try_contiguous]          ! Best try - overrides /contiguous if both p
4                     THEN
4                         true
4                     ELSE
3                         .inp_filb [filb$v_q_best_try_contiguous]);
3
4  fab [fab$v_ctg] = (IF .copy [copy$v_q_contiguous]
4                     THEN
4                         true
4                     ELSE
3                         .inp_filb [filb$v_q_contiguous]);
3
4  fab [fab$v_tef] = (IF .copy [copy$v_q_truncate]                     ! Truncate over-allocations
4                     THEN
4                         true
4                     ELSE
3                         .inp_filb [filb$v_q_truncate]);
3                                              !?? should truncate depend on explicit allocation and/or /TR
3
3  ! If /RECORD_FORMAT was given then tell him we are ignoring
3  !
3  IF    .out_filb [filb$v_rfmt_explicit]
3  THEN
4      BEGIN
4      out_filb [filb$v_rfmt_explicit] = false;
4      out_filb [filb$b_rec_format] = filb$k_rfmt_invalid;
4      $exch_signal (exch$_fil11_norec);
3      END;
3
3  ! If /CARRIAGE_CONTROL was given on either input or output then set the record attribute
3  !
3  IF .out_filb [filb$v_cctl_explicit]
3  THEN
4      fab [fab$b_rat] =  (CASE .out_filb [filb$b_car_control] FROM filb$k_cctl_lobound TO filb$k_cctl_hibound
                         SET
                                 [filb$k_cctl_cr] :       fab$m_cr;
                                 [filb$k_cctl_fortran] : fab$m_ftn;
                                 [filb$k_cctl_none] :     0;
                         TES)
ELSE IF .inp_filb [filb$v_cctl_explicit]
THEN
    fab [fab$b_rat] =  (CASE .inp_filb [filb$b_car_control] FROM filb$k_cctl_lobound TO filb$k_cctl_hibound
                         SET
                                 [filb$k_cctl_cr] :       fab$m_cr;
                                 [filb$k_cctl_fortran] : fab$m_ftn;
                                 [filb$k_cctl_none] :     0;
                         TES);

! See if we need to override the record format, variable by default.  We do not allow record format qualifie
! (except for block transfer) on Files-11 filespecs, so get all record format info from the input file.
!
IF .out_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
 OR
    .inp_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block
THEN
    BEGIN
    fab [fab$w_mrs] = 512;
    fab [fab$b_rfm] = fab$c_fix;
    END
ELSE IF .inp_filb [filb$b_rec_format] EQL filb$k_rfmt_fixed
THEN
    BEGIN
    fab [fab$w_mrs] = .inp_filb [filb$l_fixed_len];
    fab [fab$b_rfm] = fab$c_fix;
    END;

! Create and connect to the file
!
IF NOT (status = $create (fab = .fab))
THEN
    BEGIN
    exch$util_file_error (exch$_openout, .status, .fab, .fab [fab$l_stv]);
    RETURN 0;                                      ! Don't pass any status so that we won't get a chained messa
    END;                                           !  attached to the "NOTCOPIED" message

! Create the result name string in the out_filb
!
$logic_check (2, ((.nam [nam$b_rsl] LEQU filb$s_result_name) AND (.nam [nam$b_rsl] GTRU 0)), 139);
out_filb [filb$l_result_name_len] = .nam [nam$b_rsl];
CH$COPY (.nam [nam$b_rsl], .nam [nam$l_rsa], 0, filb$s_result_name, out_filb [filb$t_result_name]);

$trace_print_fao ('Created "!AF"', .out_filb [filb$l_result_name_len], out_filb [filb$t_result_name]);

IF NOT (status = $connect (rab = .rab))
THEN
    BEGIN
    exch$util_file_error (exch$_openout, .status, .fab, .rab [rab$l_stv]);
    $close (fab = .fab);
    RETURN 0;                                      ! Don't pass any status so that we won't get a chained messa
    END;                                           !  attached to the "NOTCOPIED" message

```

```
;   374         0467  3 ! Define a record stream for this file
;   375         0468  3 !
;   376         0469  3 out_filb [filb$a_record] = 0;                   ! No valid record or length
;   377         0470  3 out_filb [filb$l_record_len] = 0;
;   378         0471  3 out_filb [filb$v_files_created] = true;         ! Made a file using this filb
;   379         0472  3
;   380         0473  3 ! Make sure that the record format in the filb is correct
;   381         0474  3 !
;   382         0475  3 !?? record format is in the rms structures
;   383         0476  3
;   384         0477  3 ! Save the addresses of our routines for this volume and record format.
;   385         0478  3 !
;   386         0479  3 out_filb [filb$a_close_routine]  = exch$fil11_close_file;
;   387         0480  3 out_filb [filb$a_delete_routine] = exch$fil11_close_file;
;   388         0481  3 out_filb [filb$a_put_routine]    = exch$fil11_put;
;   389         0482  3 out_filb [filb$a_get_routine]    = 0;           ! We don't want to do this, so make it hard
;   390         0483  3
;   391         0484  2 END;     ! End of BIND to the rmsb components
;   392         0485  2
;   393         0486  2 RETURN true;
;   394         0487  2
;   395         0488  1 END;



                                                        .EXTRN  EXCH$A_GBL, EXCH$_BADLOGIC
                                                        .EXTRN  EXCH$_FIL11_NOREC
                                                        .EXTRN  SYS$CREATE, SYS$CONNECT

                                  0FFC 00000            .ENTRY  EXCH$FIL11_CREATE_FILE, Save R2,R3,R4,R5,-   ; 0239
                                                                R6,R7,R8,R9,R10,R11                          ;
                  5E      FE98  CE  9E 00002            MOVAB   -360(SP), SP                                 ;
        50 00000000G  EF          04  C1 00007            ADDL3   #4, EXCH$A_GBL, R0                           ; 0281
                  58            60  D0 0000F            MOVL    (R0), R8                                     ; 0282
                  57        14  A8  9E 00012            MOVAB   20(R8), R7                                   ;
                  56        44  A8  D0 00016            MOVL    68(R8), R6                                   ; 0285
                  52  035B00FA  8F  D0 0001A            MOVL    #56295674, R2                                ; 0291
                  51      01FF  8F  3C 00021            MOVZWL  #511, R1                                     ;
                  50            56  D0 00026            MOVL    R6, R0                                       ;
                     00000000G  EF  16 00029            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                  5A        3C  A8  D0 0002F            MOVL    60(R8), R10                                  ; 0292
                  52  035B00FA  8F  D0 00033            MOVL    #56295674, R2                                ;
                  51      020D  8F  3C 0003A            MOVZWL  #525, R1                                     ;
                  50            5A  D0 0003F            MOVL    R10, R0                                      ;
                     00000000G  EF  16 00042            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                  52  010A00F7  8F  D0 00048            MOVL    #17432823, R2                                ; 0293
                  51      0200  8F  3C 0004F            MOVZWL  #512, R1                                     ;
                  50        18  A6  D0 00054            MOVL    24(R6), R0                                   ;
                     00000000G  EF  16 00058            JSB     EXCH$UTIL_BLOCK_CHECK                        ;
                            1C  A6  D5 0005E            TSTL    28(R6)                                       ; 0294
                            13  13 00061            BEQL    1$                                           ;
                  7E        8A  8F  9A 00063            MOVZBL  #138, -(SP)                                  ;
                            01  DD 00067            PUSHL   #1                                           ;
                     00000000G  8F  DD 00069            PUSHL   #EXCH$_BADLOGIC                              ;
         00000000G  00          03  FB 0006F            CALLS   #3, LIB$STOP                                 ;
                            20  A6  D5 00076 1$:        TSTL    32(R6)                                       ; 0298
                            0D  12 00079            BNEQ    2$                                           ;
```

