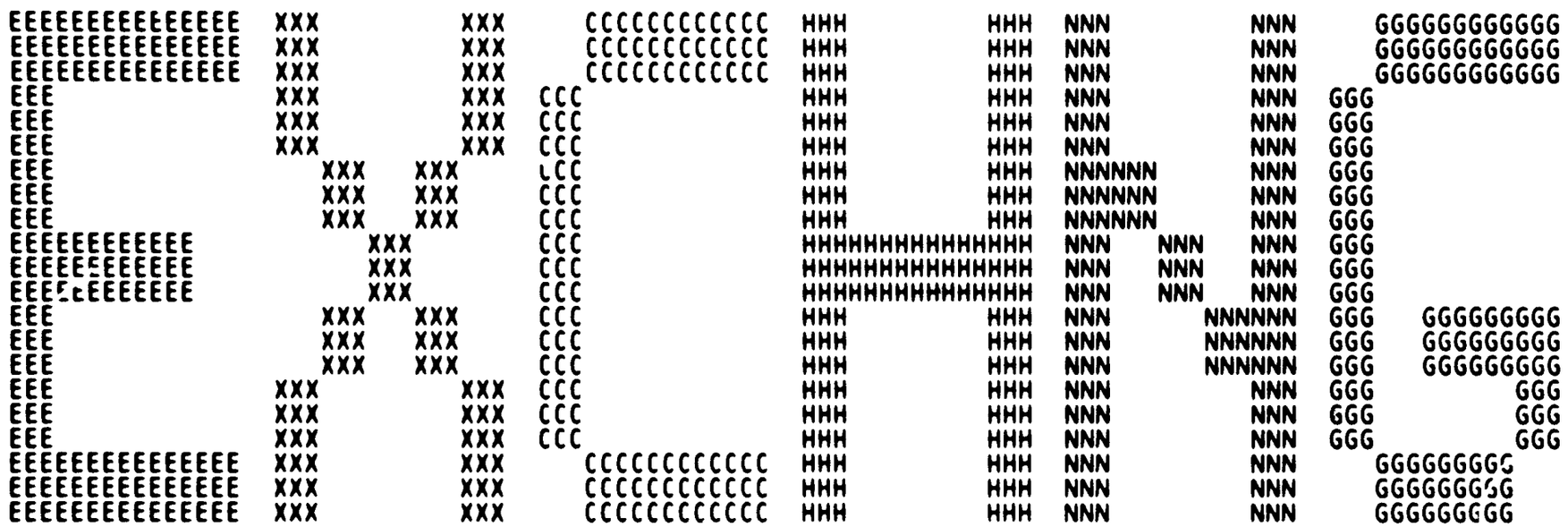
EEEEEEEEEEEEEEE XXX XXX CCCCCCCCCCCC HHH HHH NNN NNN GGGGGGGGGGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEE XXX XXX CCC HHH HHH NNNNNN NNN GGG
EEEEEEEEEEEE XXX CCC HHHHHHHHHHHHHHH NNN NNN NNN 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  RRRRRRRR    TTTTTTTTTT  NN      NN    AAAAAA    MM      MM
EEEEEEEEEE  XX      XX    CCCCCCCC  RRRRRRRR    TTTTTTTTTT  NN      NN    AAAAAA    MM      MM
EE          XX      XX  CC          RR      RR      TT      NN      NN  AA      AA  MMMM  MMMM
EE          XX      XX  CC          RR      RR      TT      NN      NN  AA      AA  MMMM  MMMM
EE            XX  XX    CC          RR      RR      TT      NNNN    NN  AA      AA  MM  MM  MM
EE            XX  XX    CC          RR      RR      TT      NNNN    NN  AA      AA  MM  MM  MM
EEEEEEEE        XX      CC          RRRRRRRR        TT      NN  NN  NN  AA      AA  MM      MM
EEEEEEEE        XX      CC          RRRRRRRR        TT      NN  NN  NN  AA      AA  MM      MM
EE            XX  XX    CC          RR  RR          TT      NN    NNNN  AAAAAAAAAA  MM      MM
EE            XX  XX    CC          RR  RR          TT      NN    NNNN  AAAAAAAAAA  MM      MM
EE          XX      XX  CC          RR    RR        TT      NN      NN  AA      AA  MM      MM
EE          XX      XX  CC          RR    RR        TT      NN      NN  AA      AA  MM      MM
EEEEEEEEEE  XX      XX    CCCCCCCC  RR      RR      TT      NN      NN  AA      AA  MM      MM
EEEEEEEEEE  XX      XX    CCCCCCCC  RR      RR      TT      NN      NN  AA      AA  MM      MM

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$rtnam                              %TITLE 'RT-11 name manipulation 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 DELETE, RENAME and other RT-11 name manipulation routines
; 35     0035  1 !
; 36     0036  1 ! ENVIRONMENT:  VAX/VMS User mode
; 37     0037  1 !
; 38     0038  1 ! AUTHOR:       CW Hobbs          CREATION DATE: 28-Nov-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 !               On a boot copy, allow an optional second parameter for
; 44     0044  1 !               compatibility with the RT11 syntax.  If present, make
; 45     0045  1 !               sure that it refers to the same device as the first parm.
; 46     0046  1 !
; 47     0047  1 !
; 48     0048  1 !--
; 49     0049  1
; 50     0050  1 ! Include files:
; 51     0051  1 !
; 52     0052  1 MACRO $module_name_string = 'exch$rtnam' %;     ! 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         rtnam_common,                       ! Common action routine for delete and rename verbs
;    62          0157  1     exch$rtnam_copy_boot,                   ! Write a boot routine on an RT-11 volume
;    63          0158  1     exch$rtnam_delete,                      ! Main entry routine for delete verb
;    64          0159  1         rtnam_delete_action,                ! Secondary action routine for delete verb
;    65          0160  1         rtnam_init              : NOVALUE,  ! Inits common to rtnam and TYPE
;    66          0161  1         rtnam_parse_cleanup     : NOVALUE,  ! Release structures and clean up after parse
;    67          0162  1         rtnam_parse_next_input,             ! Fetch and expand next input parameter
;    68          0163  1     exch$rtnam_rename,                      ! Main entry routine for RENAME verb
;    69          0164  1         rtnam_rename_action                 ! Secondary action routine for RENAME verb
;    70          0165  1     ;
;    71          0166  1
;    72          0167  1 ! EXCHANGE facility routines
;    73          0168  1 !
;    74          0169  1 EXTERNAL ROUTINE
;    75          0170  1     exch$cmd_cli_get_integer,               ! Get an integer value
;    76          0171  1     exch$cmd_related_file_parse,            ! Parse a file specification with a related file name
;    77          0172  1     exch$cmd_parse_filespec,                ! Parse a file specification
;    78          0173  1     exch$cmd_unwind_cli_syntax,             ! Return on undefined qualifiers
;    79          0174  1     exch$copy_namb_to_filb      : NOVALUE,  ! Copy fields from namb to the filb
;    80          0175  1     exch$io_rt11_read,                      ! Read blocks from RT-11 volume
;    81          0176  1     exch$io_rt11_write,                     ! Write blocks to an RT-11 volume
;    82          0177  1     exch$moun_implied_mount,                ! Do a default mount
;    83          0178  1     exch$rt11_create_file,                  ! Create and connect to an RT11 file
;    84          0179  1     exch$rt11_dircache_start    : NOVALUE,  ! Start directory write caching
;    85          0180  1     exch$rt11_dircache_stop     : NOVALUE,  ! Finish write caching and flush directory
;    86          0181  1     exch$rt11_dirseg_put,                   ! Write a directory segment
;    87          0182  1     exch$rtacp_consolidate,                 ! Compress unnecessary entries
;    88          0183  1     exch$rtacp_find_file,                   ! Find an RT11 file
;    89          0184  1     exch$rt11_open_file,                    ! Connect an RT11 file
;    90          0185  1     exch$rt11_write_cleanup     : NOVALUE,  ! Complete writing to an RT-11 volume
;    91          0186  1     exch$rt11_write_prepare     : NOVALUE,  ! Prepare to write to an RT-11 volume
;    92          0187  1     exch$util_fao_buffer,                   ! Format an fao string
;    93          0188  1     exch$util_filb_allocate,                ! Allocate file context block
;    94          0189  1     exch$util_filb_release      : NOVALUE,  ! Release file context block
;    95          0190  1     exch$util_file_error,                   ! Tell about an rms error
;    96          0191  1     exch$util_namb_release      : NOVALUE,  ! Release name block
;    97          0192  1     exch$util_radix50_from_ascii,           ! Convert ascii to radix50
;    98          0193  1     exch$util_rmsb_allocate,                ! Allocate Files-11 control block
;    99          0194  1     exch$util_rmsb_release      : NOVALUE,  ! Release Files-11 block
;   100          0195  1     exch$util_rt11ctx_allocate,             ! Allocate RT-11 context block
;   101          0196  1     exch$util_rt11ctx_release   : NOVALUE,  ! Release RT-11 block
;   102          0197  1     exch$util_vm_allocate                   ! Allocate virtual memory
;   103          0198  1     ;
;   104          0199  1
;   105          0200  1 ! Equated symbols:
;   106          0201  1 !
;   107          0202  1 !LITERAL
;   108          0203  1 !    ;
;   109          0204  1
;   110          0205  1 ! Bound declarations:
;   111          0206  1 !
;   112          0207  1 !BIND
```

; 113 0208 1 ! ;

```
;  115     0209  1 GLOBAL ROUTINE rtnam_common =   %SBTTL 'rtnam_common'
;  116     0210  2 BEGIN
;  117     0211  2 !++
;  118     0212  2 !
;  119     0213  2 ! FUNCTIONAL DESCRIPTION:
;  120     0214  2 !
;  121     0215  2 !       Common action routine for the DELETE and RENAME verbs for RT11 only
;  122     0216  2 !
;  123     0217  2 ! INPUTS:
;  124     0218  2 !
;  125     0219  2 !       none
;  126     0220  2 !
;  127     0221  2 ! IMPLICIT INPUTS:
;  128     0222  2 !
;  129     0223  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  130     0224  2 !
;  131     0225  2 ! OUTPUTS:
;  132     0226  2 !
;  133     0227  2 !       none
;  134     0228  2 !
;  135     0229  2 ! IMPLICIT OUTPUTS:
;  136     0230  2 !
;  137     0231  2 !       none
;  138     0232  2 !
;  139     0233  2 ! ROUTINE VALUE:
;  140     0234  2 !
;  141     0235  2 !       Success or worst error encountered.
;  142     0236  2 !
;  143     0237  2 ! SIDE EFFECTS:
;  144     0238  2 !
;  145     0239  2 !       Files may be created.
;  146     0240  2 !--
;  147     0241  2
;  148     0242  2 $dbgtrc_prefix ('rtnam_common> ');
;  149     0243  2
;  150     0244  2 LOCAL
;  151     0245  2     abort,
;  152     0246  2     nosys_signalled,
;  153     0247  2     status
;  154     0248  2     ;
;  155     0249  2
;  156     0250  2 BIND
;  157     0251  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock        ! Pointer to work area
;  158     0252  2     ;
;  159     0253  2
;  160     0254  2 ! Init the name used for the input file default
;  161     0255  2 !
;  162     0256  2 str$copy_dx (rtnam [rtnam$q_input_sticky_name], %ASCID '');
```

```
;  164          0257  2 ! Loop through the list of input file specifications.  Errors will be signalled.
;  165          0258  2 !
;  166          0259  2 abort = false;
;  167          0260  2 nosys_signalled = false;
;  168          0261  2
;  169          0262  3 WHILE (status = rtnam_parse_next_input ())      ! Get next input file parameter
;  170          0263  2 DO
;  171          0264  3     BEGIN
;  172          0265  3     LOCAL
;  173          0266  3         nam_len,
;  174          0267  3         typ_len,
;  175          0268  3         tot_len
;  176          0269  3         ;
;  177          0270  3     BIND
;  178          0271  3         filb = rtnam [rtnam$a_inp_filb] : $ref_bblock,
;  179          0272  3         namb = filb [filb$a_assoc_namb] : $ref_bblock,
;  180          0273  3         volb = filb [filb$a_assoc_volb] : $ref_bblock,
;  181          0274  3         ctx = filb [filb$a_context] : $ref_bblock,
;  182          0275  3         nam_nam = namb [namb$q_name] : $desc_block,
;  183          0276  3         nam_typ = namb [namb$q_type] : $desc_block
;  184          0277  3         ;
;  185          0278  3
;  186          0279  3     $block_check (2, .filb, filb, 547);
;  187          0280  3     $block_check (2, .namb, namb, 548);
;  188          0281  3     $block_check (2, .volb, volb, 549);
;  189          0282  3     $block_check (2, .ctx, rt11ctx, 550);                    ! Make sure that it is what we think
;  190          0283  3     $logic_check (3, (.ctx [rt11ctx$a_assoc_filb] EQL .filb), 206);
;  191          0284  3     $logic_check (3, (.ctx [rt11ctx$a_assoc_volb] EQL .volb), 207);
;  192          0285  3
;  193          0286  3     ! Create the result file name in the filb
;  194          0287  3     !
;  195          0288  3     nam_len = .nam_nam [dsc$w_length];
;  196          0289  3     typ_len = .nam_typ [dsc$w_length];
;  197          0290  3     tot_len = .nam_len + .typ_len;                   ! Final length of both
;  198          0291  3     filb [filb$l_result_name_len] = .volb [volb$l_vol_ident_len] + .tot_len;    ! Length of volume ident
;  199          0292  3     $logic_check (2, (.filb [filb$l_result_name_len] EQU filb$s_result_name), 208);
;  200          0293  3     CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],                  ! Volume name
;  201          0294  3                  .nam_len, .nam_nam [dsc$a_pointer], .typ_len, .nam_typ [dsc$a_pointer],
;  202          0295  3                  0, filb$s_result_name, filb [filb$t_result_name]);
;  203          0296  3
;  204          0297  3     $trace_print_fao ('looking for "!AF"', .filb [filb$l_result_name_len], filb [filb$t_result_name]);
;  205          0298  3
;  206          0299  3     ! Need to be able to write to the volume
;  207          0300  3     !
;  208          0301  3     IF NOT .volb [volb$v_write]
;  209          0302  3     THEN
;  210          0303  4         BEGIN
;  211          0304  4         status = (IF .rtnam [rtnam$v_delete_command] THEN exch$_nodellock ELSE exch$_norenlock);
;  212          0305  4         $exch_signal (.status, 2, .filb [filb$l_result_name_len], filb [filb$t_result_name]);
;  213          0306  4         abort = true;
;  214          0307  4         END
;  215          0308  4
;  216          0309  3     ELSE
;  217          0310  4         BEGIN
;  218          0311  4
;  219          0312  4         ! Engage directory write caching on the volume
;  220          0313  4         !
```

