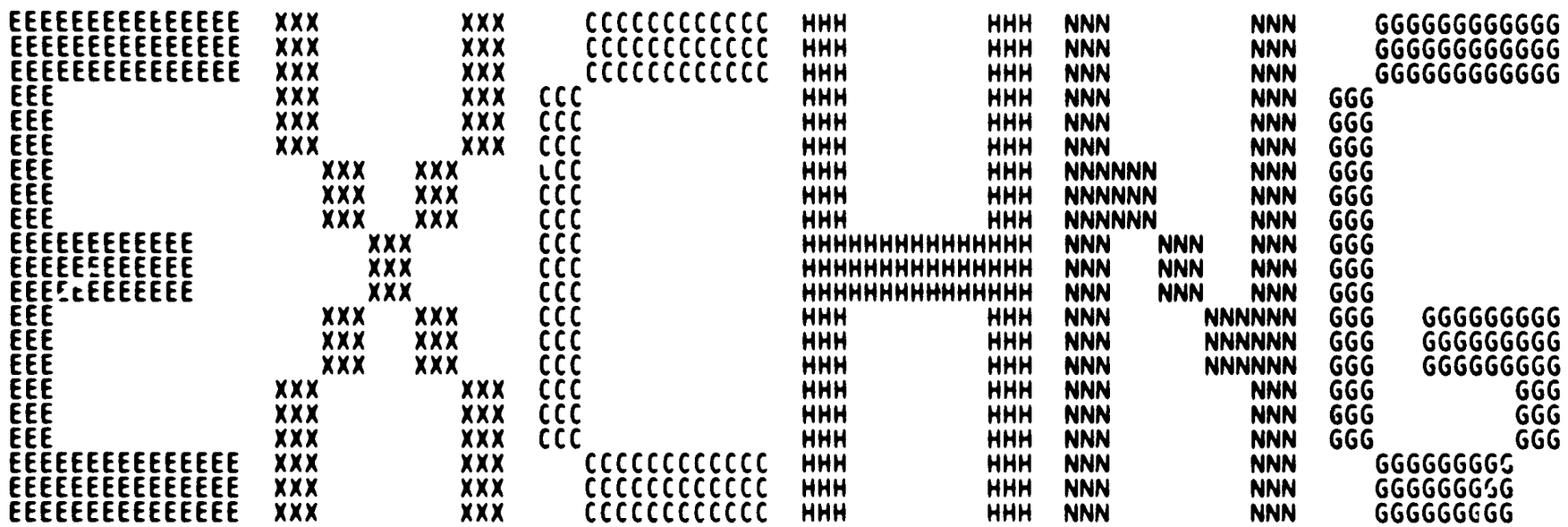
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN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

```
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      IIIIII    RRRRRRRR    EEEEEEEEEE
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      IIIIII    RRRRRRRR    EEEEEEEEEE
EE          XX      XX  CC          DD      DD      II      RR      RR  EE
EE          XX      XX  CC          DD      DD      II      RR      RR  EE
EE            XX  XX    CC          DD      DD      II      RR      RR  EE
EE            XX  XX    CC          DD      DD      II      RR      RR  EE
EEEEEEEE        XX      CC          DD      DD      II      RRRRRRRR    EEEEEEEE
EEEEEEEE        XX      CC          DD      DD      II      RRRRRRRR    EEEEEEEE
EE            XX  XX    CC          DD      DD      II      RR  RR      EE
EE            XX  XX    CC          DD      DD      II      RR  RR      EE
EE          XX      XX  CC          DD      DD      II      RR    RR    EE                ....
EE          XX      XX  CC          DD      DD      II      RR    RR    EE                ....
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      IIIIII    RR      RR  EEEEEEEEEE        ....
EEEEEEEEEE  XX      XX    CCCCCCCC  DDDDDDDD      IIIIII    RR      RR  EEEEEEEEEE        ....

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
```

```
;    1          0001  0 MODULE  exch$dire                                %TITLE 'DIRECTORY verb dispatch and misc routines'
;    2          0002  0                 (
;    3          0003  0                 IDENT = 'V04-000',
;    4          0004  0                 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
;    5          0005  0                 ) =
;    6          0006  1 BEGIN
;    7          0007  1 !
;    8          0008  1 !*****************************************************************************
;    9          0009  1 !*                                                                           *
;   10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;   11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;   12          0012  1 !*  ALL RIGHTS RESERVED.                                                      *
;   13          0013  1 !*                                                                           *
;   14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;   15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;   16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;   17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;   18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;   19          0019  1 !*  TRANSFERRED.                                                              *
;   20          0020  1 !*                                                                           *
;   21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;   22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;   23          0023  1 !*  CORPORATION.                                                              *
;   24          0024  1 !*                                                                           *
;   25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;   26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;   27          0027  1 !*                                                                           *
;   28          0028  1 !*                                                                           *
;   29          0029  1 !*****************************************************************************
;   30          0030  1
;   31          0031  1 !++
;   32          0032  1 ! FACILITY:     EXCHANGE - Foreign volume interchange facility
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT:     Primary action routines for DIRECTORY verb
;   35          0035  1 !
;   36          0036  1 ! ENVIRONMENT:  VAX/VMS User mode
;   37          0037  1 !
;   38          0038  1 ! AUTHOR:       CW Hobbs          CREATION DATE: 8-July-1982
;   39          0039  1 !
;   40          0040  1 ! MODIFIED BY:
;   41          0041  1 !
;   42          0042  1 !       V03-002 CWH3002         CW Hobbs                12-Apr-1984
;   43          0043  1 !               Signal a specific error for an attempt to do a directory
;   44          0044  1 !               on a remote node, simply saying "not permitted for files-11"
;   45          0045  1 !               could be confusing.
;   46          0046  1 !
;   47          0047  1 !
;   48          0048  1 !--
;   49          0049  1
;   50          0050  1 ! Include files:
;   51          0051  1 !
;   52          0052  1 MACRO $module_name_string = 'exch$dire' %;      ! The require file needs to know our module name
;   53          0053  1 REQUIRE 'SRC$:EXCREQ'                           ! Facility-wide require file
;   54          0054  1     ,
```

```
;    56          0151  1 %SBTTL 'Module table of contents'
;    57          0152  1
;    58          0153  1 ! Module table of contents:
;    59          0154  1 !
;    60          0155  1 FORWARD ROUTINE
;    61          0156  1           dire_close_output,                    ! Close the output file
;    62          0157  1     exch$dire_directory,                        ! Main action routine
;    63          0158  1     exch$dire_get_columns       : NOVALUE,      ! Get number of columns for listing
;    64          0159  1     exch$dire_get_width,                        ! Get output device line width
;    65          0160  1           dire_open_output,                     ! Open output file
;    66          0161  1           dire_process_parameters,              ! Loop through the parameters
;    67          0162  1     exch$dire_put               : NOVALUE,      ! Put a record to the listing file
;    68          0163  1           dire_rt11_directory,                  ! RT-11 directory
;    69          0164  1           dire_rt11_put_item    : NOVALUE,      ! Put one directory item
;    70          0165  1           dire_rt11_summary                     ! RT-11 directory summary
;    71          0166  1     ;
;    72          0167  1
;    73          0168  1 ! EXCHANGE facility routines
;    74          0169  1 !
;    75          0170  1 EXTERNAL ROUTINE
;    76          0171  1     exch$cmd_cli_get_integer,                   ! Get an integer value
;    77          0172  1     exch$cmd_match_filename,                    ! Match filename against wildcarded name
;    78          0173  1     exch$cmd_parse_filespec,                    ! Parse a file specification
;    79          0174  1     exch$dos11_directory,                       ! Directory command for DOS-11
;    80          0175  1     exch$moun_implied_mount,                    ! Do a default mount
;    81          0176  1     exch$rt11_expand_filename,                  ! Expand the RT11 file name             !?? not needed
;    82          0177  1     exch$rt11_dirseg_get,                       ! Get pointer to RT11 directory segment !?? not needed
;    83          0178  1     exch$rtacp_next_entry,                      ! Get pointer to next directory entry
;    84          0179  1     exch$util_fao_buffer,                       ! Perform the $fao service
;    85          0180  1     exch$util_file_error,                       ! Signal RMS error
;    86          0181  1     exch$util_namb_release      : NOVALUE,      ! Release name block
;    87          0182  1     exch$util_rmsb_allocate,                    ! Allocate Files-11 control block
;    88          0183  1     exch$util_rmsb_release      : NOVALUE,      ! Release Files-11 block
;    89          0184  1     exch$util_rt11ctx_allocate,                 ! Allocate RT-11 control block
;    90          0185  1     exch$util_rt11ctx_release   : NOVALUE,      ! Release RT-11 control block
;    91          0186  1     exch$util_vm_allocate_zeroed                ! Allocate virtual memory
;    92          0187  1     ;
;    93          0188  1
;    94          0189  1 ! Equated symbols:
;    95          0190  1 !
;    96          0191  1 !LITERAL
;    97          0192  1 !    ;
;    98          0193  1
;    99          0194  1 ! Bound declarations:
;   100          0195  1 !
;   101          0196  1 BIND
;   102          0197  1     null_string = %ASCID '',
;   103          0198  1     output_str  = %ASCID 'OUTPUT'
;   104          0199  1     ;
```

```
;  106     0200  1 GLOBAL ROUTINE dire_close_output =      %SBTTL 'dire_close_output'
;  107     0201  2 BEGIN
;  108     0202  2 !++
;  109     0203  2 !
;  110     0204  2 ! FUNCTIONAL DESCRIPTION:
;  111     0205  2 !
;  112     0206  2 !       Close the output file described by dire [dire$a_outfab].
;  113     0207  2 !
;  114     0208  2 ! INPUTS:
;  115     0209  2 !
;  116     0210  2 !       none
;  117     0211  2 !
;  118     0212  2 ! IMPLICIT INPUTS:
;  119     0213  2 !
;  120     0214  2 !       dire - pointer to work area
;  121     0215  2 !
;  122     0216  2 ! OUTPUTS:
;  123     0217  2 !
;  124     0218  2 !       none
;  125     0219  2 !
;  126     0220  2 ! IMPLICIT OUTPUTS:
;  127     0221  2 !
;  128     0222  2 !       none
;  129     0223  2 !
;  130     0224  2 ! ROUTINE VALUE:
;  131     0225  2 !
;  132     0226  2 !       Success or worst error encountered.
;  133     0227  2 !
;  134     0228  2 ! SIDE EFFECTS:
;  135     0229  2 !
;  136     0230  2 !       none
;  137     0231  2 !--
;  138     0232  2
;  139     0233  2 $dbgtrc_prefix ('dire_close_output> ');
;  140     0234  2
;  141     0235  2 LOCAL
;  142     0236  2     status
;  143     0237  2     ;
;  144     0238  2
;  145     0239  2 BIND
;  146     0240  2     dire = exch$a_gbl [excg$a_dire_work]        : $ref_bblock,  ! Pointer to work area
;  147     0241  2     fab  = dire [dire$a_outfab]         : $ref_bblock   ! Pointer to fab for output file
;  148     0242  2     ;
;  149     0243  2
;  150     0244  2 $block_check (2, .dire, dire, 417);
;  151     0245  2
;  152     0246  2 ! Close the file
;  153     0247  2 !
;  154     0248  3 IF NOT (status = $close (fab = .fab))
;  155     0249  2 THEN
;  156     0250  2     exch$util_file_error (exch$_closeout, .status, .fab, .fab [fab$l_stv]);
;  157     0251  2
;  158     0252  2 ! Put an RMS structure back on the available queue
;  159     0253  2 !
;  160     0254  2 exch$util_rmsb_release (.dire [dire$a_outrmsb]);
;  161     0255  2
;  162     0256  2 $debug_print_lit ('file closed');
```

```
;  163        0257  2
;  164        0258  2 RETURN .status;
;  165        0259  1 END;

                                                        .TITLE  EXCH$DIRE DIRECTORY verb dispatch and misc rout
                                                                          ines
;                                                       .IDENT  \V04-000\

                                                        .PSECT  EXCH$DIRE_PLIT,NOWRT,2

                                   00000 P.AAB:         .BLKB   0                                   ;
                         010E0000  00000 P.AAA:         .LONG   17694720                            ;
                         00000000' 00004                .ADDRESS P.AAB                              ;
      00  00  54  55  50  54  55  4F  00008 P.AAD:      .ASCII  \OUTPUT\<0><0>                      ;
                         010E0006  00010 P.AAC:         .LONG   17694726                            ;
                         00000000' 00014                .ADDRESS P.AAD                              ;

                                          NULL_STRING=          P.AAA
                                          OUTPUT_STR=           P.AAC
                                                        .EXTRN  EXCH$CMD_CLI_GET_INTEGER
                                                        .EXTRN  EXCH$CMD_MATCH_FILENAME
                                                        .EXTRN  EXCH$CMD_PARSE_FILESPEC
                                                        .EXTRN  EXCH$DOST1_DIRECTORY
                                                        .EXTRN  EXCH$MOUN_IMPLIED_MOUNT
                                                        .EXTRN  EXCH$RT11_EXPAND_FILENAME
                                                        .EXTRN  EXCH$RT11_DIRSEG_GET
                                                        .EXTRN  EXCH$RTACP_NEXT_ENTRY
                                                        .EXTRN  EXCH$UTIL_FAO_BUFFER
                                                        .EXTRN  EXCH$UTIL_FILE_ERROR
                                                        .EXTRN  EXCH$UTIL_NAMB_RELEASE
                                                        .EXTRN  EXCH$UTIL_RMSB_ALLOCATE
                                                        .EXTRN  EXCH$UTIL_RMSB_RELEASE
                                                        .EXTRN  EXCH$UTIL_RT11CTX_ALLOCATE
                                                        .EXTRN  EXCH$UTIL_RT11CTX_RELEASE
                                                        .EXTRN  EXCH$UTIL_VM_ALLOCATE_ZEROED
                                                        .EXTRN  EXCH$A_GBL, EXCH$UTIL_BLOCK_CHECK
                                                        .EXTRN  SYS$CLOSE

                                                        .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                   001C 00000           .ENTRY  DIRE_CLOSE_OUTPUT, Save R2,R3,R4    ; 0200
        50 00000000G  EF             0C   C1 00002      ADDL3   #12, EXCH$A_GBL, R0                 ; 0240
                      53             60   D0 0000A      MOVL    (R0), R3                            ; 0241
                      52  024400FE   8F   D0 0000D      MOVL    #38011134, R2                       ; 0244
                      51      01A1   8F   3C 00014      MOVZWL  #417, R1                            ;
                      50             53   D0 00019      MOVL    R3, R0                              ;
                          00000000G  EF   16 0001C      JSB     EXCH$UTIL_BLOCK_CHECK               ;
                      52        20   A3   D0 00022      MOVL    32(R3), R2                          ; 0248
                                     52   DD 00026      PUSHL   R2                                  ;
          00000000G   00             01   FB 00028      CALLS   #1, SYS$CLOSE                       ;
                      54             50   D0 0002F      MOVL    R0, STATUS                          ;
                      14             54   E8 00032      BLBS    STATUS, 1$                          ;
                                0C   A2   DD 00035      PUSHL   12(R2)                              ; 0250
                                     52   DD 00038      PUSHL   R2                                  ;
                                     54   DD 0003A      PUSHL   STATUS                              ;
                          00F81058   8F   DD 0003C      PUSHL   #16257112                           ;
```

```
                 00000000G  EF              04  FB 00042                CALLS   #4, EXCH$UTIL_FILE_ERROR                ;
                                       1C    A3  DD 00049 1$:            PUSHL   28(R3)                                  ; 0254
                 00000000G  EF              01  FB 0004C                CALLS   #1, EXCH$UTIL_RMSB_RELEASE              ;
                            50              54  D0 00053                MOVL    STATUS, R0                              ; 0258
                                                04 00056                RET                                             ; 0259

; Routine Size:  87 bytes,    Routine Base:  EXCH$DIRE_CODE + 0000
```

```
;  167          0260  1 GLOBAL ROUTINE exch$dire_directory =    %SBTTL 'exch$dire_directory'
;  168          0261  2 BEGIN
;  169          0262  2 !++
;  170          0263  2 !
;  171          0264  2 ! FUNCTIONAL DESCRIPTION:
;  172          0265  2 !
;  173          0266  2 !       Action routine for the DIRECTORY verb, parses and performs main control functions for DIRECTORY
;  174          0267  2 !
;  175          0268  2 ! INPUTS:
;  176          0269  2 !
;  177          0270  2 !       none
;  178          0271  2 !
;  179          0272  2 ! IMPLICIT INPUTS:
;  180          0273  2 !
;  181          0274  2 !       Command parameters and qualifiers as returned from CLI$xxx routines.
;  182          0275  2 !
;  183          0276  2 ! OUTPUTS:
;  184          0277  2 !
;  185          0278  2 !       none
;  186          0279  2 !
;  187          0280  2 ! IMPLICIT OUTPUTS:
;  188          0281  2 !
;  189          0282  2 !       none
;  190          0283  2 !
;  191          0284  2 ! ROUTINE VALUE:
;  192          0285  2 !
;  193          0286  2 !       Success or worst error encountered.
;  194          0287  2 !
;  195          0288  2 ! SIDE EFFECTS:
;  196          0289  2 !
;  197          0290  2 !       A directory of files may be displayed
;  198          0291  2 !--
;  199          0292  2
;  200          0293  2 $dbgtrc_prefix ('dire_directory> ');
;  201          0294  2
;  202          0295  2 LOCAL
;  203          0296  2     status,
;  204          0297  2     stat_2
;  205          0298  2     ;
;  206          0299  2
;  207          0300  2 BIND
;  208          0301  2     dire = exch$a_gbl [excg$a_dire_work] : $ref_bblock  ! Pointer to work area
;  209          0302  2     ;
;  210          0303  2
;  211          0304  2
;  212          0305  2 ! If our pointer is null, we need to allocate and initialize the work area
;  213          0306  2 !
;  214          0307  2 IF .dire EQL 0
;  215          0308  2 THEN
;  216          0309  3     BEGIN
;  217          0310  3
;  218          0311  3     ! Get the right sized chunk of memory, conveniently set to nulls
;  219          0312  3     !
;  220          0313  3     dire = exch$util_vm_allocate_zeroed (exchblk$s_dire);
;  221          0314  3
;  222          0315  3     ! Set the ident fields
;  223          0316  3     !
```

```
;  224     0317  3     $block_init (.dire, dire);
;  225     0318  3
;  226     0319  3     ! Set the descriptors up
;  227     0320  3     !
;  228     0321  3     $dyn_str_desc_init (dire [dire$q_filename]);
;  229     0322  3     $dyn_str_desc_init (dire [dire$q_output_file]);
;  230     0323  3     $dyn_str_desc_init (dire [dire$q_default_name]);
;  231     0324  3
;  232     0325  2     END;
;  233     0326  2
;  234     0327  2 ! Get some confidence that our work area is valid
;  235     0328  2 !
;  236     0329  2 $block_check (2, .dire, dire, 418);
;  237     0330  2
;  238     0331  2 ! Set the last part of the block to nulls
;  239     0332  2 !
;  240     0333  2 CH$FILL (0, exchblk$s_dire - dire$k_start_zero, .dire + dire$k_start_zero);
;  241     0334  2
;  242     0335  2 ! Get individual boolean qualifiers
;  243     0336  2 !
;  244     0337  2 dire [dire$v_q_all]       = cli$present (%ASCID 'ALL');
;  245     0338  2 !\ dire [dire$v_q_badblocks] = cli$present (%ASCID 'BADBLOCKS');
;  246     0339  2 dire [dire$v_q_blocks]    = cli$present (%ASCID 'BLOCKS');
;  247     0340  2 dire [dire$v_q_brief]     = cli$present (%ASCID 'BRIEF');
;  248     0341  2 dire [dire$v_q_date]      = cli$present (%ASCID 'DATE');
;  249     0342  2 dire [dire$v_q_deleted]   = cli$present (%ASCID 'DELETED');
;  250     0343  2 dire [dire$v_q_free]      = cli$present (%ASCID 'FREE');
;  251     0344  2 dire [dire$v_q_full]      = cli$present (%ASCID 'FULL');
;  252     0345  2 dire [dire$v_q_octal]     = cli$present (%ASCID 'OCTAL');
;  253     0346  2 dire [dire$v_q_owner]     = cli$present (%ASCID 'OWNER');
;  254     0347  2 dire [dire$v_q_printer]   = cli$present (%ASCID 'PRINTER');
;  255     0348  2 dire [dire$v_q_size]      = cli$present (%ASCID 'SIZE');
;  256     0349  2 dire [dire$v_q_summary]   = cli$present (%ASCID 'SUMMARY');
;  257     0350  2
;  258     0351  2 ! Get individual integer-valued qualifiers, routine signals on errors
;  259     0352  2 !
;  260     0353  3 IF NOT (status = exch$cmd_cli_get_integer (%ASCID 'COLUMNS', dire [dire$l_q_columns]))
;  261     0354  2 THEN                                                                                    ;
;  262     0355  2     RETURN .status;
;  263     0356  2
;  264     0357  2 ! Set any implied flags
;  265     0358  2 !
;  266     0359  2 IF .dire [dire$v_q_full]
;  267     0360  2 THEN
;  268     0361  2     dire [dire$v_q_blocks] = dire [dire$v_q_date] = dire [dire$v_q_size] = dire [dire$v_q_owner] = true;
;  269     0362  2 IF .dire [dire$v_q_brief]
;  270     0363  2 THEN
;  271     0364  2     dire [dire$v_q_blocks] = dire [dire$v_q_date] = dire [dire$v_q_size] = dire [dire$v_q_owner] = false;
;  272     0365  2
;  273     0366  2 ! Assume output to the SYS$OUTPUT device
;  274     0367  2 !
;  275     0368  2 dire [dire$a_outfab] = .exch$a_gbl [excg$a_sysout_fab];
;  276     0369  2 dire [dire$a_outrab] = .exch$a_gbl [excg$a_sysout_rab];
;  277     0370  2
;  278     0371  2 ! Get the output filename if requested
;  279     0372  2 !
;  280     0373  3 IF    (dire [dire$v_q_output] = cli$present (output_str))        ! /OUTPUT=<filespec>
```

```
;   281          0374  2    OR
;   282          0375  2        .dire [dire$v_q_printer]                    ! /PRINTER
;   283          0376  2 THEN
;   284          0377  3     IF NOT (status = dire_open_output (.dire [dire$v_q_printer]))       ! Open the output file, return on er
;   285          0378  2     THEN
;   286          0379  2         RETURN .status;
;   287          0380  2
;   288          0381  2 ! Get the listing width by doing a $getdvi on the output device
;   289          0382  2 !
;   290          0383  2 dire [dire$l_list_width] = exch$dire_get_width (.dire [dire$a_outfab]);
;   291          0384  2
;   292          0385  2 ! Loop through the list of file specifications
;   293          0386  2 !
;   294          0387  2 status = dire_process_parameters ();
;   295          0388  2
;   296          0389  2 ! Close the output file if we opened one
;   297          0390  2 !
;   298          0391  3 IF (.dire [dire$v_q_output] OR .dire [dire$v_q_printer])
;   299          0392  2 THEN
;   300          0393  3     BEGIN
;   301          0394  3     stat_2 = dire_close_output ();
;   302          0395  3     IF .status THEN status = .stat_2;          ! Make the output status the worse of process_para and close
;   303          0396  2     END;
;   304          0397  2
;   305          0398  2 RETURN .status;
;   306          0399  1 END;


                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

                                00  4C  4C  41  00018 P.AAF:   .ASCII  \ALL\<0>
                                    010E0003     0001C P.AAE:   .LONG   17694723
                                    00000000'    00020          .ADDRESS P.AAF
00  00  53  4B  43  4F  4C  42     00024 P.AAH:   .ASCII  \BLOCKS\<0><0>
                                    010E0006     0002C P.AAG:   .LONG   17694726
                                    00000000'    00030          .ADDRESS P.AAH
00  00  00  46  45  49  52  42     00034 P.AAJ:   .ASCII  \BRIEF\<0><0><0>
                                    010E0005     0003C P.AAI:   .LONG   17694725
                                    00000000'    00040          .ADDRESS P.AAJ
                                45  54  41  44  00044 P.AAL:   .ASCII  \DATE\
                                    010E0004     00048 P.AAK:   .LONG   17694724
                                    00000000'    0004C          .ADDRESS P.AAL
00  44  45  54  45  4C  45  44     00050 P.AAN:   .ASCII  \DELETED\<0>
                                    010E0007     00058 P.AAM:   .LONG   17694727
                                    00000000'    0005C          .ADDRESS P.AAN
                                45  45  52  46  00060 P.AAP:   .ASCII  \FREE\
                                    010E0004     00064 P.AAO:   .LONG   17694724
                                    00000000'    00068          .ADDRESS P.AAP
                                4C  4C  55  46  0006C P.AAR:   .ASCII  \FULL\
                                    010E0004     00070 P.AAQ:   .LONG   17694724
                                    00000000'    00074          .ADDRESS P.AAR
00  00  00  4C  41  54  43  4F     00078 P.AAT:   .ASCII  \OCTAL\<0><0><0>
                                    010E0005     00080 P.AAS:   .LONG   17694725
                                    00000000'    00084          .ADDRESS P.AAT
00  00  00  52  45  4E  57  4F     00088 P.AAV:   .ASCII  \OWNER\<0><0><0>
                                    010E0005     00090 P.AAU:   .LONG   17694725
```

```
                                    00000000'  00094            .ADDRESS P.AAV                                    ;
              00  52  45  54  4E  49  52  50  00098 P.AAX:     .ASCII  \PRINTER\<0>                              ;
                                    010E0007   000A0 P.AAW:     .LONG   17694727                                  ;
                                    00000000'  000A4            .ADDRESS P.AAX                                    ;
                              45  5A  49  53  000A8 P.AAZ:     .ASCII  \SIZE\                                    ;
                                    010E0004   000AC P.AAY:     .LONG   17694724                                  ;
                                    00000000'  000B0            .ADDRESS P.AAZ                                    ;
              00  59  52  41  4D  4D  55  53  000B4 P.ABB:     .ASCII  \SUMMARY\<0>                              ;
                                    010E0007   000BC P.ABA:     .LONG   17694727                                  ;
                                    00000000'  000C0            .ADDRESS P.ABB                                    ;
              00  53  4E  4D  55  4C  4F  43  000C4 P.ABD:     .ASCII  \COLUMNS\<0>                              ;
                                    010E0007   000CC P.ABC:     .LONG   17694727                                  ;
                                    00000000'  000D0            .ADDRESS P.ABD                                    ;

                                                                .EXTRN  EXCH$GQ_DYN_STR_TEMPLATE
                                                                .EXTRN  CLI$PRESENT

                                                                .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                        03FC 00000              .ENTRY  EXCH$DIRE_DIRECTORY, Save R2,R3,R4,R5,R6,-  ; 0260
                                                                        R7,R8,R9                                    ;
                        59 00000000G  EF  9E 00002              MOVAB   EXCH$A_GBL, R9                              ;
                        58      0000' CF  9E 00009              MOVAB   P.AAE, R8                                   ;
                        57 00000000G  00  9E 0000E              MOVAB   CLI$PRESENT, R7                             ;
            52          69            0C  C1 00015              ADDL3   #12, EXCH$A_GBL, R2                         ; 0301
                        62                D5 00019              TSTL    (R2)                                        ; 0307
                        47                12 0001B              BNEQ    1$                                          ;
                        7E       0244 8F  3C 0001D              MOVZWL  #580, -(SP)                                 ; 0313
              00000000G EF            01  FB 00022              CALLS   #1, EXCH$UTIL_VM_ALLOCATE_ZEROED            ;
                        62            50  D0 00029              MOVL    R0, (R2)                                    ;
                    08  A0       0244 8F  B0 0002C              MOVW    #580, 8(R0)                                 ; 0317
                    0A  A0            02  8E 00032              MNEGB   #2, 10(R0)                                  ;
                        50            62  D0 00036              MOVL    (R2), R0                                    ; 0321
                        53 00000000G  EF  D0 00039              MOVL    TMPL, R3                                    ;
                        60            53  D0 00040              MOVL    R3, (R0)                                    ;