```
                        00000000G  EF            00  FB 0007B         CALLS   #0, EXCH$UTIL_RMSB_ALLOCATE                 ; 0300
                               20  A6            50  D0 00082         MOVL    R0, 32(R6)                                  ;
                                                 16  11 00086         BRB     3$                                          ;
                                   52  031600F6  8F  D0 00088 2$:     MOVL    #51773686, R2                               ; 0302
                                   51      0201  8F  3C 0008F         MOVZWL  #513, R1                                    ;
                                   50        20  A6  D0 00094         MOVL    32(R6), R0                                  ;
                                      00000000G  EF  16 00098         JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                                   6E        20  A6  D0 0009E 3$:     MOVL    32(R6), (SP)                                ; 0306
                              3A   A6            67  3C 000A2         MOVZWL  (R7), 58(R6)                                ; 0312
0100   8F            00       04   B7            67  2C 000A6         MOVC5   (R7), @4(R7), #0, #256, 90(R6)              ; 0314
                                             5A  A6     000AE                                                             ;
                                             08  AE  9F 000B0         PUSHAB  RFP                                         ; 0321
                                             5A  AA  9F 000B3         PUSHAB  90(R10)                                     ;
                                             3A  AA  DD 000B6         PUSHL   58(R10)                                     ;
                                             04  A7  DD 000B9         PUSHL   4(R7)                                       ;
                                   7E            67  3C 000BC         MOVZWL  (R7), -(SP)                                 ;
                        00000000G  EF            05  FB 000BF         CALLS   #5, EXCH$CMD_RELATED_FILE_PARSE             ;
                               04  AE            50  D0 000C6         MOVL    R0, STATUS                                  ;
                                   1B        04  AE  E8 000CA         BLBS    STATUS, 4$                                  ;
                                   52  00F810A0  8F  D0 000CE         MOVL    #16257184, TEMP                             ; 0324
                                             04  AE  DD 000D5         PUSHL   STATUS                                      ;
                                                 57  DD 000D8         PUSHL   R7                                          ;
                                                 01  DD 000DA         PUSHL   #1                                          ;
                                                 52  DD 000DC         PUSHL   TEMP                                        ;
                        00000000G  00            04  FB 000DE         CALLS   #4, LIB$SIGNAL                              ;
                                   50            52  D0 000E5         MOVL    TEMP, R0                                    ;
                                                     04 000E8         RET                                                 ;
                     50            6E            10  C1 000E9 4$:     ADDL3   #16, (SP), R0                               ; 0339
                                   57            60  D0 000ED         MOVL    (R0), R7                                    ;
0050   8F            00            6E            00  2C 000F0         MOVC5   #0, (SP), #0, #80, (R7)                     ;
                                                 67     000F7                                                             ;
                                   67      5003  8F  B0 000F8         MOVW    #20483, (R7)                                ;
                              04   A7        40  8F  9A 000FD         MOVZBL  #64, 4(R7)                                  ;
                              16   A7      4341  8F  B0 00102         MOVW    #17217, 22(R7)                              ;
                              1E   A7      0202  8F  B0 00108         MOVW    #514, 30(R7)                                ;
                     50            6E            18  C1 0010E         ADDL3   #24, (SP), R0                               ;
                                   59            60  D0 00112         MOVL    (R0), R9                                    ;
                              28   A7            59  D0 00115         MOVL    R9, 40(R7)                                  ;
                              2C   A7        14  AE  D0 00119         MOVL    RFP+12, 44(R7)                              ;
                              34   A7        13  AE  90 0011E         MOVB    RFP+11, 52(R7)                              ;
                     50            6E            14  C1 00123         ADDL3   #20, (SP), R0                               ; 0345
                                   5B            60  D0 00127         MOVL    (R0), R11                                   ;
0044   8F            00            6E            00  2C 0012A         MOVC5   #0, (SP), #0, #68, (R11)                    ;
                                                 6B     00131                                                             ;
                                   6B      4401  8F  B0 00132         MOVW    #17409, (R11)                               ;
                              04   AB      0400  8F  3C 00137         MOVZWL  #1024, 4(R11)                               ;
                                             1E  AB  94 0013D         CLRB    30(R11)                                     ;
                              36   AB            02  90 00140         MOVB    #2, 54(R11)                                 ;
                              3C   AB            57  D0 00144         MOVL    R7, 60(R11)                                 ;
0060   8F            00            6E            00  2C 00148         MOVC5   #0, (SP), #0, #96, (R9)                     ; 0351
                                                 69     0014F                                                             ;
                                   69      6002  8F  B0 00150         MOVW    #24578, (R9)                                ;
                              02   A9            01  8E 00155         MNEGB   #1, 2(R9)                                   ;
                     50            6E            20  C1 00159         ADDL3   #32, (SP), R0                               ;
                              04   A9            60  D0 0015D         MOVL    (R0), 4(R9)                                 ;
                              0A   A9            01  8E 00161         MNEGB   #1, 10(R9)                                  ;
                     50            6E            1C  C1 00165         ADDL3   #28, (SP), R0                               ;
```

```
EXCH$FIL11          Files-11 volume specific routines               B  5
V04-000             exch$fil11_create_file                         16-Sep-1984 00:56:31    VAX-11 Bliss-32 V4.0-742                Page  11
                                                                   14-Sep-1984 12:29:04    [EXCHNG.SRC]EXCFIL11.B32;1                   (4)

                                   0C    A9           60 D0 00169        MOVL    (R0), 12(R9)                  :
                                         6E     2B    A6 9E 0016D        MOVAB   43(R6), (SP)                  ; 0355
            50        00   BE            01           05 EF 00171        EXTZV   #5, #1, @0(SP), R0            :
                                         50           50 D2 00177        MCOML   R0, R0                        :
      04    A7             01            01           50 F0 0017A        INSV    R0, #1, #1, 4(R7)             :
                                                24    A8 D5 00180        TSTL    36(R8)                        ; 0361
                                                      06 13 00183        BEQL    5$                            :
                                         50     24    A8 D0 00185        MOVL    36(R8), R0                    ; 0363
                                                      0F 11 00189        BRB     7$                            :
                                                2D    AA D5 0018B 5$:    TSTL    45(R10)                       ; 0364
                                                      06 13 0018E        BEQL    6$                            :
                                         50     2D    AA D0 00190        MOVL    45(R10), R0                   ; 0366
                                                      04 11 00194        BRB     7$                            :
                                         50     3E    AA D0 00196 6$:    MOVL    62(R10), R0                   ; 0368
                                   10    A7           50 D0 0019A 7$:    MOVL    R0, 16(R7)                    ; 0361
                                                28    A8 D5 0019E        TSTL    40(R8)                        ; 0370
                                                      06 13 001A1        BEQL    8$                            :
                                         50     28    A8 D0 001A3        MOVL    40(R8), R0                    ; 0372
                                                      04 11 001A7        BRB     9$                            :
                                         50     31    AA D0 001A9 8$:    MOVL    49(R10), R0                   ; 0374
                                   14    A7           50 B0 001AD 9$:    MOVW    R0, 20(R7)                    ; 0370
                                         05     30    A8 E9 001B1        BLBC    48(R8), 10$                   ; 0376
                                         50           01 D0 001B5        MOVL    #1, R0                        :
                                                      06 11 001B8        BRB     11$                           :
            50        2C   AA            01           00 EF 001BA 10$:   EXTZV   #0, #1, 44(R10), R0           ; 0380
      06    A7             01            05           50 F0 001C0 11$:   INSV    R0, #5, #1, 6(R7)             ; 0376
                           05      30    A8           01 E1 001C6        BBC     #1, 48(R8), 12$               ; 0382
                                         50           01 D0 001CB        MOVL    #1, R0                        :
                                                      06 11 001CE        BRB     13$                           :
            50        2C   AA            01           01 EF 001D0 12$:   EXTZV   #1, #1, 44(R10), R0           ; 0386
      06    A7             01            04           50 F0 001D6 13$:   INSV    R0, #4, #1, 6(R7)             ; 0382
                           05      31    A8           02 E1 001DC        BBC     #2, 49(R8), 14$               ; 0388
                                         50           01 D0 001E1        MOVL    #1, R0                        :
                                                      06 11 001E4        BRB     15$                           :
            50        2C   AA            01           02 EF 001E6 14$:   EXTZV   #2, #1, 44(R10), R0           ; 0392
      07    A7             01            04           50 F0 001EC 15$:   INSV    R0, #4, #1, 7(R7)             ; 0388
                                         14     00    BE E9 001F2        BLBC    @0(SP), 16$                   ; 0397
                                   00    BE           01 8A 001F6        BICB2   #1, @0(SP)                    ; 0400
                                                28    A6 94 001FA        CLRB    40(R6)                        ; 0401
                                         00000000G    8F DD 001FD        PUSHL   #EXCH$_FIL11_NOREC            ; 0402
                            00000000G    00           01 FB 00203        CALLS   #1, LIB$SIGNAL                :
                           0D      00    BE           01 E1 0020A 16$:   BBC     #1, @0(SP), 18$               ; 0407
                           02            00     2A    A6 8F 0020F        CASEB   42(R6), #0, #2                ; 0409
                         0022          001D         0018    00214 17$:   .WORD   20$-17$,-                     :
                                                                                 21$-17$,-                     :
                                                                                 22$-17$                       :
                                                      10 11 0021A        BRB     20$                           :
                           1B      2B    AA           01 E1 0021C 18$:   BBC     #1, 43(R10), 24$              ; 0415
                           02            00     2A    AA 8F 00221        CASEB   42(R10), #0, #2               ; 0417
                         0010          000B         0006    00226 19$:   .WORD   20$-19$,-                     :
                                                                                 21$-19$,-                     :
                                                                                 22$-19$                       :
                                         50           02 D0 0022C 20$:   MOVL    #2, R0                        :
                                                      07 11 0022F        BRB     23$                           :
                                         50           01 D0 00231 21$:   MOVL    #1, R0                        :
                                                      02 11 00234        BRB     23$                           :
                                                      50 D4 00236 22$:   CLRL    R0                            :
```