```
;  221     0314  4          exch$rt11_dircache_start (.volb);
;  222     0315  4
;  223     0316  4          WHILE exch$rtacp_find_file (.ctx, filb [filb$t_result_name] + .volb [volb$l_vol_ident_len], .tot_len
;  224     0317  4          DO
;  225     0318  4
;  226     0319  4              ! Two kinds of files need special treatment, so filter them before we perform any action
;  227     0320  4              !
;  228     0321  4              IF
;  229     0322  4
;  230     0323  4                  ! Files with a file type of .SYS will not be touched unless /SYSTEM has been given
;  231     0324  4                  !
;  232     0325  6                  (BEGIN
;  233     0326  6                  IF .ctx [rt11ctx$w_filetype] EQL r50_sys
;  234     0327  6                  THEN
;  235     0328  7                      BEGIN
;  236     0329  7
;  237     0330  7                      ! Give a warning message the first time we pass over a .SYS file
;  238     0331  7                      !
;  239     0332  8                      IF  (NOT .rtnam [rtnam$v_q_system])
;  240     0333  7                        AND
;  241     0334  8                          (NOT .nosys_signalled)
;  242     0335  7                      THEN
;  243     0336  8                          BEGIN
;  244     0337  8                          nosys_signalled = true;
;  245     0338  8                          $exch_signal (exch$_nosysact);
;  246     0339  7                          END;
;  247     0340  7                      .rtnam [rtnam$v_q_system]   ! Block has the value of the qualifier /SYSTEM
;  248     0341  7                      END
;  249     0342  6                  ELSE
;  250     0343  6                      1                           ! File not .SYS, file is "found"
;  251     0344  5                  END)
;  252     0345  5
;  253     0346  5                  ! And now for the second class of file.  Both blocks must return true for the file to be "found
;  254     0347  5                  !
;  255     0348  4                  AND
;  256     0349  4
;  257     0350  4                      ! Files with a file type of .BAD will not be touched by wildcard names
;  258     0351  4                      !
;  259     0352  6                      (BEGIN
;  260     0353  6                      IF .ctx [rt11ctx$w_filetype] EQL r50_bad
;  261     0354  6                      THEN
;  262     0355  7                          BEGIN
;  263     0356  7                          IF  .namb [namb$v_wild_name]                ! If the found file was not explicitly named
;  264     0357  7                              OR                                      !  then skip to the next file by calling
;  265     0358  7                              .namb [namb$v_wild_type]                !  ourselves again
;  266     0359  7                          THEN
;  267     0360  7                              0       ! Wildcard, block returns false, file is not "found"
;  268     0361  7                          ELSE
;  269     0362  7                              1       ! No wildcard, block returns true, file is "found"
;  270     0363  7                          END
;  271     0364  6                      ELSE
;  272     0365  6                          1           ! File not .BAD, file is "found"
;  273     0366  5                      END)
;  274     0367  5
;  275     0368  5              ! Now the THEN clause, we have truly found this file, now work on it
;  276     0369  5              !
;  277     0370  4              THEN
```

```
;  278     0371  5                BEGIN
;  279     0372  5                LOCAL
;  280     0373  5                    res_buf : $bvector [filb$s_result_name],          ! A buffer in which to build the act
;  281     0374  5                    res_len;
;  282     0375  5
;  283     0376  5                ! Create the result file name in the filb
;  284     0377  5                !
;  285     0378  5                nam_len = .ctx [rt11ctx$l_exp_fullname_len];
;  286     0379  5                res_len = .volb [volb$l_vol_ident_len] + .nam_len;          ! Length of volume ident plu
;  287     0380  5                $logic_check (2, (.res_len LEQU filb$s_result_name), 125);
;  288     0381  5                CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],          ! Volume nam
;  289     0382  5                         .nam_len, ctx [rt11ctx$t_exp_fullname],
;  290     0383  5                         0, filb$s_result_name, res_buf);
;  291     0384  5
;  292     0385  5                $debug_print_fao ('file "!AF" exists', .res_len, res_buf);
;  293     0386  5
;  294     0387  5                filb [filb$v_files_found] = true;       ! Remember that we have found a file, whether or not
;  295     0388  5
;  296     0389  5                ! If control/c has been hit, leave the loop now
;  297     0390  5                !
;  298     0391  5                IF .exch$a_gbl [excg$v_control_c]
;  299     0392  5                THEN
;  300     0393  6                    BEGIN
;  301     0394  6                    $exch_signal ($info_stat_copy (exch$_canceled));
;  302     0395  6                    abort = true;
;  303     0396  6                    EXITLOOP;
;  304     0397  5                    END;
;  305     0398  5
;  306     0399  5                ! Call the appropriate secondary action routine, depending on the command
;  307     0400  5                !
;  308     0401  5                IF .rtnam [rtnam$v_delete_command]
;  309     0402  5                THEN
;  310     0403  5                    status = rtnam_delete_action (.res_len, res_buf)
;  311     0404  5                ELSE
;  312     0405  6                    BEGIN
;  313     0406  6                    status = rtnam_rename_action (.res_len, res_buf);
;  314     0407  6                    SELECTONE .status OF
;  315     0408  6                    SET
;  316     0409  6                        [exch$_parseerr, exch$_norendev, exch$_badfilename] :
;  317     0410  6                                                    abort = true;
;  318     0411  6                        [OTHERWISE] :
;  319     0412  6                                                    ;
;  320     0413  6                    TES;
;  321     0414  5                    END;
;  322     0415  5
;  323     0416  5                IF .abort THEN EXITLOOP;
;  324     0417  4                END;
;  325     0418  4
;  326     0419  4            ! If no files were found, then scream and shout
;  327     0420  4            !
;  328     0421  4            IF NOT .filb [filb$v_files_found]
;  329     0422  4            THEN
;  330     0423  5                BEGIN
;  331     0424  5                LOCAL
;  332     0425  5                    fao_desc : VECTOR [2, LONG];
;  333     0426  5
;  334     0427  5                ! Turn the expanded name into a descriptor
```

```
;  335          0428  5              !
;  336          0429  5              fao_desc [0] = .filb [filb$l_result_name_len];
;  337          0430  5              fao_desc [1] = filb [filb$t_result_name];
;  338          0431  5
;  339          0432  5              $exch_signal (exch$_filenotfound, 1, fao_desc);
;  340          0433  5
;  341          0434  4              END;
;  342          0435  4
;  343          0436  4 !???     IF .abort THEN EXITLOOP;
;  344          0437  3          END;
;  345          0438  3
;  346          0439  3      rtnam_parse_cleanup ();                    ! Release namb, clean up after parse
;  347          0440  3      IF .abort THEN EXITLOOP;
;  348          0441  2      END;
;  349          0442  2
;  350          0443  2 RETURN .status;
;  351          0444  1 END;


                                                        .TITLE  EXCH$RTNAM RT-11 name manipulation routines
                                                        .IDENT  \V04-000\

                                                        .PSECT  EXCH$RTNAM_PLIT,NOWRT,2

                                              00000 P.AAB:  .BLKB   0                                          ;
                                          010E0000  00000 P.AAA:  .LONG   17694720                             ;
                                          00000000' 00004         .ADDRESS P.AAB                               ;

                                                        .EXTRN  EXCH$CMD_CLI_GET_INTEGER
                                                        .EXTRN  EXCH$CMD_RELATED_FILE_PARSE
                                                        .EXTRN  EXCH$CMD_PARSE_FILESPEC
                                                        .EXTRN  EXCH$CMD_UNWIND_CLI_SYNTAX
                                                        .EXTRN  EXCH$COPY_NAMB_TO_FILB
                                                        .EXTRN  EXCH$IO_RT11_READ
                                                        .EXTRN  EXCH$IO_RT11_WRITE
                                                        .EXTRN  EXCH$MOON_IMPLIED_MOUNT
                                                        .EXTRN  EXCH$RT11_CREATE_FILE
                                                        .EXTRN  EXCH$RT11_DIRCACHE_START
                                                        .EXTRN  EXCH$RT11_DIRCACHE_STOP
                                                        .EXTRN  EXCH$RT11_DIRSEG_PUT
                                                        .EXTRN  EXCH$RTACP_CONSOLIDATE
                                                        .EXTRN  EXCH$RTACP_FIND_FILE
                                                        .EXTRN  EXCH$RT11_OPEN_FILE
                                                        .EXTRN  EXCH$RT11_WRITE_CLEANUP
                                                        .EXTRN  EXCH$RT11_WRITE_PREPARE
                                                        .EXTRN  EXCH$UTIL_FAO_BUFFER
                                                        .EXTRN  EXCH$UTIL_FILB_ALLOCATE
                                                        .EXTRN  EXCH$UTIL_FILB_RELEASE
                                                        .EXTRN  EXCH$UTIL_FILE_ERROR
                                                        .EXTRN  EXCH$UTIL_NAMB_RELEASE
                                                        .EXTRN  EXCH$UTIL_RADIX50_FROM_ASCII
                                                        .EXTRN  EXCH$UTIL_RMSB_ALLOCATE
                                                        .EXTRN  EXCH$UTIL_RMSB_RELEASE
                                                        .EXTRN  EXCH$UTIL_RT11CTX_ALLOCATE
                                                        .EXTRN  EXCH$UTIL_RT11CTX_RELEASE
                                                        .EXTRN  EXCH$UTIL_VM_ALLOCATE
                                                        .EXTRN  EXCH$A_GBC, STR$COPY_DX
```

```
                                        .EXTRN  EXCH$UTIL_BLOCK_CHECK
                                        .EXTRN  EXCH$_BADLOGIC, EXCH$_NODELLOCK
                                        .EXTRN  EXCH$_NORENLOCK
                                        .EXTRN  EXCH$_NOSYSACT, EXCH$_CANCELED
                                        .EXTRN  EXCH$_PARSEERR, EXCH$_NORENDEV
                                        .EXTRN  EXCH$_BADFILENAME
                                        .EXTRN  EXCH$_FILENOTFOUND

                                        .PSECT  EXCH$RTNAM_CODE,NOWRT,2

                                0FFC 00000      .ENTRY  RTNAM_COMMON, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0209
                                                        R10,R11
                        5E      FED8    CE  9E 00002    MOVAB   -296(SP), SP
            50 00000000G        EF      18  C1 00007    ADDL3   #24, EXCH$A_GBL, R0                   ; 0251
                                0000'   CF  9F 0000F    PUSHAB  P.AAA                                 ; 0256
                        58              60  D0 00013    MOVL    (R0), R8
                                  14    A8  9F 00016    PUSHAB  20(R8)
                00000000G       00      02  FB 00019    CALLS   #2, STR$COPY_DX
                                  14    AE  7C 00020    CLRQ    NOSYS_SIGNALLED                       ; 0260
                    0000V       CF      00  FB 00023 1$: CALLS  #0, RTNAM_PARSE_NEXT_INPUT            ; 0262
                      10        AE      50  D0 00028    MOVL    R0, STATUS
                                03  10  AE  E8 0002C    BLBS    STATUS, 2$
                                    0282    31 00030    BRW     21$
                                56  24  A8  D0 00033 2$: MOVL    36(R8), R6                            ; 0272
                          0C    AE  18  A6  D0 00037    MOVL    24(R6), 12(SP)                        ; 0275
            50            0C    AE 00000050 8F  C1 0003C    ADDL3   #80, 12(SP), R0
                                  5A        60  9E 00045    MOVAB   (R0), R10
            50            0C    AE 00000058 8F  C1 00048    ADDL3   #88, 12(SP), R0                   ; 0276
                                  59        60  9E 00051    MOVAB   (R0), R9
                                  52 035B00FA 8F  D0 00054    MOVL    #56295674, R2                     ; 0279
                                  51    0223  8F  3C 0005B    MOVZWL  #547, R1
                                  50        56  D0 00060    MOVL    R6, R0
                                  00000000G   EF  16 00063    JSB     EXCH$UTIL_BLOCK_CHECK
                                  52 010A00F7 8F  D0 00069    MOVL    #17432823, R2                     ; 0280
                                  51    0224  8F  3C 00070    MOVZWL  #548, R1
                                  50      0C  AE  D0 00075    MOVL    12(SP), R0
                                  00000000G   EF  16 00079    JSB     EXCH$UTIL_BLOCK_CHECK
                                  57      1C  A6  D0 0007F    MOVL    28(R6), R7                        ; 0281
                                  52 041B00F3 8F  D0 00083    MOVL    #68878579, R2
                                  51    0225  8F  3C 0008A    MOVZWL  #549, R1
                                  50        57  D0 0008F    MOVL    R7, R0
                                  00000000G   EF  16 00092    JSB     EXCH$UTIL_BLOCK_CHECK
                                  5B      20  A6  D0 00098    MOVL    32(R6), R11                       ; 0282
                                  52 008200F4 8F  D0 0009C    MOVL    #8519924, R2
                                  51    0226  8F  3C 000A3    MOVZWL  #550, R1
                                  50        5B  D0 000A8    MOVL    R11, R0
                                  00000000G   EF  16 000AB    JSB     EXCH$UTIL_BLOCK_CHECK
                          08    AE          6A  3C 000B1    MOVZWL  (R10), NAM_LEN                    ; 0288
                          04    AE          69  3C 000B5    MOVZWL  (R9), TYP_LEN                     ; 0289
                6E        08    AE      04  AE  C1 000B9    ADDL3   TYP_LEN, NAM_LEN, TOT_LEN         ; 0290
            3A  A6        65    A7          6E  C1 000BF    ADDL3   TOT_LEN, 101(R7), 58(R6)          ; 0291
                    00000100    8F      3A  A6  D1 000C5    CMPL    58(R6), #256                      ; 0292
                                            13  1B 000CD    BLEQU   3$
                                7E      D0  8F  9A 000CF    MOVZBL  #208, -(SP)
                                            01  DD 000D3    PUSHL   #1
                                00000000G   8F  DD 000D5    PUSHL   #EXCH$_BADLOGIC
                    00000000G       00      03  FB 000DB    CALLS   #3, LIB$STOP
```