                        51 00000000G  EF  D0 00043              MOVL    TMPL+4, R1                                  ;
                    04  A0            51  D0 0004A              MOVL    R1, 4(R0)                                   ;
            50          62            0C  C1 0004E              ADDL3   #12, (R2), R0                               ; 0322
                        60            53  D0 00052              MOVL    R3, (R0)                                    ;
                    04  A0            51  D0 00055              MOVL    R1, 4(R0)                                   ;
            50          62            14  C1 00059              ADDL3   #20, (R2), R0                               ; 0323
                        60            53  D0 0005D              MOVL    R3, (R0)                                    ;
                    04  A0            51  D0 00060              MOVL    R1, 4(R0)                                   ;
                        56            62  D0 00064 1$:          MOVL    (R2), R6                                    ; 0329
                        52 024400FE   8F  D0 00067              MOVL    #38011134, R2                               ;
                        51       01A2 8F  3C 0006E              MOVZWL  #418, R1                                    ;
                        50            56  D0 00073              MOVL    R6, R0                                      ;
                           00000000G  EF  16 00076              JSB     EXCH$UTIL_BLOCK_CHECK                       ;
0228   8F           00          6E        00  2C 0007C              MOVC5   #0, (SP), #0, #552, 28(R6)                  ; 0333
                                    1C    A6     00083                                                              ;
                        52         2C     A6  9E 00085              MOVAB   44(R6), R2                                  ; 0337
                                          58  DD 00089              PUSHL   R8                                          ;
                        67                01  FB 0008B              CALLS   #1, CLI$PRESENT                             ;
       62           01  00                50  F0 0008E              INSV    R0, #0, #1, (R2)                            ;
                                   10     A8  9F 00093              PUSHAB  P.AAG                                       ; 0339
                        67                01  FB 00096              CALLS   #1, CLI$PRESENT                             ;
       62           01  01                50  F0 00099              INSV    R0, #1, #1, (R2)                            ;
```

```
                                                20  A8  9F 0009E         PUSHAB  P.AAI                                           ; 0340
                                            67      01  FB 000A1         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  02      50  F0 000A4         INSV    R0, #2, #1, (R2)                                ;
                                                2C  A8  9F 000A9         PUSHAB  P.AAK                                           ; 0341
                                            67      01  FB 000AC         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  03      50  F0 000AF         INSV    R0, #3, #1, (R2)                                ;
                                                3C  A8  9F 000B4         PUSHAB  P.AAM                                           ; 0342
                                            67      01  FB 000B7         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  04      50  F0 000BA         INSV    R0, #4, #1, (R2)                                ;
                                                48  A8  9F 000BF         PUSHAB  P.AAO                                           ; 0343
                                            67      01  FB 000C2         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  05      50  F0 000C5         INSV    R0, #5, #1, (R2)                                ;
                                                54  A8  9F 000CA         PUSHAB  P.AAQ                                           ; 0344
                                            67      01  FB 000CD         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  06      50  F0 000D0         INSV    R0, #6, #1, (R2)                                ;
                                                64  A8  9F 000D5         PUSHAB  P.AAS                                           ; 0345
                                            67      01  FB 000D8         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  07      50  F0 000DB         INSV    R0, #7, #1, (R2)                                ;
                                                74  A8  9F 000E0         PUSHAB  P.AAU                                           ; 0346
                                            67      01  FB 000E3         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  09      50  F0 000E6         INSV    R0, #9, #1, (R2)                                ;
                                              0084  C8  9F 000EB         PUSHAB  P.AAW                                           ; 0347
                                            67      01  FB 000EF         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  0A      50  F0 000F2         INSV    R0, #10, #1, (R2)                               ;
                                              0090  C8  9F 000F7         PUSHAB  P.AAY                                           ; 0348
                                            67      01  FB 000FB         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  0B      50  F0 000FE         INSV    R0, #11, #1, (R2)                               ;
                                              00A0  C8  9F 00103         PUSHAB  P.ABA                                           ; 0349
                                            67      01  FB 00107         CALLS   #1, CLI$PRESENT                                 ;
                                62      01  0C      50  F0 0010A         INSV    R0, #12, #1, (R2)                               ;
                                                28  A6  9F 0010F         PUSHAB  40(R6)                                          ; 0353
                                              00B0  C8  9F 00112         PUSHAB  P.ABC                                           ;
                    00000000G   EF                  02  FB 00116         CALLS   #2, EXCH$CMD_CLI_GET_INTEGER                    ;
                                53                  50  D0 0011D         MOVL    R0, STATUS                                      ;
                                65                  53  E9 00120         BLBC    STATUS, 7$                                      ;
                            05  62                  06  E1 00123         BBC     #6, (R2), 2$                                    ; 0359
                                62              0A0A  8F  A8 00127         BISW2   #2570, (R2)                                     ; 0361
                            05  62                  02  E1 0012C 2$:      BBC     #2, (R2), 3$                                    ; 0362
                                62              0A0A  8F  AA 00130         BICW2   #2570, (R2)                                     ; 0364
                                50                  69  D0 00135 3$:      MOVL    EXCH$A_GBL, R0                                  ; 0368
                                    20  A6      00D0  C0  7D 00138         MOVQ    208(R0), 32(R6)                                 ;
                                                  F4  A8  9F 0013E         PUSHAB  OUTPUT_STR                                      ; 0373
                                            67      01  FB 00141         CALLS   #1, CLI$PRESENT                                 ;
01  A2                          01              00  50  F0 00144         INSV    R0, #0, #1, 1(R2)                               ;
                                04                  50  E8 0014A         BLBS    R0, 4$                                          ;
                            10  62                  0A  E1 0014D         BBC     #10, (R2), 5$                                   ; 0375
                        7E  62      01              0A  EF 00151 4$:      EXTZV   #10, #1, (R2), -(SP)                            ; 0377
                                        0000V   CF      01  FB 00156         CALLS   #1, DIRE_OPEN_OUTPUT                            ;
                                53                  50  D0 0015B         MOVL    R0, STATUS                                      ;
                                27                  53  E9 0015E         BLBC    STATUS, 7$                                      ;
                                                20  A6  DD 00161 5$:      PUSHL   32(R6)                                          ; 0383
                                        0000V   CF      01  FB 00164         CALLS   #1, EXCH$DIRE_GET_WIDTH                         ;
                                            38  A6      50  D0 00169         MOVL    R0, 56(R6)                                      ;
                                        0000V   CF      00  FB 0016D         CALLS   #0, DIRE_PROCESS_PARAMETERS                     ; 0387
                                53                  50  D0 00172         MOVL    R0, STATUS                                      ;
                                04                  01  A2  E8 00175         BLBS    1(R2), 6$                                       ; 0391
                            0B  62                  0A  E1 00179         BBC     #10, (R2), 7$                                   ;
```

```
                FE27    CF              00  FB 0017D 6$:    CALLS   #0, DIRE_CLOSE_OUTPUT          ; 0394
                        03              53  E9 00182        BLBC    STATUS, 7$                     ; 0395
                        53              50  D0 00185        MOVL    STAT_2, STATUS                 ;
                        50              53  D0 00188 7$:    MOVL    STATUS, R0                     ; 0398
                                            04 0018B        RET                                    ; 0399

; Routine Size:  396 bytes,    Routine Base:  EXCH$DIRE_CODE + 0057
```

```
;  308     0400  1 GLOBAL ROUTINE exch$dire_get_columns : NOVALUE =        %SBTTL 'exch$dire_get_columns'
;  309     0401  2 BEGIN
;  310     0402  2 !++
;  311     0403  2 !
;  312     0404  2 ! FUNCTIONAL DESCRIPTION:
;  313     0405  2 !
;  314     0406  2 !       Determine the number of columns for the listing
;  315     0407  2 !
;  316     0408  2 ! INPUTS:
;  317     0409  2 !
;  318     0410  2 !       none
;  319     0411  2 !
;  320     0412  2 ! IMPLICIT INPUTS:
;  321     0413  2 !
;  322     0414  2 !       dire block
;  323     0415  2 !
;  324     0416  2 ! OUTPUTS:
;  325     0417  2 !
;  326     0418  2 !       none
;  327     0419  2 !
;  328     0420  2 ! IMPLICIT OUTPUTS:
;  329     0421  2 !
;  330     0422  2 !       none
;  331     0423  2 !
;  332     0424  2 ! ROUTINE VALUE:
;  333     0425  2 !
;  334     0426  2 !       none
;  335     0427  2 !
;  336     0428  2 ! SIDE EFFECTS:
;  337     0429  2 !
;  338     0430  2 !       none
;  339     0431  2 !--
;  340     0432  2
;  341     0433  2 $dbgtrc_prefix ('dire_get_columns> ');
;  342     0434  2
;  343     0435  2 REGISTER
;  344     0436  2     cols,
;  345     0437  2     override
;  346     0438  2     ;
;  347     0439  2
;  348     0440  2 BIND
;  349     0441  2     dire = exch$a_gbl [excg$a_dire_work] : $ref_bblock
;  350     0442  2     ;
;  351     0443  2
;  352     0444  2
;  353     0445  2 ! Make of copy of the number of columns
;  354     0446  2 !
;  355     0447  2 cols = .dire [dire$l_q_columns];
;  356     0448  2
;  357     0449  2 ! Compare listing width with requested value
;  358     0450  2 !
;  359     0451  3 IF   (.cols GTR 0 )
;  360     0452  2   AND
;  361     0453  3      (((.cols * .dire [dire$l_item_width]) + ((.cols-1) * dire$k_item_spacing)) GTRU .dire [dire$l_list_widt
;  362     0454  2 THEN
;  363     0455  3     BEGIN
;  364     0456  3     override = true;
```

```
;  365          0457  3     $exch_signal (exch$_toomanycol);
;  366          0458  3     END
;  367          0459  2 ELSE
;  368          0460  2     override = false;
;  369          0461  2
;  370          0462  2 ! If /COLUMNS defaulted or needs to be fixed, compute the number of columns
;  371          0463  2 !
;  372          0464  2 IF (.cols EQL 0) OR .override
;  373          0465  2 THEN
;  374          0466  3     BEGIN
;  375          0467  3
;  376          0468  3     ! First guess, just divide listing width by item width
;  377          0469  3     !
;  378          0470  3     cols = .dire [dire$l_list_width] / .dire [dire$l_item_width];
;  379          0471  3
;  380          0472  3     ! Now, include the spacing, and decrement cols until it fits
;  381          0473  3     !
;  382          0474  3     WHILE ((.cols * .dire [dire$l_item_width]) + ((.cols-1) * dire$k_item_spacing)) GTRU .dire [dire$l_list_
;  383          0475  3     DO
;  384          0476  3         cols = .cols - 1;
;  385          0477  3
;  386          0478  3     $logic_check (2, (.cols GTRU 0), 108);
;  387          0479  3
;  388          0480  3     dire [dire$l_q_columns] = .cols;
;  389          0481  3
;  390          0482  2     END;
;  391          0483  2
;  392          0484  2 $debug_print_fao ('number of columns !UL', .cols);
;  393          0485  2
;  394          0486  2 RETURN;
;  395          0487  1 END;


                                                                .EXTRN  EXCH$_TOOMANYCOL
                                                                .EXTRN  EXCH$_BADLOGIC

                                                001C 00000      .ENTRY  EXCH$DIRE_GET_COLUMNS, Save R2,R3,R4            ; 0400
                50 00000000G   EF                 0C   C1 00002  ADDL3   #12, EXCH$A_GBL, R0                      ; 0441
                               54                 60   D0 0000A  MOVL    (R0), R4                                 ; 0447
                               52           28    A4   D0 0000D  MOVL    40(R4), COLS                             ;
                                                  26   15 00011  BLEQ    1$                                       ; 0451
                50             52           34    A4   C5 00013  MULL3   52(R4), COLS, R0                         ; 0453
                51             52                 03   C5 00018  MULL3   #3, COLS, R1                            ;
                               50           FD A140   9E 0001C  MOVAB   -3(R1)[R0], R0                           ;
                        38     A4                 50   D1 00021  CMPL    R0, 56(R4)                               ;
                                                  12   1B 00025  BLEQU   1$                                       ;
                               53                 01   D0 00027  MOVL    #1, OVERRIDE                             ; 0456
                                    00000000G     8F   DD 0002A  PUSHL   #EXCH$_TOOMANYCOL                        ; 0457
                00000000G      00                 01   FB 00030  CALLS   #1, LIB$SIGNAL                           ;
                                                  02   11 00037  BRB     2$                                       ; 0451
                                                  53   D4 00039 1$:    CLRL    OVERRIDE                         ; 0460
                                                  52   D5 0003B 2$:    TSTL    COLS                             ; 0464
                                                  03   13 0003D        BEQL    3$                               ;
                               39                 53   E9 0003F        BLBC    OVERRIDE, 7$                     ;
                52       38    A4           34    A4   C7 00042 3$:    DIVL3   52(R4), 56(R4), COLS             ; 0470
                50             52           34    A4   C5 00048 4$:    MULL3   52(R4), COLS, R0                 ; 0474
```

```
                51              52              03  C5 0004D           MULL3   #3, COLS, R1                                ;
                                50          FD A140  9E 00051           MOVAB   -3(R1)[R0], R0                              ;
                          38    A4              50  D1 00056           CMPL    R0, 56(R4)                                  ;
                                                04  1B 0005A           BLEQU   5$                                          ;
                                                52  D7 0005C           DECL    COLS                                        ; 0476
                                                E8  11 0005E           BRB     4$                                          ;
                                                52  D5 00060 5$:       TSTL    COLS                                        ; 0478
                                                13  12 00062           BNEQ    6$                                          ;
                              7E        6C      8F  9A 00064           MOVZBL  #108, -(SP)                                 ;
                                                01  DD 00068           PUSHL   #1                                          ;
                                00000000G       8F  DD 0006A           PUSHL   #EXCH$_BADLOGIC                             ;
                    00000000G   00              03  FB 00070           CALLS   #3, LIB$STOP                                ;
                          28    A4              52  D0 00077 6$:       MOVL    COLS, 40(R4)                                ; 0480
                                                04  0007B 7$:       RET                                                 ; 0487

; Routine Size:  124 bytes,    Routine Base:  EXCH$DIRE_CODE + 01E3
```

```
;  397     0488  1 GLOBAL ROUTINE exch$dire_get_width (fab : $ref_bblock) =        %SBTTL 'exch$dire_get_width'
;  398     0489  2 BEGIN
;  399     0490  2 !++
;  400     0491  2 !
;  401     0492  2 ! FUNCTIONAL DESCRIPTION:
;  402     0493  2 !
;  403     0494  2 !       Get the line width of the output device
;  404     0495  2 !
;  405     0496  2 ! INPUTS:
;  406     0497  2 !
;  407     0498  2 !       fab - pointer to a fab with the opened output file
;  408     0499  2 !
;  409     0500  2 ! IMPLICIT INPUTS:
;  410     0501  2 !
;  411     0502  2 !       nam block hanging from the fab
;  412     0503  2 !
;  413     0504  2 ! OUTPUTS:
;  414     0505  2 !
;  415     0506  2 !       none
;  416     0507  2 !
;  417     0508  2 ! IMPLICIT OUTPUTS:
;  418     0509  2 !
    419     0510  2 !       none
    420     0511  2 !
    421     0512  2 ! ROUTINE VALUE:
    422     0513  2 !
            0514  2 !       width of output line
            0515  2 !
      4     0516  2 ! SIDE EFFECTS:
      4     0517  2 !
      4     0518  2 !       none
            0519  2 !--
            0520  2
;     4,    0521  2 $dbgtrc_prefix ('dire_get_width> ');
;     43    0522  2
      4       3  2 LOCAL
              24  2     dev_item : VECTOR [4, LONG],
            0525  2     dev_name : VECTOR [2, LONG],
            0526  2     status,
            0527  2     width
              28  2     ;
              29  2
;       39  0530  2 BIND
;       40  0531  2     nam = .fab [fab$l_nam] : $bblock            ! Nam block for this fab
        41  0532  2     ;
      442   0533  2
      443   0534  2
      444   0535  2 ! Initialize the item list for the $GETDVI
        45  0536  2 !
;     446   0537  2 %IF switch_variant GEQ 3 %THEN width = 0; %FI          ! Suppress uninit reference message while debugging
;     447   0538  2 dev_item [0] = (dvi$_devbufsiz^16 OR 4);          ! Device buffer size, output length 4
;     448   0539  2 dev_item [1] = width;                            ! Address of output buffer
;     449   0540  2 dev_item [2] = 0;                                ! No returned length
;     450   0541  2 dev_item [3] = 0;                                ! End of GETDVI item list
;     451   0542  2
;     452   05-3  2 ! Pull the device name from the name block
;     453   0544  2 !
```

```
;  454          0545  2 dev_name [0] = .nam [nam$b_dev];                ! Get the size
;  455          0546  2 dev_name [1] = .nam [nam$l_dev];                !  and the address
;  456          0547  2
;  457          0548  2 $debug_print_fao ('Attempting GETDVI on !AS', dev_name);
;  458          0549  2
;  459          0550  2 ! Get the device information
;  460          0551  2 !
;  461          0552  3 IF NOT (status = $getdviw (efn=0, devnam=dev_name, itmlst=dev_item))
;  462          0553  2 THEN
;  463          0554  2     $exch_signal_stop (.status);
;  464          0555  2
;  465          0556  2 ! If the output is a disk, the buffer will show up as 512.  Reduce it to 132
;  466          0557  2 !
;  467          0558  2 IF .width EQL 512
;  468          0559  2 THEN
;  469          0560  2     width = 132;
;  470          0561  2
;  471          0562  2 $debug_print_fao ('Device width is !UL', .width);
;  472          0563  2
;  473          0564  2 RETURN .width
;  474          0565  1 END;
; INFO#250                 L1:0558
; Referenced LOCAL symbol WIDTH is probably not initialized


                                                        .EXTRN  SYS$GETDVIW, LIB$STOP

                                        0000 00000      .ENTRY  EXCH$DIRE_GET_WIDTH, Save nothing               ; 0488
                                    5E  0000 C2 00002   SUBL2   #28, SP                                         ;
                                    50  04   AC D0 00005   MOVL    FAB, R0                                     ; 0531
                                    50  28   A0 D0 00009   MOVL    40(R0), R0                                  ;
                                0C  AE  00080004 8F D0 0000D   MOVL    #524292, DEV_ITEM                       ; 0538
                                10  AE       6E 9E 00015   MOVAB   WIDTH, DEV_ITEM+4                           ; 0539
                                        14  AE 7C 00019   CLRQ    DEV_ITEM+8                                  ; 0540
                                04  AE      39 A0 9A 0001C MOVZBL  57(R0), DEV_NAME                            ; 0545
                                08  AE      44 A0 D0 00021 MOVL    68(R0), DEV_NAME+4                          ; 0546
                                           7E 7C 00026   CLRQ    -(SP)                                       ; 0552
                                           7E 7C 00028   CLRQ    -(SP)                                       ;
                                        1C  AE 9F 0002A   PUSHAB  DEV_ITEM                                    ;
                                        18  AE 9F 0002D   PUSHAB  DEV_NAME                                    ;
                                           7E 7C 00030   CLRQ    -(SP)                                       ;
                    00000000G   00         08 FB 00032   CALLS   #8, SYS$GETDVIW                              ;
                                0A         50 E8 00039   BLBS    STATUS, 1$                                  ;
                                           50 DD 0003C   PUSHL   STATUS                                      ; 0554
                    00000000G   00         01 FB 0003E   CALLS   #1, LIB$STOP                                 ;
                                           04 00045      RET                                                 ;
                    00000200    8F         6E D1 00046 1$:    CMPL    WIDTH, #512                             ; 0558
                                           04 12 0004D   BNEQ    2$                                          ;
                                6E      84 8F 9A 0004F   MOVZBL  #132, WIDTH                                  ; 0560
                                50         6E D0 00053 2$:    MOVL    WIDTH, R0                               ; 0564
                                           04 00056      RET                                                 ; 0565

; Routine Size:  87 bytes,    Routine Base:  EXCH$DIRE_CODE + 025F
```

```
;  476     0566    GLOBAL ROUTINE dire_open_output (spool_output) =        %SBTTL 'dire_open_output'
;  477     0567  2 BEGIN
;  478     0568  2 !++
;  479     0569  2 !
;  480     0570  2 ! FUNCTIONAL DESCRIPTION:
;  481     0571  2 !
;  482     0572  2 !       Open the output file described on the /OUTPUT qualifier.  Memory is allocated for RMS data structure
;  483     0573  2 !
;  484     0574  2 ! INPUTS:
;  485     0575  2 !
;  486     0576  2 !       spool_output - True if the file should be spooled and deleted
;  487     0577  2 !
;  488     0578  2 ! IMPLICIT INPUTS:
;  489     0579  2 !
;  490     0580  2 !       Command qualifier value as returned from CLI$xxx routines.
;  491     0581  2 !
;  492     0582  2 ! OUTPUTS:
;  493     0583  2 !
;  494     0584  2 !       none
;  495     0585  2 !
;  496     0586  2 ! IMPLICIT OUTPUTS:
;  497     0587  2 !
;  498     0588  2 !       dire [dire$a_outfab] - a module-wide pointer to the fab which has the opened output file
;  499     0589  2 !       dire [dire$a_outrab] - a module-wide pointer to the rab which has the connected output stream
;  500     0590  2 !
;  501     0591  2 ! ROUTINE VALUE:
;  502     0592  2 !
;  503     0593  2 !       true, or error code if one occurred
;  504     0594  2 !
;  505     0595  2 ! SIDE EFFECTS:
;  506     0596  2 !
;  507     0597  2 !       Memory is allocated, a file is opened
;  508     0598  2 !--
;  509     0599  2
;  510     0600  2 $dbgtrc_prefix ('dire_open_output> ');
;  511     0601  2
;  512     0602  2 LOCAL
;  513     0603  2     status
;  514     0604  2     ;
;  515     0605  2
;  516     0606  2 BIND
;  517     0607  2     dire = exch$a_gbl [excg$a_dire_work] : $ref_bblock, ! Pointer to work area
;  518     0608  2     fab  = dire [dire$a_outfab]          : $ref_bblock, ! Pointer to fab for output file
;  519     0609  2     outdesc = dire [dire$q_output_file] : $bblock,      ! Descriptor for name of output file
;  520     0610  2     rab  = dire [dire$a_outrab]          : $ref_bblock, ! Pointer to rab for output file
;  521     0611  2     rmsb = dire [dire$a_outrmsb]         : $ref_bblock  ! Pointer to $RMSB for output file
;  522     0612  2     ;
;  523     0613  2
;  524     0614  2 $block_check (2, .dire, dire, 419);
;  525     0615  2
;  526     0616  2 ! Get the output filename, it can be null and we don't care
;  527     0617  2 !
;  528     0618  2 cli$get_value (output_str, outdesc);
;  529     0619  2
;  530     0620  2 $debug_print_fao ('original output file name is "!AS"', outdesc);
;  531     0621  2
;  532     0622  2 ! Get a pointer to the RMS structures for the output file
```

```
;  533          0623  2 !
;  534          0624  2 rmsb = exch$util_rmsb_allocate ();
;  535          0625  2
;  536          0626  2 ! Set the FAB and RAB pointers for the listing file to what we are opening
;  537          0627  2 !
;  538          0628  2 fab = .rmsb [rmsb$a_fab];
;  539          0629  2 rab = .rmsb [rmsb$a_rab];
;  540          0630  2
;  541          0631  2 ! Initialize the RMS structures
;  542          0632  2 !
;  543        P 0633  2 $fab_init (
;  544        P 0634  2           FAB = .rmsb [rmsb$a_fab],                   ! Output file FAB
;  545        P 0635  2           DNA = UPLIT BYTE ('EXCHDIRE.LIS'),          !  Default name address
;  546        P 0636  2           DNS = 12,                                   !  Default name size
;  547        P 0637  2           FNA = .outdesc [dsc$a_pointer],             !  Set name addr
;  548        P 0638  2           FNS = .outdesc [dsc$w_length],              !  Set name size
;  549        P 0639  2           FOP = (NAM,SQO),                            !  Sequential only
;  550        P 0640  2           MRS = 132,                                  !  Maximum record size
;  551        P 0641  2           RAT = CR,                                   !  Carriage return record attribute