```
                        1E   A7                50  90 00238 23$:   MOVB    R0, 30(R7)                          ;
                             01          29    A6  91 0023C 24$:   CMPB    41(R6), #1                          ; 0427
                                               06  13 00240        BEQL    25$                                 ;
                             01          29    AA  91 00242        CMPB    41(R10), #1                         ; 0429
                                               08  12 00246        BNEQ    26$                                 ;
                        36   A7        0200    8F  B0 00248 25$:   MOVW    #512, 54(R7)                        ; 0432
                                               0B  11 0024E        BRB     27$                                 ; 0433
                             02          28    AA  91 00250 26$:   CMPB    40(R10), #2                         ; 0435
                                               09  12 00254        BNEQ    28$                                 ;
                        36   A7          35    AA  B0 00256        MOVW    53(R10), 54(R7)                     ; 0438
                        1F   A7                01  90 0025B 27$:   MOVB    #1, 31(R7)                          ; 0439
                                               57  DD 0025F 28$:   PUSHL   R7                                  ; 0444
                 00000000G   00                01  FB 00261        CALLS   #1, SYS$CREATE                      ;
                        04   AE                50  D0 00268        MOVL    R0, STATUS                          ;
                             17          04    AE  E8 0026C        BLBS    STATUS, 29$                         ;
                                         0C    A7  DD 00270        PUSHL   12(R7)                              ; 0447
                                               57  DD 00273        PUSHL   R7                                  ;
                                         0C    AE  DD 00275        PUSHL   STATUS                              ;
                                   00F810A0    8F  DD 00278        PUSHL   #16257184                           ;
                 00000000G   EF                04  FB 0027E        CALLS   #4, EXCH$UTIL_FILE_ERROR            ;
                                               78  11 00285        BRB     32$                                 ; 0448
                             52          03    A9  9A 00287 29$:   MOVZBL  3(R9), R2                           ; 0453
                                               13  12 0028B        BNEQ    30$                                 ;
                             7E          8B    8F  9A 0028D        MOVZBL  #139, -(SP)                         ;
                                               01  DD 00291        PUSHL   #1                                  ;
                                  00000000G    8F  DD 00293        PUSHL   #EXCH$_BADLOGIC                     ;
                 00000000G   00                03  FB 00299        CALLS   #3, LIB$STOP                        ;
                        3A   A6                52  D0 002A0 30$:   MOVL    R2, 58(R6)                          ; 0454
   0100   8F         00   04   B9              52  2C 002A4        MOVC5   R2, @4(R9), #0, #256, 90(R6)        ; 0455
                                         5A    A6     002AC                                                    ;
                                               5B  DD 002AE        PUSHL   R11                                 ; 0459
                 00000000G   00                01  FB 002B0        CALLS   #1, SYS$CONNECT                     ;
                        04   AE                50  D0 002B7        MOVL    R0, STATUS                          ;
                             20          04    AE  E8 002BB        BLBS    STATUS, 31$                         ;
                                         0C    AB  DD 002BF        PUSHL   12(R11)                             ; 0462
                                               57  DD 002C2        PUSHL   R7                                  ;
                                         0C    AE  DD 002C4        PUSHL   STATUS                              ;
                                   00F810A0    8F  DD 002C7        PUSHL   #16257184                           ;
                 00000000G   EF                04  FB 002CD        CALLS   #4, EXCH$UTIL_FILE_ERROR            ;
                                               57  DD 002D4        PUSHL   R7                                  ; 0463
                 00000000G   00                01  FB 002D6        CALLS   #1, SYS$CLOSE                       ;
                                               20  11 002DD        BRB     32$                                 ; 0464
                                         42    A6  7C 002DF 31$:   CLRQ    66(R6)                              ; 0470
                        00   BE                10  88 002E2        BISB2   #16, @0(SP)                         ; 0471
                        4A   A6        FCA1    CF  9E 002E6        MOVAB   EXCH$FIL11_CLOSE_FILE, 74(R6)       ; 0479
                        4E   A6        FC9B    CF  9E 002EC        MOVAB   EXCH$FIL11_CLOSE_FILE, 78(R6)       ; 0480
                        56   A6       0000V    CF  9E 002F2        MOVAB   EXCH$FIL11_PUT, 86(R6)              ; 0481
                                         52    A6  D4 002F8        CLRL    82(R6)                              ; 0482
                             50                01  D0 002FB        MOVL    #1, R0                              ; 0486
                                                   04 002FE        RET                                         ;
                                               50  D4 002FF 32$:   CLRL    R0                                  ; 0488
                                                   04 00301        RET                                         ;

; Routine Size:  770 bytes,    Routine Base:  EXCH$FIL11_CODE + 0075
```

```
1 GLOBAL ROUTINE exch$fil11_get (filb : $ref_bblock) =     %SBTTL 'exch$fil11_get (filb)'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Return a pointer to the next fixed-length record in the file
2 !
2 ! INPUTS:
2 !
2 !       filb - pointer to filb for an open Files-11 file
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       none
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if success, false if any error
2 !
2 ! SIDE EFFECTS:
2 !
2 !       error conditions will be signaled
2 !--
2
2 $dbgtrc_prefix ('fil11_get> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     namb = filb [filb$a_assoc_namb]     : $ref_bblock,
2     ctx  = filb [filb$a_context]        : $ref_bblock,
2     fab  = ctx [rmsb$a_fab]             : $ref_bblock,
2     rab  = ctx [rmsb$a_rab]             : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .filb, filb, 500);              !?? definitely over-zealous checking
2 $block_check (2, .namb, namb, 508);
2 $block_check (2, .ctx, rmsb, 501);
2
2 ! Set the user buffer fields in the rab
2 !
2 rab [rab$l_ubf] = filb [filb$t_record_buffer];  ! buffer address
2
2 ! Read a single record from SYS$INPUT
```