```
                        20   AE      0100    8F   3C 000E2 3$:     MOVZWL  #256, 32(SP)                                  ; 0293
                        24   AE        5A    A6   9E 000E8        MOVAB   90(R6), 36(SP)                                 ; 0295
                        1C   AE        24    AE   D0 000ED        MOVL    36(SP), 28(SP)                                 ;
    20  AE          00    69   A7        65    A7   2C 000F2        MOVC5   101(R7), 105(R7), #0, 32(SP), @28(SP)          ;
                                           1C   BE      000FA                                                           ;
                                                2A   18 000FC        BGEQ    4$                                         ;
                        1C   AE        65    A7   C0 000FE        ADDL2   101(R7), 28(SP)                                ;
                        20   AE        65    A7   C2 00103        SUBL2   101(R7), 32(SP)                                ;
    20  AE          00    04   BA        08    AE   2C 00108        MOVC5   NAM_LEN, @4(R10), #0, 32(SP), @28(SP)          ;
                                           1C   BE      00110                                                           ;
                                                14   18 00112        BGEQ    4$                                         ;
                        1C   AE        08    AE   C0 00114        ADDL2   NAM_LEN, 28(SP)                                ;
                        20   AE        08    AE   C2 00119        SUBL2   NAM_LEN, 32(SP)                                ;
    20  AE          00    04   B9        04    AE   2C 0011E        MOVC5   TYP_LEN, @4(R9), #0, 32(SP), @28(SP)           ;
                                           1C   BE      00126                                                           ;
                    30      48   A7                  05   E0 00128 4$:     BBS     #5, 72(R7), 7$                     ; 0301
                    0A      20   A8                  01   E1 0012D        BBC     #1, 32(R8), 5$                     ; 0304
                            10   AE  00000000G      8F   D0 00132        MOVL    #EXCH$_NODELLOCK, STATUS                   ;
                                                08   11 0013A        BRB     6$                                         ;
                            10   AE  00000000G      8F   D0 0013C 5$:     MOVL    #EXCH$_NORENLOCK, STATUS                   ;
                                           24   AE   DD 00144 6$:     PUSHL   36(SP)                                     ; 0305
                                           3A   A6   DD 00147        PUSHL   58(R6)                                     ;
                                                02   DD 0014A        PUSHL   #2                                         ;
                                           1C   AE   DD 0014C        PUSHL   STATUS                                     ;
                00000000G   00                  04   FB 0014F        CALLS   #4, LIB$SIGNAL                             ;
                        18   AE                  01   D0 00156        MOVL    #1, ABORT                                  ; 0306
                                                014C 31 00159        BRW     20$                                        ; 0301
                                                57   DD 0015D 7$:     PUSHL   R7                                         ; 0314
                00000000G   EF                  01   FB 0015F        CALLS   #1, EXCH$RT11_DIRCACHE_START               ;
                                                6E   DD 00166 8$:     PUSHL   TOT_LEN                                    ; 0316
                    50              56      65    A7   C1 00168        ADDL3   101(R7), R6, R0                            ;
                                           5A   A0   9F 0016D        PUSHAB  90(R0)                                     ;
                                                5B   DD 00170        PUSHL   R11                                        ;
                00000000G   EF                  03   FB 00172        CALLS   #3, EXCH$RTACP_FIND_FILE                   ;
                                    03          50   E8 00179        BLBS    R0, 9$                                     ;
                                                0109 31 0017C        BRW     19$                                        ;
                        7ABB    8F      3E      AB   B1 0017F 9$:     CMPW    62(R11), #31419                            ; 0326
                                                1F   12 00185        BNEQ    11$                                        ;
                1A          1C   A8                  03   E0 00187        BBS     #3, 28(R8), 11$                    ; 0332
                                    11          14   AE   E8 0018C        BLBS    NOSYS_SIGNALLED, 10$               ; 0334
                            14   AE                  01   D0 00190        MOVL    #1, NOSYS_SIGNALLED                ; 0337
                                            00000000G   8F   DD 00194        PUSHL   #EXCH$_NOSYSACT                    ; 0338
                00000000G   00                  01   FB 0019A        CALLS   #1, LIB$SIGNAL                             ;
                C0          1C   A8                  03   E1 001A1 10$:     BBC     #3, 28(R8), 8$                     ; 0340
                        0CAC    8F      3E      AB   B1 001A6 11$:     CMPW    62(R11), #3244                             ; 0353
                                                1A   12 001AC        BNEQ    12$                                        ;
                50          0C   AE  0000006C      8F   C1 001AE        ADDL3   #108, 12(SP), R0                           ; 0356
                AB                  60              01   E0 001B7        BBS     #1, (R0), 8$                               ;
                51          0C   AE  0000006C      8F   C1 001BB        ADDL3   #108, 12(SP), R1                           ; 0358
                9E                  61              02   E0 001C4        BBS     #2, (R1), 8$                               ;
                            08   AE        46    AB   D0 001C8 12$:     MOVL    70(R11), NAM_LEN                           ; 0378
    20  AE          65    A7        08    AE   C1 001CD        ADDL3   NAM_LEN, 101(R7), RES_LEN                      ; 0379
                00000100    8F      20      AE   D1 001D4        CMPL    RES_LEN, #256                              ; 0380
                                                13   1B 001DC        BLEQU   13$                                        ;
                                    7E      7D      8F   9A 001DE        MOVZBL  #125, -(SP)                                ;
                                                01   DD 001E2        PUSHL   #1                                         ;
                                            00000000G   8F   DD 001E4        PUSHL   #EXCH$_BADLOGIC                    ;
```

```
                00000000G  00              03  FB 001EA          CALLS   #3, LIB$STOP                                    ;
                        5A        0100      8F  3C 001F1 13$:    MOVZWL  #256, R10                                       ; 0382
                        59          28      AE  9E 001F6          MOVAB   RES_BUF, R9                                     ;
           5A         00          69  A7      65  A7 2C 001FA          MOVC5   101(R7), 105(R7), #0, R10, (R9)         ;
                                               69     00201                                                            ;
                                               10  18 00202          BGEQ    14$                                     ;
                        59          65      A7  C0 00204          ADDL2   101(R7), R9                                     ;
                        5A          65      A7  C2 00208          SUBL2   101(R7), R10                                    ;
           5A         00          54      A8      08  AE 2C 0020C          MOVC5   NAM_LEN, 84(R11), #0, R10, (R9)     ;
                                               69     00213                                                            ;
                                   2B      A6      08  88 00214 14$:    BISB2   #8, 43(R6)                              ; 0387
                                   1B 000000000G   FF  E9 00218          BLBC    @EXCH$A_GBL, 15$                        ; 0391
                                   50 000000000G   8F  D0 0021F          MOVL    #EXCH$_CANCELED, STATUS2                ; 0394
           50         03              00          03  F0 00226          INSV    #3, #0, #3, STATUS2                     ;
                                                  50  DD 0022B          PUSHL   STATUS2                                 ;
                00000000G  00                     01  FB 0022D          CALLS   #1, LIB$SIGNAL                          ;
                        18  AE                    01  D0 00234          MOVL    #1, ABORT                               ; 0395
                                                  4E  11 00238          BRB     19$                                     ; 0393
                11         20  A8                 01  E1 0023A 15$:    BBC     #1, 32(R8), 16$                         ; 0401
                                           28  AE  9F  00023F          PUSHAB  RES_BUF                                 ; 0403
                                           24  AE  DD  00242          PUSHL   RES_LEN                                 ;
                        0000V  CF                  02  FB 00245          CALLS   #2, RTNAM_DELETE_ACTION                 ;
                            10  AE                 50  D0 0024A          MOVL    R0, STATUS                              ;
                                                   31  11 0024E          BRB     18$                                     ;
                                           28  AE  9F  00250 16$:    PUSHAB  RES_BUF                                 ; 0406
                                           24  AE  DD  00253          PUSHL   RES_LEN                                 ;
                        0000V  CF                  02  FB 00256          CALLS   #2, RTNAM_RENAME_ACTION                 ;
                            10  AE                 50  D0 0025B          MOVL    R0, STATUS                              ;
                00000000G  8F          10  AE      D1  00025F          CMPL    STATUS, #EXCH$_PARSEERR                 ; 0409
                                                   14  13 00267          BEQL    17$                                     ;
                00000000G  8F          10  AE      D1  00269          CMPL    STATUS, #EXCH$_NORENDEV                 ;
                                                   0A  13 00271          BEQL    17$                                     ;
                00000000G  8F          10  AE      D1  00273          CMPL    STATUS, #EXCH$_BADFILENAME              ;
                                                   04  12 0027B          BNEQ    18$                                     ;
                        18  AE                    01  D0 0027D 17$:    MOVL    #1, ABORT                               ; 0410
                            03          18  AE      E8  00281 18$:    BLBS    ABORT, 19$                              ; 0416
                                              FEDE  31  00285          BRW     8$                                      ;
           1C           2B  A6                     03  E0 00288 19$:    BBS     #3, 43(R6), 20$                         ; 0421
                        F8  AD          3A  A6      D0  0028D          MOVL    58(R6), FAO_DESC                        ; 0429
                        FC  AD          24  AE      D0  00292          MOVL    36(SP), FAO_DESC+4                      ; 0430
                                        F8  AD      9F  00297          PUSHAB  FAO_DESC                                ; 0432
                                                    01  DD 0029A          PUSHL   #1                                      ;
                                    000000000G      8F  DD 0029C          PUSHL   #EXCH$_FILENOTFOUND                     ;
                00000000G  00                     03  FB 002A2          CALLS   #3, LIB$SIGNAL                          ;
                    0000V  CF                     00  FB 002A9 20$:    CALLS   #0, RTNAM_PARSE_CLEANUP                 ; 0439
                            03          18  AE      E8  002AE          BLBS    ABORT, 21$                              ; 0440
                                              FD6E  31  002B2          BRW     1$                                      ;
                                    50      10  AE      D0  002B5 21$:    MOVL    STATUS, R0                              ; 0443
                                                    04  002B9          RET                                             ; 0444

; Routine Size:  698 bytes,    Routine Base:  EXCH$RTNAM_CODE + 0000
```

```
;  353     0445  1 GLOBAL ROUTINE exch$rtnam_copy_boot =   %SBTTL 'exch$rtnam_copy_boot'
;  354     0446  2 BEGIN
;  355     0447  2 !++
;  356     0448  2 !
;  357     0449  2 ! FUNCTIONAL DESCRIPTION:
;  358     0450  2 !
;  359     0451  2 !       Write a boot block on the RT-11 volume.
;  360     0452  2 !
;  361     0453  2 ! INPUTS:
;  362     0454  2 !
;  363     0455  2 !       none
;  364     0456  2 !
;  365     0457  2 ! IMPLICIT INPUTS:
;  366     0458  2 !
;  367     0459  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  368     0460  2 !
;  369     0461  2 ! OUTPUTS:
;  370     0462  2 !
;  371     0463  2 !       none
;  372     0464  2 !
;  373     0465  2 ! IMPLICIT OUTPUTS:
;  374     0466  2 !
;  375     0467  2 !       none
;  376     0468  2 !
;  377     0469  2 ! ROUTINE VALUE:
;  378     0470  2 !
;  379     0471  2 !       Success or worst error encountered.
;  380     0472  2 !
;  381     0473  2 ! SIDE EFFECTS:
;  382     0474  2 !
;  383     0475  2 !       Block 0 and 2-5 of the target volume will be written
;  384     0476  2 !--
;  385     0477  2
;  386     0478  2 $dbgtrc_prefix ('rtnam_copy_boot> ');
;  387     0479  2
;  388     0480  2 LOCAL
;  389     0481  2     check,
;  390     0482  2     buf : $bvector [512*5],                     ! five block buffer for monitor file blocks
;  391     0483  2     status
;  392     0484  2     ;
;  393     0485  2
;  394     0486  2 BIND
;  395     0487  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock        ! Pointer to work area
;  396     0488  2     ;
;  397     0489  2
;  398     0490  2 ! Allocate and/or initialize the work area
;  399     0491  2 !
;  400     0492  2 rtnam_init ();
```

```
;  402     0493  2 ! Get the monitor file specification.  Errors will be signalled.
;  403     0494  2 !
;  404     0495  2 str$copy_dx (rtnam [rtnam$q_input_sticky_name], %ASCID '.SYS'); ! Default file type to .SYS
;  405     0496  3 IF (status = rtnam_parse_next_input ())
;  406     0497  2 THEN
;  407     0498  3     BEGIN
;  408     0499  3     LOCAL
;  409     0500  3         out_file : $desc_block,
;  410     0501  3         out_namb : $ref_bblock,
;  411     0502  3         nam_len,
;  412     0503  3         typ_len,
;  413     0504  3         tot_len
;  414     0505  3         ;
;  415     0506  3
;  416     0507  3     BIND
;  417     0508  3         filb = rtnam [rtnam$a_inp_filb] : $ref_bblock,
;  418     0509  3         namb = filb [filb$a_assoc_namb] : $ref_bblock,
;  419     0510  3         volb = filb [filb$a_assoc_volb] : $ref_bblock,
;  420     0511  3         ctx = filb [filb$a_context] : $ref_bblock,
;  421     0512  3         nam_nam = namb [namb$q_name] : $desc_block,
;  422     0513  3         nam_typ = namb [namb$q_type] : $desc_block
;  423     0514  3         ;
;  424     0515  3
;  425     0516  3     $block_check (2, .filb, filb, 638);
;  426     0517  3     $block_check (2, .namb, namb, 639);
;  427     0518  3     $block_check (2, .volb, volb, 640);
;  428     0519  3     $block_check (2, .ctx, rt11ctx, 641);                    ! Make sure that it is what we think
;  429     0520  3     $logic_check (3, (.ctx [rt11ctx$a_assoc_filb] EQL .filb), 273);
;  430     0521  3     $logic_check (3, (.ctx [rt11ctx$a_assoc_volb] EQL .volb), 274);
;  431     0522  3
;  432     0523  3     ! See if the optional output parameter is present.  If so, verify that it refers to the same volb as
;  433     0524  3     ! the input.  This lets us keep the syntax for COPY /BOOT compatible with RT-11.
;  434     0525  3     !
;  435     0526  3     $dyn_str_desc_init (out_file);
;  436     0527  4     IF (status = exch$cmd_parse_filespec (%ASCID 'OUTPUT', 0, 0, out_file, out_namb))
;  437     0528  3     THEN
;  438     0529  4         BEGIN
;  439     0530  4         REGISTER
;  440     0531  4             ignore,
;  441     0532  4             out_volb;
;  442     0533  4         out_volb = .out_namb [namb$a_assoc_volb];
;  443     0534  7         ignore = (0 NEQ (.out_namb [namb$l_nameflags] AND (namb$m_explicit_node OR namb$m_explicit_directory
;  444     0535  4                                 OR namb$m_explicit_name OR namb$m_explicit_type OR namb$m_explicit_version
;  445     0536  4         exch$util_namb_release (.out_namb);
;  446     0537  4         IF .out_volb NEQ .volb                   ! If volb addresses different it is not the same device
;  447     0538  4         THEN
;  448     0539  5             BEGIN
;  449     0540  5             status = exch$_nocopyboot;
;  450     0541  5             $exch_signal (.status, 0, exch$_notsamedev);
;  451     0542  5             rtnam_parse_cleanup ();
;  452     0543  5             RETURN .status;
;  453     0544  4             END;
;  454     0545  4         IF .ignore
;  455     0546  4         THEN
;  456     0547  4             $exch_signal (exch$_devonly, 1, out_file);                                              ;
;  457     0548  3         END;
;  458     0549  3
```