;  552          0642  2           NAM = .rmsb [rmsb$a_nam]);                  !  Name block
;  553        P 0643  2 $rab_init (
;  554        P 0644  2           RAB = .rmsb [rmsb$a_rab],                   ! Output file RAB
;  555        P 0645  2           MBF = 2,                                    !  Multi-buffer count
;  556        P 0646  2           RAC = SEQ,                                  !  Sequential only
;  557        P 0647  2           ROP = WBH,                                  !  Write behind
;  558          0648  2           FAB = .rmsb [rmsb$a_fab]);                  !  FAB addr
;  559        P 0649  2 $nam_init (
;  560        P 0650  2           NAM = .rmsb [rmsb$a_nam],                   ! File name block
;  561        P 0651  2           RSA = .rmsb [rmsb$a_rsbuf],                 !  Result name addr
;  562        P 0652  2           RSS = nam$c_maxrss,                         !  Result name size
;  563        P 0653  2           ESA = .rmsb [rmsb$a_esbuf],                 !  Expanded name addr
;  564          0654  2           ESS = nam$c_maxrss);                        !  Expanded name size
;  565          0655  2
;  566          0656  2 ! If the file is to be spooled, then set some more bits in the fab (Delete,Name block,Spool,Sequential only)
;  567          0657  2 !
;  568          0658  2 IF .spool_output
;  569          0659  2 THEN
;  570          0660  2     fab [fab$l_fop] = (fab$m_dlt OR fab$m_nam OR fab$m_spl OR fab$m_sqo);
;  571          0661  2
;  572          0662  2 ! Create and connect to the output file, free memory if any errors
;  573          0663  2 !
;  574          0664  3 IF NOT (status = $create (fab = .fab))
;  575          0665  2 THEN
;  576          0666  3     BEGIN
;  577          0667  3     exch$util_file_error (exch$_openout, .status, .fab, .fab [fab$l_stv]);
;  578          0668  3     exch$util_rmsb_release (.rmsb);                 ! Free the RMS structure
;  579          0669  3     RETURN exch$_openout;
;  580          0670  2     END;
;  581          0671  3 IF NOT (status = $connect (rab = .rab))
;  582          0672  2 THEN
;  583          0673  3     BEGIN
;  584          0674  3     exch$util_file_error (exch$_openout, .status, .fab, .rab [rab$l_stv]);
;  585          0675  3     exch$util_rmsb_release (.rmsb);                 ! Free the RMS structure
;  586          0676  3     RETURN exch$_openout;
;  587          0677  2     END;
;  588          0678  2
;  589          0679  3 BEGIN
```

```
;  590          0680  3     BIND
;  591          0681  3         nam = .rmsb [rmsb$a_nam] : $bblock;
;  592          0682  3     $trace_print_fao ('output file name is ''.AF''', .nam [nam$b_rsl], .nam [nam$l_rsa]);
;  593          0683  2 END;
;  594          0684  2
;  595          0685  2 RETURN true;
;  596          0686  1 END;


                                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

        53 49 4C 2E 45 52 49 44 48 43 58 45  000D4 P.ABE:    .ASCII  \EXCHDIRE.LIS\                                  ;

                                                                .EXTRN  CLI$GET_VALUE, SYS$CREATE
                                                                .EXTRN  SYS$CONNECT

                                                                .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                                     0FFC 00000 .ENTRY  DIRE_OPEN_OUTPUT, Save R2,R3,R4,R5,R6,R7,-   ; 0566
                                                                        R8,R9,R10,R11
                53 00000000G   EF                    0C   C1 00002 ADDL3   #12, EXCH$A_GBL, R3                      ; 0607
                5B             63                    20   C1 0000A ADDL3   #32, (R3), R11                         ; 0608
                5A             63                    0C   C1 0000E ADDL3   #12, (R3), R10                         ; 0609
                7E             63                    24   C1 00012 ADDL3   #36, (R3), -(SP)                       ; 0610
                54             63                    1C   C1 00016 ADDL3   #28, (R3), R4                         ; 0611
                               52 024400FE           8F   D0 0001A MOVL    #38011134, R2                          ; 0614
                               51     01A3           8F   3C 00021 MOVZWL  #419, R1
                               50                    63   D0 00026 MOVL    (R3), R0
                                  00000000G          EF   16 00029 JSB     EXCH$UTIL_BLOCK_CHECK
                                                     5A   DD 0002F PUSHL   R10                                     ; 0618
                                        0000'        CF   9F 00031 PUSHAB  OUTPUT_STR
                   00000000G   00                    02   FB 00035 CALLS   #2, CLI$GET_VALUE
                   00000000G   EF                    00   FB 0003C CALLS   #0, EXCH$UTIL_RMSB_ALLOCATE             ; 0624
                               64                    50   D0 00043 MOVL    R0, (R4)
                               59                    64   D0 00046 MOVL    (R4), R9                               ; 0628
                               56       10           A9   D0 00049 MOVL    16(R9), R6
                               6B                    56   D0 0004D MOVL    R6, (R11)
                               58       14           A9   D0 00050 MOVL    20(R9), R8                             ; 0629
                         00    BE                    58   D0 00054 MOVL    R8, @0(SP)
0050  8F                00     6E                    00   2C 00058 MOVC5   #0, (SP), #0, #80, (R6)                 ; 0642
                                                     66        0005F
                               66       5003         8F   B0 00060 MOVW    #20483, (R6)
                         04    A6 01000040           8F   D0 00065 MOVL    #16777280, 4(R6)
                         16    A6                    02   90 0006D MOVB    #2, 22(R6)
                         1E    A6       0202         8F   B0 00071 MOVW    #514, 30(R6)
                               57       18           A9   D0 00077 MOVL    24(R9), R7
                         28    A6                    57   D0 0007B MOVL    R7, 40(R6)
                         2C    A6       04           AA   D0 0007F MOVL    4(R10), 44(R6)
                         30    A6       0000'        CF   9E 00084 MOVAB   P.ABE, 48(R6)
                         34    A6                    6A   90 0008A MOVB    (R10), 52(R6)
                         35    A6                    0C   90 0008E MOVB    #12, 53(R6)
                         36    A6       84           8F   9B 00092 MOVZBW  #132, 54(R6)
0044  8F                00     6E                    00   2C 00097 MOVC5   #0, (SP), #0, #68, (R8)                 ; 0648
                                                     68        0009E
                               68       4401         8F   B0 0009F MOVW    #17409, (R8)
                         04    A8       0400         8F   3C 000A4 MOVZWL  #1024, 4(R8)
```

```
                                  1E    A8 94 000AA          CLRB    30(R8)                              :
                         36  A8         02 90 000AD          MOVB    #2, 54(R8)                          :
                         3C  A8         56 D0 000B1          MOVL    R6, 60(R8)                          :
   0060  8F          00      6E         00 2C 000B5          MOVC5   #0, (SP), #0, #96, (R7)             : 0654
                             67            000BC                                                         :
                             67   6002  8F B0 000BD          MOVW    #24578, (R7)                        :
                         02  A7         01 8E 000C2          MNEGB   #1, 2(R7)                           :
                         04  A7     20  A9 D0 000C6          MOVL    32(R9), 4(R7)                       :
                         0A  A7         01 8E 000CB          MNEGB   #1, 10(R7)                          :
                         0C  A7     1C  A9 D0 000CF          MOVL    28(R9), 12(P7)                      :
                             0B     04  A  E9 000D4          BLBC    SPOOL_OUTPUT, 1$                    : 0658
                             50         6B D0 000D8          MOVL    (R11), R0                           : 0660
                         04  A0 0100A040 8F D0 000DB         MOVL    #16818240, 4(R0)                    :
                             53         6B D0 000E3 1$:      MOVL    (R11), R3                           : 0664
                                        53 DD 000E6          PUSHL   R3                                  :
                  00000000G  00         01 FB 000E8          CALLS   #1, SYS$CREATE                      :
                             54         50 D0 000EF          MOVL    R0, STATUS                          :
                             05         54 E8 000F2          BLBS    STATUS, 2$                          :
                                    0C  A3 DD 000F5          PUSHL   12(R3)                              : 0667
                                        16 11 000F8          BRB     3$                                  :
                             52     00  BE D0 000FA 2$:      MOVL    @0(SP), R2                          : 0671
                                        52 DD 000FE          PUSHL   R2                                  :
                  00000000G  00         01 FB 00100          CALLS   #', SYS$CONNECT                     :
                             54         50 D0 00107          MOVL    R0, STATUS                          :
                             25         54 E8 0010A          BLBS    STATUS, 4$                          :
                                    0C  A2 DD 0010D          PUSHL   12(R2)                              : 0674
                                        53 DD 00110 3$:      PUSHL   R3                                  :
                                        54 DD 00112          PUSHL   STATUS                              :
                               00F810A0 8F DD 00114          PUSHL   #16257184                           :
                  00000000G  EF         04 FB 0011A          CALLS   #4, EXCH$UTIL_FILE_ERROR            :
                                        59 DD 00121          PUSHL   R9                                  : 0675
                  00000000G  EF         01 FB 00123          CALLS   #1, EXCH$UTIL_RMSB_RELEASE          :
                             50 00F810A0 8F D0 0012A         MOVL    #16257184, R0                       : 0676
                                        04 00131             RET                                         :
                             50         01 D0 00132 4$:      MOVL    #1, R0                              : 0685
                                        04 00135             RET                                         : 0686

; Routine Size:  310 bytes,    Routine Base:  EXCH$DIRE_CODE + 02B6
```

```
;  598          0687  1 GLOBAL ROUTINE dire_process_parameters =        %SBTTL 'dire_process_parameters'
;  599          0688  2 BEGIN
;  600          0689  2 !++
;  601          0690  2 !
;  602          0691  2 ! FUNCTIONAL DESCRIPTION:
;  603          0692  2 !
;  604          0693  2 !       Process the list of directory parameters.
;  605          0694  2 !
;  606          0695  2 ! INPUTS:
;  607          0696  2 !
;  608          0697  2 !       none
;  609          0698  2 !
;  610          0699  2 ! IMPLICIT INPUTS:
;  611          0700  2 !
;  612          0701  2 !       Command qualifier value as returned from CLI$xxx routines.
;  613          0702  2 !
;  614          0703  2 ! OUTPUTS:
;  615          0704  2 !
;  616          0705  2 !       none
;  617          0706  2 !
;  618          0707  2 ! IMPLICIT OUTPUTS:
;  619          0708  2 !
;  620          0709  2 !       none
;  621          0710  2 !
;  622          0711  2 ! ROUTINE VALUE:
;  623          0712  2 !
;  624          0713  2 !       Success or worst error encountered.
;  625          0714  2 !
;  626          0715  2 ! SIDE EFFECTS:
;  627          0716  2 !
;  628          0717  2 !       none
;  629          0718  2 !--
;  630          0719  2
;  631          0720  2 $dbgtrc_prefix ('dire_process_parameters> ');
;  632          0721  2
;  633          0722  2 LOCAL
;  634          0723  2     namb : $ref_bblock,
;  635          0724  2     volb : $ref_bblock,
;  636          0725  2     namb_head : $ref_bblock,
;  637          0726  2     namb_tail : $ref_bblock,
;  638          0727  2     format,
;  639          0728  2     status
;  640          0729  2     ;
;  641          0730  2
;  642          0731  2 BIND
;  643          0732  2     dire = exch$a_gbl [excg$a_dire_work] : $ref_bblock, ! Pointer to work area
;  644          0733  2     fab  = dire [dire$a_outfab]          : $ref_bblock  ! Pointer to fab for output file
;  645          0734  2     ;
;  646          0735  2
;  647          0736  2 $block_check (2, .dire, dire, 420);
;  648          0737  2
;  649          0738  2 status = $severe_stat_copy (exch$_badlogic);    ! If no parameter then we have a problem
```

```
;  651          0739  2 ! Get the work list of nambs by reading all filename parameters and placing the items in a list
;  652          0740  2 !
;  653          0741  2 str$copy_dx (dire [dire$q_default_name], %ASCID '*.*');          ! Init sticky name
;  654          0742  2 namb_head = 0;
;  655          0743  2 %IF switch_variant GEQ 3 %THEN namb_tail = 0; %FI       ! Suppress uninit reference message while debugging
;  656          0744  2
;  657          0745  2 !?? The following loops do not clean up nambs when they see an error - should be cleaned up
;  658          0746  2 !
;  659          0747  2 WHILE 1
;  660          0748  2 DO
;  661          0749  3     BEGIN
;  662          0750  3
;  663          0751  3     ! Fetch the filename and a pointer to a namb
;  664          0752  3     !
;  665          0753  3     status = exch$cmd_parse_filespec (%ASCID 'FILENAME', dire [dire$q_default_name], 0, dire [dire$q_filenam
;  666          0754  3     IF  .status EQL 0
;  667          0755  3       OR
;  668          0756  3         .exch$a_gbl [excg$v_control_c]
;  669          0757  3     THEN
;  670          0758  3         EXITLOOP;                                   ! No more files in the list
;  671          0759  3     IF NOT .status
;  672          0760  3     THEN
;  673          0761  3         $exch_signal_return (exch$_parseerr, 1, dire [dire$q_filename], .status);
;  674          0762  3     $debug_print_fao ('input parameter is "!AS"', dire [dire$q_filename]);
;  675          0763  3
;  676          0764  3     ! If the device is not mounted, then perform an implied mount
;  677          0765  3     !
;  678          0766  3     IF .namb [namb$a_assoc_volb] EQL 0
;  679          0767  3       AND
;  680          0768  5         (BEGIN
;  681          0769  5          BIND
;  682          0770  5            dev = namb [namb$l_fabdev] : $bblock;
;  683          0771  6          .dev [dev$v_for] OR (NOT (.dev [dev$v_mnt]))
;  684          0772  4          END)
;  685          0773  3     THEN
;  686          0774  4         IF NOT (status = exch$moun_implied_mount (.namb))
;  687          0775  3         THEN
;  688          0776  3             RETURN .status;
;  689          0777  3
;  690          0778  3     ! Now make sure that we can do a directory of this volume type
;  691          0779  3     !
;  692          0780  3     volb = .namb [namb$a_assoc_volb];
;  693          0781  3     IF .volb NEQ 0                                  ! If the volb is there, then the interesting format
;  694          0782  3     THEN
;  695          0783  3         format = .volb [volb$b_vol_format]          !  is that of the volb.  Otherwise, it
;  696          0784  3     ELSE
;  697          0785  3         format = .namb [namb$b_vol_format];         !  is that of the namb.
;  698          0786  3                                                                                                    54
;  699          0787  3     CASE .format FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF                                20
;  700          0788  3     SET
;  701          0789  3         [volb$k_vfmt_files:1] :
;  702          0790  4                                 BEGIN
;  703          0791  4                                 LOCAL
;  704          0792  4                                     errstat;                                                   20
;  705          0793  4                                 IF .namb [namb$v_explicit_node]                                00
;  706          0794  4                                 THEN
;  707          0795  5                                     BEGIN
```

```
;  708     0796  5                                errstat = exch$_noremote;
;  709     0797  5                                $exch_signal (.errstat, 1, dire [dire$q_filename]);
;  710     0798  5                                END
;  711     0799  4                            ELSE
;  712     0800  5                                BEGIN
;  713     0801  5                                errstat = exch$_opnotperf11;
;  714     0802  5                                $exch_signal (.errstat, 1, namb [namb$q_device]);
;  715     0803  4                                END;
;  716     0804  4                            exch$util_namb_release (.namb);                 ! Release the bad namb
;  717     0805  4                            WHILE .namb_head NEQ 0                          ! Release any other nambs
;  718     0806  4                            DO
;  719     0807  5                                BEGIN
;  720     0808  5                                REGISTER
;  721     0809  5                                    temp;
;  722     0810  5                                temp = .namb_head;                          ! Save current namb
;  723     0811  5                                namb_head = .namb_head [namb$a_next];       ! Advance to next namb
;  724     0812  5                                exch$util_namb_release (.temp);             ! Release the saved namb
;  725     0813  4                                END;
;  726     0814  4                            RETURN .errstat;
;  727     0815  3                            END;
;  728     0816  3        [INRANGE] :                                                         ! These we can do
;  729     0817  3        [volb$k_vfmt_invalid, OUTRANGE] :
;  730     0818  3                            $logic_check (0, (false), 239);
;  731     0819  3    TES;
;  732     0820  3
;  733     0821  3    ! We should now have a valid volb, but we still should check
;  734     0822  3    !
;  735     0823  3    $block_check (2, .volb, volb, 421);
;  736     0824  3
;  737     0825  3    ! Now copy the full name to the default name for proper stickiness
;  738     0826  3    !
;  739     0827  3    str$copy_dx (dire [dire$q_default_name], namb [namb$q_fullname]);
;  740     0828  3
;  741     0829  3    ! Add this namb to the tail of the work list
;  742     0830  3    !
;  743     0831  3    IF .namb_head EQL 0
;  744     0832  3    THEN
;  745     0833  4        BEGIN
;  746     0834  4        namb_head = .namb;
;  747     0835  4        namb_tail = .namb;
;  748     0836  4        END
;  749     0837  3    ELSE
;  750     0838  4        BEGIN
;  751     0839  4        namb_tail [namb$a_next] = .namb;        ! Ignore uninit reference message on this assignment
;  752     0840  4        namb_tail = .namb;
;  753     0841  3        END;
;  754     0842  3
;  755     0843  2    END;
```

```
;  757          0844  2 ! Now print the volumes.  We collect nambs for a given volume by chaining them from the volb.  We then print
;  758          0845  2 ! volume.  Repeat until the work list is done.
;  759          0846  2 !
;  760          0847  2 WHILE .namb_head NEQ 0
;  761          0848  2 DO
;  762          0849  3     BEGIN
;  763          0850  3
;  764          0851  3     ! Pull the first namb from the work list
;  765          0852  3     !
;  766          0853  3     namb = .namb_head;                          ! Grab a pointer to it
;  767          0854  3     namb_head = .namb [namb$a_next];            !  and remove it from the work list
;  768          0855  3
;  769          0856  3     ! Get the volb for this first namb
;  770          0857  3     !
;  771          0858  3     volb = .namb [namb$a_assoc_volb];           ! The namb has a pointer to its volb
;  772          0859  3     volb [volb$a_namb_head] = .namb;            !  and now the volb can find the namb
;  773          0860  3     namb_tail = .namb;                          ! Tail of the chain on this volb now
;  774          0861  3     namb_tail [namb$a_next] = 0;                ! End of the chain
;  775          0862  3
;  776          0863  3     ! Now, chain any additional nambs which reference this same volb, onto this volb
;  777          0864  3     !
;  778          0865  3     WHILE 1                                     ! This could have been coded as
;  779          0866  3     DO                                          !  " WHILE  ((.namb_head NEQ 0) AND
;  780          0867  4         BEGIN                                   !       (.namb_head [namb$a_assoc_volb] EQL .volb)) "
;  781          0868  4         IF .namb_head EQL 0                     !  except that we must control the order of the tests
;  782          0869  4         THEN
;  783          0870  4             EXITLOOP;
;  784          0871  4         IF .namb_head [namb$a_assoc_volb] NEQ .volb
;  785          0872  4         THEN
;  786          0873  4             EXITLOOP;
;  787          0874  4
;  788          0875  4         ! This namb is for the same volume, add it to the end of the chain
;  789          0876  4         !
;  790          0877  4         namb = .namb_head;                      ! Remove the namb from the work list
;  791          0878  4         namb_head = .namb [namb$a_next];
;  792          0879  4         namb_tail [namb$a_next] = .namb;        ! Add it to the tail of the chain hanging from the volb
;  793          0880  4         namb_tail = .namb;
;  794          0881  4         namb_tail [namb$a_next] = 0;            ! It is the end of the chain
;  795          0882  3         END;
;  796          0883  3
;  797          0884  3     ! Now print the directory
;  798          0885  3     !
;  799          0886  3     CASE .volb [volb$b_vol_format] FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF
;  800          0887  3     SET
;  801          0888  3         [volb$k_vfmt_dos11]     : status = exch$dos11_directory (.volb);
;  802          0889  3         [volb$k_vfmt_rt11]      : status = dire_rt11_directory (.volb);
;  803          0890  3 !\      [volb$k_vfmt_rtmt]      : $exch_signal_stop (exch$_notimplement);
;  804          0891  3         [OUTRANGE,INRANGE]      : $logic_check (0, (false), 240);
;  805          0892  3     TES;
;  806          0893  3
;  807          0894  3     ! Now release all the nambs hanging off this volb
;  808          0895  3     !
;  809          0896  3     namb = .volb [volb$a_namb_head];
;  810          0897  3     WHILE .namb NEQ 0
;  811          0898  3     DO
;  812          0899  4         BEGIN
;  813          0900  4         REGISTER
```

```
;  814          0901  4              temp;
;  815          0902  4          temp = .namb;
;  816          0903  4          namb = .namb [namb$a_next];
;  817          0904  4          exch$util_namb_release (.temp);
;  818          0905  3          END;
;  819          0906  3
;  820          0907  2      END;
;  821          0908  2
;  822          0909  2 RETURN .status;
;  823          0910  1 END;
; INFO#250                 L1:0839
; Referenced LOCAL symbol NAMB_TAIL is probably not initialized


                                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

                                                00  2A  2E  2A  000E0 P.ABG:   .ASCII  \*.*\<0>                                ;
                                                  010E0003  000E4 P.ABF:   .LONG   17694723                                   ;
                                                  00000000' 000E8          .ADDRESS P.ABG                                     ;
                                        45  4D  41  4E  45  4C  49  46  000EC P.ABI:   .ASCII  \FILENAME\                        ;
                                                  010E0008  000F4 P.ABH:   .LONG   17694728                                   ;
                                                  00000000' 000F8          .ADDRESS P.ABI                                     ;

                                                                .EXTRN  STR$COPY_DX, EXCH$_PARSEERR
                                                                .EXTRN  EXCH$_NOREMOTE, EXCH$_OPNOTPERF11

                                                                .PSEC   EXCH$DIRE_CODE,NOWRT,2

                                                    0FFC 00000          .ENTRY  DIRE_PROCESS_PARAMETERS, Save R2,R3,R4,R5,- ; 0687
                                                                        R6,R7,R8,R9,R10,R11
                                              5B 00000000G  8F  D0 00002        MOVL    #EXCH$_BADLOGIC, R11                      ;
                                              5E            04  C2 00009        SUBL2   #4, SP                                    ;
                                        58 00000000G        EF  C1 0000C        ADDL3   #12, EXCH$A_GBL, R8                       ; 0732
                                              52 024400FE   8F  D0 00014        MOVL    #38011134, R2                             ; 0736
                                              51     01A4   8F  3C 0001B        MOVZWL  #420, R1                                  ;
                                              50            68  D0 00020        MOVL    (R8), R0                                  ;
                                                 00000000G  EF  16 00023        JSB     EXCH$UTIL_BLOCK_CHECK                     ;
                                              50            5B  D0 00029        MOVL    R11, STATUS2                              ; 0738
                                50      03    00            04  F0 0002C        INSV    #4, #0, #3, STATUS2                       ;
                                              59            50  D0 00031        MOVL    STATUS2, STATUS                           ;
                                                    0000'   CF  9F 00034        PUSHAB  P.ABF                                     ; 0741
                                        57    68            14  C1 00038        ADDL3   #20, (R8), R7                             ;
                                              57            DD  0003C          PUSHL   R7                                        ;
                                    00000000G 00            02  FB 0003E        CALLS   #2, STR$COPY_DX                           ;
                                                            56  D4 00045        CLRL    NAMB_HEAD                                 ; 0742
                                                            5E  DD 00047 1$:     PUSHL   SP                                        ; 0753
                                                            68  DD 00049        PUSHL   (R8)                                      ;
                                                            7E  D4 0004B        CLRL    -(SP)                                     ;
                                                            57  DD 0004D        PUSHL   R7                                        ;
                                                    0000'   CF  9F 0004F        PUSHAB  P.ABH                                     ;
                                    00000000G       EF  05  FB 00053        CALLS   #5, EXCH$CMD_PARSE_FILESPEC                ;
                                              59            50  D0 0005A        MOVL    R0, STATUS                                ;
                                                            03  12 0005D        BNEQ    3$                                        ; 0754