```
2 !
3 status = (IF .rab [rab$v_bio]                         ! If we are doing block I/O to the file
3                THEN
4                    BEGIN
4                    rab [rab$w_usz] = 512;              ! Buffer size
5                    $read (rab = .rab)                  ! Physical uses block i/o
4                    END
3                ELSE
4                    BEGIN
4                    rab [rab$w_usz] = filb$s_record_buffer;      ! buffer size
5                    $get (rab = .rab)                   ! Everything else is record i/o
2                    END);
2
2 ! Since we are using locate mode, RMS can return a record which is larger than our buffer.  We check the
2 ! returned record length and simulate an RMS$_RTB error if we see such an animal.
2 !
2 IF .rab [rab$w_rsz] GTRU filb$s_record_buffer
2 THEN
3    BEGIN
3    status = rms$_rtb;                                  ! Status is record too big
3    rab [rab$l_stv] = .rab [rab$w_rsz];                 ! STV contains the record size for the signal
2    END;
2
2 ! Signal any rms (or simulated rms) errors
2 !
2 IF NOT .status
2 THEN
3    BEGIN
3
3    filb [filb$a_record]     = 0;                       ! Invalidate record descriptor
3    filb [filb$l_record_len] = 0;
3
3    ! If the error is anything but end of file then signal
3    !
3    IF .status NEQ rms$_eof
3    THEN
4       BEGIN
4       exch$util_file_error (exch$_readerr, .status, .fab, .rab [rab$l_stv]);
4       RETURN .status;                                  ! Return the RMS error
4       END
4
4    ! Normal exit, return 0
4    !
3    ELSE
3       RETURN false;
2    END;
2
2 ! Return the address and length of the record
2 !
2 filb [filb$a_record]     = .rab [rab$l_rbf];
2 filb [filb$l_record_len] = .rab [rab$w_rsz];
2
2 RETURN true;
2
1 END;
```

**F

```
                                                     .EXTRN  SYS$READ, SYS$GET

                                   007C 00000        .ENTRY  EXCH$FIL11_GET, Save R2,R3,R4,R5,R6     ; 0489
                 56  00000000G  EF   9E 00002        MOVAB   EXCH$UTIL_BLOCK_CHECK, R6
                 53          04  AC   D0 00009        MOVL    FILB, R3                                ; 0529
  55         20  A3              10   C1 0000D        ADDL3   #16, 32(R3), R5                         ; 0531
  54         20  A3              14   C1 00012        ADDL3   #20, 32(R3), R4                         ; 0532
                 52   035B00FA  8F   D0 00017        MOVL    #56295674, R2                           ; 0537
                 51       01F4  8F   3C 0001E        MOVZWL  #500, R1
                 50              53   D0 00023        MOVL    R3, R0
                                66   16 00026        JSB     EXCH$UTIL_BLOCK_CHECK
                 52   010A00F7  8F   D0 00028        MOVL    #17432823, R2                           ; 0538
                 51       01FC  8F   3C 0002F        MOVZWL  #508, R1
                 50         18  A3   D0 00034        MOVL    24(R3), R0
                                66   16 00038        JSB     EXCH$UTIL_BLOCK_CHECK
                 52   031600F6  8F   D0 0003A        MOVL    #51773686, R2                           ; 0539
                 51       01F5  8F   3C 00041        MOVZWL  #501, R1
                 50         20  A3   D0 00046        MOVL    32(R3), R0
                                66   16 0004A        JSB     EXCH$UTIL_BLOCK_CHECK
                 52              64   D0 0004C        MOVL    (R4), R2                                ; 0543
             24  A2       015A  C3   9E 0004F        MOVAB   346(R3), 36(R2)
  11         05  A2              03   E1 00055        BBC     #3, 5(R2), 1$                           ; 0547
             20  A2       0200  8F   B0 0005A        MOVW    #512, 32(R2)                            ; 0550
                                52   DD 00060        PUSHL   R2                                      ; 0551
      00000000G  00              01   FB 00062        CALLS   #1, SYS$READ
                                0F   11 00069        BRB     2$
             20  A2       0200  8F   B0 0006B 1$:    MOVW    #512, 32(R2)                            ; 0555
                                52   DD 00071        PUSHL   R2                                      ; 0556
      00000000G  00              01   FB 00073        CALLS   #1, SYS$GET
                 54              50   D0 0007A 2$:    MOVL    R0, STATUS
           0200  8F         22  A2   B1 0007D        CMPW    34(R2), #512                            ; 0562
                                0C   1B 00083        BLEQU   3$
                 54   000181A8  8F   D0 00085        MOVL    #98728, STATUS                          ; 0565
             0C  A2         22  A2   3C 0008C        MOVZWL  34(R2), 12(R2)                          ; 0566
                 24              54   E8 00091 3$:    BLBS    STATUS, 4$                              ; 0571
                            42  A3   7C 00094        CLRQ    66(R3)                                  ; 0576
       0001827A  8F              54   D1 00097        CMPL    STATUS, #98938                          ; 0580
                                26   13 0009E        BEQL    5$
                            0C  A2   DD 000A0        PUSHL   12(R2)                                  ; 0583
                                65   DD 000A3        PUSHL   (R5)
                                54   DD 000A5        PUSHL   STATUS
                      00F810B0  8F   DD 000A7        PUSHL   #16257200
      00000000G  EF              04   FB 000AD        CALLS   #4, EXCH$UTIL_FILE_ERROR
                 50              54   D0 000B4        MOVL    STATUS, R0                              ; 0590
                                     04 000B7        RET
             46  A3         28  A2   D0 000B8 4$:    MOVL    40(R2), 70(R3)                          ; 0595
             42  A3         22  A2   3C 000BD        MOVZWL  34(R2), 66(R3)                          ; 0596
                 50              01   D0 000C2        MOVL    #1, R0                                  ; 0598
                                     04 000C5        RET
                                50   D4 000C6 5$:    CLRL    R0                                      ; 0600
                                     04 000C8        RET

; Routine Size:  201 bytes,    Routine Base:  EXCH$FIL11_CODE + 0377
```

```
1 GLOBAL ROUTINE exch$fil11_open_file =   %SBTTL 'exch$fil11_open_file'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Perform Files-11 volume specific open processing
2 !
2 ! INPUT/OUTPUT:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       copy verb work area
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       none
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if able to open a file, false otherwise
2 !
2 ! SIDE EFFECTS:
2 !
2 !       file is opened, copy work area modified
2 !--
2
2 $dbgtrc_prefix ('fil11_open_file> ');
2
2 LOCAL
2     xab : $bblock [xab$c_fhclen],              ! File header char xab so that we can read the size of the f
2     status
2     ;
2
2 BIND
2     copy     = exch$a_gbl [excg$a_copy_work]    : $ref_bblock,
2     inp_filb = copy [copy$a_inp_filb]           : $ref_bblock,
2     out_filb = copy [copy$a_out_filb]           : $ref_bblock,
2     ctx  = inp_filb [filb$a_context]     : $ref_bblock,
2     namb = inp_filb [filb$a_assoc_namb] : $ref_bblock
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check_if_nonzero (2, .out_filb, filb, 577);
2 $block_check (2, .inp_filb, filb, 502);
2 $block_check (2, .namb, namb, 503);
2 $logic_check (2, (.inp_filb [filb$a_assoc_volb] EQL 0), 136);
2
2 ! If the context block is null, then allocate an RMSB
2 !
```