```
;  459          0550  3     ! Create the result file name in the filb, concatenate name and type together
;  460          0551  3     !
;  461          0552  3     nam_len = .nam_nam [dsc$w_length];
;  462          0553  3     typ_len = .nam_typ [dsc$w_length];
;  463          0554  3     tot_len = .nam_len + .typ_len;                  Final length of both
;  464          0555  3     filb [filb$l_result_name_len] = .volb [volb$l_vol_ident_len] + .tot_len;    ! Length of volume ident
;  465          0556  3     $logic_check (2, (.filb [filb$l_result_name_len] [EQU filb$s_result_name), 289);
;  466          0557  3     CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],                     ! Volume name
;  467          0558  3                     .nam_len, .nam_nam [dsc$a_pointer], .typ_len, .nam_typ [dsc$a_pointer],
;  468          0559  3                     0, filb$s_result_name, filb [filb$t_result_name]);
;  469          0560  3
;  470          0561  3     $trace_print_fao ('looking for "!AF"', .filb [filb$l_result_name_len], filb [filb$t_result_name]);
;  471          0562  3
;  472          0563  3     ! Need to be able to write to the volume
;  473          0564  3     !
;  474          0565  3     IF NOT .volb [volb$v_write]
;  475          0566  3     THEN
;  476          0567  4         BEGIN
;  477          0568  4         status = exch$_nocopyboot;
;  478        P 0569  4         $exch_signal (.status, 0, $warning_stat_copy (exch$_writelock),
;  479          0570  4                       2, .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;  480          0571  4         END
;  481          0572  4
;  482          0573  4     ! We can write, now see if we can find the monitor file
;  483          0574  4     !
;  484          0575  3     ELSE
;  485          0576  4         BEGIN
;  486          0577  4
;  487          0578  4         IF exch$rtacp_find_file (.ctx, filb [filb$t_result_name] + .volb [volb$l_vol_ident_len], .tot_len)
;  488          0579  4         THEN
;  489          0580  5             BEGIN
;  490          0581  5
;  491          0582  5             ! Remember the status of the read-check bit, then force read-checking on
;  492          0583  5             !
;  493          0584  5             check = .volb [volb$v_read_check];
;  494          0585  5             volb [volb$v_read_check] = true;
;  495          0586  5
;  496          0587  5             ! Now we can do what we came here to do, read the first 5 blocks of the monitor file
;  497          0588  5             !
;  498        L 0589  5             $logic_check (0, (ctx$k_buffer_blocks GEQU 5), 314);        ! Need a big enough buffer for read-
; %PRINT:       assumption 314 verified during compilation
;  499          0590  5             status = exch$io_rt11_read (.volb, .ctx [rt11ctx$l_start_block], 5, buf [0]);
;  500          0591  5             volb [volb$v_read_check] = .check;  ! Restore check state
;  501          0592  5             IF NOT .status
;  502          0593  5             THEN
;  503          0594  5                 RETURN .status;
;  504          0595  5
;  505          0596  5             ! Remember the status of the write-check bit, then force write-checking on
;  506          0597  5             !
;  507          0598  5             check = .volb [volb$v_write_check];
;  508          0599  5             volb [volb$v_write_check] = true;
;  509          0600  5
;  510          0601  5             ! Write the first block of the monitor file to block 0 of the volume
;  511          0602  5             !
;  512          0603  6             IF NOT (status = exch$io_rt11_write (.volb, 0, 1, buf [0]))
;  513          0604  5             THEN
;  514          0605  6                 BEGIN
```

```
;   515          0606  6                  volb [volb$v_write_check] = .check;     ! Restore the write-checking state
;   516          0607  6                  RETURN .status;
;   517          0608  5                  END;
;   518          0609  5
;   519          0610  5              ! Write the next four blocks of the monitor file to blocks 2 to 5 of the volume
;   520          0611  5              !
;   521          0612  5              status = exch$io_rt11_write (.volb, 2, 4, buf [512]);
;   522          0613  5              volb [volb$v_write_check] = .check; ! Restore the write-checking state
;   523          0614  5              IF NOT .status
;   524          0615  5              THEN
;   525          0616  5                  RETURN .status;
;   526          0617  5
;   527          0618  5              ! Log the action if requested
;   528          0619  5              !
;   529          0620  5              IF .rtnam [rtnam$v_q_log]
;   530          0621  5              THEN
;   531        P 0622  5                  $exch_signal (exch$_copyboot, 4, .ctx [rt11ctx$l_exp_fullname_len], ctx [rt11ctx$t_exp_fulln
;   532          0623  5                               .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
;   533          0624  5              END
;   534          0625  5
;   535          0626  5          ! The monitor file was not found, scream and shout
;   536          0627  5          !
;   537          0628  4          ELSE
;   538          0629  5              BEGIN
;   539          0630  5              LOCAL
;   540          0631  5                  fao_desc : VECTOR [2, LONG];
;   541          0632  5
;   542          0633  5              ! Turn the expanded name into a descriptor
;   543          0634  5              !
;   544          0635  5              fao_desc [0] = .filb [filb$l_result_name_len];
;   545          0636  5              fao_desc [1] = filb [filb$t_result_name];
;   546          0637  5
;   547          0638  5              $exch_signal (exch$_filenotfound, 1, fao_desc);
;   548          0639  5
;   549          0640  4                  END;
;   550          0641  3              END;
;   551          0642  3
;   552          0643  3          rtnam_parse_cleanup ();                     ! Release namb, clean up after parse
;   553          0644  2          END;
;   554          0645  2
;   555          0646  2 RETURN .status;
;   556          0647  1 END;


                                                        .PSECT  EXCH$RTNAM_PLIT,NOWRT,2

                                    53  59  53  2E  00008 P.AAD:  .ASCII  \.SYS\                                              ;
                                      010E0004  0000C P.AAC:  .LONG   17694724                                              ;
                                      00000000' 00010         .ADDRESS P.AAD                                                ;
                00  00  54  55  50  54  55  4F  00014 P.AAF:  .ASCII  \OUTPUT\<0><0>                                         ;
                                      010E0006  0001C P.AAE:  .LONG   17694726                                              ;
                                      00000000' 00020         .ADDRESS P.AAF                                                ;

                                                        .EXTRN  EXCH$GQ_DYN_STR_TEMPLATE
                                                        .EXTRN  EXCH$_NOCOPYBOOT
                                                        .EXTRN  EXCH$_NOTSAMEDEV
```

```
                                                        .EXTRN  EXCH$_DEVONLY, EXCH$_WRITELOCK
                                                        .EXTRN  EXCH$_COPYBOOT

                                                        .PSECT  EXCH$RTNAM_CODE,NOWRT,2

                                                        .ENTRY  EXCH$RTNAM_COPY_BOOT, Save R2,R3,R4,R5,R6,-      ; 0445
                                                                R7,R8,R9,RT0,R1T
                        5E      F5D0    CE  9E 00002            MOVAB   -2608(SP), SP
                52 00000000G    EF              18  C1 00007     ADDL3   #24, EXCH$A_GBL, R2                     ; 0487
                      0000V     CF              00  FB 0000F     CALLS   #0, RTNAM_INIT                          ; 0492
                                        0000'   CF  9F 00014     PUSHAB  P.AAC                                   ; 0495
                          08    AE              62  D0 00018     MOVL    (R2), 8(SP)
                50        08    AE              14  C1 0001C     ADDL3   #20, 8(SP), R0
                                                50  DD 00021     PUSHL   R0
                   00000000G    00              02  FB 00023     CALLS   #2, STR$COPY_DX
                      0000V     CF              0C  FB 0002A     CALLS   #0, RTNAM_PARSE_NEXT_INPUT              ; 0496
                                6E              50  D0 0002F     MOVL    R0, STATUS
                                03              6E  E8 00032     BLBS    STATUS, 1$
                                                    024B  31 00035  BRW     15$
                50        04    AE              24  C1 00038 1$:  ADDL3   #36, 4(SP), R0                        ; 0509
                                56              60  D0 0003D     MOVL    (R0), R6
                5A        18    A6 00000050     8F  C1 00040     ADDL3   #80, 24(R6), R10                        ; 0512
                59        18    A6 00000058     8F  C1 00049     ADDL3   #88, 24(R6), R9                         ; 0513
                                52 035B00FA     8F  D0 00052     MOVL    #56295674, R2                           ; 0516
                                51      027E    8F  3C 00059     MOVZWL  #638, R1
                                50              56  D0 0005E     MOVL    R6, R0
                                   00000000G    EF  16 00061     JSB     EXCH$UTIL_BLOCK_CHECK
                                52 010A00F7     8F  D0 00067     MOVL    #17432823, R2                           ; 0517
                                51      027F    8F  3C 0006E     MOVZWL  #639, R1
                                50        18    A6  D0 00073     MOVL    24(R6), R0
                                   00000000G    EF  16 00077     JSB     EXCH$UTIL_BLOCK_CHECK
                                57        1C    A6  D0 0007D     MOVL    28(R6), R7                              ; 0518
                                52 041B00F3     8F  D0 00081     MOVL    #68878579, R2
                                51      0280    8F  3C 00088     MOVZWL  #640, R1
                                50              57  D0 0008D     MOVL    R7, R0
                                   00000000G    EF  16 00090     JSB     EXCH$UTIL_BLOCK_CHECK
                                5B        20    A6  D0 00096     MOVL    32(R6), R11                             ; 0519
                                52 008200F4     8F  D0 0009A     MOVL    #8519924, R2
                                51      0281    8F  3C 000A1     MOVZWL  #641, R1
                                50              5B  D0 000A6     MOVL    R11, R0
                                   00000000G    EF  16 000A9     JSB     EXCH$UTIL_BLOCK_CHECK
                          28    AE 00000000G    EF  7D 000AF     MOVQ    TMPL, DESC                              ; 0526
                                          1C    AE  9F 000B7     PUSHAB  OUT_NAMB                                ; 0527
                                          2C    AE  9F 000BA     PUSHAB  OUT_FILE
                                                7E  7C 000BD     CLRQ    -(SP)
                                        0000'   CF  9F 000BF     PUSHAB  P.AAE
00000000G                       EF              05  FB 000C3     CALLS   #5, EXCH$CMD_PARSE_FILESPEC
                                6E              50  D0 000CA     MOVL    R0, STATUS
                                4F              6E  E9 000CD     BLBC    STATUS, 4$
                                50        1C    AE  D0 000D0     MOVL    OUT_NAMB, R0                            ; 0533
                                53        74    A0  D0 000D4     MOVL    116(R0), OUT_VOLB
                                                51  D4 000D8     CLRL    R1                                      ; 0534
                      0F40      8F        6C    A0  B3 000DA     BITW    108(R0), #3904
                                                02  13 000E0     BEQL    2$
                                                51  D6 000E2     INCL    R1
                                52              51  D0 000E4 2$:  MOVL    R1, IGNORE
                                                50  DD 000E7     PUSHL   R0                                      ; 0536
```