                                                        00E8  31 0005F 2$:      BRW     19$                                       ;
                                              F6 00000000G  FF  E8 00062 3$:    BLBS    @EXCH$A_GBL, 2$                           ; 0756
                                              19            59  E8 00069        BLBS    STATUS, 4$                                ; 0759
```

```
                                52 00000000G   8F   D0 0006C          MOVL    #EXCH$_PARSEERR, TEMP                   ; 0761
                                               59   DD 00073          PUSHL   STATUS
                                               68   DD 00075          PUSHL   (R8)
                                               01   DD 00077          PUSHL   #1
                                               52   DD 00079          PUSHL   TEMP
                          00000000G   00             04   FB 0007B          CALLS   #4, LIB$SIGNAL
                                            008D     31   00082          BRW     15$
                                54                 6E   D0 00085 4$:      MOVL    NAMB, R4                        ; 0766
                                         74        A4   D5 00088          TSTL    116(R4)
                                                   1B   12 0008B          BNEQ    6$
                                 5               6B    A4   E8 0008D          BLBS    107(R4), 5$                     ; 0771
                          12         6A          A4         03   E0 00091          BBS     #3, 106(R4), 6$
                                                   54   DD 00096 5$:      PUSHL   R4                              ; 0774
                          00000000G   EF             01   FB 00098          CALLS   #1, EXCH$MOUN_IMPLIED_MOUNT
                                59                 50   D0 0009F          MOVL    R0, STATUS
                                03                 59   E8 000A2          BLBS    STATUS, 6$
                                            0138     31   000A5          BRW     31$
                                55         74      A4   D0 000A8 6$:      MOVL    116(R4), VOLB                   ; 0780
                                                   06   13 000AC          BEQL    7$                              ; 0781
                                5A         58      A5   9A 000AE          MOVZBL  88(VOLB), FORMAT                ; 0783
                                                   04   11 000B2          BRB     8$
                                5A         7A      A4   9A 000B4 7$:      MOVZBL  122(R4), FORMAT                 ; 0785
                             03                    00   5A   CF 000B8 8$:      CASEL   FORMAT, #0, #3                  ; 0787
                    005A          0019          005A         0008     000BC 9$:      .WORD   10$-9$,-
                                                                                             16$-9$,-
                                                                                             11$-9$,-
                                                                                             16$-9$
                                7E         EF      8F   9A 000C4 10$:      MOVZBL  #239, -(SP)                     ; 0818
                                                   01   DD 000C8          PUSHL   #1
                                                   5B   DD 000CA          PUSHL   R11
                          00000000G   00             03   FB 000CC          CALLS   #3, LIB$STOP
                                                   41   11 000D3          BRB     16$
                    0B          6C    A4                 06   E1 000D5 11$:      BBC     #6, 108(R4), 12$        ; 0793
                                52 00000000G   8F   D0 000DA          MOVL    #EXCH$_NOREMOTE, ERRSTAT                ; 0796
                                               68   DD 000E1          PUSHL   (R8)                            ; 0797
                                               0A   11 000E3          BRB     13$
                                52 00000000G   8F   D0 000E5 12$:      MOVL    #EXCH$_OPNOTPERF11, ERRSTAT             ; 0801
                                         40    A4   9F 000EC          PUSHAB  64(R4)                          ; 0802
                                               01   DD 000EF 13$:      PUSHL   #1
                                               52   DD 000F1          PUSHL   ERRSTAT
                          00000000G   00             03   FB 000F3          CALLS   #3, LIB$SIGNAL
                                               54   DD 000FA          PUSHL   R4                              ; 0804
                          00000000G   EF             01   FB 000FC 14$:      CALLS   #1, EXCH$UTIL_NAMB_RELEASE
                                               56   D5 00103          TSTL    NAMB_HEAD                       ; 0805
                                               0B   13 00105          BEQL    15$
                                50             56   D0 00107          MOVL    NAMB_HEAD, TEMP                 ; 0810
                                56         70  A6   D0 0010A          MOVL    112(NAMB_HEAD), NAMB_HEAD       ; 0811
                                               50   DD 0010E          PUSHL   TEMP                            ; 0812
                                               EA   11 00110          BRB     14$
                                50             52   D0 00112 15$:      MOVL    ERRSTAT, R0                     ; 0814
                                               04   00115            RET
                                52 041B00F3    8F   D0 00116 16$:      MOVL    #68878579, R2                   ; 0823
                                51      01A5   8F   3C 0011D          MOVZWL  #421, R1
                                50             55   D0 00122          MOVL    VOLB, R0
                                      00000000G EF   16 00125          JSB     EXCH$UTIL_BLOCK_CHECK
                                            18   A4   9F 0012B          PUSHAB  24(R4)                          ; 0827
                                               57   DD 0012E          PUSHL   R7
```

```
                00000000G  00          02  FB 00130          CALLS   #2, STR$COPY_DX                          ;
                                        56  D5 00137          TSTL    NAMB_HEAD                              ; 0831
                                        05  12 00139          BNEQ    17$                                    ;
                           56           54  D0 0013B          MOVL    R4, NAMB_HEAD                          ; 0834
                                        04  11 0013E          BRB     18$                                    ; 0831
                        70 A3           54  D0 00140 17$:     MOV     R4, 112(NAMB_TAIL)                     ; 0839
                        53              54  D0 00144 18$:     MC      R4, NAMB_TAIL                          ; 0835
                                      FEFD  31 00147          BRw     1$                                     ; 0747
                                        56  D5 0014A 19$:     TSTL    NAMB_HEAD                              ; 0847
                                        03  12 0014C          BNEQ    20$                                    ;
                                      008F  31 0014E          BRW     31$                                    ;
                           6E             56  D0 00151 20$:     MOVL    NAMB_HEAD, NAMB                        ; 0853
                           50             6E  D0 00154          MOVL    NAMB, R0                               ; 0854
                           56          70 A0  D0 00157          MOVL    112(R0), NAMB_HEAD                     ;
                           55          74 A0  D0 0015B          MOVL    116(R0), VOLB                          ; 0858
                        4C A5             50  D0 0015F          MOVL    R0, 76(VOLB)                           ; 0859
                           53             50  D0 00163          MOVL    R0, NAMB_TAIL                          ; 0860
                                       70 A3  D4 00166          CLRL    112(NAMB_TAIL)                         ; 0861
                           51          70 A3  9E 00169          MOVAB   112(R3), R1                            ; 0879
                                        56  D5 0016D 21$:     TSTL    NAMB_HEAD                              ; 0868
                                        1E  13 0016F          BEQL    22$                                    ;
                           55          74 A6  D1 00171          CMPL    116(NAMB_HEAD), VOLB                   ; 0871
                                        18  12 00175          BNEQ    22$                                    ;
                           6E             56  D0 00177          MOVL    NAMB_HEAD, NAMB                        ; 0877
                           50             6E  D0 0017A          MOVL    NAMB, R0                               ; 0878
                           56          70 A0  D0 0017D          MOVL    112(R0), NAMB_HEAD                     ;
                           61             50  D0 00181          MOVL    R0, (R1)                               ; 0879
                           53             50  D0 00184          MOVL    R0, NAMB_TAIL                          ; 0880
                           51          70 A3  9E 00187          MOVAB   112(R3), R1                            ; 0881
                                        61  D4 0018B          CLRL    (R1)                                   ;
                                        DE  11 0018D          BRB     21$                                    ; 0865
                     03                 00  58 A5  8F 0018F 22$:     CASEB   88(VOLB), #0, #3                       ; 0886
0024                 0008                   0019      0008     00194 23$:     .WORD   24$-23$,-                      ;
                                                                                25$-23$,-                      ;
                                                                                24$-23$,-                      ;
                                                                                26$-23$                        ;
                           7E          F0  8F  9A 0019C 24$:     MOVZBL  #240, -(SP)                            ; 0891
                                        01  DD 001A0          PUSHL   #1                                     ;
                                        5B  DD 001A2          PUSHL   R11                                    ;
                00000000G  00           03  FB 001A4          CALLS   #3, LIB$STOP                           ;
                                        15  11 001AB          BRB     28$                                    ;
                                        55  DD 001AD 25$:     PUSHL   VOLB                                   ; 0888
                00000000G  EF           01  FB 001AF          CALLS   #1, EXCH$DOS11_DIRECTORY               ;
                                        07  11 001B6          BRB     27$                                    ;
                                        55  DD 001B8 26$:     PUSHL   VOLB                                   ; 0889
                     0000V  CF          01  FB 001BA          CALLS   #1, DIRE_RT11_DIRECTORY                ;
                           59             50  D0 001BF 27$:     MOVL    R0, STATUS                             ;
                           6E          4C A5  D0 001C2 28$:     MOVL    76(VOLB), NAMB                         ; 0896
                           51             6E  D0 001C6 29$:     MOVL    NAMB, R1                               ; 0897
                                        03  12 001C9          BNEQ    30$                                    ;
                                      FF7C  31 001CB          BRW     19$                                    ;
                           50             51  D0 001CE 30$:     MOVL    R1, TEMP                               ; 0902
                           6E          70 A1  D0 001D1          MOVL    112(R1), NAMB                          ; 0903
                                        50  DD 001D5          PUSHL   TEMP                                   ; 0904
                00000000G  EF           01  FB 001D7          CALLS   #1, EXCH$UTIL_NAMB_RELEASE             ;
                                        E6  11 001DE          BRB     29$                                    ; 0897
                           50             59  D0 001E0 31$:     MOVL    STATUS, R0                             ; 0909
```

```
                                    04 001E3        RET                                     ; 0910

; Routine Size:  484 bytes,    Routine Base:  EXCH$DIRE_CODE + 03EC
```

```
;  825          0911  1 GLOBAL ROUTINE exch$dire_put (desc : $ref_bblock, cancel_ctrl_o) : NOVALUE =    %SBTTL 'exch$dire_put (desc,
;  826          0912  2 BEGIN
;  827          0913  2 !++
;  828          0914  2 !
;  829          0915  2 ! FUNCTIONAL DESCRIPTION:
;  830          0916  2 !
;  831          0917  2 !       Put a sirgle record to the output listing file
;  832          0918  2 !
;  833          0919  2 ! INPUTS:
;  834          0920  2 !
;  835          0921  2 !       desc - a string descriptor for the record
;  836          0922  2 !       cancel_ctrl_o - flag describing whether to cancel control o (optional)
;  837          0923  2 !
;  838          0924  2 ! IMPLICIT INPUTS:
;  839          0925  2 !
;  840          0926  2 !       dire - directory work area
;  841          0927  2 !
;  842          0928  2 ! OUTPUTS:
;  843          0929  2 !
;  844          0930  2 !       none
;  845          0931  2 !
;  846          0932  2 ! IMPLICIT OUTPUTS:
;  847          0933  2 !
;  848          0934  2 !       none
;  849          0935  2 !
;  850          0936  2 ! ROUTINE VALUE:
;  851          0937  2 !
;  852          0938  2 !       true if put succeeded, false if not
;  853          0939  2 !
;  854          0940  2 ! SIDE EFFECTS:
;  855          0941  2 !
;  856          0942  2 !       error conditions will be signaled
;  857          0943  2 !--
;  858          0944  2
;  859          0945  2 $dbgtrc_prefix ('dire_put> ');
;  860          0946  2
;  861          0947  2 BUILTIN
;  862          0948  2     ACTUALCOUNT
;  863          0949  2     ;
;  864          0950  2
;  865          0951  2 LOCAL                                                                                                    66
;  866          0952  2     status                                                                                               73
;  867          0953  2     ;
;  868          0954  2
;  869          0955  2 BIND
;  870          0956  2     dire = exch$a_gbl [excg$a_dire_work]        : $ref_bblock,  ! Pointer to work area                   74
;  871          0957  2     fab  = dire [dire$a_outfab]         : $ref_bblock,  ! Pointer to fab for output file
;  872          0958  2     rab  = dire [dire$a_outrab]         : $ref_bblock   ! Pointer to rab for output file
;  873          0959  2     ;
;  874          0960  2                                                                                                          20
;  875          0961  2                                                                                                          74
;  876          0962  2 $block_check (2, .dire, dire, 422);
;  877          0963  2
;  878          0964  2 ! Copy the record size and address to the RAB
;  879          0965  2 !
;  880          0966  2 rab [rab$w_rsz] = .desc [dsc$w_length];                                                                  20
;  881          0967  2 rab [rab$l_rbf] = .desc [dsc$a_pointer];                                                                 69
```

```
;  882          0968  2
;  883          0969  2 ! Set the bit for control o
;  884          0970  2 !
;  885          0971  2 IF ACTUALCOUNT () EQL 2
;  886          0972  2 THEN
;  887          0973  2     rab [rab$v_cco] = .cancel_ctrl_o;
;  888          0974  2
;  889          0975  2 ! Now write the record
;  890          0976  2 !
;  891          0977  3 IF NOT (status = $put (rab = .rab))
;  892          0978  2 THEN
;  893          0979  2     exch$util_file_error ($severe_stat_copy (exch$_writeerr), .status, .fab, .rab [rab$l_stv]);
;  894          0980  2
;  895          0981  2 rab [rab$v_cco] = false;                        ! Force it off
;  896          0982  2
;  897          0983  2 RETURN;
;  898          0984  1 END;


                                                                .EXTRN  SYS$PUT

                                                003C 00000      .ENTRY  EXCH$DIRE_PUT, Save R2,R3,R4,R5                 ; 0911
                53 00000000G   EF               0C   C1 00002  ADDL3   #12, EXCH$A_GBL, R3                             ; 0956
                55             63               20   C1 0000A  ADDL3   #32, (R3), R5                                 ; 0957
                54             63               24   C1 0000E  ADDL3   #36, (R3), R4                                 ; 0958
                               52 024400FE      8F   D0 00012  MOVL    #38011134, R2                                  ; 0962
                               51     01A6      8F   3C 00019  MOVZWL  #422, R1                                       ;
                               50               63   D0 0001E  MOVL    (R3), R0                                       ;
                                  00000000G     EF   16 00021  JSB     EXCH$UTIL_BLOCK_CHECK                          ;
                               52               64   D0 00027  MOVL    (R4), R2                                       ; 0966
                               50         04    AC   D0 0002A  MOVL    DESC, R0                                       ;
                                     22   A2    60   B0 0002E  MOVW    (R0), 34(R2)                                   ;
                                     28   A2    04   A0 00032  MOVL    4(R0), 40(R2)                                  ; 0967
                               02               6C   91 00037  CMPB    (AP), #2                                       ; 0971
                                                07   12 0003A  BNEQ    1$                                             ;
      07   A2           01                     07   08   AC   F0 0003C  INSV    CANCEL_CTRL_O, #7, #1, 7(R2)           ; 0973
                                                52   DD 00043  1$:    PUSHL   R2                                     ; 0977
                                  00000000G     00   01   FB   00045  CALLS   #1, SYS$PUT                             ;
                               1C               50   E8 0004C  BLBS    STATUS, 2$                                     ;
                                           0C   A2   DD 0004F  PUSHL   12(R2)                                         ; 0979
                                                65   DD 00052  PUSHL   (R5)                                          ;
                                                50   DD 00054  PUSHL   STATUS                                       ;
                               50 00F810D0      8F   D0 00056  MOVL    #16257232, STATUS2                             ;
         50            03                     00   04   F0 0005D  INSV    #4, #0, #3, STATUS2                            ;
                                                50   DD 00062  PUSHL   STATUS2                                      ;
                                  00000000G     EF   04   FB   00064  CALLS   #4, EXCH$UTIL_FILE_ERROR                 ;
                                        07   A2   80   8F   8A 0006B  2$:    BICB2   #128, 7(R2)                            ; 0981
                                                04   00070  RET                                                       ; 0984

; Routine Size:  113 bytes,    Routine Base:  EXCH$DIRE_CODE + 05D0
```

```
;  900          0985  1 GLOBAL ROUTINE dire_rt11_directory (volb : $ref_bblock) =       %SBTTL 'dire_rt11_directory'
;  901          0986  2 BEGIN
;  902          0987  2 !++
;  903          0988  2 !
;  904          0989  2 ! FUNCTIONAL DESCRIPTION:
;  905          0990  2 !
;  906          0991  2 !       Traverse the directory and display the contents
;  907          0992  2 !
;  908          0993  2 ! INPUTS:
;  909          0994  2 !
;  910          0995  2 !       volb - pointer to volb which has been connected to the RT-11 device
;  911          0996  2 !
;  912          0997  2 ! IMPLICIT INPUTS:
;  913          0998  2 !
;  914          0999  2 !       dire - directory work area
;  915          1000  2 !
;  916          1001  2 ! OUTPUTS:
;  917          1002  2 !                                                                                                          ; R
;  918          1003  2 !       none
;  919          1004  2 !
;  920          1005  2 ! IMPLICIT OUTPUTS:
;  921          1006  2 !
;  922          1007  2 !       none
;  923          1008  2 !
;  924          1009  2 ! ROUTINE VALUE:
;  925          1010  2 !
;  926          1011  2 !       true if valid, false if not
;  927          1012  2 !
;  928          1013  2 ! SIDE EFFECTS:
;  929          1014  2 !
;  930          1015  2 !       error conditions will be signaled
;  931          1016  2 !--
;  932          1017  2
;  933          1018  2 $dbgtrc_prefix ('dire_rt11_directory> ');
;  934          1019  2
;  935          1020  2 LOCAL
;  936          1021  2     ent : $ref_bblock,                          ! a pointer to the current directory entry
;  937          1022  2     nam : $ref_bblock,                          ! a pointer to the namb block
;  938          1023  2     ctx : $ref_bblock,                          ! a pointer to an RT-11 context block
;  939          1024  2     wid,
;  940          1025  2     blocks_displayed,
;  941          1026  2     blocks_avail,
;  942          1027  2     large_avail,
;  943          1028  2     files_displayed,
;  944          1029  2     faop : $ref_bblock,                         ! pointer to an fao output
;  945          1030  2     status,
;  946          1031  2     flags
;  947          1032  2     ;
;  948          1033  2
;  949          1034  2 BIND
;  950          1035  2     dire = exch$a_gbl [excg$a_dire_work]        : $ref_bblock,  ! pointer to work area
;  951          1036  2     fab  = dire [dire$a_outfab]         : $ref_bblock,  ! pointer to fab for output file
;  952          1037  2     rab  = dire [dire$a_outrab]         : $ref_bblock,  ! pointer to rab for output file
;  953          1038  2     rtv  = volb [volb$a_vfmt_specific]  : $ref_bblock   ! pointer to rt-11 specific information
;  954          1039  2     ;
```

```
;  956          1040  2 $block_check (2, .dire, dire, 423);                ! Check that the directory work area is valid
;  957          1041  2 $block_check (2, .volb, volb, 424);                ! Check the input parameter to make sure it is a volb
;  958          1042  2 $block_check (2, .rtv,  rt11, 430);                ! And make sure it is an rt-11 volb
;  959          1043  2
;  960        L 1044  2 %IF switch_check                                     ! If checking is on look at all the nambs
;  961          1045  2 %THEN
;  962          1046  2     nam = .volb [volb$a_namb_head];                   ! Get the address of the first namb chained to the volb
;  963          1047  2     $logic_check (2, (.nam NEQ 0), 109);
;  964          1048  2     WHILE .nam NEQ 0
;  965          1049  2     DO
;  966          1050  3         BEGIN
;  967          1051  3         $block_check (2, .nam, namb, 425);
;  968          1052  3         nam = .nam [namb$a_next];
;  969          1053  2         END;
;  970          1054  2 %FI
;  971          1055  2
;  972          1056  2 ! Determine the width of a directory item
;  973          1057  2 !
;  974          1058  2 wid = rt11ctx$s_exp_fullname + rt11ctx$s_exp_protected;  ! "P " and name are always there
;  975          1059  2 IF .dire [dire$v_q_size]                               ! Is /SIZE present?
;  976          1060  2 THEN
;  977          1061  2     wid = .wid + (IF .dire [dire$v_q_octal] THEN 7 ELSE 5);
;  978          1062  2 IF .dire [dire$v_q_date]                               !  or /DATE?
;  979          1063  2 THEN
;  980          1064  2     wid = .wid + 12;
;  981          1065  2 IF .dire [dire$v_q_blocks]                             !  or /BLOCKS?
;  982          1066  2 THEN
;  983          1067  2     wid = .wid + (IF .dire [dire$v_q_octal] THEN 7 ELSE 6);
;  984          1068  2 dire [dire$l_item_width] = .wid;                       ! Put the value in the work area
;  985          1069  2
;  986          1070  2 exch$dire_get_columns ();                             ! Fetch the number of columns
;  987          1071  2
;  988          1072  2 ! Print the header information
;  989          1073  2 !
;  990          1074  2 exch$dire_put (null_string);                          ! Print a blank line
;  991          1075  2 faop = exch$util_fao_buffer (%ASCID 'Directory of RT-11 volume !AF      !%D',   ! Format the header line
;  992          1076  2                 .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident], 0);
;  993          1077  2 faop [dsc$w_length] = .faop [dsc$w_length] - 6;        ! Remove the seconds field from the date-tim
;  994          1078  2 exch$dire_put (.faop);                                 ! Print the header line
;  995          1079  2 IF .volb [volb$v_virtual]                             ! If virtual, then give the file name too
;  996          1080  2 THEN
;  997          1081  3     BEGIN
;  998          1082  3     BIND
;  999          1083  3         nam = .volb [volb$a_nam] : $bblock;                          ! RMS nam block for the open
; 1000          1084  3     faop = exch$util_fao_buffer (%ASCID '                  using !AF',  ! Format the header line
; 1001          1085  3                     .nam [nam$b_rsl], .nam [nam$l_rsa]);
; 1002          1086  3     exch$dire_put (.faop);                                       ! Print the header line
; 1003          1087  2     END;
; 1004          1088  2 IF .dire [dire$v_q_owner]
; 1005          1089  2 THEN
; 1006          1090  3     BEGIN
; 1007          1091  3     faop = exch$util_fao_buffer (
; 1008          1092  3                 %ASCID 'Volume ID: !AF   Volume Owner: !AF   System ID: !AF',  ! FAO string
; 1009          1093  3                 rt11$s_volume_id, rtv [rt11$t_volume_id],                    ! Volume id string
; 1010          1094  3                 rt11$s_owner_name, rtv [rt11$t_owner_name],                  ! Owner name string
; 1011          1095  3                 rt11$s_system_id, rtv [rt11$t_system_id]);                   ! System id string
; 1012          1096  3     exch$dire_put (.faop);                                       ! Print the id line
```

```
;  1013          1097  2     END;
;  1014          1098  2 exch$dire_put (null_string);                                    ! And print another blank li
;  1015          1099  2
;  1016          1100  2 ! If we are doing a summary, call the summary routine
;  1017          1101  2 !
;  1018          1102  2 IF .dire [dire$v_q_summary]
;  1019          1103  2 THEN
;  1020          1104  2     RETURN dire_rt11_summary (.volb);
;  1021          1105  2
;  1022          1106  2 ! Initialize the local variables
;  1023          1107  2 !
;  1024          1108  2 status = true;                                  ! Assume that we will find some files
;  1025          1109  2 blocks_displayed = 0;
;  1026          1110  2 blocks_avail = 0;
;  1027          1111  2 large_avail = 0;
;  1028          1112  2 files_displayed = 0;
;  1029          1113  2
;  1030          1114  2 ! Purchase an RT11 context block
;  1031          1115  2 !
;  1032          1116  2 ctx = exch$util_rt11ctx_allocate (.volb, 0);
;  1033          1117  2
;  1034          1118  2 ! Loop through all the segments in the directory
;  1035          1119  2 !