```
  2 IF .ctx EQL 0
  2 THEN
  2     ctx = exch$util_rmsb_allocate ()            ! Get a fresh one
  2 ELSE
  2     $block_check (2, .ctx, rmsb, 504);          ! Check the old one
  2
  2 ! Use the RTL routine to find the next file matched by the input name, unless we are reopening in which case
  2 ! is ready
  2 !
  2 IF NOT .copy [copy$v_reopen_input]
  2 THEN
  3     BEGIN
P 3     $trace_print_fao ('before find_file:  fullname !AS, inpname !AS, wcc !XL',
  3             namb [namb$q_fullname], inp_filb [filb$q_name_string], .inp_filb [filb$a_fil11_wcc]);
  3     status = lib$find_file (namb [namb$q_fullname], inp_filb [filb$q_name_string], inp_filb [filb$a_fil11_wc
P 3     $trace_print_fao ('find_file status !XL, fullname !AS, inpname !AS, wcc !XL',
  3             .status, namb [namb$q_fullname], inp_filb [filb$q_name_string], .inp_filb [filb$a_fil11_wcc]
  3     IF NOT .status
  3     THEN
  4         BEGIN
  4
  4         IF NOT .inp_filb [filb$v_files_found]           ! If no files were found, then scream and shout
  4         THEN
  4             $exch_signal (exch$_filenotfound, 1, namb [namb$q_fullname], .status);
  4
  4         IF  .status EQL rms$_nmf                        ! rms$_nmf means that we are done with this filespec
  4           OR
  6            (BEGIN                                       ! Or if the error is severe
  6             BIND
  6                 sb = status : $bblock;
  6             .sb [sts$v_severity] EQL sts$k_severe
  5             END)
  4         THEN
  4             status = 0;                             ! 0 status terminates the outer loop
  4
  4         RETURN .status;
  3         END;
  3     END;
  2
  3 BEGIN
  3 BIND
  3     fab = ctx [rmsb$a_fab] : $ref_bblock,
  3     rab = ctx [rmsb$a_rab] : $ref_bblock,
  3     nam = ctx [rmsb$a_nam] : $ref_bblock,
  3     res = inp_filb [filb$q_name_string] : $desc_block;
  3
  3 ! Initialize the RMS structures
  3 !
P 3 $fab_init (
P 3         FAB = .fab,                                 ! Input file FAB
P 3         FAC = (BRO,GET),                            !  Get only, block I/O in case we can do things faster that
P 3         FNA = .res [dsc$a_pointer],                 !  Set name addr
P 3         FNS = .res [dsc$w_length],                  !  Set name size
P 3         FOP = $QO,                                  !  Sequential only
P 3         NAM = .nam,                                 !  Name block
P 3         SHR = (GET,PUT,UPI),                        !  Allow other readers/writers
  3         XAB = xab);                                 !  A file header char xab so that we can read the file size
```

```
P  3 $rab_init (
P  3           RAB = .rab,                                 ! Input file RAB
P  3           MBF = 2,                                    !  Multi-buffer count (MBC from default)
P  3           RAC = SEQ,                                  !  Sequential only
P  3           ROP = (LOC,RAH),                            !  Locate mode, read ahead
   3           FAB = .fab);                                !  FAB addr
P  3 $nam_init (
P  3           NAM = .nam,                                 ! File name block
P  3           RSA = .ctx [rmsb$a_rsbuf],                  !  Result name addr
P  3           RSS = nam$c_maxrss,                         !  Result name size
P  3           ESA = .ctx [rmsb$a_esbuf],                  !  Expanded name addr
   3           ESS = nam$c_maxrss);                        !  Expanded name size
P  3 $xabfhc_init (                                        ! File header char xab so that we can read the file size
   3           XAB = xab);                                 !  RMS will fill it in when we open
   3
   3 ! If this is a block transfer mode read, set the block i/o bit
   3 !
   5 rab [rab$v_bio] = ((.inp_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block)
   4                   OR
   5                      (IF .out_filb EQL 0
   5                       THEN
   5                          0
   5                       ELSE
   3                          .out_filb [filb$b_transfer_mode] EQL filb$k_xfrm_block));
   3
   3 ! Open and connect to the file
   3 !
   3 $trace_print_fao ('opening, fab=!XL', .fab);
   4 IF NOT (status = $open (fab = .fab))
   3 THEN
   4     BEGIN
   4     exch$util_file_error (exch$_openin, .status, .fab, .fab [fab$l_stv]);
   4     RETURN .status;
   3     END;
   4 IF NOT (status = $connect (rab = .rab))
   3 THEN
   4     BEGIN
   4     exch$util_file_error (exch$_openin, .status, .fab, .rab [rab$l_stv]);
   4     $close (fab = .fab);
   4     RETURN .status;
   3     END;
   3
   3
   3 ! Create the result name string in the filb
   3 !
   3 $logic_check (2, ((.nam [nam$b_rsl] LEQU filb$s_result_name) AND (.nam [nam$b_rsl] GTRU 0)), 137);
   3 inp_filb [filb$l_result_name_len] = .nam [nam$b_rsl];
   3 CH$COPY (.nam [nam$b_rsl], .nam [nam$l_rsa], 0, filb$s_result_name, inp_filb [filb$t_result_name]);
   3
   3 $trace_print_fao ('Found "!AF"', .inp_filb [filb$l_result_name_len], inp_filb [filb$t_result_name]);
   3
   3 ! Define a record stream for this file
   3 !
   3 inp_filb [filb$a_record] = 0;                         ! No valid record or length
   3 inp_filb [filb$l_record_len] = 0;
   3 inp_filb [filb$v_files_found] = true;                 ! Found a file using this filb
   3 inp_filb [filb$l_block_count] = .xab [xab$l_ebk] -        ! Put the file size in the filb where any routine ca
```

```
;  681          0772  4                       (IF .xab [xab$w_ffb] NEQ 0  ! (Eof block is one too high if the first free byte is zero)
;  682          0773  3                        THEN 0 ELSE 1);
;  683          0774  3  fab [fab$l_xab] = 0;                        ! Remove the xab from the fab, won't be valid after return
;  684          0775  3
;  685          0776  3  ! Save the addresses of our routines for this volume and record format.
;  686          0777  3  !
;  687          0778  3  inp_filb [filb$a_close_routine] = exch$fil11_close_file;
;  688          0779  3  inp_filb [filb$a_put_routine]   = 0;          ! Make it very hard to do a PUT
;  689          0780  3  inp_filb [filb$a_get_routine]   = exch$fil11_get;
;  690          0781  3
;  691          0782  2  END;     ! End of BIND to the rmsb components
;  692          0783  2
;  693          0784  2  RETURN true;
;  694          0785  2
;  695          0786  1  END;


                                                        .EXTRN  LIB$FIND_FILE, EXCH$_FILENOTFOUND
                                                        .EXTRN  SYS$OPEN

                                       0FFC 00000       .ENTRY  EXCH$FIL11_OPEN_FILE, Save R2,R3,R4,R5,R6,-   ; 0601
                                                                R7,R8,R9,R10,R11                              ;
                          5E       2C  C2 00002         SUBL2   #44, SP                                       ;
          50 00000000G    EF       04  C1 00005         ADDL3   #4, EXCH$A_GBL, R0                            ; 0642
                          53       60  D0 0000D         MOVL    (R0), R3                                      ; 0643
                          58    3C A3  D0 00010         MOVL    60(R3), R8                                    ; 0645
                                44 A3  DD 00014         PUSHL   68(R3)                                        ; 0651
                                   15  13 00017         BEQL    1$                                            ;
                    52 035B00FA    8F  D0 00019         MOVL    #56295674, R2                                 ;
                    51     0241    8F  3C 00020         MOVZWL  #577, R1                                      ;
                    50             6E  D0 00025         MOVL    (SP), R0                                      ;
                       00000000G   EF  16 00028         JSB     EXCH$UTIL_BLOCK_CHECK                         ;
                    52 035B00FA    8F  D0 0002E  1$:    MOVL    #56295674, R2                                 ; 0652
                    51     01F6    8F  3C 00035         MOVZWL  #502, R1                                      ;
                    50             58  D0 0003A         MOVL    R8, R0                                        ;
                       00000000G   EF  16 0003D         JSB     EXCH$UTIL_BLOCK_CHECK                         ;
                    52 010A00F7    8F  D0 00043         MOVL    #17432823, R2                                 ; 0653
                    51     01F7    8F  3C 0004A         MOVZWL  #503, R1                                      ;
                    50       18    A8  D0 0004F         MOVL    24(R8), R0                                    ;
                       00000000G   EF  16 00053         JSB     EXCH$UTIL_BLOCK_CHECK                         ;
                             1C    A8  D5 00059         TSTL    28(R8)                                        ; 0654
                                   13  13 0005C         BEQL    2$                                            ;
                    7E       88    8F  9A 0005E         MOVZBL  #136, -(SP)                                   ;
                                   01  DD 00062         PUSHL   #1                                            ;
                       00000000G   8F  DD 00064         PUSHL   #EXCH$_BADLOGIC                               ;
            00000000G  00          03  FB 0006A         CALLS   #3, LIB$STOP                                  ;
                             20    A8  D5 00071  2$:    TSTL    32(R8)                                        ; 0658
                                   0D  12 00074         BNEQ    3$                                            ;
            00000000G  EF          00  FB 00076         CALLS   #0, EXCH$UTIL_RMSB_ALLOCATE                   ; 0660
                   20  A8          50  D0 0007D         MOVL    R0, 32(R8)                                    ;
                                   16  11 00081         BRB     4$                                            ;
                    52 031600F6    8F  D0 00083  3$:    MOVL    #51773686, R2                                 ; 0662
                    51     01F8    8F  3C 0008A         MOVZWL  #504, R1                                      ;
                    50       20    A8  D0 0008F         MOVL    32(R8), R0                                    ;
                       00000000G   EF  16 00093         JSB     EXCH$UTIL_BLOCK_CHECK                         ;
          47     34    A3          02  E0 00099  4$:    BBS     #2, 52(R3), 8$                                ; 0667
```