```
                    00000000G  EF              01  FB 000E9          CALLS   #1, EXCH$UTIL_NAMB_RELEASE
                               57              53  D1 000F0          CMPL    OUT_VOLB, R7                        ; 0537
                                               15  13 000F3          BEQL    3$
                               6E 00000000G    8F  D0 000F5          MOVL    #EXCH$_NOCOPYBOOT, STATUS           ; 0540
                                  00000000G    8F  DD 000FC          PUSHL   #EXCH$_NOTSAMEDEV                   ; 0541
                                               7E  D4 00102          CLRL    -(SP)
                                        08     AE  DD 00104          PUSHL   STATUS
                                           016D     31 00107          BRW     13$
                               12              52  E9 0010A 3$:      BLBC    IGNORE, 4$                          ; 0545
                                        28     AE  9F 0010D          PUSHAB  OUT_FILE                            ; 0547
                                               01  DD 00110          PUSHL   #1
                                  00000000G    8F  DD 00112          PUSHL   #EXCH$_DEVONLY
                    00000000G  00              03  FB 00118          CALLS   #3, LIB$SIGNAL
                           18  AE              6A  3C 0011F 4$:      MOVZWL  (R10), NAM_LEN                      ; 0552
                           10  AE              69  3C 00123          MOVZWL  (R9), TYP_LEN                       ; 0553
                 14    AE  18  AE          10  AE  C1 00127          ADDL3   TYP_LEN, NAM_LEN, TOT_LEN           ; 0554
                 3A    A6  65  A7          14  AE  C1 0012E          ADDL3   TOT_LEN, 101(R7), 58(R6)            ; 0555
                    00000100  8F          3A  A6  D1 00135          CMPL    58(R6), #256                        ; 0556
                                               14  1B 0013D          BLEQU   5$
                               7E      0121    8F  3C 0013F          MOVZWL  #289, -(SP)
                                               01  DD 00144          PUSHL   #1
                                  00000000G    8F  DD 00146          PUSHL   #EXCH$_BADLOGIC
                    00000000G  00              03  FB 0014C          CALLS   #3, LIB$STOP
                           08  AE          0100    8F  3C 00153 5$:      MOVZWL  #256, 8(SP)                     ; 0557
                           0C  AE          5A  A6  9E 00159          MOVAB   90(R6), 12(SP)                      ; 0559
                               58          0C  AE  D0 0015E          MOVL    12(SP), R8
     08    AE                 00          69  A7  65  A7  2C 00162  MOVC5   101(R7), 105(R7), #0, 8(SP), (R8)
                                                   68      0016A
                                                   26  18 0016B          BGEQ    6$
                                           58      65  A7  C0 0016D          ADDL2   101(R7), R8
                                        08     AE  65  A7  C2 00171          SUBL2   101(R7), 8(SP)
     08    AE                 00          04  BA      18  AE  2C 00176  MOVC5   NAM_LEN, @4(R10), #0, 8(SP), (R8)
                                                   68      0017E
                                                   12  18 0017F          BGEQ    6$
                                           58      18  AE  C0 00181          ADDL2   NAM_LEN, R8
                                        08     AE  18  AE  C2 00185          SUBL2   NAM_LEN, 8(SP)
     08    AE                 00          04  B9      10  AE  2C 0018A  MOVC5   TYP_LEN, @4(R9), #0, 8(SP), (R8)
                                                   68      00192
                               52          48  A7  9E 00193 6$:      MOVAB   72(R7), R2                          ; 0565
                        23     62              05  E0 00197          BBS     #5, (R2), 7$
                               6E 00000000G    8F  D0 0019B          MOVL    #EXCH$_NOCOPYBOOT, STATUS           ; 0568
                                           69  A7  9F 001A2          PUSHAB  105(R7)                             ; 0570
                                           65  A7  DD 001A5          PUSHL   101(R7)
                                               02  DD 001A8          PUSHL   #2
                               50 00000000G    8F  D0 001AA          MOVL    #EXCH$_WRITELOCK, STATUS2
                               50              07  8A 001B1          BICB2   #7, STATUS2
                                               50  DD 001B4          PUSHL   STATUS2
                                               7E  D4 001B6          CLRL    -(SP)
                                        14     AE  DD 001B8          PUSHL   STATUS
                                           009B     31 001BB          BRW     11$
                                        14     AE  DD 001BE 7$:      PUSHL   TOT_LEN                            ; 0578
                        50                 56      65  A7  C1 001C1          ADDL3   101(R7), R6, R0
                                           5A      A0  9F 001C6          PUSHAB  90(R0)
                                                   5B  DD 001C9          PUSHL   R11
                          00000000G        EF          03  FB 001CB          CALLS   #3, EXCH$RTACP_FIND_FILE
                                           03          50  E8 001D2          BLBS    R0, 8$
                                                   008A    31 001D5          BRW     12$
```

```
                53              62              01                  01  EF 001D8 8$:     EXTZV   #1, #1, (R2), CHECK                ; 0584
                                                62                  02  88 001DD         BISB2   #2, (R2)                            ; 0585
                                                            30      AE  9F 001E0         PUSHAB  BUF                                 ; 0590
                                                                    05  DD 001E3         PUSHL   #5
                                                            72      AB  DD 001E5         PUSHL   114(R11)
                                                                    57  DD 001E8         PUSHL   R7
                                    00000000G   EF                  04  FB 001EA         CALLS   #4, EXCH$IO_RT11_READ
                                                6E                  50  D0 001F1         MOVL    R0, STATUS
                62              01              01                  53  F0 001F4         INSV    CHECK, #1, #1, (R2)                 ; 0591
                                                3E                  6E  E9 001F9         BLBC    STATUS, 10$                         ; 0592
                53              62              01                  02  EF 001FC         EXTZV   #2, #1, (R2), CHECK                 ; 0598
                                                62                  04  88 00201         BISB2   #4, (R2)                            ; 0599
                                                            30      AE  9F 00204         PUSHAB  BUF                                 ; 0603
                                                                    01  DD 00207         PUSHL   #1
                                                                    7E  D4 00209         CLRL    -(SP)
                                                                    57  DD 0020B         PUSHL   R7
                                    00000000G   EF                  04  FB 0020D         CALLS   #4, EXCH$IO_RT11_WRITE
                                                6E                  50  D0 00214         MOVL    R0, STATUS
                                                07                  6E  E8 00217         BLBS    STATUS, 9$
                62              01              02                  53  F0 0021A         INSV    CHECK, #2, #1, (R2)                 ; 0606
                                                                    62  11 0021F         BRB     15$                                 ; 0607
                                                        0230        CE  9F 00221 9$:     PUSHAB  BUF+512                             ; 0612
                                                                    04  DD 00225         PUSHL   #4
                                                                    02  DD 00227         PUSHL   #2
                                                                    57  DD 00229         PUSHL   R7
                                    00000000G   EF                  04  FB 0022B         CALLS   #4, EXCH$IO_RT11_WRITE
                                                6E                  50  D0 00232         MOVL    R0, STATUS
                62              01              02                  53  F0 00235         INSV    CHECK, #2, #1, (R2)                 ; 0613
                                                46                  6E  E9 0023A 10$:    BLBC    STATUS, 15$                         ; 0614
                                52      04      AE                  1C  C1 0023D         ADDL3   #28, 4(SP), R2                      ; 0620
                                                39                  62  E9 00242         BLBC    (R2), 14$
                                                            69      A7  9F 00245         PUSHAB  105(R7)                             ; 0623
                                                            65      A7  DD 00248         PUSHL   101(R7)
                                                            54      AB  9F 0024B         PUSHAB  84(R11)
                                                            46      AB  DD 0024E         PUSHL   70(R11)
                                                                    04  DD 00251         PUSHL   #4
                                                    00000000G       8F  DD 00253         PUSHL   #EXCH$_COPYBOOT
                        00000000G   00                              06  FB 00259 11$:    CALLS   #6, LIB$SIGNAL
                                                                    1C  11 00260         BRB     14$                                 ; 0578
                            20      AE          3A      A6          D0 00262 12$:    MOVL    58(R6), FAO_DESC                    ; 0635
                            24      AE          0C      AE          D0 00267         MOVL    12(SP), FAO_DESC+4                  ; 0636
                                                20      AE          9F 0026C         PUSHAB  FAO_DESC                            ; 0638
                                                                    01  DD 0026F         PUSHL   #1
                                                    00000000G       8F  DD 00271         PUSHL   #EXCH$_FILENOTFOUND
                        00000000G   00                              03  FB 00277 13$:    CALLS   #3, LIB$SIGNAL
                            0000V   CF                              00  FB 0027E 14$:    CALLS   #0, RTNAM_PARSE_CLEANUP             ; 0643
                                    50                              6E  D0 00283 15$:    MOVL    STATUS, R0                          ; 0646
                                                                        04  00286         RET                                         ; 0647

; Routine Size:  647 bytes,    Routine Base:  EXCH$RTNAM_CODE + 02BA
```

```
;  558          0648  1 GLOBAL ROUTINE exch$rtnam_delete =      %SBTTL 'exch$rtnam_delete'
;  559          0649  2 BEGIN
;  560          0650  2 !++
;  561          0651  2 !
;  562          0652  2 ! FUNCTIONAL DESCRIPTION:
;  563          0653  2 !
;  564          0654  2 !       Entry routine for the DELETE verb for RT11 only
;  565          0655  2 !
;  566          0656  2 ! INPUTS:
;  567          0657  2 !
;  568          0658  2 !       none
;  569          0659  2 !
;  570          0660  2 ! IMPLICIT INPUTS:
;  571          0661  2 !
;  572          0662  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  573          0663  2 !
;  574          0664  2 ! OUTPUTS:
;  575          0665  2 !
;  576          0666  2 !       none
;  577          0667  2 !
;  578          0668  2 ! IMPLICIT OUTPUTS:
;  579          0669  2 !
;  580          0670  2 !       none
;  581          0671  2 !
;  582          0672  2 ! ROUTINE VALUE:
;  583          0673  2 !
;  584          0674  2 !       Success or worst error encountered.
;  585          0675  2 !
;  586          0676  2 ! SIDE EFFECTS:
;  587          0677  2 !
;  588          0678  2 !       Files may be deleted.
;  589          0679  2 !--
;  590          0680  2
;  591          0681  2 $dbgtrc_prefix ('rtnam_delete> ');
;  592          0682  2
;  593          0683  2 LOCAL
;  594          0684  2     rtnam : $ref_bblock                        ! Pointer to work area
;  595          0685  2     ;
;  596          0686  2
;  597          0687  2
;  598          0688  2 ! Allocate and/or initialize the work area
;  599          0689  2 !
;  600          0690  2 rtnam_init ();
;  601          0691  2
;  602          0692  2 ! Get pointers that we need.  Have to wait until work area allocated by init call
;  603          0693  2 !
;  604          0694  2 rtnam = .exch$a_gbl [excg$a_rtnam_work];             ! Pointer to work area
;  605          0695  2 rtnam [rtnam$v_delete_command] = true;
;  606          0696  2
;  607          0697  2 ! Do the rest from the common routine
;  608          0698  2 !
;  609          0699  2 RETURN rtnam_common ();
;  610          0700  1 END;
```

```
                                    0000 00000          .ENTRY  EXCH$RTNAM_DELETE, Save nothing                 ; 0648
                0000V   CF                  00   FB 00002  CALLS   #0, RTNAM_INIT                             ; 0690
                        50 00000000G        EF   D0 00007  MOVL    EXCH$A_GBL, R0                              ; 0694
                        50          18      A0   D0 0000E  MOVL    24(R0), RTNAM                               ;
                  20    A0                  02   88 00012  BISB2   #2, 32(RTNAM)                               ; 0695
                FAA4    CF                  00   FB 00016  CALLS   #0, RTNAM_COMMON                            ; 0699
                                            04   0001B  RET                                                     ; 0700

; Routine Size:  28 bytes,    Routine Base:  EXCH$RTNAM_CODE + 0541
```

```
;  612     0701  1 GLOBAL ROUTINE rtnam_delete_action (res_len, res_buf) = %SBTTL 'rtnam_delete_action (res_len, res_buf)'
;  613     0702  2 BEGIN
;  614     0703  2 !++
;  615     0704  2 !
;  616     0705  2 ! FUNCTIONAL DESCRIPTION:
;  617     0706  2 !
;  618     0707  2 !       Secondary action routine for the DELETE verb for RT11 only
;  619     0708  2 !
;  620     0709  2 ! INPUTS:
;  621     0710  2 !
;  622     0711  2 !       res_len - length of result name
;  623     0712  2 !       res_buf - address of result name
;  624     0713  2 !
;  625     0714  2 ! IMPLICIT INPUTS:
;  626     0715  2 !
;  627     0716  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  628     0717  2 !
;  629     0718  2 ! OUTPUTS:
;  630     0719  2 !
;  631     0720  2 !       none
;  632     0721  2 !
;  633     0722  2 ! IMPLICIT OUTPUTS:
;  634     0723  2 !
;  635     0724  2 !       none
;  636     0725  2 !
;  637     0726  2 ! ROUTINE VALUE:
;  638     0727  2 !
;  639     0728  2 !       Success or worst error encountered.
;  640     0729  2 !
;  641     0730  2 ! SIDE EFFECTS:
;  642     0731  2 !
;  643     0732  2 !       Files may be deleted.
;  644     0733  2 !--
;  645     0734  2
;  646     0735  2 $dbgtrc_prefix ('rtnam_delete_action> ');
;  647     0736  2
;  648     0737  2 LOCAL
;  649     0738  2     status
;  650     0739  2     ;
;  651     0740  2
;  652     0741  2 BIND
;  653     0742  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock,       ! Pointer to work area
;  654     0743  2     filb = rtnam [rtnam$a_inp_filb] : $ref_bblock,                                          ;
;  655     0744  2     volb = filb [filb$a_assoc_volb] : $ref_bblock,
;  656     0745  2     ctx = filb [filb$a_context] : $ref_bblock,
;  657     0746  2     ent = ctx [rt11ctx$a_ent_address] : $ref_bblock
;  658     0747  2     ;
;  659     0748  2
;  660     0749  2 $block_check (2, .filb, filb, 554);
;  661     0750  2 $block_check (2, .volb, volb, 555);
;  662     0751  2 $block_check (2, .ctx, rt11ctx, 556);                       ! Make sure that it is what we think it is
;  663     0752  2
;  664     0753  2 ! Verify that it is ok to delete the existing file
;  665     0754  2 !
;  666     0755  2 IF .ctx [rt11ctx$v_typ_protected]       ! Can't delete protected files
;  667     0756  2 THEN
;  668     0757  3     $exch_signal_return (exch$_notdeleted, 2, .res_len, .res_buf, exch$_rtprotect)
```

```
;  669          0758  3
;  670          0759  3 ! Looks good, delete it now
;  671          0760  3 !
;  672          0761  2 ELSE
;  673          0762  2     BEGIN
;  674          0763  2
;  675          0764  2     $trace_print_fao ('deleting "!AF"', .res_len, .res_buf);
;  676          0765  2
;  677          0766  2     ! Mark the entry as deleted
;  678          0767  2     !
;  679          0768  2     ent [rt11ent$b_type_byte] = rt11ent$m_typ_empty;
;  680          0769  2
;  681          0770  2     ! Also change the type in the context buffer, since it will be used by EXCH$RTACP_CHECK_POSITION
;  682          0771  2     !
;  683          0772  2     ctx [rt11ctx$b_type_byte] = rt11ent$m_typ_empty;
;  684          0773  2
;  685          0774  2     exch$rt11_dirseg_put (.volb, .ctx [rt11ctx$l_seg_number]);          ! Write the directory segment
;  686          0775  2
;  687          0776  2     IF .rtnam [rtnam$v_q_log]
;  688          0777  2     THEN
;  689          0778  2         $exch_signal (exch$_deleted, 2, .res_len, .res_buf);
;  690          0779  2
;  691          0780  2     END;
;  692          0781  2
;  693          0782  2 RETURN true;
;  694          0783  1 END;


                                                        .EXTRN  EXCH$_NOTDELETED
                                                        .EXTRN  EXCH$_RTPROTECT
                                                        .EXTRN  EXCH$_DELETED