;  1036          1120  2 flags = rtnxt$m_permanent OR rtnxt$m_empty OR rtnxt$m_tentative OR rtnxt$m_unknown OR rtnxt$m_skip_check;
;  1037          1121  3 WHILE ((ent = exch$rtacp_next_entry (.ctx, .flags)) NEQ 0)
;  1038          1122  2 DO
;  1039          1123  3     BEGIN
;  1040          1124  3
;  1041          1125  3     ! Print this file if that is what is called for
;  1042          1126  3     !
;  1043          1127  3     CASE .ent [rt11ent$v_type] FROM 0 TO rt11ent$m_typ_end_segment OF
;  1044          1128  3     SET
;  1045          1129  3
;  1046          1130  3     [rt11ent$m_typ_tentative, rt11ent$m_typ_empty] :
;  1047          1131  4             BEGIN
;  1048          1132  4
;  1049          1133  4             blocks_avail = .blocks_avail + .ent [rt11ent$w_blocks];          ! Update our statistics
;  1050          1134  4             IF .ent [rt11ent$w_blocks] GTRU .large_avail
;  1051          1135  4             THEN
;  1052          1136  4                 large_avail = .ent [rt11ent$w_blocks];
;  1053          1137  4
;  1054          1138  4             IF .dire [dire$v_q_deleted]                          ! For /DELETED we display whatever
;  1055          1139  4             THEN                                                 !  filename is in the entry
;  1056          1140  5                 BEGIN
;  1057          1141  5
;  1058          1142  5                 IF .ctx [rt11ctx$l_exp_fullname_len] EQL 1       ! If we have a namelen of 1,
;  1059          1143  5                 THEN                                            !  it is the period alone,
;  1060          1144  6                     BEGIN                                       !  these with "< noname >".
;  1061          1145  6                     ctx [rt11ctx$l_exp_fullname_len] = 10;
;  1062          1146  6                     CH$MOVE (10, UPLIT BYTE ('< noname >'), ctx [rt11ctx$t_exp_fullname]);
;  1063          1147  5                     END;
;  1064          1148  5
;  1065          1149  5                 dire_rt11_put_item (.ctx);                       ! print the item
;  1066          1150  5                 files_displayed = .files_displayed + 1;          ! update trailer statistics
;  1067          1151  5                 blocks_displayed = .blocks_displayed + .ent [rt11ent$w_blocks];
;  1068          1152  5
;  1069          1153  5                 END
```

```
;  1070          1154  5
;  1071          1155  4              ELSE IF .dire [dire$v_q_free] OR .dire [dire$v_q_all]                ! For /FREE or /ALL we list
;  1072          1156  4              THEN                                                                !  file as "< unused >".
;  1073          1157  5                  BEGIN
;  1074          1158  5                  ctx [rt11ctx$l_exp_fullname_len] = 10;
;  1075          1159  5                  CH$MOVE (10, UPLIT BYTE ('< unused >'), ctx [rt11ctx$t_exp_fullname]);
;  1076          1160  5                  dire_rt11_put_item (.ctx);
;  1077          1161  5                  files_displayed = .files_displayed + 1;                         ! update trailer statistics
;  1078          1162  5                  blocks_displayed = .blocks_displayed + .ent [rt11ent$w_blocks];
;  1079          1163  4                  END;
;  1080          1164  3              END;
;  1081          1165  3
;  1082          1166  3      [rt11ent$m_typ_permanent] :
;  1083          1167  4              BEGIN
;  1084          1168  4
;  1085          1169  6              IF    ((NOT .dire [dire$v_q_free])                                  ! Don't print permanent file
;  1086          1170  5                 AND                                                            !  if we've seen /FREE or /D
;  1087          1171  5                     (NOT .dire [dire$v_q_deleted]))
;  1088          1172  4              THEN
;  1089          1173  5                  BEGIN
;  1090          1174  5
;  1091          1175  5                  nam = .volb [volb$a_namb_head];                                 ! Get first namb
;  1092          1176  5                  WHILE .nam NEQ 0                                                !  and loop through them all
;  1093          1177  5                  DO
;  1094          1178  6                      BEGIN
;  1095          1179  6                      BIND                                                        ! Need the name and type fro
;  1096          1180  6                          nam_nam = nam [namb$q_name] : $desc_block,              !  command input because we
;  1097          1181  6                          nam_typ = nam [namb$q_type] : $desc_block;              !  to compare this name agai
;  1098          1182  6                                                                                  !  any selections in the com
;  1099          1183  6                      IF exch$cmd_match_filename
;  1100          1184  6                              (.ctx [rt11ctx$l_exp_fullname_len], ctx [rt11ctx$t_exp_fullname],
;  1101          1185  6                              .nam_nam [dsc$w_length] + .nam_typ [dsc$w_length], .nam_nam [dsc$a_pointer])
;  1102          1186  6                      THEN
;  1103          1187  7                          BEGIN
;  1104          1188  7                          dire_rt11_put_item (.ctx);
;  1105          1189  7                          files_displayed = .files_displayed + 1;                 ! update trailer statistics
;  1106          1190  7                          blocks_displayed = .blocks_displayed + .ent [rt11ent$w_blocks];
;  1107          1191  7                          EXITLOOP;
;  1108          1192  6                          END;
;  1109          1193  6                      nam = .nam [namb$a_next];                                   ! to next namb in chain
;  1110          1194  5                      END;
;  1111          1195  4                  END;
;  1112          1196  3              END;
;  1113          1197  3
;  1114          1198  3      [INRANGE, OUTRANGE] :
;  1115          1199  3              ;
;  1116          1200  3
;  1117          1201  3      TES;
;  1118          1202  3
;  1119          1203  3      ! Leave the loop if control/c has been hit
;  1120          1204  3      !
;  1121          1205  3      IF .exch$a_gbl [excg$v_control_c]
;  1122          1206  3      THEN
;  1123          1207  3          EXITLOOP;
;  1124          1208  2      END;
;  1125          1209  2
;  1126          1210  2 ! If we aborted, make the signal
```

```
;  1127          1211  2 !
;  1128          1212  2 IF .exch$a_gbl [excg$v_control_c]
;  1129          1213  2 THEN
;  1130          1214  2     $exch_signal ($info_stat_copy (exch$_canceled))
;  1131          1215  2
;  1132          1216  2 ! Otherwise, finish the directory
;  1133          1217  2 !
;  1134          1218  2 ELSE
;  1135          1219  3     BEGIN
;  1136          1220  3
;  1137          1221  3     IF .dire [dire$l_cur_column] NEQ 0          ! If anything is waiting, print it
;  1138          1222  3     THEN
;  1139          1223  3         dire_rt11_put_item (0);
;  1140          1224  3
;  1141          1225  3     ! If we printed some items, then add an extra blank line
;  1142          1226  3     !
;  1143          1227  3     IF .dire [dire$v_items_printed]
;  1144          1228  3     THEN
;  1145          1229  3         exch$dire_put (null_string, true)               ! Cancel CTRL/O with second parm
;  1146          1230  3
;  1147          1231  3     ! Nothing was printed, return a warning status so that command procedures can see whether files were fou
;  1148          1232  3     !
;  1149          1233  3     ELSE
;  1150          1234  3         status = $info_stat_copy (rms$_fnf);    ! Let a guy see that $ EXC DIR CS1:DEFBOO.CMD didn't work
;  1151          1235  3
;  1152          1236  3     ! Print the statistics line
;  1153          1237  3     !
;  1154          1238  4     faop = (IF .dire [dire$v_q_octal]
;  1155          1239  4             THEN
;  1156          1240  4                 %ASCID 'Total of !OW file!%S, !OW block!%S.  Free space !OW block!%S, largest !OW (octal).'
;  1157          1241  4             ELSE
;  1158          1242  3                 %ASCID 'Total of !UL file!%S, !UL block!%S.  Free space !UL block!%S, largest !UL.');
;  1159          1243  3     faop = exch$util_fao_buffer (.faop, .files_displayed, .blocks_displayed, .blocks_avail, .large_avail);
;  1160          1244  3     exch$dire_put (.faop);
;  1161          1245  2     END;
;  1162          1246  2
;  1163          1247  2 ! Free up the context block.
;  1164          1248  2 !
;  1165          1249  2 exch$util_rt11ctx_release (.ctx);
;  1166          1250  2
;  1167          1251  2 RETURN .status;
;  1168          1252  1 END;


                                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

54  52  20  66  6F  20  79  72  6F  74  63  65  72  69  44  000FC P.ABK:   .ASCII  \Directory of RT-11 volume !AF      !%D-     ;
20  46  41  21  20  65  6D  75  6C  6F  76  20  31  31  2D  0010B                 \<0>                                           ;
                        00  44  25  21  20  20  20  20  20  0011A                                                            ;
                                                    00      00123         .ASCII  <0>                                        ;
                                                            00124 P.ABJ:    .LONG   17694758                                   ;
                                                            00128           .ADDRESS P.ABK                                     ;
20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  0012C P.ABM:    .ASCII  \                    using !AF\<0><0>      ;
00  46  41  21  20  67  6E  69  73  75  20  20  20  20  20  0013B                                                            ;
                                                    00      0014A                                                          ;
                                                    00      0014B         .ASCII  <0>                                        ;
```

```
                                                    010E001D  0014C P.ABL:   .LONG   17694749                                                ;
                                                    00000000'  00150          .ADDRESS P.ABM                                                ;
20 46 41 21 20 3A 44 49 20 65 6D 75 6C 6F 56  00154 P.ABO:   .ASCII  \Volume ID: !AF   Volume Owner: !AF   Sys\                          ;
3A 72 65 6E 77 4F 20 65 6D 75 6C 6F 56 20 20  00163                                                                                       ;
                  73 79 53 20 20 20 46 41 21 20  00172                                                                                    ;
               00 46 41 21 20 3A 44 49 20 6D 65 74  0017C          .ASCII  \tem ID: !AF\<0>                                               ;
                                                    010E0033  00188 P.ABN:   .LONG   17694771                                             ;
                                                    00000000'  0018C          .ADDRESS P.ABO                                              ;
                  3E 20 65 6D 61 6E 6F 6E 20 3C  00190 P.ABP:   .ASCII  \< noname >\                                                      ;
                  3E 20 64 65 73 75 6E 75 20 3C  0019A P.ABQ:   .ASCII  \< unused >\                                                      ;
69 66 20 57 4F 21 20 66 6F 20 6C 61 74 6F 54  001A4 P.ABS:   .ASCII  \Total of !OW file!%S, !OW block!%S.  Fre\                          ;
63 6F 6C 62 20 57 4F 21 20 2C 53 25 21 65 6C  001B3                                                                                       ;
                  65 72 46 20 20 2E 53 25 21 6B  001C2                                                                                    ;
6F 6C  2 20 57 4F 21 20 65 63 61 70 73 20 65  001CC          .ASCII  \e space !OW block!%S, largest !OW (octal\                          ;
20 74 73 65 67 72 61 6C 20 2C 53 25 21 6B 63  001DB                                                                                       ;
                  6C 61 74 63 6F 28 20 57 4F 21  001EA                                                                                    ;
                              00 00 2E 29  001F4          .ASCII  \).\<0><0>                                                            ;
                                                    010E0052  001F8 P.ABR:   .LONG   17694802                                             ;
                                                    00000000'  001FC          .ADDRESS P.ABS                                              ;
69 66 20 4C 55 21 20 66 6F 20 6C 61 74 6F 54  00200 P.ABU:   .ASCII  \Total of !UL file!%S, !UL block!%S.  Fre\                          ;
63 6F 6C 62 20 4C 55 21 20 2C 53 25 21 65 6C  0020F                                                                                       ;
                  65 72 46 20 20 2E 53 25 21 6B  0021E                                                                                    ;
6F 6C 62 20 4C 55 21 20 65 63 61 70 73 20 65  00228          .ASCII  \e space !UL block!%S, largest !UL.\<0>                             ;
20 74 73 65 67 72 61 6C 20 2C 53 25 21 6B 63  00237                                                                                       ;
                              00 2E 4C 55 21  00246                                                                                       ;
                                    00  0024B          .ASCII  <0>                                                                    ;
                                                    010E004A  0024C P.ABT:   .LONG   17694794                                             ;
                                                    00000000'  00250          .ADDRESS P.ABU                                              ;

                                                              .EXTRN  EXCH$_CANCELED

                                                              .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                                 0FFC 00000   .ENTRY  DIRE_RT11_DIRECTORY, Save R2,R3,R4,R5,R6,-                       ; 0985
                                                              R7,R8,R9,R10,R11                                                            ;
                  5E                        20  C2 00002   SUBL2   #32, SP                                                             ;
50 00000000G      EF                        0C  C1 00005   ADDL3   #12, EXCH$A_GBL, R0                                                ; 1035
                  5B                        60  D0 0000D   MOVL    (R0), R11                                                           ; 1036
                  57                  04    AC  D0 00010   MOVL    VOLB, R7                                                            ; 1038
                  52 024400FE         8F    D0 00014   MOVL    #38011134, R2                                                       ; 1040
                  51      01A7        8F    3C 0001B   MOVZWL  #423, R1                                                            ;
                  50                  5B    D0 00020   MOVL    R11, R0                                                             ;
                     00000000G        EF    16 00023   JSB     EXCH$UTIL_BLOCK_CHECK                                               ;
                  52 041B00F3         8F    D0 00029   MOVL    #68878579, R2                                                       ; 1041
                  51      01A8        8F    3C 00030   MOVZWL  #424, R1                                                            ;
                  50                  57    D0 00035   MOVL    R7, R0                                                              ;
                     00000000G        EF    16 00038   JSB     EXCH$UTIL_BLOCK_CHECK                                               ;
                  52 880E00F5         8F    D0 0003E   MOVL    #-2012348171, R2                                                    ; 1042
                  51      01AE        8F    3C 00045   MOVZWL  #430, R1                                                            ;
                  50            54    A7    D0 0004A   MOVL    84(R7), R0                                                          ;
                     00000000G        EF    16 0004E   JSB     EXCH$UTIL_BLOCK_CHECK                                               ;
                  5A            4C    A7    D0 00054   MOVL    76(R7), NAM                                                         ; 1046
                                      13    12 00058   BNEQ    1$                                                                  ; 1047
                  7E            6D    8F    9A 0005A   MOVZBL  #109, -(SP)                                                         ;
                                      01    DD 0005E   PUSHL   #1                                                                  ;
                     00000000G        8F    DD 00060   PUSHL   #EXCH$_BADLOGIC                                                     ;
   00000000G      00                  03    FB 00066   CALLS   #3, LIB$STOP                                                        ;
```

```
                                5A  D5 0006D 1$:     TSTL    NAM                                  ; 1048
                                1B  13 0006F         BEQL    2$                                   ;
              52 010A00F7       8F  D0 00071         MOVL    #17432823, R2                        ; 1051
              51     01A9       8F  3C 00078         MOVZWL  #425, R1                             ;
              50                5A  D0 0007D         MOVL    NAM, R0                              ;
               00000000G        EF  16 00080         JSB     EXCH$UTIL_BLOCK_CHECK                ;
              5A       70       AA  D0 00086         MOVL    112(NAM), NAM                        ; 1052
                                E1  11 0008A         BRB     1$                                   ; 1048
              51                0C  D0 0008C 2$:     MOVL    #12, WID                             ; 1058
              6E       2C       AB  9E 0008F         MOVAB   44(R11), (SP)                        ; 1059
      10      00  BE                0B  E1 00093         BBC     #11, @0(SP), 5$                      ;
                       00       BE  95 00098         TSTB    @0(SP)                               ; 1061
                                05  18 0009B         BGEQ    3$                                   ;
              50                07  D0 0009D         MOVL    #7, R0                               ;
                                03  11 000A0         BRB     4$                                   ;
              50                05  D0 000A2 3$:     MOVL    #5, R0                               ;
              51                50  C0 000A5 4$:     ADDL2   R0, WID                              ;
      03      00  BE                03  E1 000A8 5$:     BBC     #3, @0(SP), 6$                       ; 1062
              51                0C  C0 000AD         ADDL2   #12, WID                             ; 1064
      10      00  BE                01  E1 000B0 6$:     BBC     #1, @0(SP), 9$                       ; 1065
                       00       BE  95 000B5         TSTB    @0(SP)                               ; 1067
                                05  18 000B8         BGEQ    7$                                   ;
              50                07  D0 000BA         MOVL    #7, R0                               ;
                                03  11 000BD         BRB     8$                                   ;
              50                06  D0 000BF 7$:     MOVL    #6, R0                               ;
              51                50  C0 000C2 8$:     ADDL2   R0, WID                              ;
           34 AB                51  D0 000C5 9$:     MOVL    WID, 52(R11)                         ; 1068
         FAD4 CF                00  FB 000C9         CALLS   #0, EXCH$DIRE_GET_COLUMNS            ; 1070
                     0000'      CF  9F 000CE         PUSHAB  NULL_STRING                          ; 1074
         FEB8 CF                01  FB 000D2         CALLS   #1, EXCH$DIRE_PUT                    ;
                                7E  D4 000D7         CLRL    -(SP)                                ; 1076
                       69       A7  9F 000D9         PUSHAB  105(R7)                              ;
                       65       A7  DD 000DC         PUSHL   101(R7)                              ;
                     0000'      CF  9F 000DF         PUSHAB  P.ABJ                                ; 1075
    00000000G EF                04  FB 000E3         CALLS   #4, EXCH$UTIL_FAO_BUFFER             ; 1076
           04 AE                50  D0 000EA         MOVL    R0, FAOP                             ;
           04 BE                06  A2 000EF         SUBW2   #6, @FAOP                            ; 1077
                       04       AE  DD 000F2         PUSHL   FAOP                                 ; 1078
         FE95 CF                01  FB 000F5         CALLS   #1, EXCH$DIRE_PUT                    ;
      22   48 A7                04  E1 000FA         BBC     #4, 72(R7), 10$                      ; 1079
           50          18       A7  D0 000FF         MOVL    24(R7), R0                           ; 1083
                       04       A0  DD 00103         PUSHL   4(R0)                                ; 1085
           7E          03       A0  9A 00106         MOVZBL  3(R0), -(SP)                         ;
                     0000'      CF  9F 0010A         PUSHAB  P.ABL                                ; 1084
    00000000G EF                03  FB 0010E         CALLS   #3, EXCH$UTIL_FAO_BUFFER             ;
           04 AE                50  D0 00115         MOVL    R0, FAOP                             ;
                       04       AE  DD 00119         PUSHL   FAOP                                 ; 1086
         FE6E CF                01  FB 0011C         CALLS   #1, EXCH$DIRE_PUT                    ;
      38   00 BE                09  E1 00121 10$:    BBC     #9, @0(SP), 11$                      ; 1088
      7E   54 A7 000003FE       8F  C1 00126         ADDL3   #1022, 84(R7), -(SP)                 ; 1095
                                0C  DD 0012F         PUSHL   #12                                  ;
      7E   54 A7 000003F2       8F  C1 00131         ADDL3   #1010, 84(R7), -(SP)                 ; 1094
                                0C  DD 0013A         PUSHL   #12                                  ; 1095
      7E   54 A7 000003E6       8F  C1 0013C         ADDL3   #998, 84(R7), -(SP)                  ; 1093
                                0C  DD 00145         PUSHL   #12                                  ; 1095
                     0000'      CF  9F 00147         PUSHAB  P.ABN                                ; 1091
    00000000G EF                07  FB 0014B         CALLS   #7, EXCH$UTIL_FAO_BUFFER             ; 1095
```

```
                     04   AE          50  D0 00152       MOVL    R0, FAOP                          ;
                                04    AE  DD 00156       PUSHL   FAOP                              ; 1096
                   FE31   CF          01  FB 00159       CALLS   #1, EXCH$DIRE_PUT                 ;
                              0000'   CF  9F 0015E 11$:  PUSHAB  NULL_STRING                       ; 1098
                   FE28   CF          01  FB 00162       CALLS   #1, EXCH$DIRE_PUT                 ;
           08        00   BE          0C  E1 00167       BBC     #12, @0(SP), 12$                  ; 1102
                                      57  DD 0016C       PUSHL   R7                                ; 1104
                  0000V   CF          01  FB 0016E       CALLS   #1, DIRE_RT11_SUMMARY             ;
                                          04 00173       RET                                       ;
                                08    AE  7C 00174 12$:  CLRQ    FILES_DISPLAYED                   ; 1112
                     14   AE          01  7D 00177       MOVQ    #1, STATUS                        ; 1108
                                10    AE  D4 0017B       CLRL    LARGE_AVAIL                       ; 1111
                                      7E  D4 0017E       CLRL    -(SP)                             ; 1116
                                      57  DD 00180       PUSHL   R7                                ;
              00000000G   EF          02  FB 00182       CALLS   #2, EXCH$UTIL_RT11CTX_ALLOCATE    ;
                          56          50  D0 00189       MOVL    R0, CTX                           ;
                     1C   AE          1F  D0 0018C       MOVL    #31, FLAGS                        ; 1120
                                1C    AE  DD 00190 13$:  PUSHL   FLAGS                             ; 1121
                                      56  DD 00193       PUSHL   CTX                               ;
              00000000G   EF          02  FB 00195       CALLS   #2, EXCH$RTACP_NEXT_ENTRY         ;
                          58          50  D0 0019C       MOVL    R0, ENT                           ;
                                      03  12 0019F       BNEQ    14$                               ;
                                    00C6  31 001A1       BRW     26$                               ;
    7E        01  A8          04      00  EF 001A4 14$:  EXTZV   #0, #4, 1(ENT), -(SP)             ; 1127
                  08          00      8E  CF 001AA       CASEL   (SP)+, #0, #8                     ;
  00B2          0014        0014    00B2     001AE 15$:  .WORD   25$-15$,-                         ;
  00B2          00B2        00B2    0062     001B6               16$-15$,-                         ;
                                    00B2     001BE               16$-15$,-                         ;
                                                                 25$-15$,-                         ;
                                                                 22$-15$,-                         ;
                                                                 25$-15$,-                         ;
                                                                 25$-15$,-                         ;
                                                                 25$-15$,-                         ;
                                                                 25$-15$                           ;
                                      4C  11 001C0       BRB     21$                               ;
                          59    08    A8  3C 001C2 16$:  MOVZWL  8(ENT), R9                        ; 1133
                     18   AE          59  C0 001C6       ADDL2   R9, BLOCKS_AVAIL                  ;
                     10   AE          59  D1 001CA       CMPL    R9, LARGE_AVAIL                   ; 1134
                                      04  1B 001CE       BLEQU   17$                               ;
                     10   AE          59  D0 001D0       MOVL    R9, LARGE_AVAIL                   ; 1136
           13        00   BE          04  E1 001D4 17$:  BBC     #4, @0(SP), 18$                   ; 1138
                          01    46    A6  D1 001D9       CMPL    70(CTX), #1                       ; 1142
                                      21  12 001DD       BNEQ    20$                               ;
                     46   A6          0A  D0 001DF       MOVL    #10, 70(CTX)                      ; 1145
    54        A6      0000'   CF          0A  28 001E3       MOVC3   #10, P.ABP, 84(CTX)           ; 1146
                                      14  11 001EA       BRB     20$                               ; 1149
           04        00   BE          05  E0 001EC 18$:  BBS     #5, @0(SP), 19$                   ; 1155
                          6B    00    BE  E9 001F1       BLBC    @0(SP), 25$                       ;
                     46   A6          0A  D0 001F5 19$:  MOVL    #10, 70(CTX)                      ; 1158