```
                                                       24   A8  9F  0009E        PUSHAB  36(R8)                           ; 0672
                                                       10   A8  9F  000A1        PUSHAB  16(R8)                           ;
                                 52        18  A8           18  C1  000A4        ADDL3   #24, 24(R8), R2                  ;
                                                            52  DD  000A9        PUSHL   R2                               ;
                                    00000000G  00           03  FB  000AB        CALLS   #3, LIB$FIND_FILE                ;
                                               5B           50  D0  000B2        MOVL    R0, STATUS                       ;
                                               2D           5B  E8  000B5        BLBS    STATUS, 8$                       ; 0675
                                 13        2B  A8           03  E0  000B8        BBS     #3, 43(R8), 5$                   ; 0679
                                                     0804   8F  BB  000BD        PUSHR   #^M<R2,R11>                      ; 0681
                                                            01  DD  000C1        PUSHL   #1                               ;
                                                00000000G   8F  DD  000C3        PUSHL   #EXCH$_FILENOTFOUND              ;
                                    00000000G  00           04  FB  000C9        CALLS   #4, LIB$SIGNAL                   ;
                                     000182CA  8F           5B  D1  000D0  5$:   CMPL    STATUS, #99018                   ; 0683
                                                            07  13  000D7        BEQL    6$                               ;
                 04              5B            03           00  ED  000D9        CMPZV   #0, #3, SB, #4                   ; 0688
                                                            02  12  000DE        BNEQ    7$                               ;
                                                            5B  D4  000E0  6$:   CLRL    STATUS                           ; 0691
                                                          0108  31  000E2  7$:   BRW     13$                              ; 0693
                                           5A          20   A8  D0  000E5  8$:   MOVL    32(R8), R10                      ; 0699
                                           56          10   A8  9E  000E9        MOVAB   16(R8), R6                       ; 0702
                                           57          10   AA  D0  000ED        MOVL    16(R10), R7                      ; 0714
       0050      8F              00                    6E   00  2C  000F1        MOVC5   #0, (SP), #0, #80, (R7)          ;
                                                            67      000F8                                                 ;
                                               67    5003   8F  B0  000F9        MOVW    #20483, (R7)                     ;
                                           04  A7      40   8F  9A  000FE        MOVZBL  #64, 4(R7)                       ;
                                           16  A7    4342   8F  B0  00103        MOVW    #17218, 22(R7)                   ;
                                           1F  A7           02  90  00109        MOVB    #2, 31(R7)                       ;
                                           24  A7      04   AE  9E  0010D        MOVAB   XAB, 36(R7)                      ;
                                               59      18   AA  D0  00112        MOVL    24(R10), R9                      ;
                                           28  A7           59  D0  00116        MOVL    R9, 40(R7)                       ;
                                           2C  A7      04   A6  D0  0011A        MOVL    4(R6), 44(R7)                    ;
                                           34  A7           66  90  0011F        MOVB    (R6), 52(R7)                     ;
                                               56      14   AA  D0  00123        MOVL    20(R10), R6                      ; 0720
       0044      8F              00                    6E   00  2C  00127        MOVC5   #0, (SP), #0, #68, (R6)          ;
                                                            66      0012E                                                 ;
                                               66    4401   8F  B0  0012F        MOVW    #17409, (R6)                     ;
                                           04  A6 00010200  8F  D0  00134        MOVL    #66048, 4(R6)                    ;
                                                       1E   A6  94  0013C        CLRB    30(R6)                           ;
                                           36  A6           02  90  0013F        MOVB    #2, 54(R6)                       ;
                                           3C  A6           57  D0  00143        MOVL    R7, 60(R6)                       ;
       0060      8F              00                    6E   00  2C  00147        MOVC5   #0, (SP), #0, #96, (R9)          ; 0726
                                                            69      0014E                                                 ;
                                               69    6002   8F  B0  0014F        MOVW    #24578, (R9)                     ;
                                           02  A9           01  8E  00154        MNEGB   #1, 2(R9)                        ;
                                           04  A9      20   AA  D0  00158        MOVL    32(R10), 4(R9)                   ;
                                           0A  A9           01  8E  0015D        MNEGB   #1, 10(R9)                       ;
                                           0C  A9      1C   AA  D0  00161        MOVL    28(R10), 12(R9)                  ;
                 2C              00                    6E   00  2C  00166        MOVC5   #0, (SP), #0, #44, $RMS_PTR      ; 0728
                                                       04   AE      0016B                                                 ;
                                           04  AE    2C1D   8F  B0  0016D        MOVW    #11293, $RMS_PTR                 ;
                                                            51  D4  00173        CLRL    R1                               ; 0732
                                               01      29   A8  91  00175        CMPB    41(R8), #1                       ;
                                                            02  12  00179        BNEQ    9$                               ;
                                                            51  D6  0017B        INCL    R1                               ;
                                                            6E  D5  0017D  9$:   TSTL    (SP)                             ; 0734
                                                            04  12  0017F        BNEQ    10$                              ;
                                                            50  D4  00181        CLRL    R0                               ;
```

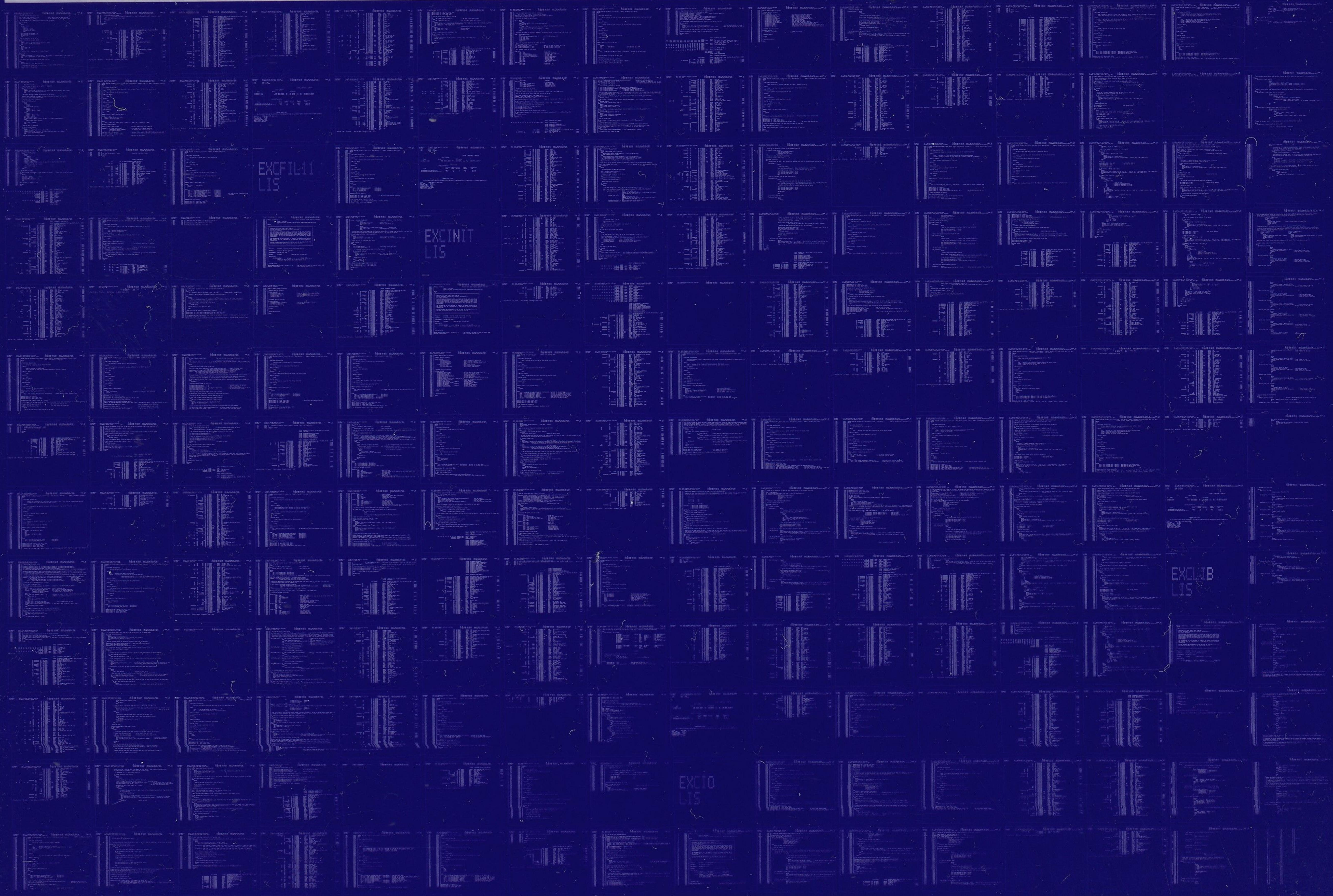
```
                                         0D  11 00183        BRB     11$                                  ;
                                         50  D4 00185 10$:   CLRL    R0                                   ; 0738
                  52           6E        29  C1 00187        ADDL3   #41, (SP), R2                        ;
                               01        62  91 0018B        CMPB    (R2), #1                             ;
                                         02  12 0018E        BNEQ    11$                                  ;
                                         50  D6 00190        INCL    R0                                   ;
                  53           50        51  89 00192 11$:   BISB3   R1, R0, R3                           ; 0734
    05   A6       01           03        53  F0 00196        INSV    R3, #3, #1, 5(R6)                    ;
                                         57  DD 0019C        PUSHL   R7                                   ; 0743
                   00000000G   00        01  FB 0019E        CALLS   #1, SYS$OPEN                         ;
                               5B        50  D0 001A5        MOVL    R0, STATUS                           ;
                               16        5B  E8 001A8        BLBS    STATUS, 12$                          ;
                                     0C  A7  DD 001AB        PUSHL   12(R7)                               ; 0746
                                         57  DD 001AE        PUSHL   R7                                   ;
                                         5B  DD 001B0        PUSHL   STATUS                               ;
                               00F81098  8F  DD 001B2        PUSHL   #16257176                            ;
                   00000000G   EF        04  FB 001B8        CALLS   #4, EXCH$UTIL_FILE_ERROR             ;
                                         2C  11 001BF        BRB     13$                                  ; 0747
                                         56  DD 001C1 12$:   PUSHL   R6                                   ; 0749
                   00000000G   00        01  FB 001C3        CALLS   #1, SYS$CONNECT                      ;
                               5B        50  D0 001CA        MOVL    R0, STATUS                           ;
                               21        5B  E8 001CD        BLBS    STATUS, 14$                          ;
                                     0C  A6  DD 001D0        PUSHL   12(R6)                               ; 0752
                                         57  DD 001D3        PUSHL   R7                                   ;
                                         5B  DD 001D5        PUSHL   STATUS                               ;
                               00F81098  8F  DD 001D7        PUSHL   #16257176                            ;
                   00000000G   EF        04  FB 001DD        CALLS   #4, EXCH$UTIL_FILE_ERROR             ;
                                         57  DD 001E4        PUSHL   R7                                   ; 0753
                   00000000G   00        01  FB 001E6        CALLS   #1, SYS$CLOSE                        ;
                               50        5B  D0 001ED 13$:   MOVL    STATUS, R0                           ; 0754
                                             04 001F0        RET                                          ;
                               52    03  A9  9A 001F1 14$:   MOVZBL  3(R9), R2                            ; 0760
                                         13  12 001F5        BNEQ    15$                                  ;
                               7E    89  8F  9A 001F7        MOVZBL  #137, -(SP)                          ;
                                         01  DD 001FB        PUSHL   #1                                   ;
                               00000000G 8F  DD 001FD        PUSHL   #EXCH$_BADLOGIC                      ;
                   00000000G   00        03  FB 00203        CALLS   #3, LIB$STOP                         ;
                          3A   A8        52  D0 0020A 15$:   MOVL    R2, 58(R8)                           ; 0761
  0100   8F       00      04   B9        52  2C 0020E        MOVC5   R2, @4(R9), #0, #256, 90(R8)         ; 0762
                                     5A  A8     00216                                                     ;
                                     42  A8  7C 00218        CLRQ    66(R8)                               ; 0769
                          2B   A8        08  88 0021B        BISB2   #8, 43(R8)                           ; 0770
                                     18  AE  B5 0021F        TSTW    XAB+20                               ; 0772
                                         04  13 00222        BEQL    16$                                  ;
                                         50  D4 00224        CLRL    R0                                   ;
                                         03  11 00226        BRB     17$                                  ;
                               50        01  D0 00228 16$:   MOVL    #1, R0                               ;
           3E     A8      14   AE        50  C3 0022B 17$:   SUBL3   R0, XAB+16, 62(R8)                   ;
                                     24  A7  D4 00231        CLRL    36(R7)                               ; 0774
                          4A   A8  F988  CF  9E 00234        MOVAB   EXCH$FIL11_CLOSE_FILE, 74(R8)        ; 0778
                                     56  A8  D4 0023A        CLRL    86(R8)                               ; 0779
                          52   A8  FCF6  CF  9E 0023D        MOVAB   EXCH$FIL11_GET, 82(R8)               ; 0780
                               50        01  D0 00243        MOVL    #1, R0                               ; 0784
                                             04 00246        RET                                          ; 0786