                                    00FC 00000          .ENTRY  RTNAM_DELETE_ACTION, Save R2,R3,R4,R5,R6,R7 ; 0701
                    57 00000000G    00   9E 00002      MOVAB   LIB$SIGNAL, R7                              ;
                    56 00000000G    EF   9E 00009      MOVAB   EXCH$UTIL_BLOCK_CHECK, R6                   ;
          50 00000000G              EF   18 C1 00010      ADDL3   #24, EXCH$A_GBL, R0                      ; 0742
                                    54   60 D0 00018      MOVL    (R0), R4                                 ; 0743
          55         24             A4   1C C1 0001B      ADDL3   #28, 36(R4), R5                          ; 0744
          50         24             A4   20 C1 00020      ADDL3   #32, 36(R4), R0                          ; 0745
                                    53   60 D0 00025      MOVL    (R0), R3                                 ; 0746
                    52 035B00FA     8F   D0 00028      MOVL    #56295674, R2                               ; 0749
                    51      022A    8F   3C 0002F      MOVZWL  #554, R1                                    ;
                    50        24    A4   D0 00034      MOVL    36(R4), R0                                  ;
                                    66   16 00038      JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                    52 041B00F3     8F   D0 0003A      MOVL    #68878579, R2                               ; 0750
                    51      022B    8F   3C 00041      MOVZWL  #555, R1                                    ;
                    50              65   D0 00046      MOVL    (R5), R0                                    ;
                                    66   16 00049      JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                    52 008200F4     8F   D0 0004B      MOVL    #8519924, R2                                ; 0751
                    51      022C    8F   3C 00052      MOVZWL  #556, R1                                    ;
                    50              53   D0 00057      MOVL    R3, R0                                      ;
                                    66   16 0005A      JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                              39    A3   95 0005C      TSTB    57(R3)                                      ; 0755
                                    1C   18 0005F      BGEQ    1$                                          ;
                    52 00000000G    8F   D0 00061      MOVL    #EXCH$_NOTDELETED, TEMP                     ; 0757
                       00000000G    8F   DD 00068      PUSHL   #EXCH$_RTPROTECT                            ;
```

```
                         7E        04    AC  7D 0006E           MOVQ    RES_LEN, -(SP)                          ;
                                         02  DD 00072           PUSHL   #2                                      ;
                                         52  DD 00074           PUSHL   TEMP                                    ;
                         67              05  FB 00076           CALLS   #5, LIB$SIGNAL                          ;
                         50              52  D0 00079           MOVL    TEMP, R0                                ;
                                         04  00007C            RET                                             ;
                         50        7E    A3  D0 0007D 1$:       MOVL    126(R3), R0                             ; 0768
                   01    A0              02  90 00081           MOVB    #2, 1(R0)                               ;
                   39    A3              02  90 00085           MOVB    #2, 57(R3)                              ; 0772
                                   76    A3  DD 00089           PUSHL   118(R3)                                 ; 0774
                                         65  DD 0008C           PUSHL   (R5)                                    ;
             00000000G   EF              02  FB 0008E           CALLS   #2, EXCH$RT11_DIRSEG_PUT                ;
                         0F        1C    A4  E9 00095           BLBC    28(R4), 2$                              ; 0776
                         7E        04    AC  7D 00099           MOVQ    RES_LEN, -(SP)                          ; 0778
                                         02  DD 0009D           PUSHL   #2                                      ;
                              00000000G  8F  DD 0009F           PUSHL   #EXCH$_DELETED                          ;
                         67              04  FB 000A5           CALLS   #4, LIB$SIGNAL                          ;
                         50              01  D0 000A8 2$:       MOVL    #1, R0                                  ; 0782
                                         04  000AB             RET                                             ; 0783

; Routine Size:  172 bytes,    Routine Base:  EXCH$RTNAM_CODE + 055D
```

```
;  696          0784  1 GLOBAL ROUTINE rtnam_init : NOVALUE =   %SBTTL 'rtnam_init'
;  697          0785  2 BEGIN
;  698          0786  2 !++
;  699          0787  2 !
;  700          0788  2 ! FUNCTIONAL DESCRIPTION:
;  701          0789  2 !
;  702          0790  2 !       Common init routine for the rtnam verbs
;  703          0791  2 !
;  704          0792  2 ! INPUTS:
;  705          0793  2 !
;  706          0794  2 !       none
;  707          0795  2 !
;  708          0796  2 ! IMPLICIT INPUTS:
;  709          0797  2 !
;  710          0798  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  711          0799  2 !
;  712          0800  2 ! OUTPUTS:
;  713          0801  2 !
;  714          0802  2 !       none
;  715          0803  2 !
;  716          0804  2 ! IMPLICIT OUTPUTS:
;  717          0805  2 !
;  718          0806  2 !       none
;  719          0807  2 !
;  720          0808  2 ! ROUTINE VALUE:
;  721          0809  2 !
;  722          0810  2 !       none
;  723          0811  2 !
;  724          0812  2 ! SIDE EFFECTS:
;  725          0813  2 !
;  726          0814  2 !       Files may be created.
;  727          0815  2 !--
;  728          0816  2
;  729          0817  2 $dbgtrc_prefix ('rtnam_init> ');
;  730          0818  2
;  731          0819  2 LOCAL
;  732          0820  2     status
;  733          0821  2     ;
;  734          0822  2
;  735          0823  2 BIND
;  736          0824  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock        ! Pointer to work area
;  737          0825  2     ;
;  738          0826  2
;  739          0827  2 ! The COPY /BOOT command does not support /LOG or /SYSTEM.  Set up a handler to do a RETURN when we would
;  740          0828  2 ! normally get the MSG$_SYNTAX error.
;  741          0829  2 !
;  742          0830  2 ENABLE
;  743          0831  2     exch$cmd_unwind_cli_syntax;
;  744          0832  2
;  745          0833  2 ! If our pointer is null, we need to allocate and initialize the work area
;  746          0834  2 !
;  747          0835  2 IF .rtnam EQL 0
;  748          0836  2 THEN
;  749          0837  3     BEGIN
;  750          0838  3
;  751          0839  3     ! Get the right sized chunk of memory
;  752          0840  3     !
```

```
;  753          0841  3     rtnam = exch$util_vm_allocate (exchblk$s_rtnam);
;  754          0842  3
;  755          0843  3     ! Set the ident fields
;  756          0844  3     !
;  757          0845  3     $block_init (.rtnam, rtnam);
;  758          0846  3
;  759          0847  3     ! Set the dynamic strings
;  760          0848  3     !
;  761          0849  3     $dyn_str_desc_init (rtnam [rtnam$q_input_filename]);
;  762          0850  3     $dyn_str_desc_init (rtnam [rtnam$q_output_filename]);
;  763          0851  3     $dyn_str_desc_init (rtnam [rtnam$q_input_sticky_name]);
;  764          0852  3
;  765          0853  3     END
;  766          0854  2 ELSE
;  767          0855  3     BEGIN
;  768          0856  3
;  769          0857  3     ! Free the dynamic strings and the Chicago 7
;  770          0858  3     !
;  771          0859  3     str$free1_dx (rtnam [rtnam$q_input_filename]);
;  772          0860  3     str$free1_dx (rtnam [rtnam$q_output_filename]);
;  773          0861  3     str$free1_dx (rtnam [rtnam$q_input_sticky_name]);
;  774          0862  3
;  775          0863  3     END;
;  776          0864  2
;  777          0865  2 ! Get some confidence that our work area is valid
;  778          0866  2 !
;  779          0867  2 $block_check (2, .rtnam, rtnam, 541);
;  780          0868  2
;  781          0869  2 ! Set the last part of the block to nulls
;  782          0870  2 !
;  783          0871  2 CH$FILL (0, rtnam$k_end_zero - rtnam$k_start_zero, .rtnam + rtnam$k_start_zero);
;  784          0872  2
;  785          0873  2 ! Get the global boolean qualifiers common to all (COPY /BOOT, DELETE, RENAME) commands
;  786          0874  2 !
;  787          0875  2 rtnam [rtnam$v_q_log]           = cli$present (%ASCID 'LOG');                                   ! global
;  788          0876  2
;  789          0877  2 ! Get global booleans common to Delete and Rename only, COPY /BOOT will unwind, so nothing that COPY/BOOT ne
;  790          0878  2 ! can be past this point
;  791          0879  2 !
;  792          0880  2 rtnam [rtnam$v_q_system]        = cli$present (%ASCID 'SYSTEM');                                ! global
;  793          0881  2 !\ rtnam [rtnam$v_q_confirm]    = cli$present (%ASCID 'CONFIRM');                               ! global
;  794          0882  2
;  795          0883  2 RETURN;
;  796          0884  1 END;


                                                        .PSECT  EXCH$RTNAM_PLIT,NOWRT,2

                                    00  47  4F  4C  00024 P.AAH:  .ASCII  \LOG\<0>                                    ;
                                                    010E0003  00028 P.AAG:  .LONG   17694723                        ;
                                                    00000000' 0002C         .ADDRESS P.AAH                          ;
                            00  00  4D  45  54  53  59  53  00030 P.AAJ:  .ASCII  \SYSTEM\<0><0>                      ;
                                                    010E0006  00038 P.AAI:  .LONG   17694726                        ;
                                                    00000000' 0003C         .ADDRESS P.AAJ                          ;