    54        A6      0000'   CF          0A  28 001F9       MOVC3   #10, P.ABQ, 84(CTX)           ; 1159
                                      56  DD 00200 20$:  PUSHL   CTX                               ; 1160
                  0000V   CF          01  FB 00202       CALLS   #1, DIRE_RT11_PUT_ITEM            ;
                                08    AE  D6 00207       INCL    FILES_DISPLAYED                   ; 1161
                     0C   AE          59  C0 0020A       ADDL2   R9, BLOCKS_DISPLAYED              ; 1162
                                      50  11 0020D 21$:  BRB     25$                               ; 1127
           4B        00   BE          05  E0 00210 22$:  BBS     #5, @0(SP), 25$                   ; 1169
           46        00   BE          04  E0 00215       BBS     #4, @0(SP), 25$                   ; 1171
```

```
                                5A        4C   A7  D0 0021A          MOVL    76(R7), NAM                                    ; 1175
                                52        54   A6  9E 0021E          MOVAB   84(R6), R2                                     ; 1184
                                                 5A  D5 00222 23$:    TSTL    NAM                                            ; 1176
                                                 3A  13 00224         BEQL    25$
                                50        50     AA  9E 00226         MOVAB   80(NAM), R0                                    ; 1180
                                          04     A0  DD 0022A         PUSHL   4(R0)                                          ; 1185
                                51                60  3C 0022D         MOVZWL  (R0), R1
                                53        58     AA  3C 00230         MOVZWL  88(NAM), R3
                                               6341  9F 00234         PUSHAB  (R3)[R1]
                                                 52  DD 00237         PUSHL   R2                                             ; 1184
                                          46     A6  DD 00239         PUSHL   70(CTX)
                                00000000G EF           04  FB 0023C         CALLS   #4, EXCH$CMD_MATCH_FILENAME
                                14                     50  E9 00243         BLBC    R0, 24$
                                                       56  DD 00246         PUSHL   CTX                                    ; 1188
                                   0000V  CF           01  FB 00248         CALLS   #1, DIRE_RT11_PUT_ITEM
                                          08     AE   D6  00024D         INCL    FILES_DISPLAYED                        ; 1189
                                50        08     A8  3C 00250         MOVZWL  8(ENT), R0                             ; 1190
                                   0C     AE           50  C0 00254         ADDL2   R0, BLOCKS_DISPLAYED
                                                       06  11 00258         BRB     25$                                    ; 1187
                                5A        70     AA  D0 0025A 24$:    MOVL    112(NAM), NAM                          ; 1193
                                                 C2  11 0025E         BRB     23$                                    ; 1176
                                03 00000000G     FF  E8 00260 25$:    BLBS    @EXCH$A_G3L, 26$                       ; 1205
                                               FF26  31 00267         BRW     13$
                                17 00000000G     FF  E9 0026A 26$:    BLBC    @EXCH$A_GBL, 27$                       ; 1212
                                50 00000000G     8F  D0 00271         MOVL    #EXCH$_CANCELED, STATUS2               ; 1214
                50              03               00  03  F0 00278         INSV    #3, #0, #3, STATUS2
                                                     50  DD 0027D         PUSHL   STATUS2
                                   00000000G     00  01  FB 0027F         CALLS   #1, LIB$SIGNAL
                                                 63  11 00286         BRB     33$
                                          40     AB  D5 00288 27$:    TSTL    64(R11)                                ; 1221
                                                 07  13 0028B         BEQL    28$
                                                 7E  D4 0028D         CLRL    -(SP)                                  ; 1223
                                   0000V  CF           01  FB 0028F         CALLS   #1, DIRE_RT11_PUT_ITEM
                                ;0D       30     AB  01  E1 00294 28$:    BBC     #1, 48(RT1), 29$                       ; 1227
                                                 01  DD 00299         PUSHL   #1                                     ; 1229
                                          0000'  CF  9F 0029B         PUSHAB  NULL_STRING
                                   FCEB   CF           02  FB 0029F         CALLS   #2, EXCH$DIRE_PUT
                                                       10  11 002A4         BRB     30$
                                50 00018292      8F  D0 002A6 29$:    MOVL    #98962, STATUS2                        ; 1234
                50              03               00  03  F0 002AD         INSV    #3, #0, #3, STATUS2
                                   14     AE           50  D0 002B2         MOVL    STATUS2, STATUS
                                          00     BE  95  002B6 30$:    TSTB    @0(SP)                                 ; 1238
                                                 08  18  002B9         BGEQ    31$
                                   04     AE   0000'  CF  9E 002BB         MOVAB   P.ABR, FAOP                            ; 1239
                                                 06  11 002C1         BRB     32$
                                   04     AE   0000'  CF  9E 002C3 31$:    MOVAB   P.ABT, FAOP                            ; 1241
                                          10     AE  DD 002C9 32$:    PUSHL   LARGE_AVAIL                            ; 1243
                                          1C     AE  DD 002CC         PUSHL   BLOCKS_AVAIL
                                          14     AE  DD 002CF         PUSHL   BLOCKS_DISPLAYED
                                          14     AE  DD 002D2         PUSHL   FILES_DISPLAYED
                                          14     AE  DD 002D5         PUSHL   FAOP
                                   00000000G     EF  05  FB 002D8         CALLS   #5, EXCH$UTIL_FAO_BUFFER
                                      04     AE           50  D0 002DF         MOVL    R0, FAOP
                                          04     AE  DD 002E3         PUSHL   FAOP                                   ; 1244
                                   FCA4   CF           01  FB 002E6         CALLS   #1, EXCH$DIRE_PUT
                                                       56  DD 002EB 33$:    PUSHL   CTX                                    ; 1249
                                   00000000G     EF  01  FB 002ED         CALLS   #1, EXCH$UTIL_RT11CTX_RELEASE
```

```
                                50       14    AE  D0 002F4          MOVL    STATUS, R0                              ; 1251
                                                   04 002F8          RET                                             ; 1252

; Routine Size:  761 bytes,    Routine Base:  EXCH$DIRE_CODE + 0641
```

```
;1170        1253  1 GLOBAL ROUTINE dire_rt11_summary (volb : $ref_bblock) = %SBTTL 'dire_rt11_summary'
;1171        1254  2 BEGIN
;1172        1255  2 !++
;1173        1256  2 !
;1174        1257  2 ! FUNCTIONAL DESCRIPTION:
;1175        1258  2 !
;1176        1259  2 !       Display a summary of the directory
;1177        1260  2 !
;1178        1261  2 ! INPUTS:
;1179        1262  2 !
;1180        1263  2 !       volb - pointer to volb which has been connected to the RT-11 device
;1181        1264  2 !
;1182        1265  2 ! IMPLICIT INPUTS:
;1183        1266  2 !
;1184        1267  2 !       dire - directory work area
;1185        1268  2 !
;1186        1269  2 ! OUTPUTS:
;1187        1270  2 !
;1188        1271  2 !       none
;1189        1272  2 !
;1190        1273  2 ! IMPLICIT OUTPUTS:
;1191        1274  2 !
;1192        1275  2 !       none
;1193        1276  2 !
;1194        1277  2 ! ROUTINE VALUE:
;1195        1278  2 !
;1196        1279  2 !       true if valid, false if not
;1197        1280  2 !
;1198        1281  2 ! SIDE EFFECTS:
;1199        1282  2 !
;1200        1283  2 !       error conditions will be signaled
;1201        1284  2 !--
;1202        1285  2
;1203        1286  2 $dbgtrc_prefix ('dire_rt11_summary> ');
;1204        1287  2
;1205        1288  2 LOCAL
;1206        1289  2     seg : $ref_bblock,                         ! a pointer to the current directory segment
;1207        1290  2     ent : $ref_bblock,                         ! a pointer to the current directory entry
;1208        1291  2     ctx : $ref_bblock,                         ! a pointer to an RT-11 context block
;1209        1292  2     seg_num,
;1210        1293  2     faop : $ref_bblock,                        ! pointer to an fao output
;1211        1294  2     flags,
;1212        1295  2     max_ent,                                   ! entries per segment for summary
;1213        1296  2     empt_cnt,                                  ! count of empty entries
;1214        1297  2     perm_cnt,                                  ! count of permanent entries
;1215        1298  2     unkn_cnt,                                  ! count of unknown entries
;1216        1299  2     empt_tot,                                  ! total count of empty entries
;1217        1300  2     perm_tot,                                  ! total count of permanent entries
;1218        1301  2     unkn_tot                                   ! total count of unknown entries
;1219        1302  2     ;
;1220        1303  2
;1221        1304  2 BIND
;1222        1305  2     dire = exch$a_gbl [excg$a_dire_work]       : $ref_bblock,  ! pointer to work area
;1223        1306  2     fab  = dire [dire$a_outfab]        : $ref_bblock,  ! pointer to fab for output file
;1224        1307  2     rab  = dire [dire$a_outrab]        : $ref_bblock   ! pointer to rab for output file
;1225        1308  2     ;
```

```
;  1227          1309  2 ROUTINE seg_stat (num, perm, empt, unkn, maxe) : NOVALUE =
;  1228          1310  2 BEGIN
;  1229          1311  2 LOCAL
;  1230          1312  2     faop : $ref_bblock,                          . pointer to an fao output
;  1231          1313  2     use;
;  1232          1314  2
;  1233          1315  2 faop = (IF .num EQL 0
;  1234          1316  2         THEN
;  1235          1317  2             %ASCID '        Total for all segments:'
;  1236          1318  2         ELSE
;  1237          1319  2             exch$util_fao_buffer (%ASCID '        Segment !2UL', .num));
;  1238          1320  2 exch$dire_put (.faop, (.num EQL 0));
;  1239          1321  2
;  1240          1322  2 faop = exch$util_fao_buffer (%ASCID '                    Permanent files: !4UL', .perm);
;  1241          1323  2 exch$dire_put (.faop);
;  1242          1324  2
;  1243          1325  2 faop = exch$util_fao_buffer (%ASCID '                    Unused files:    !4UL', .empt);
;  1244          1326  2 exch$dire_put (.faop);
;  1245          1327  2                                                                                          ; R
;  1246          1328  2 IF .unkn NEQ 0
;  1247          1329  2 THEN
;  1248          1330  3     BEGIN
;  1249          1331  3     faop = exch$util_fao_buffer (%ASCID '                        Unknown entries: !4UL', .unkn);
;  1250          1332  3     exch$dire_put (.faop);
;  1251          1333  3     END;
;  1252          1334  2
;  1253          1335  2 use = .perm + .empt + .unkn;
;  1254          1336  2 faop = exch$util_fao_buffer (%ASCID '                    Entries in use:  !4UL     Entries available: !UL',
;  1255          1337  2                                 .use, .maxe - .use);
;  1256          1338  2 exch$dire_put (.faop);
;  1257          1339  2
;  1258          1340  2 exch$dire_put (null_string);            ! Print a blank line
;  1259          1341  2
;  1260          1342  2 RETURN;
;  1261          1343  1 END;


                                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

66 20 6C 61 74 6F 54 20 20 20 20 20 20 20 20 20  00254 P.ABW:  .ASCII  \        Total for all segments:\<0>
73 74 6E 65 6D 67 65 73 20 6C 6C 61 20 72 6F     00263
                                    00 3A        00272
                                 010E001F        00274 P.ABV:  .LONG   17694751
                                 00000000'       00278         .ADDRESS P.ABW
74 6E 65 6D 67 65 53 20 20 20 20 20 20 20 20 20  0027C P.ABY:  .ASCII  \        Segment !2UL\
                              4C 55 32 21 20     0028B
                                 010E0014        00290 P.ABX:  .LONG   17694740
                                 00000000'       00294         .ADDRESS P.ABY
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  00298 P.ACA:  .ASCII  \                    Permanent files: !4\
74 6E 65 6E 61 6D 72 65 50 20 20 20 20 20 20 20  002A7
                  34 21 20 3A 73 65 6C 69 66 20  002B6
                        00 00 4C 55              002C0         .ASCII  \UL\<0><0>
                                 010E002A        002C4 P.ABZ:  .LONG   17694762
                                 00000000'       002C8         .ADDRESS P.ACA
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  002CC P.ACC:  .ASCII  \                    Unused files:    !4\
69 66 20 64 65 73 75 6E 55 20 20 20 20 20 20 20  002DB
```

```
                    34 21 20 20 20 20 3A 73 65 6C  002EA
                                      00 00 4C 55  002F4          .ASCII  \UL\<0><0>                                      ;
                                        010E002A  002F8 P.ACB:    .LONG   17694762                                        ;
                                        00000000' 002FC           .ADDRESS P.ACC                                          ;
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  00300 P.ACE:    .ASCII  \                       Unknown entries: !4\    ;
65 20 6E 77 6F 6E 6B 6E 55 20 20 20 20 20 20  0030F                                                           ;
                    34 21 20 3A 73 65 69 72 74 6E  0031E                                                           ;
                                      00 00 4C 55  00328          .ASCII  \UL\<0><0>                                      ;
                                        010E002A  0032C P.ACD:    .LONG   17694762                                        ;
                                        00000000' 00330           .ADDRESS P.ACE                                          ;
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  00334 P.ACG:    .ASCII  \                       Entries in use:  !4\    ;
69 20 73 65 69 72 74 6E 45 20 20 20 20 20 20  00343                                                           ;
                    34 21 20 20 3A 65 73 75 20 6E  00352                                                           ;
20 73 65 69 72 74 6E 45 20 20 20 20 20 4C 55  0035C           .ASCII  \UL     Entries available: !UL\<0><0>          ;
00 4C 55 21 20 3A 65 6C 62 61 6C 69 61 76 61  0036B                                                           ;
                                              00  0037A                                                           ;
                                              00  0037B           .ASCII  <0>                                             ;
                                        010E0045  0037C P.ACF:    .LONG   17694789                                        ;
                                        00000000' 00380           .ADDRESS P.ACG                                          ;


                                                                  .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                            003C 00000 SEG_STAT:
                                                                  .WORD   Save R2,R3,R4,R5                                ; 1309
                       55       FC90  CF  9E 00002           MOVAB   EXCH$DIRE_PUT, R5                               ;
                       54  00000000G  EF  9E 00007           MOVAB   EXCH$UTIL_FAO_BUFFER, R4                        ;
                                      52  D4 0000E           CLRL    R2                                              ; 1315
                                04    AC  D5 00010           TSTL    NUM                                             ;
                                      09  12 00013           BNEQ    1$                                              ;
                                      52  D6 00015           INCL    R2                                              ;
                       53       0000'  CF  9E 00017           MOVAB   P.ABV, FAOP                                     ; 1316
                                      0D  11 0001C           BRB     2$                                              ;
                                04    AC  DD 0001E 1$:       PUSHL   NUM                                             ; 1319
                              0000'   CF  9F 00021           PUSHAB  P.ABX                                           ;
                       64             02  FB 00025           CALLS   #2, EXCH$UTIL_FAO_BUFFER                        ;
                       53             50  D0 00028           MOVL    R0, FAOP                                        ;
                                      52  DD 0002B 2$:       PUSHL   R2                                              ; 1320
                                      53  DD 0002D           PUSHL   FAOP                                            ;
                       65             02  FB 0002F           CALLS   #2, EXCH$DIRE_PUT                               ;
                                08    AC  DD 00032           PUSHL   PERM                                            ; 1322
                              0000'   CF  9F 00035           PUSHAB  P.ABZ                                           ;
                       64             02  FB 00039           CALLS   #2, EXCH$UTIL_FAO_BUFFER                        ;
                       53             50  D0 0003C           MOVL    R0, FAOP                                        ;
                                      53  DD 0003F           PUSHL   FAOP                                            ; 1323
                       65             01  FB 00041           CALLS   #1, EXCH$DIRE_PUT                               ;
                                0C    AC  DD 00044           PUSHL   EMPT                                            ; 1325
                              0000'   CF  9F 00047           PUSHAB  P.ACB                                           ;
                       64             02  FB 0004B           CALLS   #2, EXCH$UTIL_FAO_BUFFER                        ;
                       53             50  D0 0004E           MOVL    R0, FAOP                                        ;
                                      53  DD 00051           PUSHL   FAOP                                            ; 1326
                       65             01  FB 00053           CALLS   #1, EXCH$DIRE_PUT                               ;
                       52       10    AC  D0 00056           MOVL    UNKN, R2                                        ; 1328
                                      11  13 0005A           BEQL    3$                                              ;
                                      52  DD 0005C           PUSHL   R2                                              ; 1331
                              0000'   CF  9F 0005E           PUSHAB  P.ACD                                           ;
```

```
                                64              02  FB 00062          CALLS   #2, EXCH$UTIL_FAO_BUFFER                    ;
                                53              50  D0 00065          MOVL    R0, FAOP                                    ;
                                                53  DD 00068          PUSHL   FAOP                                        ; 1332
                                65              01  FB 0006A          CALLS   #1, EXCH$DIRE_PUT                           ;
                    50      08  AC      0C      AC  C1 0006D 3$:      ADDL3   EMPT, PERM, R0                              ; 1335
                                50              52  C0 00073          ADDL2   R2, USE                                     ;
                    7E      14  AC              50  C3 00076          SUBL3   USE, MAXE, -(SP)                            ; 1337
                                                50  DD 0007B          PUSHL   USE                                         ;
                                        0000'   CF  9F 0007D          PUSHAB  P.ACF                                       ; 1336
                                64              03  FB 00081          CALLS   #3, EXCH$UTIL_FAO_BUFFER                    ;
                                53              50  D0 00084          MOVL    R0, FAOP                                    ;
                                                53  DD 00087          PUSHL   FAOP                                        ; 1338
                                65              01  FB 00089          CALLS   #1, EXCH$DIRE_PUT                           ;
                                        0000'   CF  9F 0008C          PUSHAB  NULL_STRING                                 ; 1340
                                65              01  FB 00090          CALLS   #1, EXCH$DIRE_PUT                           ;
                                                    04 00093          RET                                                 ; 1343

; Routine Size:  148 bytes,    Routine Base:  EXCH$DIRE_CODE + 093A
```

```
;  1263          1344  2 ! Get a context block for the loop
;  1264          1345  2 !
;  1265          1346  2 ctx = exch$util_rt11ctx_allocate (.volb, 0);
;  1266          1347  2
;  1267          1348  2 ! Initialize the local variables
;  1268          1349  2 !
;  1269          1350  2 seg_num = 1;                                    ! Start with the first directory segment
;  1270          1351  2 empt_cnt = 0;
;  1271          1352  2 perm_cnt = 0;
;  1272          1353  2 unkn_cnt = 0;
;  1273          1354  2 empt_tot = 0;
;  1274          1355  2 perm_tot = 0;
;  1275          1356  2 unkn_tot = 0;
;  1276          1357  2
;  1277          1358  2 ! Need some info from the root segment, grab a pointer
;  1278          1359  2 !
;  1279          1360  2 seg = exch$rt11_dirseg_get (.volb, 1);
;  1280          1361  2 $logic_check (2, (.seg NEQ 0), 126);
;  1281          1362  2
;  1282          1363  2 ! Print the header for the summary
;  1283          1364  2 !
;  1284          1365  3 max_ent =  (rt11$k_dirseglen -                 ! Maximum number of entries is the length of entire segment
;  1285          1366  3              rt11hdr$k_length -                 !  less the length of the segment header
;  1286          1367  3              2)                                 !  less two bytes for the end-of-segment marker
;  1287          1368  2              /                                  ! Divided by
;  1288          1369  3              (rt11ent$k_length +                !  the length of an entry - the standard length
;  1289          1370  2              .seg [rt11hdr$w_extra_bytes]);     !  plus any extra bytes
;  1290          1371  2 faop = exch$util_fao_buffer (%ASCID '    Directory segments allocated: !2UL     Maximum files per segment:!4
;  1291          1372  2                                                .seg [rt11hdr$w_num_segs], .max_ent);
;  1292          1373  2 exch$dire_put (.faop);                         ! Print the header line
;  1293          1374  2 faop = exch$util_fao_buffer (%ASCID '    Directory segments in use:    !2UL     Maximum files per volume: !4
;  1294          1375  2                                                .seg [rt11hdr$w_high_seg], .max_ent * .seg [rt11hdr$w_num_se
;  1295          1376  2 exch$dire_put (.faop);                         ! Print the header line
;  1296          1377  2 exch$dire_put (null_string);                           !  and another blank line
;  1297          1378  2
;  1298          1379  2 ! Loop through all the segments in the directory
;  1299          1380  2 !
;  1300          1381  2 flags = rtnxt$m_permanent OR rtnxt$m_empty OR rtnxt$m_tentative OR rtnxt$m_unknown OR rtnxt$m_skip_check;
;  1301          1382  3 WHILE ((ent = exch$rtacp_next_entry (.ctx, .flags)) NEQ 0)
;  1302          1383  2 DO
;  1303          1384  3     BEGIN
;  1304          1385  3
;  1305          1386  3     ! If we have been canceled, leave the loop
;  1306          1387  3     !
;  1307          1388  3     IF .exch$a_gbl [excg$v_control_c]
;  1308          1389  3     THEN
;  1309          1390  3         EXITLOOP;
;  1310          1391  3
;  1311          1392  3     ! Look for the split between segments
;  1312          1393  3     !
;  1313          1394  3     IF .seg_num NEQ .ctx [rt11ctx$l_seg_number]
;  1314          1395  3     THEN
;  1315          1396  4         BEGIN
;  1316          1397  4         seg_stat (.seg_num, .perm_cnt, .empt_cnt, .unkn_cnt, .max_ent);
;  1317          1398  4         empt_cnt = 0;
;  1318          1399  4         perm_cnt = 0;
;  1319          1400  4         unkn_cnt = 0;
```

```
;  1320          1401  4          seg_num = .ctx [rt11ctx$l_seg_number];
;  1321          1402  3          END;
;  1322          1403  3
;  1323          1404  3      ! Now print this file if that is what is called for
;  1324          1405  3      !
;  1325          1406  3      CASE .ent [rt11ent$v_type] FROM 0 TO rt11ent$m_typ_end_segment OF
;  1326          1407  3      SET
;  1327          1408  3
;  1328          1409  3          [rt11ent$m_typ_tentative, rt11ent$m_typ_empty] :
;  1329          1410  4                  BEGIN
;  1330          1411  4                  empt_cnt = .empt_cnt + 1;
;  1331          1412  4                  empt_tot = .empt_tot + 1;
;  1332          1413  3                  END;
;  1333          1414  3
;  1334          1415  3          [rt11ent$m_typ_permanent] :
;  1335          1416  4                  BEGIN
;  1336          1417  4                  perm_cnt = .perm_cnt + 1;
;  1337          1418  4                  perm_tot = .perm_tot + 1;
;  1338          1419  3                  END;
;  1339          1420  3
;  1340          1421  3          [INRANGE, OUTRANGE] :
;  1341          1422  4                  BEGIN
;  1342          1423  4                  unkn_cnt = .unkn_cnt + 1;
;  1343          1424  4                  unkn_tot = .unkn_tot + 1;
;  1344          1425  3                  END;
;  1345          1426  3          TES;
;  1346          1427  2      END;
;  1347          1428  2
;  1348          1429  2  ! If we have been canceled, print the message
;  1349          1430  2  !
;  1350          1431  2  IF .exch$a_gbl [excg$v_control_c]
;  1351          1432  2  THEN
;  1352          1433  2      $exch_signal ($info_stat_copy (exch$_canceled))
;  1353          1434  2
;  1354          1435  2  ELSE
;  1355          1436  3      BEGIN
;  1356          1437  3
;  1357          1438  3      ! Print the stats for the last segment
;  1358          1439  3      !
;  1359          1440  3      seg_stat (.seg_num, .perm_cnt, .empt_cnt, .unkn_cnt, .max_ent);
;  1360          1441  3
;  1361          1442  3      ! If we have more than one segment, print the grand total
;  1362          1443  3      !
;  1363          1444  3      IF .seg_num NEQ 1                          ! Will be equal if we only have 1 segment
;  1364          1445  3      THEN
;  1365          1446  3          seg_stat (0, .perm_tot, .empt_tot, .unkn_tot, .max_ent * .seg [rt11hdr$w_num_segs]);
;  1366          1447  2      END;
;  1367          1448  2
;  1368          1449  2  ! Free up the context block.
;  1369          1450  2  !
;  1370          1451  2  exch$util_rt11ctx_release (.ctx)
;  1371          1452  2
;  1372          1453  2  RETURN true;
;  1373          1454  1  END;
```