; Routine Size:  583 bytes,     Routine Base:  EXCH$FIL11_CODE + 0440
```

```
1 GLOBAL ROUTINE exch$fil11_put = %SBTTL 'exch$fil11_put'
2 BEGIN
2 !++
2 !
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Add the next record to the file
2 !
2 ! INPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT INPUTS:
2 !
2 !       copy [copy$a_out_filb]   - out_filb - pointer to filb for an open Files-11 output file
2 !       copy [copy$a_inp_filb]   - inp_filb - pointer to the input filb containing the record info
2 !       inp_filb [filb$l_record_len] - len  - length of the record
2 !       inp_filb [filb$a_record]     - buf  - address of the record
2 !
2 ! OUTPUTS:
2 !
2 !       none
2 !
2 ! IMPLICIT OUTPUTS:
2 !
2 !       out_filb will get updated
2 !
2 ! ROUTINE VALUE:
2 !
2 !       true if success, false if any error
2 !
2 ! SIDE EFFECTS:
2 !
2 !       error conditions will be signaled
2 !--
2
2 $dbgtrc_prefix ('fil11_put> ');
2
2 LOCAL
2     status
2     ;
2
2 BIND
2     copy = exch$a_gbl [excg$a_copy_work]: $ref_bblock,    ! COPY verb work area
2     out_filb = copy [copy$a_out_filb]   : $ref_bblock,    ! pointer to filb for an open Files-11 output file
2     inp_filb = copy [copy$a_inp_filb]   : $ref_bblock,    ! pointer to the input filb with the record info
2     len  = inp_filb [filb$l_record_len],                  ! length of the record
2     buf  = inp_filb [filb$a_record],                      ! address of the record
2     ctx  = out_filb [filb$a_context]    : $ref_bblock,    ! output file context block
2     namb = out_filb [filb$a_assoc_namb] : $ref_bblock,    ! associated output namb structure
2     fab  = ctx [rmsb$a_fab]             : $ref_bblock,    ! RMS FAB for the file
2     rab  = ctx [rmsb$a_rab]             : $ref_bblock     ! RMS RAB
2     ;
2
2 $debug_print_lit ('entry');
2
2 $block_check (2, .out_filb, filb, 505);          !?? definitely over-zealous checking
```