                                                        .EXTRN  STR$FREE1_DX, CLI$PRESENT
```

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

                                                    01FC 00000          .ENTRY  RTNAM_INIT, Save R2,R3,R4,R5,R6,R7,R8       ; 0784
                                          58 00000000G  00   9E 00002          MOVAB   CLI$PRESENT, R8
                                          57 00000000G  00   9E 00009          MOVAB   STR$FREE1_DX, R7
                                52 00000000G      EF        18   C1 00010          ADDL3   #24, EXCH$A_GBL, R2                 ; 0824
                                                  6D      0096   CF   DE 00018          MOVAL   3$, (FP)
                                                                 62   D5 0001D          TSTL    (R2)                        ; 0835
                                                                 44   12 0001F          BNEQ    1$
                                                                 34   DD 00021          PUSHL   #52                         ; 0841
                                     00000000G    EF             01   FB 00023          CALLS   #1, EXCH$UTIL_VM_ALLOCATE
                                                  62             50   D0 0002A          MOVL    R0, (R2)
                                            08    A0             34   B0 0002D          MOVW    #52, 8(R0)                  ; 0845
                                            0A    A0             0E   8E 00031          MNEGB   #14, 10(R0)
                                                  50             62   D0 00035          MOVL    (R2), R0                    ; 0849
                                                  53 00000000G   EF   D0 00038          MOVL    TMPL, R3
                                                  60             53   D0 0003F          MOVL    R3, (R0)
                                                  51 00000000G   EF   D0 00042          MOVL    TMPL+4, R1
                                            04    A0             51   D0 00049          MOVL    R1, 4(R0)
                                50                62             0C   C1 0004D          ADDL3   #12, (R2), R0               ; 0850
                                                  60             53   D0 00051          MOVL    R3, (R0)
                                            04    A0             51   D0 00054          MOVL    R1, 4(R0)
                                50                62             14   C1 00058          ADDL3   #20, (R2), R0               ; 0851
                                                  60             53   D0 0005C          MOVL    R3, (R0)
                                            04    A0             51   D0 0005F          MOVL    R1, 4(R0)
                                                                 13   11 00063          BRB     2$                          ; 0835
                                                                 62   DD 00065  1$:     PUSHL   (R2)                        ; 0859
                                                  67             01   FB 00067          CALLS   #1, STR$FREE1_DX
                                7E                62             0C   C1 0006A          ADDL3   #12, (R2), -(SP)            ; 0860
                                                  67             01   FB 0006E          CALLS   #1, STR$FREE1_DX
                                7E                62             14   C1 00071          ADDL3   #20, (R2), -(SP)            ; 0861
                                                  67             01   FB 00075          CALLS   #1, STR$FREE1_DX
                                                  56             62   D0 00078  2$:     MOVL    (R2), R6                    ; 0867
                                                  52 003400F2    8F   D0 0007B          MOVL    #3408114, R2
                                                  51      021D   8F   3C 00082          MOVZWL  #541, R1
                                                  50             56   D0 00087          MOVL    R6, R0
                                                     00000000G   EF   16 0008A          JSB     EXCH$UTIL_BLOCK_CHECK
                                      18                  00    6E        00   2C 00090          MOVC5   #0, (SP), #0, #24, 28(R6)   ; 0871
                                                                      1C   A6   00095
                                                                 0000'  CF   9F 00097          PUSHAB  P.AAG                       ; 0875
                                                                 68             01   FB 0009B          CALLS   #1, CLI$PRESENT
                                1C    A6                  01             00   F0 0009E          INSV    R0, #0, #1, 28(R6)
                                                                 0000'  CF   9F 000A4          PUSHAB  P.AAI                       ; 0880
                                                                 68             01   FB 000A8          CALLS   #1, CLI$PRESENT
                                1C    A6                  01             03   F0 000AB          INSV    R0, #3, #1, 28(R6)
                                                                                04   000B1          RET                         ; 0884
                                                                           0000 000B2  3$:     .WORD   Save nothing            ; 0824
                                                                 7E             D4 000B4          CLRL    -(SP)
                                                                 5E             DD 000B6          PUSHL   SP
                                                  7E        04   AC             7D 000B8          MOVQ    4(AP), -(SP)
                                     00000000G    EF             03   FB 000BC          CALLS   #3, EXCH$CMD_UNWIND_CLI_SYNTAX
                                                                                04   000C3          RET

; Routine Size:  196 bytes,    Routine Base:  EXCH$RTNAM_CODE + 0609
```

```
;  798          0885  1 GLOBAL ROUTINE rtnam_parse_cleanup : NOVALUE =  %SBTTL 'rtnam_parse_cleanup'
;  799          0886  2 BEGIN
;  800          0887  2 !++
;  801          0888  2 !
;  802          0889  2 ! FUNCTIONAL DESCRIPTION:
;  803          0890  2 !
;  804          0891  2 !       Clean up after a successful parse.  Release the namb and other structures.
;  805          0892  2 !
;  806          0893  2 ! INPUTS:
;  807          0894  2 !
;  808          0895  2 !       none
;  809          0896  2 !
;  810          0897  2 ! IMPLICIT INPUTS:
;  811          0898  2 !
;  812          0899  2 !       rtnam$a_inp_namb field in rtnam work area
;  813          0900  2 !
;  814          0901  2 ! OUTPUTS:
;  815          0902  2 !
;  816          0903  2 !       none
;  817          0904  2 !
;  818          0905  2 ! IMPLICIT OUTPUTS:
;  819          0906  2 !
;  820          0907  2 !       none
;  821          0908  2 !
;  822          0909  2 ! ROUTINE VALUE:
;  823          0910  2 !
;  824          0911  2 !       none
;  825          0912  2 !
;  826          0913  2 ! SIDE EFFECTS:
;  827          0914  2 !
;  828          0915  2 !       none
;  829          0916  2 !--
;  830          0917  2
;  831          0918  2 $dbgtrc_prefix ('rtnam_parse_cleanup> ');
;  832          0919  2
;  833          0920  2 BIND
;  834          0921  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock,       ! Pointer to work area
;  835          0922  2     filb = rtnam [rtnam$a_inp_filb]       : $ref_bblock, ! Filb for the input
;  836          0923  2     namb = filb [filb$a_assoc_namb]       : $ref_bblock, ! Namb for the input
;  837          0924  2     volb = filb [filb$a_assoc_volb]       : $ref_bblock, ! Volb for the input
;  838          0925  2     ctx = filb [filb$a_context]           : $ref_bblock  ! Volume specific context
;  839          0926  2     ;
;  840          0927  2
;  841          0928  2 $debug_print_lit ('entry');
;  842          0929  2 $block_check (2, .filb, filb, 543);
;  843          0930  2 $block_check (2, .namb, namb, 544);
;  844          0931  2 $block_check (2, .volb, volb, 545);
;  845          0932  2 $block_check (2, .ctx, rtlctx, 546);
```

```
;  847          0933  2 ! If we have been able to write, then clean things up
;  848          0934  2 !
;  849          0935  2 IF .volb [volb$v_write]
;  850          0936  2 THEN
;  851          0937  3     BEGIN
;  852          0938  3
;  853          0939  3     ! Compress unnecessary entries from the directory.  0 means not to restructure the directory.
;  854          0940  3     !
;  855          0941  3     exch$rtacp_consolidate (.volb, 0);
;  856          0942  3
;  857          0943  3     ! Turn caching off and flush any modified directory segments
;  858          0944  3     !
;  859          0945  3     exch$rt11_dircache_stop (.volb);
;  860          0946  3
;  861          0947  2     END;
;  862          0948  2
;  863          0949  2 ! Release the context block
;  864          0950  2 !
;  865          0951  2 exch$util_rt11ctx_release (.ctx);
;  866          0952  2
;  867          0953  2 ! Release the input namb
;  868          0954  2 !
;  869          0955  2 exch$util_namb_release (.namb);
;  870          0956  2
;  871          0957  2 ! Release the input filb
;  872          0958  2 !
;  873          0959  2 exch$util_filb_release (.filb);
;  874          0960  2
;  875          0961  2 RETURN;
;  876          0962  1 END;


                                  00FC 00000          .ENTRY  RTNAM_PARSE_CLEANUP, Save R2,R3,R4,R5,R6,R7 ; 0885
                     57 00000000G EF   9E 00002       MOVAB   EXCH$UTIL_BLOCK_CHECK, R7                   ;
          50 00000000G EF           18   C1 00009       ADDL3   #24, EXCH$A_GBL, R0                         ; 0921
          54           60           24   C1 00011       ADDL3   #36, (R0), R4                               ; 0922
          56           64           18   C1 00015       ADDL3   #24, (R4), R6                               ; 0923
          53           64           1C   C1 00019       ADDL3   #28, (R4), R3                               ; 0924
          55           64           20   C1 0001D       ADDL3   #32, (R4), R5                               ; 0925
                     52 035B00FA    8F   D0 00021       MOVL    #56295674, R2                               ; 0929
                     51      021F   8F   3C 00028       MOVZWL  #543, R1                                    ;
                     50             64   D0 0002D       MOVL    (R4), R0                                    ;
                                    67   16 00030       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                     52 010A00F7    8F   D0 00032       MOVL    #17432823, R2                               ; 0930
                     51      0220   8F   3C 00039       MOVZWL  #544, R1                                    ;
                     50             66   D0 0003E       MOVL    (R6), R0                                    ;
                                    67   16 00041       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                     53             63   D0 00043       MOVL    (R3), R3                                    ; 0931
                     52 041B00F3    8F   D0 00046       MOVL    #68878579, R2                               ;
                     51      0221   8F   3C 0004D       MOVZWL  #545, R1                                    ;
                     50             53   D0 00052       MOVL    R3, R0                                      ;
                                    67   16 00055       JSB     EXCH$UTIL_BLOCK_CHECK                       ;
                     52 008200F4    8F   D0 00057       MOVL    #8519924, R2                                ; 0932
                     51      0222   8F   3C 0005E       MOVZWL  #546, R1                                    ;
```

```
                                50          65  D0 00063          MOVL    (R5), R0                                        ;
                                            67  16 00066          JSB     EXCH$UTIL_BLOCK_CHECK                           ;
                     14     48  A3          05  E1 00068          BBC     #5, 72(R3), 1$                                  ; 0935
                                            7E  D4 0006D          CLRL    -(SP)                                           ; 0941
                                            53  DD 0006F          PUSHL   R3                                              ;
                00000000G   EF              02  FB 00071          CALLS   #2, EXCH$RTACP_CONSOLIDATE                      ;
                                            53  DD 00078          PUSHL   R3                                              ; 0945
                00000000G   EF              01  FB 0007A          CALLS   #1, EXCH$RT11_DIRCACHE_STOP                     ;
                                            65  DD 00081 1$:      PUSHL   (R5)                                            ; 0951
                00000000G   EF              01  FB 00083          CALLS   #1, EXCH$UTIL_RT11CTX_RELEASE                   ;
                                            66  DD 0008A          PUSHL   (R6)                                            ; 0955
                00000000G   EF              01  FB 0008C          CALLS   #1, EXCH$UTIL_NAMB_RELEASE                      ;
                                            64  DD 00093          PUSHL   (R4)                                            ; 0959
                00000000G   EF              01  FB 00095          CALLS   #1, EXCH$UTIL_FILB_RELEASE                      ;
                                                04 0009C          RET                                                     ; 0962

; Routine Size:  157 bytes,    Routine Base:  EXCH$RTNAM_CODE + 06CD
```

```
;  878          0963  1 GLOBAL ROUTINE rtnam_parse_next_input = %SBTTL 'rtnam_parse_next_input'
;  879          0964  2 BEGIN
;  880          0965  2 !++
;  881          0966  2 !
;  882          0967  2 ! FUNCTIONAL DESCRIPTION:
;  883          0968  2 !
;  884          0969  2 !       Fetch the next input parameter.  Parse the filename and initialize the input file work region.
;  885          0970  2 !
;  886          0971  2 ! INPUTS:
;  887          0972  2 !
;  888          0973  2 !       none
;  889          0974  2 !
;  890          0975  2 ! IMPLICIT INPUTS:
;  891          0976  2 !
;  892          0977  2 !       Command qualifier value as returned from CLI$xxx routines.  rtnam command work area.
;  893          0978  2 !
;  894          0979  2 ! OUTPUTS:
;  895          0980  2 !
;  896          0981  2 !       none
;  897          0982  2 !
;  898          0983  2 ! IMPLICIT OUTPUTS:
;  899          0984  2 !
;  900          0985  2 !       Command work area receives parse info
;  901          0986  2 !
;  902          0987  2 ! ROUTINE VALUE:
;  903          0988  2 !
;  904          0989  2 !       Success or worst error encountered.
;  905          0990  2 !
;  906          0991  2 ! SIDE EFFECTS:
;  907          0992  2 !
;  908          0993  2 !       none
;  909          0994  2 !--
;  910          0995  2
;  911          0996  2 $dbgtrc_prefix ('rtnam_parse_next_input> ');
;  912          0997  2
;  913          0998  2 LOCAL
;  914          0999  2     ctx : $ref_bblock,
;  915          1000  2     volb : $ref_bblock,
;  916          1001  2     format,
;  917          1002  2     status
;  918          1003  2     ;
;  919          1004  2
;  920          1005  2 BIND
;  921          1006  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock,       ! Pointer to work area
;  922          1007  2     inp_filb = rtnam [rtnam$a_inp_filb]  : $ref_bblock, ! Filb for the input
;  923          1008  2     inp_namb = rtnam [rtnam$a_inp_namb]  : $ref_bblock  ! Namb for the input
;  924          1009  2     ;
;  925          1010  2
;  926          1011  2
;  927          1012  2 $block_check (2, .rtnam, rtnam, 542);
;  928          1013  2
;  929          1014  2 ! Fetch the filename and a pointer to a namb
;  930          1015  2 !
;  931          1016  3 IF NOT (status = exch$cmd_parse_filespec (%ASCID 'INPUT', rtnam [rtnam$q_input_sticky_name], 0,
;  932          1017  3                                          rtnam [rtnam$q_input_filename], inp_namb))
;  933          1018  2 THEN
;  934          1019  3     BEGIN
```

```
;  935          1020  3     IF .status NEQ 0
;  936          1021  3     THEN
;  937          1022  3         $exch_signal (exch$_parseerr, 1, rtnam [rtnam$q_input_filename], .status);
;  938          1023  3     RETURN .status;                                 ! No more files to rtnam, or error in parse
;  939          1024  2     END;
;  940          1025  2 $trace_print_fao ('input parameter is "!AS"', rtnam [rtnam$q_input_filename]);
;  941          1026  2
;  942          1027  2 ! If if the input potentially describes multiple files, then set the bit
;  943          1028  2 !
;  944          1029  2 IF .inp_namb [namb$v_more_files] OR .inp_namb [namb$v_wildcard]
;  945          1030  2 THEN
;  946          1031  2     rtnam [rtnam$v_multiple_files] = true;
;  947          1032  2
;  948          1033  2 ! If a foreign device is not mounted, then perform an implied mount
;  949          1034  2 !
;  950          1035  3 IF   (.inp_namb [namb$a_assoc_volb] EQL 0)
;  951          1036  2   AND
;  952          1037  4      (BEGIN
;  953          1038  4       BIND
;  954          1039  4         dev = inp_namb [namb$l_fabdev] : $bblock;
;  955          1040  5       .dev [dev$v_for] OR (NOT (.dev [dev$v_mnt]))
;  956          1041  3       END)
;  957          1042  2   AND
;  958          1043  4      ((.inp_namb [namb$b_devclass] EQL dc$_disk)
;  959          1044  3       OR
;  960          1045  3       (.inp_namb [namb$b_devclass] EQL dc$_tape))
;  961          1046  2 THEN
;  962          1047  3     BEGIN
;  963          1048  3
;  964          1049  4     IF NOT (status = exch$moun_implied_mount (.inp_namb))
;  965          1050  3     THEN
;  966          1051  4         BEGIN
;  967          1052  4         exch$util_namb_release (.inp_namb);
;  968          1053  4         RETURN .status;
;  969          1054  3         END;
;  970          1055  2     END;
;  971          1056  2
;  972          1057  2 volb = .inp_namb [namb$a_assoc_volb];
;  973          1058  2 IF .volb NEQ 0                              ! If the volb is there, then the interesting format
;  974          1059  2 THEN
;  975          1060  2     format = .volb [volb$b_vol_format]  !  is that of the volb.  Otherwise, it
;  976          1061  2 ELSE
;  977          1062  2     format = .inp_namb [namb$b_vol_format];     !  is that of the namb.
;  978          1063  2
;  979          1064  2 CASE .format FROM volb$k_vfmt_lobound TO volb$k_vfmt_hibound OF
;  980          1065  2 SET
;  981          1066  2     [volb$k_vfmt_files11] :
;  982          1067  3                         BEGIN
;  983          1068  3                         LOCAL
;  984          1069  3                             errstat;
;  985          1070  3                         IF .inp_namb [namb$v_explicit_node]
;  986          1071  3                         THEN
;  987          1072  4                             BEGIN
;  988          1073  4                             errstat = exch$_noremote;
;  989          1074  4                             $exch_signal (.errstat, 1, rtnam [rtnam$q_input_filename]);
;  990          1075  4                             END
;  991          1076  3                         ELSE
```

```
;  992          1077  4                          BEGIN
;  993          1078  4                          errstat = exch$_opnotperf11;
;  994          1079  4                          $exch_signal (.errstat, 1, inp_namb [namb$q_device]);
;  995          1080  3                          END;
;  996          1081  3                      exch$util_namb_release (.inp_namb);                 ! Release the bad namb
;  997          1082  3                      RETURN .errstat;
;  998          1083  2                      END;
;  999          1084  2
; 1000          1085  2      [volb$k_vfmt_dos11] :
; 1001          1086  3                      BEGIN
; 1002          1087  3                      LOCAL
; 1003          1088  3                          errstat;
; 1004          1089  3                      errstat = exch$_opnotperdos;
; 1005          1090  3                      $exch_signal (.errstat, 1, inp_namb [namb$q_device]);
; 1006          1091  3                      exch$util_namb_release (.inp_namb);                 ! Release the bad namb
; 1007          1092  3                      RETURN .errstat;
; 1008          1093  2                      END;
; 1009          1094  2
; 1010          1095  2      [volb$k_vfmt_rt11] :
; 1011          1096  2                      ;                                                   ! These we can do
; 1012          1097  2
; 1013          1098  2 !\   [volb$k_vfmt_rtmt] :
; 1014          1099  2 !\                   BEGIN
; 1015          1100  2 !\                   LOCAL
; 1016          1101  2 !\                       errstat;
; 1017          1102  2 !\                   errstat = exch$_opnotperrtmt;
; 1018          1103  2 !\                   $exch_signal (.errstat, 1, inp_namb [namb$q_device]);
; 1019          1104  2 !\                   exch$util_namb_release (.inp_namb);                 ! Release the bad namb
; 1020          1105  2 !\                   RETURN .errstat;
; 1021          1106  2 !\                   END;
; 1022          1107  2
; 1023          1108  2      [INRANGE, OUTRANGE] :
; 1024          1109  2                      $logic_check (0, (false), 238);
; 1025          1110  2 TES;
; 1026          1111  2
; 1027          1112  2 $logic_check (3, (.volb NEQ 0), 119);
; 1028          1113  2
; 1029          1114  2 ! Now copy the full name to the default name for proper stickiness
; 1030          1115  2 !
; 1031          1116  2 str$copy_dx (rtnam [rtnam$q_input_sticky_name], inp_namb [namb$q_fullname]);
; 1032          1117  2
; 1033          1118  2 ! Allocate a file block to contain the input information
; 1034          1119  2 !
; 1035          1120  2 inp_filb = exch$util_filb_allocate ();
; 1036          1121  2 exch$copy_namb_to_filb (.inp_namb, .inp_filb);  ! Copy from the namb to the filb
; 1037          1122  2
; 1038          1123  2 ! Put an RT-11 context block on the filb
; 1039          1124  2 !
; 1040          1125  2 ctx = exch$util_rt11ctx_allocate (.volb, .inp_filb);
; 1041          1126  2 inp_filb [filb$a_context] = .ctx;
; 1042          1127  2
; 1043          1128  2 RETURN .status;
; 1044          1129  1 END;


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