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

73 20 79 72 6F 74 63 65 72 69 44 20 20 20 20 00384 P.ACI:  .ASCII  \    Directory segments allocated: !2UL  \
74 61 63 6F 6C 6C 61 20 73 74 6E 65 6D 67 65 00393
                  20 20 4C 55 32 21 20 3A 64 65 003A2
65 6C 69 66 20 6D 75 6D 69 78 61 4D 20 20 20 003AC        .ASCII  \   Maximum files per segment:!4UL\<0><0>
21 3A 74 6E 65 6D 67 65 73 20 72 65 70 20 73 003BB
                              00 00 4C 55 34 003CA
                                    00 003CF        .ASCII  <0>
                                 010E0049 003D0 P.ACH:  .LONG   17694793
                                 00000000' 003D4        .ADDRESS P.ACI
73 20 79 72 6F 74 63 65 72 69 44 20 20 20 20 003D8 P.ACK:  .ASCII  \    Directory segments in use:    !2UL  \
3A 65 73 75 20 6E 69 20 73 74 6E 65 6D 67 65 003E7
                  20 20 4C 55 32 21 20 20 20 20 003F6
65 6C 69 66 20 6D 75 6D 69 78 61 4D 20 20 20 00400        .ASCII  \   Maximum files per volume: !4UL\<0><0>
21 20 3A 65 6D 75 6C 6F 76 20 72 65 70 20 73 0040F
                              00 00 4C 55 34 0041E
                                    00 00423        .ASCII  <0>
                                 010E0049 00424 P.ACJ:  .LONG   17694793
                                 00000000' 00428        .ADDRESS P.ACK



                                                                        .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                       0FFC 00000       .ENTRY  DIRE_RT11_SUMMARY, Save R2,R3,R4,R5,R6,R7,- ; 1253
                                                                R8,R9,R10,R11
                                 5E     08  C2 00002       SUBL2   #8, SP
                  50 00000000G   EF     0C  C1 00005       ADDL3   #12, EXCH$A_GBL, R0                         ; 1305
                                 7E     7E  D4 00008       CLRL    -(SP)                                       ; 1346
                                        04  AC DD 0000F    PUSHL   VOLB
                      00000000G  EF     02  FB 00012       CALLS   #2, EXCH$UTIL_RT11CTX_ALLOCATE
                                 54     50  D0 00019       MOVL    R0, CTX
                                        58  7C 0001C       CLRQ    PERM_CNT                                    ; 1352
                                 56     01  7D 0001E       MOVQ    #1, SEG_NUM                                 ; 1350
                                        6E  7C 00021       CLRQ    PERM_TOT                                    ; 1355
                                        5B  D4 00023       CLRL    UNKN_TOT                                    ; 1356
                                        01  DD 00025       PUSHL   #1                                          ; 1360
                                  04    AC  DD 00027       PUSHL   VOLB
                   00000000G      EF    02  FB 0002A       CALLS   #2, EXCH$RT11_DIRSEG_GET
                                 53     50  D0 00031       MOVL    R0, SEG
                                        13  12 00034       BNEQ    1$                                          ; 1361
                                 7E      7E 8F 9A 00036    MOVZBL  #126, -(SP)
                                        01  DD 0003A       PUSHL   #1
                              00000000G 8F  DD 0003C       PUSHL   #EXCH$_BADLOGIC
                   00000000G  00     03  FB 00042       CALLS   #3, LIB$STOP
                                 50     06  A3 3C 00049 1$: MOVZWL  6(SEG), R0                                 ; 1369
                                 50     0E  C0 0004D       ADDL2   #14, R0
                  52 000003F4    8F     50  C7 00050       DIVL3   R0, #1012, MAX_ENT
                                        52  DD 00058       PUSHL   MAX_ENT                                     ; 1372
                                 7E     63  3C 0005A       MOVZWL  (SEG), -(SP)
                                  0000' CF  9F 0005D       PUSHAB  P.ACH                                       ; 1371
                   00000000G      EF    03  FB 00061       CALLS   #3, EXCH$UTIL_FAO_BUFFER
                                 55     50  D0 00068       MOVL    R0, FAOP
                                        55  DD 0006B       PUSHL   FAOP                                        ; 1373
                       FB90      CF    01  FB 0006D       CALLS   #1, EXCH$DIRE_PUT
                                 5A     63  3C 00072       MOVZWL  (SEG), R10                                  ; 1375
```

```
                              5A          52 C4 00075         MULL2   MAX_ENT, R10                          ;
                                          5A DD 00078         PUSHL   R10                                   ;
                              7E     04   A3 3C 0007A         MOVZWL  4(SEG), -(SP)                         ;
                                  0000'   CF 9F 0007E         PUSHAB  P.ACJ                                 ;
                      00000000G   EF      03 FB 00082         CALLS   #3, EXCH$UTIL_FAO_BUFFER              ; 1374
                                  55      50 D0 00089         MOVL    R0, FAOP                              ;
                                          55 DD 0008C         PUSHL   FAOP                                  ; 1376
                           FB6F   CF      01 FB 0008E         CALLS   #1, EXCH$DIRE_PUT                     ;
                                  0000'   CF 9F 00093         PUSHAB  NULL_STRING                           ; 1377
                           FB66   CF      01 FB 00097         CALLS   #1, EXCH$DIRE_PUT                     ;
                                  53      1F D0 0009C         MOVL    #31, FLAGS                            ; 1381
                                          53 DD 0009F 2$:     PUSHL   FLAGS                                 ; 1382
                                          54 DD 000A1         PUSHL   CTX                                   ;
                      00000000G   EF      02 FB 000A3         CALLS   #2, EXCH$RTACP_NEXT_ENTRY             ;
                                  55      50 D0 000AA         MOVL    R0, ENT                               ;
                                          51 13 000AD         BEQL    8$                                    ;
                              51 00000000G FF E8 000AF         BLBS    @EXCH$A_GBL, 9$                       ; 1388
                             76   A4      56 D1 000B6         CMPL    SEG_NUM, 118(CTX)                     ; 1394
                                          15 13 000BA         BEQL    3$                                    ;
                                          52 DD 000BC         PUSHL   MAX_ENT                               ; 1397
                                          57 DD 000BE         PUSHL   UNKN_CNT                              ;
                                  0340    8F BB 000C0         PUSHR   #^M<R6,R8,R9>                         ;
                           FEA3   CF      05 FB 000C4         CALLS   #5, SEG_STAT                          ;
                                          58 7C 000C9         CLRQ    PERM_CNT                              ; 1399
                                          57 D4 000CB         CLRL    UNKN_CNT                              ; 1400
                              56     76   A4 D0 000CD         MOVL    118(CTX), SEG_NUM                     ; 1401
    50       01   A5          04          00 EF 000D1 3$:     EXTZV   #0, #4, 1(ENT), R0                    ; 1406
                  08          00          50 CF 000D7         CASEL   R0, #0, #8                            ;
  0012          0018        0018        0012    000DB 4$:     .WORD   5$-4$,-                               ;
  0012          0012        0012        001F    000E3                 6$-4$,-                               ;
                                        0012    000EB                 6$-4$,-                               ;
                                                                      5$-4$,-                               ;
                                                                      7$-4$,-                               ;
                                                                      5$-4$,-                               ;
                                                                      5$-4$,-                               ;
                                                                      5$-4$,-                               ;
                                                                      5$-4$                                 ;
                                          57 D6 000ED 5$:     INCL    UNKN_CNT                              ; 1423
                                          5B D6 000EF         INCL    UNKN_TOT                              ; 1424
                                          AC 11 000F1         BRB     2$                                    ; 1406
                                          59 D6 000F3 6$:     INCL    EMPT_CNT                              ; 1411
                                     04   AE D6 000F5         INCL    EMPT_TOT                              ; 1412
                                          A5 11 000F8         BRB     2$                                    ; 1406
                                          58 D6 000FA 7$:     INCL    PERM_CNT                              ; 1417
                                          6E D6 000FC         INCL    PERM_TOT                              ; 1418
                                          9F 11 000FE         BRB     2$                                    ; 1382
                              17 00000000G FF E9 00100 8$:     BLBC    @EXCH$A_GBL, 10$                      ; 1431
                              50 00000000G 8F D0 00107 9$:     MOVL    #EXCH$_CANCELED, STATUS2              ; 1433
    50            03          00          03 F0 0010E         INSV    #3, #0, #3, STATUS2                   ;
                                          50 DD 00113         PUSHL   STATUS2                               ;
                      00000000G   00      01 FB 00115         CALLS   #1, LIB$SIGNAL                        ;
                                          23 11 0011C         BRB     11$                                   ;
                                          52 DD 0011E 10$:    PUSHL   MAX_ENT                               ; 1440
                                          57 DD 00120         PUSHL   UNKN_CNT                              ;
                                  0340    8F BB 00122         PUSHR   #^M<R6,R8,R9>                         ;
                           FE41   CF      05 FB 00126         CALLS   #5, SEG_STAT                          ;
                                  01      56 D1 0012B         CMPL    SEG_NUM, #1                           ; 1444
```