```
;  754          0844  2 $block_check (2, .inp_filb, filb, 526);
;  755          0845  2 $block_check (2, .namb, namb, 506);
;  756          0846  2 $block_check (2, .ctx, rmsb, 507);
;  757          0847  2
;  758          0848  2 ! Set the record buffer fields in the rab
;  759          0849  2 !
;  760          0850  2 IF    .fab [fab$b_rfm] EQL fab$c_fix              ! If we have fixed-length output
;  761          0851  2   AND
;  762          0852  2       .fab [fab$w_mrs] NEQ .len                   ! And the input length isn't correct
;  763          0853  2 THEN
;  764          0854  3     BEGIN
;  765          0855  3     CH$COPY (.len, .buf, .inp_filb [filb$b_pad_char], .fab [fab$w_mrs], out_filb [filb$t_record_buffer]);
;  766          0856  3     rab [rab$l_rbf] = out_filb [filb$t_record_buffer];
;  767          0857  3     rab [rab$w_rsz] = .fab [fab$w_mrs];
;  768          0858  3     END
;  769          0859  2 ELSE                                              ! Otherwise just point the rab at the record
;  770          0860  3     BEGIN
;  771          0861  3     rab [rab$l_rbf] = .buf;                       ! buffer address
;  772          0862  3     rab [rab$w_rsz] = .len;                       ! buffer size
;  773          0863  2     END;
;  774          0864  2
;  775          0865  2 ! Write a single record to the output filb
;  776          0866  2 !
;  777          0867  3 IF NOT (status = $put (rab = .rab))
;  778          0868  2 THEN
;  779          0869  3     BEGIN
;  780          0870  3
;  781          0871  3     exch$util_file_error (exch$_writeerr, .status, .fab, .rab [rab$l_stv]);
;  782          0872  3     RETURN .status;
;  783          0873  3
;  784          0874  2     END;
;  785          0875  2
;  786          0876  2 RETURN true;
;  787          0877  1 END;


                                                                      .EXTRN   SYS$PUT

                                                      03FC 00000      .ENTRY   EXCH$FIL11_PUT, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0787
                                                                               R9                                           ;
                              59 00000000G   EF    9E 00002      MOVAB    EXCH$UTIL_BLOCK_CHECK, R9                    ;
              50 00000000G    EF             04    C1 00009      ADDL3    #4, EXCH$A_GBL, R0                           ; 0830
              51              60 00000044    8F    C1 00011      ADDL3    #68, (R0), R1                                ; 0831
              50              60             3C    C1 00019      ADDL3    #60, (R0), R0                                ; 0832
                              53             60    D0 0001D      MOVL     (R0), R3                                     ; 0833
                              56             61    D0 00020      MOVL     (R1), R6                                     ; 0835
              54        20    A6             10    C1 00023      ADDL3    #16, 32(R6), R4                              ; 0837
              58        20    A6             14    C1 00028      ADDL3    #20, 32(R6), R8                              ; 0838
                              52 035B00FA    8F    D0 0002D      MOVL     #56295674, R2                                ; 0843
                              51       01F9  8F    3C 00034      MOVZWL   #505, R1                                     ;
                              50             56    D0 00039      MOVL     R6, R0                                       ;
                                             69    16 0003C      JSB      EXCH$UTIL_BLOCK_CHECK                        ;
                              52 035B00FA    8F    D0 0003E      MOVL     #56295674, R2                                ; 0844
                              51       020E  8F    3C 00045      MOVZWL   #526, R1                                     ;
                              50             53    D0 0004A      MOVL     R3, R0                                       ;
                                             69    16 0004D      JSB      EXCH$UTIL_BLOCK_CHECK                        ;
                              52 010A00F7   8F  D0 0004F        MOVL    #17432823, R2                                 : 0845
                              51     01FA   8F  3C 00056        MOVZWL  #506, R1                                      :
                              50       18   A6  D0 0005B        MOVL    24(R6), R0                                    :
                                            69  16 0005F        JSB     EXCH$UTIL_BLOCK_CHECK                         :
                              52 031600F6   8F  D0 00061        MOVL    #51773686, R2                                 : 0846
                              51     01FB   8F  3C 00068        MOVZWL  #507, R1                                      :
                              50       20   A6  D0 0006D        MOVL    32(R6), R0                                    :
                                            69  16 00071        JSB     EXCH$UTIL_BLOCK_CHECK                         :
                              57            64  D0 00073        MOVL    (R4), R7                                      : 0850
                              01       1F   A7  91 00076        CMPB    31(R7), #1                                    :
                                            25  12 0007A        BNEQ    1$                                            :
     42   A3      36   A7     10            00  ED 0007C        CMPZV   #0, #16, 54(R7), 66(R3)                       : 0852
                                            1C  13 00083        BEQL    1$                                            :
     36   A7      39   A3     46   B3      42   A3  2C 00085    MOVC5   66(R3), @70(R3), 57(R3), 54(R7), 346(R6)      : 0855
                                     015A   C6     0008E                                                              :
                              52            68  D0 00091        MOVL    (R8), R2                                      : 0856
                              28   A2    015A   C6  9E 00094    MOVAB   346(R6), 40(R2)                               :
                              22   A2      36   A7  B0 0009A    MOVW    54(R7), 34(R2)                                : 0857
                                            0D  11 0009F        BRB     2$                                            : 0850
                              52            68  D0 000A1 1$:    MOVL    (R8), R2                                      : 0861
                              28   A2      46   A3  D0 000A4    MOVL    70(R3), 40(R2)                                :
                              22   A2      42   A3  B0 000A9    MOVW    66(R3), 34(R2)                                : 0862
                                            52  DD 000AE 2$:    PUSHL   R2                                            : 0867
                      00000000G   00        01  FB 000B0        CALLS   #1, SYS$PUT                                   :
                              53            50  D0 000B7        MOVL    R0, STATUS                                    :
                              18            53  E8 000BA        BLBS    STATUS, 3$                                    :
                                       0C   A2  DD 000BD        PUSHL   12(R2)                                        : 0871
                                     0088   8F  BB 000C0        PUSHR   #^M<R3,R7>                                    :
                                 00F810D0   8F  DD 000C4        PUSHL   #16257232                                     :
                      00000000G   EF        04  FB 000CA        CALLS   #4, EXCH$UTIL_FILE_ERROR                      :
                              50            53  D0 000D1        MOVL    STATUS, R0                                    : 0872
                                                04 000D4        RET                                                   :
                              50            01  D0 000D5 3$:    MOVL    #1, R0                                        : 0876
                                                04 000D8        RET                                                   : 0877

; Routine Size:  217 bytes,    Routine Base:  EXCH$FIL11_CODE + 0687
;  789           0878  1 END
;  790           0879  0 ELUDOM




                                                                .EXTRN  LIB$SIGNAL, LIB$STOP

;                                 PSECT SUMMARY
;
;        Name                        Bytes                      Attributes
;
;  EXCH$FIL11_CODE                       1888  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      117          0          1000         00:01.8
;  _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1     1151       86          7            79         00:01.3




;                                COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:EXCFIL11/OBJ=OBJ$:EXCFIL11 MSRC$:EXCFIL11/UPDATE=(ENH$:EXCFIL11)

; Size:          1888 code + 0 data bytes
; Run Time:         00:40.1
; Elapsed Time:     02:12.9
; Lines/CPU Min:      1315
; Lexemes/CPU-Min: 30904
; Memory Used:   282 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


EXCFIL11
LIS

EXCINIT
LIS

EXCLIB
LIS

EXCIO
LIS