```
                                        11  13 0012E          BEQL    11$                                             ;
                                        5A  DD 00130          PUSHL   R10                                             ; 1446
                                        5B  DD 00132          PUSHL   UNKN_TOT                                        ;
                                    0C  AE  DD 00134          PUSHL   EMPT_TOT                                        ;
                                    0C  AE  DD 00137          PUSHL   PERM_TOT                                        ;
                                        7E  D4 0013A          CLRL    -(SP)                                           ;
                                FE2B    CF  05  FB 0013C      CALLS   #5, SEG_STAT                                    ;
                                        54  DD 00141 11$:     PUSHL   CTX                                             ; 1451
                            00000000G   EF  01  FB 00143      CALLS   #1, EXCH$UTIL_RT11!CTX_RELEASE                  ;
                                        50  01  DO 0014A      MOVL    #1, R0                                          ; 1453
                                            04 0014D          RET                                                     ; 1454

; Routine Size:  334 bytes,    Routine Base:  EXCH$DIRE_CODE + 09CE
```

```
;1375     1455  1 GLOBAL ROUTINE dire_rt11_put_item (ctx : $ref_bblock) : NOVALUE =       %SBTTL 'dire_rt11_put_item'
;1376     1456  2 BEGIN
;1377     1457  2 !++
;1378     1458  2 !
;1379     1459  2 ! FUNCTIONAL DESCRIPTION:
;1380     1460  2 !
;1381     1461  2 !       Add a single item to the output listing, writing the record if we hit the column count
;1382     1462  2 !
;1383     1463  2 ! INPUTS:
;1384     1464  2 !
;1385     1465  2 !       ctx  - an RT11 directory entry context block, if address is zero, means write buffer.
;1386     1466  2 !
;1387     1467  2 ! IMPLICIT INPUTS:
;1388     1468  2 !
;1389     1469  2 !       dire - directory work area
;1390     1470  2 !
;1391     1471  2 ! OUTPUTS:
;1392     1472  2 !
;1393     1473  2 !       none
;1394     1474  2 !
;1395     1475  2 ! IMPLICIT OUTPUTS:
;1396     1476  2 !
;1397     1477  2 !       none
;1398     1478  2 !
;1399     1479  2 ! ROUTINE VALUE:
;1400     1480  2 !
;1401     1481  2 !       true if put succeeded, false if not
;1402     1482  2 !
;1403     1483  2 ! SIDE EFFECTS:
;1404     1484  2 !
;1405     1485  2 !       error conditions will be signaled
;1406     1486  2 !--
;1407     1487  2
;1408     1488  2 $dbgtrc_prefix ('dire_rt11_put_item> ');
;1409     1489  2
;1410     1490  2 LOCAL
;1411     1491  2     desc : VECTOR [2, LONG],
;1412     1492  2     status
;1413     1493  2     ;
;1414     1494  2
;1415     1495  2 REGISTER
;1416     1496  2     wid                                        ! current listing width
;1417     1497  2     ;
;1418     1498  2
;1419     1499  2 BIND
;1420     1500  2     dire = exch$a_gbl [excg$a_dire_work]       : $ref_bblock,                  ! pointer to work area
;1421     1501  2     buf = dire [dire$t_list_buffer] : VECTOR [dire$s_list_buffer, BYTE] ! output buffer
;1422     1502  2     ;
;1423     1503  2
;1424     1504  2
;1425     1505  2 $block_check (2, .dire, dire, 427);
;1426     1506  2
;1427     1507  2 ! Perform a couple of sanity checks
;1428     1508  2 !
;1429     1509  2 $logic_check (2, (.dire [dire$l_cur_column] LSSU .dire [dire$l_q_columns]), 110);
;1430     1510  2 $logic_check (2, (.dire [dire$l_cur_width]  LSSU .dire [dire$l_list_width]), 111);
;1431     1511  2
```

```
;  1432          1512  2 ! Perform required initializations
;  1433          1513  2 !
;  1434          1514  2 dire [dire$v_items_printed] = true;            ! one or more items has been printed
;  1435          1515  2 wid = .dire [dire$l_cur_width];               ! a fast register for the current width
;  1436          1516  2
;  1437          1517  2 ! An input address of 0 means that we should flush the buffer
;  1438          1518  2 !
;  1439          1519  3 IF (.ctx EQL 0)
;  1440          1520  2 THEN
;  1441          1521  3     BEGIN
;  1442          1522  3
;  1443          1523  3     IF .wid GTR 0
;  1444          1524  3     THEN
;  1445          1525  4         BEGIN
;  1446          1526  4         desc [0] = .wid;
;  1447          1527  4         desc [1] = buf [0];
;  1448          1528  4         exch$dire_put (desc);
;  1449          1529  3         END;
;  1450          1530  3
;  1451          1531  3     dire [dire$l_cur_column] = 0;
;  1452          1532  3     dire [dire$l_cur_width] = 0;
;  1453          1533  3     RETURN true;
;  1454          1534  2     END;
;  1455          1535  2
;  1456          1536  2 ! Add the separator to the buffer
;  1457          1537  2 !
;  1458          1538  2 IF .dire [dire$l_cur_column] NEQ 0
;  1459          1539  2 THEN
;  1460          1540  3     BEGIN
;  1461          1541  3     CH$FILL (%C ' ', dire$k_item_spacing, buf [.wid]);
;  1462          1542  3     wid = .wid + dire$k_item_spacing;
;  1463          1543  2     END;
;  1464          1544  2
;  1465          1545  2 ! We always add the protection and filename to the buffer
;  1466          1546  2 !
;  1467          1547  2 CH$MOVE (12, ctx [rt11ctx$t_exp_protected], buf [.wid]);
;  1468          1548  2 wid = .wid + 12;
;  1469          1549  2
;  1470          1550  2 ! Add the file size if requested
;  1471          1551  2 !
;  1472          1552  2 IF .dire [dire$v_q_size]
;  1473          1553  2 THEN
;  1474          1554  3     BEGIN
;  1475          1555  3     LOCAL
;  1476          1556  3         addwid,
;  1477          1557  3         ctrstr;
;  1478          1558  3     IF .dire [dire$v_q_octal]
;  1479          1559  3     THEN
;  1480          1560  4         BEGIN
;  1481          1561  4         addwid = 7;
;  1482          1562  4         ctrstr = %ASCID ' !6OW';
;  1483          1563  4         END
;  1484          1564  3     ELSE
;  1485          1565  4         BEGIN
;  1486          1566  4         addwid = 5;
;  1487          1567  4         ctrstr = %ASCID '!5UL';
;  1488          1568  3         END;
```

```
;  1489          1569  3     desc [0] = .addwid;
;  1490          1570  3     desc [1] = buf [.wid];
;  1491          1571  4     IF NOT (status = $fao (.ctrstr, 0, desc, .ctx [rt11ctx$w_blocks]))
;  1492          1572  3     THEN
;  1493          1573  3         $exch_signal_stop (.status);
;  1494          1574  3
;  1495          1575  3     wid = .wid + .addwid;
;  1496          1576  2     END;
;  1497          1577  2
;  1498          1578  2 ! Add the date if requested
;  1499          1579  2 !
;  1500          1580  2 IF .dire [dire$v_q_date]
;  1501          1581  2 THEN
;  1502          1582  3     BEGIN
;  1503          1583  3     CH$MOVE (1, UPLIT BYTE (' '), buf [.wid]);
;  1504          1584  3     wid = .wid + 1;
;  1505          1585  3     CH$MOVE (rt11ctx$s_exp_date, ctx [rt11ctx$t_exp_date], buf [.wid]);
;  1506          1586  3     wid = .wid + rt11ctx$s_exp_date;
;  1507          1587  2     END;
;  1508          1588  2
;  1509          1589  2 ! Add the start block if requested
;  1510          1590  2 !
;  1511          1591  2 IF .dire [dire$v_q_blocks]
;  1512          1592  2 THEN
;  1513          1593  3     BEGIN
;  1514          1594  3     LOCAL
;  1515          1595  3         addwid,
;  1516          1596  3         ctrstr;
;  1517          1597  3     IF .dire [dire$v_q_octal]
;  1518          1598  3     THEN
;  1519          1599  4         BEGIN
;  1520          1600  4         addwid = 7;
;  1521          1601  4         ctrstr = %ASCID ' !6OW';
;  1522          1602  4         END
;  1523          1603  3     ELSE
;  1524          1604  4         BEGIN
;  1525          1605  4         addwid = 6;
;  1526          1606  4         ctrstr = %ASCID ' !5UL';
;  1527          1607  3         END;                                                                    ; F
;  1528          1608  3     desc [0] = .addwid;
;  1529          1609  3     desc [1] = buf [.wid];
;  1530          1610  4     IF NOT (status = $fao (.ctrstr, 0, desc, .ctx [rt11ctx$l_start_block]))
;  1531          1611  3     THEN
;  1532          1612  3         $exch_signal_stop (.status);
;  1533          1613  3
;  1534          1614  3     wid = .wid + .addwid;
;  1535          1615  3
;  1536          1616  2     END;
;  1537          1617  2
;  1538          1618  2 ! Do we need to flush?
;  1539          1619  2 !
;  1540          1620  2 IF (dire [dire$l_cur_column] = .dire [dire$l_cur_column] + 1) GEQU .dire [dire$l_q_columns]
;  1541          1621  2 THEN
;  1542          1622  3     BEGIN
;  1543          1623  3
;  1544          1624  3     desc [0] = .wid;
;  1545          1625  3     desc [1] = buf [0];
```

```
;  1546          1626  3     exch$dire_put (desc);
;  1547          1627  3     dire [dire$l_cur_column] = 0;
;  1548          1628  3     wid = 0;
;  1549          1629  2     END;
;  1550          1630  2
;  1551          1631  2 dire [dire$l_cur_width] = .wid;
;  1552          1632  2
;  1553          1633  2 RETURN;
;  1554          1634  1 END;


                                                                .PSECT  EXCH$DIRE_PLIT,NOWRT,2

                        00  00  00  57  4F  36  21  20  0042C P.ACM:   .ASCII  \ !60W\<0><0><0>                          ;
                                            010E0005  00434 P.ACL:   .LONG   17694725                                   ;
                                            00000000' 00438          .ADDRESS P.ACM                                     ;
                                        4C  55  35  21  0043C P.ACO:   .ASCII  \!5UL\                                   ;
                                            010E0004  00440 P.ACN:   .LONG   17694724                                   ;
                                            00000000' 00444          .ADDRESS P.ACO                                     ;
                                                20  00448 P.ACP:   .ASCII  \ \                                         ;
                                                    00449          .BLKB   3                                           ;
                        00  00  00  57  4F  36  21  20  0044C P.ACR:   .ASCII  \ !60W\<0><0><0>                          ;
                                            010E0005  00454 P.ACQ:   .LONG   17694725                                   ;
                                            00000000' 00458          .ADDRESS P.ACR                                     ;
                        00  00  00  4C  55  35  21  20  0045C P.ACT:   .ASCII  \ !5UL\<0><0><0>                          ;
                                            010E0005  00464 P.ACS:   .LONG   17694725                                   ;
                                            00000000' 00468          .ADDRESS P.ACT                                     ;

                                                                .EXTRN  SYS$FAO

                                                                .PSECT  EXCH$DIRE_CODE,NOWRT,2

                                                    0FFC 00000          .ENTRY  DIRE_RT11_PUT_ITEM, Save R2,R3,R4,R5,R6,R7,-; 1455
                                                                                R8,R9,R10,R11                               ;
                            5E                      0C   C2 00002       SUBL2   #12, SP                                 ;
                50 00000000G   EF                    0C   C1 00005       ADDL3   #12, EXCH$A_GBL, R0                     ; 1500
                            57                      60   D0 0000D       MOVL    (R0), R7                                ; 1501
                            59              44      A7   9E 00010       MOVAB   68(R7), R9                              ;
                            52  024400FE          8F   D0 00014       MOVL    #38011134, R2                           ; 1505
                            51      01AB            8F   3C 0001B       MOVZWL  #427, R1                                ;
                            50                      57   D0 00020       MOVL    R7, R0                                  ;
                                00000000G           EF   16 00023       JSB     EXCH$UTIL_BLOCK_CHECK                   ;
                            6E              40      A7   9E 00029       MOVAB   64(R7), (SP)                            ; 1509
                        28  A7                  00  BE   D1 0002D       CMPL    @0(SP), 40(R7)                          ;
                                                    13   1F 00032       BLSSU   1$                                      ;
                            7E              6E      8F   9A 00034       MOVZBL  #110, -(SP)                             ;
                                                    01   DD 00038       PUSHL   #1                                      ;
                                00000000G           8F   DD 0003A       PUSHL   #EXCH$_BADLOGIC                         ;
                00000000G   00                      03   FB 00040       CALLS   #3, LIB$STOP                            ;
                        38  A7                  3C  A7   D1 00047 1$:    CMPL    60(R7), 56(R7)                          ; 1510
                                                    13   1F 0004C       BLSSU   2$                                      ;
                            7E              6F      8F   9A 0004E       MOVZBL  #111, -(SP)                             ;
                                                    01   DD 00052       PUSHL   #1                                      ;
                                00000000G           8F   DD 00054       PUSHL   #EXCH$_BADLOGIC                         ;
                00000000G   00                      03   FB 0005A       CALLS   #3, LIB$STOP                            ;
                        30  A7                      02   88 00061 2$:    BISB2   #2, 48(R7)                              ; 1514
```

```
                                    56        3C    A7  D0 00065          MOVL    60(R7), WID                                 ; 1515
                                    58        04    AC  D0 00069          MOVL    CTX, R8                                     ; 1519
                                                    1B  12 0006D          BNEQ    4$
                                                    56  D5 0006F          TSTL    WID                                         ; 1523
                                                    10  15 00071          BLEQ    3$
                                          04        AE  56 D0 00073          MOVL    WID, DESC                               ; 1526
                                          08        AE  59 D0 00077          MOVL    R9, DESC+4                              ; 1527
                                                    04  AE 9F 0007B          PUSHAB  DESC                                    ; 1528
                                       FA31         CF  01 FB 0007E          CALLS   #1, EXCH$DIRE_PUT
                                                    00  BE D4 00083 3$:      CLRL    @0(SP)                                  ; 1531
                                                    3C  A7 D4 00086          CLRL    60(R7)                                  ; 1532
                                                        04 00089             RET                                             ; 1533
                                                    00  BE D5 0008A 4$:      TSTL    @0(SP)                                  ; 1538
                                                        0D 13 0008D          BEQL    5$
                 8649                18             00 00202020 8F F0 0008F          INSV    #2105376, #0, #24, (WID)+[R9]           ; 1541
                                                    56  02 C0 00099          ADDL2   #2, WID                                 ; 1542
                                     6649       52  A8  0C 28 0009C 5$:      MOVC3   #12, 82(R8), (WID)[R9]                  ; 1547
                                                    56  0C C0 000A2          ADDL2   #12, WID                                ; 1548
                                                    5A        2C A7 9E 000A5          MOVAB   44(R7), R10                             ; 1552
                                      3A            6A  0B E1 000A9          BBC     #11, (R10), 8$
                                                        6A 95 000AD          TSTB    (R10)                                   ; 1558
                                                        0A 18 000AF          BGEQ    6$
                                                    52  07 D0 000B1          MOVL    #7, ADDWID                              ; 1561
                                                    50        0000' CF 9E 000B4          MOVAB   P.ACL, CTRSTR                           ; 1562
                                                        08 11 000B9          BRB     7$                                      ; 1558
                                                    52  05 D0 000BB 6$:      MOVL    #5, ADDWID                              ; 1566
                                                    50        0000' CF 9E 000BE          MOVAB   P.ACN, CTRSTR                           ; 1567
                                          04        AE  52 D0 000C3 7$:      MOVL    ADDWID, DESC                            ; 1569
                                08        AE          59  56 C1 000C7          ADDL3   WID, R9, DESC+4                         ; 1570
                                                    7E        40 A8 3C 000CC          MOVZWL  64(R8), -(SP)                           ; 1571
                                                    08  AE 9F 000D0          PUSHAB  DESC
                                                    7E  D4 000D3          CLRL    -(SP)
                                                    50  DD 000D5          PUSHL   CTRSTR
                                 00000000G          00  04 FB 000D7          CALLS   #4, SYS$FAO
                                                    5B  50 D0 000DE          MOVL    R0, STATUS
                                                    50  5B E9 000E1          BLBC    STATUS, 12$
                                                    56  52 C0 000E4          ADDL2   ADDWID, WID                             ; 1575
                                      0F            6A  03 E1 000E7 8$:      BBC     #3, (R10), 9$                           ; 1580
                                                8649        0000' CF 90 000EB          MOVB    P.ACP, (WID)+[R9]               ; 1583
                                     6649       67   A8  0B 28 000F1          MOVC3   #11, 103(R8), (WID)[R9]                 ; 1585
                                                    56  0B C0 000F7          ADDL2   #11, WID                                ; 1586
                                      43            6A  01 E1 000FA 9$:      BBC     #1, (R10), 14$                          ; 1591
                                                        6A 95 000FE          TSTB    (R10)                                   ; 1597
                                                        0A 18 00100          BGEQ    10$
                                                    52  07 D0 00102          MOVL    #7, ADDWID                              ; 1600
                                                    50        0000' CF 9E 00105          MOVAB   P.ACQ, CTRSTR                           ; 1601
                                                        08 11 0010A          BRB     11$                                     ; 1597
                                                    52  06 D0 0010C 10$:      MOVL    #6, ADDWID                              ; 1605
                                                    50        0000' CF 9E 0010F          MOVAB   P.ACS, CTRSTR                           ; 1606
                                          04        AE  52 D0 00114 11$:      MOVL    ADDWID, DESC                            ; 1608
                                08        AE          59  56 C1 00118          ADDL3   WID, R9, DESC+4                         ; 1609
                                                        72 A8 DD 0011D          PUSHL   114(R8)                                 ; 1610
                                                    08  AE 9F 00120          PUSHAB  DESC
                                                    7E  D4 00123          CLRL    -(SP)
                                                    50  DD 00125          PUSHL   CTRSTR
                                 00000000G          00  04 FB 00127          CALLS   #4, SYS$FAO
                                                    5B  50 D0 0012E          MOVL    R0, STATUS
```

```
                                0A              5B  E8 00131               BLBS    STATUS, 13$                             ;
                                                5B  DD 00134 12$:           PUSHL   STATUS                                  ; 1612
                     00000000G  00              01  FB 00136               CALLS   #1, LIB$STOP                            ;
                                                    04 0013D               RET                                             ;
                                56              52  C0 0013E 13$:          ADDL2   ADDWID, WID                             ; 1614
                50      00      BE              01  C1 00141 14$:          ADDL3   #1, @0(SP), R0                          ; 1620
                        00      BE              50  D0 00146               MOVL    R0, @0(SP)                              ;
                        28      A7              50  D1 0014A               CMPL    R0, 40(R7)                              ;
                                                15  1F 0014E               BLSSU   15$                                     ;
                        04      AE              56  D0 00150               MOVL    WID, DESC                               ; 1624
                        08      AE              59  D0 00154               MOVL    R9, DESC+4                              ; 1625
                                        04      AE  9F 00158               PUSHAB  DESC                                    ; 1626
                      F954      CF              01  FB 0015B               CALLS   #1, EXCH$DIRE_PUT                       ;
                                        00      BE  D4 00160               CLRL    @0(SP)                                  ; 1627
                                                56  D4 00163               CLRL    WID                                     ; 1628
                        3C      A7              56  D0 00165 15$:          MOVL    WID, 60(R7)                             ; 1631
                                                    04 00169               RET                                             ; 1634

; Routine Size:  362 bytes,    Routine Base:  EXCH$DIRE_CODE + 0B1C
```

```
; 1556          1635  1 END
; 1557          1636  0 ELUDOM


                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  EXCH$DIRE_PLIT                 1132  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  EXCH$DIRE_CODE                 3206  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       103         0        1000        00:01.9
;  _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1    1151       136        11          79        00:01.4



; Information:  2
; Warnings:     0
; Errors:       0


;                                       COMMAND QUALIFIERS

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

; Size:          3206 code + 1132 data bytes
; Run Time:         01:05.5
; Elapsed Time:     03:17.6
; Lines/CPU Min:     1497
; Lexemes/CPU-Min: 20454
; Memory Used:  298 pages
; Compilation Complete
```

EXCDEFS
LIS

EXCDIRE
LIS

EXCDOS11
LIS

EXCDBG
LIS