```
                00 00 00 54 55 50 4E 49  00040 P.AAL:  .ASCII  \INPUT\<0><0><0>
                              010E0005   00048 P.AAK:  .LONG   17694725
                              00000000'  0004C         .ADDRESS P.AAL

                                                       .EXTRN  EXCH$_NOREMOTE, EXCH$_OPNOTPERF11
                                                       .EXTRN  EXCH$_OPNOTPERDOS

                                                       .PSECT  EXCH$RTNAM_CODE,NOWRT,2

                                  01FC 00000           .ENTRY  RTNAM_PARSE_NEXT_INPUT, Save R2,R3,R4,R5,-      ; 0963
                                                               R6,R7,R8
                          58 00000000G   EF  9E 00002  MOVAB   EXCH$UTIL_NAMB_RELEASE, R8
                          57 00000000G   00  9E 00009  MOVAB   LIB$SIGNAL, R7
                50 00000000G   EF        18  C1 00010  ADDL3   #24, EXCH$A_GBL, R0                          ; 1006
                               53        60  D0 00018  MOVL    (R0), R3                                    ; 1007
                               52 003400F2  8F  D0 0001B  MOVL    #3408114, R2                                ; 1012
                               51     021E  8F  3C 00022  MOVZWL  #542, R1
                               50        53  D0 00027  MOVL    R3, R0
                                  00000000G  EF  16 0002A  JSB     EXCH$UTIL_BLOCK_CHECK
                                         28  A3  9F 00030  PUSHAB  40(R3)                                     ; 1017
                                             53  DD 00033  PUSHL   R3
                                             7E  D4 00035  CLRL    -(SP)
                                         14  A3  9F 00037  PUSHAB  20(R3)                                     ; 1016
                                       0000' CF  9F 0003A  PUSHAB  P.AAK
                   00000000G   EF        05  FB 0003E  CALLS   #5, EXCH$CMD_PARSE_FILESPEC                  ; 1017
                               56        50  D0 00045  MOVL    R0, STATUS
                               13        56  E8 00048  BLBS    STATUS, 1$
                                         50  13 0004B  BEQL    6$                                          ; 1020
                                     0048  8F  BB 0004D  PUSHR   #^M<R3,R6>                                 ; 1022
                                         01  DD 00051  PUSHL   #1
                                  00000000G  8F  DD 00053  PUSHL   #EXCH$_PARSEERR
                                     67      04  FB 00059  CALLS   #4, LIB$SIGNAL
                                             3F  11 0005C  BRB     6$                                          ; 1023
                                     52  28  A3  D0 0005E  1$:    MOVL    40(R3), R2                          ; 1029
                                         6D  A2  95 00062  TSTB    109(R2)
                                             04  19 00065  BLSS    2$
                                     04  6C  A2  E9 00067  BLBC    108(R2), 3$
                                20   A3      01  88 0006B  2$:    BISB2   #1, 32(R3)                          ; 1031
                                         74  A2  D5 0006F  3$:    TSTL    116(R2)                             ; 1035
                                             2C  12 00072  BNEQ    7$
                                     05  6B  A2  E8 00074  BLBS    107(R2), 4$                                 ; 1040
                            23   6A  A2      03  E0 00078  BBS     #3, 106(R2), 7$
                                     01  78  A2  91 0007D  4$:    CMPB    120(R2), #1                         ; 1043
                                             06  13 00081  BEQL    5$
                                     02  78  A2  91 00083  CMPB    120(R2), #2                                 ; 1045
                                             17  12 00087  BNEQ    7$
                                             52  DD 00089  5$:    PUSHL   R2                                  ; 1049
                      00000000G   EF         01  FB 0008B  CALLS   #1, EXCH$MOUN_IMPLIED_MOUNT
                                  56         50  D0 00092  MOVL    R0, STATUS
                                  08         56  E8 00095  BLBS    STATUS, 7$
                                             52  DD 00098  PUSHL   R2                                  ; 1052
                                     68      01  FB 0009A  CALLS   #1, EXCH$UTIL_NAMB_RELEASE
                                         0098  31  0009D  6$:    BRW     18$                                 ; 1053
                                     55  74  A2  D0 000A0  7$:    MOVL    116(R2), VOLB                       ; 1057
                                             06  13 000A4  BEQL    8$                                          ; 1058
                                     50  58  A5  9A 000A6  MOVZBL  88(VOLB), FORMAT                          ; 1060
```

```
                                         04  11 000AA       BRB     9$                               ;
                          50       7A    A2  9A 000AC 8$:   MOVZBL  122(R2), FORMAT                  ; 1062
               03         00             50  CF 000B0 9$:   CASEL   FORMAT, #0, #3                   ; 1064
    0050       001D       0036         0008     000B4 10$:  .WORD   11$-10$,-                        ;
                                                                    14$-10$,-                        ;
                                                                    12$-10$,-                        ;
                                                                    17$-10$                          ;
                          7E       EE    8F  9A 000BC 11$:  MOVZBL  #238, -(SP)                      ; 1109
                                         01  DD 000C0       PUSHL   #1                               ;
                             00000000G   8F  DD 000C2       PUSHL   #EXCH$_BADLOGIC                  ;
               00000000G  00             03  FB 000C8       CALLS   #3, LIB$STOP                     ;
                                         33  11 000CF       BRB     17$                              ;
               0B         6C       A2    06  E1 000D1 12$:  BBC     #6, 108(R2), 13$                 ; 1070
                          54 00000000G   8F  D0 000D6       MOVL    #EXCH$_NOREMOTE, ERRSTAT         ; 1073
                                         53  DD 000DD       PUSHL   R3                               ; 1074
                                         13  11 000DF       BRB     16$                              ;
                          54 00000000G   8F  D0 000E1 13$:  MOVL    #EXCH$_OPNOTPERF11, ERRSTAT      ; 1078
                                         07  11 000E8       BRB     15$                              ; 1079
                          54 00000000G   8F  D0 000EA 14$:  MOVL    #EXCH$_OPNOTPERDOS, ERRSTAT      ; 1089
                                   40    A2  9F 000F1 15$:  PUSHAB  64(R2)                           ; 1090
                                         01  DD 000F4 16$:  PUSHL   #1                               ;
                                         54  DD 000F6       PUSHL   ERRSTAT                          ;
                          67             03  FB 000F8       CALLS   #3, LIB$SIGNAL                   ;
                                         52  DD 000FB       PUSHL   R2                               ; 1091
                          68             01  FB 000FD       CALLS   #1, EXCH$UTIL_NAMB_RELEASE       ;
                          50             54  D0 00100       MOVL    ERRSTAT, R0                      ; 1092
                                             04 00103       RET                                      ;
                                   18    A2  9F 00104 17$:  PUSHAB  24(R2)                           ; 1116
                                   14    A3  9F 00107       PUSHAB  20(R3)                           ;
               00000000G  00             02  FB 0010A       CALLS   #2, STR$COPY_DX                  ;
               00000000G  EF             00  FB 00111       CALLS   #0, EXCH$UTIL_FILB_ALLOCATE      ; 1120
                      24  A3             50  D0 00118       MOVL    R0, 36(R3)                       ;
                          53       24    A3  D0 0011C       MOVL    36(R3), R3                       ; 1121
                                         0C  BB 00120       PUSHR   #^M<R2,R3>                       ;
               00000000G  EF             02  FB 00122       CALLS   #2, EXCH$COPY_NAMB_TO_FILB       ;
                                         53  DD 00129       PUSHL   R3                               ; 1125
                                         55  DD 0012B       PUSHL   VOLB                             ;
               00000000G  EF             02  FB 0012D       CALLS   #2, EXCH$UTIL_RT11CTX_ALLOCATE   ;
                      20  A3             50  D0 00134       MOVL    CTX, 32(R3)                      ; 1126
                          50             56  D0 00138 18$:  MOVL    STATUS, R0                       ; 1128
                                             04 0013B       RET                                      ; 1129
```

; Routine Size: 316 bytes,   Routine Base: EXCH$RTNAM_CODE + 076A

```
;  1046     1130  1 GLOBAL ROUTINE exch$rtnam_rename =      %SBTTL 'exch$rtnam_rename'
;  1047     1131  2 BEGIN
;  1048     1132  2 !++
;  1049     1133  2 !
;  1050     1134  2 ! FUNCTIONAL DESCRIPTION:
;  1051     1135  2 !
;  1052     1136  2 !       Entry routine for the RENAME verb for RT11 only
;  1053     1137  2 !
;  1054     1138  2 ! INPUTS:
;  1055     1139  2 !
;  1056     1140  2 !       none
;  1057     1141  2 !
;  1058     1142  2 ! IMPLICIT INPUTS:
;  1059     1143  2 !
;  1060     1144  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  1061     1145  2 !
;  1062     1146  2 ! OUTPUTS:
;  1063     1147  2 !
;  1064     1148  2 !       none
;  1065     1149  2 !
;  1066     1150  2 ! IMPLICIT OUTPUTS:
;  1067     1151  2 !
;  1068     1152  2 !       none
;  1069     1153  2 !
;  1070     1154  2 ! ROUTINE VALUE:
;  1071     1155  2 !
;  1072     1156  2 !       Success or worst error encountered.
;  1073     1157  2 !
;  1074     1158  2 ! SIDE EFFECTS:
;  1075     1159  2 !
;  1076     1160  2 !       Files may be renamed.
;  1077     1161  2 !--
;  1078     1162  2
;  1079     1163  2 $dbgtrc_prefix ('rtnam_rename> ');
;  1080     1164  2
;  1081     1165  2 LOCAL
;  1082     1166  2     protect,
;  1083     1167  2     rtnam : $ref_bblock,                        ! pointer to work area
;  1084     1168  2     status
;  1085     1169  2     ;
;  1086     1170  2
;  1087     1171  2 ! Allocate and/or initialize the work area
;  1088     1172  2 !
;  1089     1173  2 rtnam_init ();
;  1090     1174  2
;  1091     1175  2 ! Get pointers that we need.  Have to wait until work area allocated by init call
;  1092     1176  2 !
;  1093     1177  2 rtnam = .exch$a_gbl [excg$a_rtnam_work];              ! Pointer to work area
;  1094     1178  2 rtnam [rtnam$v_rename_command] = true;
;  1095     1179  2
;  1096     1180  2 ! For /PROTECT, we need to know whether it was specified or defaulted
;  1097     1181  2 !
;  1098     1182  2 protect = cli$present (%ASCID 'PROTECT');
;  1099     1183  2 rtnam [rtnam$v_q_protect]          = .protect;                    ! Simply value of low bit
;  1100     1184  4 rtnam [rtnam$v_q_protect_explicit] = ((.protect EQL cli$_present)    ! Either /PROTECT or /NOPROT
;  1101     1185  2                                     OR (.protect EQL cli$_negated));    !  must be there
;  1102     1186  2
```

```
;  1103          1187  2 ! Do the rest from the common routine
;  1104          1188  2 !
;  1105          1189  2 status = rtnam_common ();
;  1106          1190  2
;  1107          1191  2 IF .rtnam [rtnam$a_out_namb] NEQ 0
;  1108          1192  2 THEN
;  1109          1193  2     exch$util_namb_release (.rtnam [rtnam$a_out_namb]);          ! Return the namb to the available p
;  1110          1194  2
;  1111          1195  2 RETURN .status;
;  1112          1196  1 END;


                                                        .PSECT  EXCH$RTNAM_PLIT,NOWRT,2

                    00  54  43  45  54  4F  52  50  00050 P.AAN:   .ASCII  \PROTECT\<0>                                    ;
                                  010E0007  00058 P.AAM:   .LONG   17694727                                             ;
                                  00000000' 0005C          .ADDRESS P.AAN                                               ;

                                                        .EXTRN  CLI$_PRESENT, CLI$_NEGATED

                                                        .PSECT  EXCH$RTNAM_CODE,NOWRT,2

                                      001C 00000        .ENTRY  EXCH$RTNAM_RENAME, Save R2,R3,R4                     ; 1130
                FD5C    CF                00  FB 00002  CALLS   #0, RTNAM_INIT                                       ; 1173
                        50 00000000G      EF  D0 00007  MOVL    EXCH$A_GBL, R0                                       ; 1177
                        52          18    A0  D0 0000E  MOVL    24(R0), RTNAM                                        ;
                  20    A2                04  88 00012  BISB2   #4, 32(RTNAM)                                       ; 1178
                                0000'     CF  9F 00016  PUSHAB  P.AAM                                                ; 1182
                00000000G       00        01  FB 0001A  CALLS   #1, CLI$PRESENT                                     ;
     1C   A2              01                50  F0 00021  INSV    PROTECT, #1, #1, 28(RTNAM)                       ; 1183
                                            53  D4 00027  CLRL    R3                                               ; 1184
                00000000G       8F        50  D1 00029  CMPL    PROTECT, #CLI$_PRESENT                           ;
                                          02  12 00030  BNEQ    1$                                               ;
                                          53  D6 00032  INCL    R3                                               ;
                                          51  D4 00034  1$:     CLRL    R1                                       ; 1185
                00000000G       8F        50  D1 00036  CMPL    PROTECT, #CLI$_NEGATED                           ;
                                          02  12 0003D  BNEQ    2$                                               ;
                                          51  D6 0003F  INCL    R1                                               ;
                      54                  53  89 00041  2$:     BISB3   R3, R1, R4                               ;
     1C   A2          01                  54  F0 00045  INSV    R4, #2, #1, 28(RTNAM)                            ;
                    F70A    CF            00  FB 0004B  CALLS   #0, RTNAM_COMMON                                  ; 1189
                            53            50  D0 00050  MOVL    R0, STATUS                                       ;
                                    30    A2  D5 00053  TSTL    48(RTNAM)                                        ; 1191
                                          0A  13 00056  BEQL    3$                                               ;
                                    30    A2  DD 00058  PUSHL   48(RTNAM)                                        ; 1193
                00000000G       EF        01  FB 0005B  CALLS   #1, EXCH$UTIL_NAMB_RELEASE                        ;
                                50            53  D0 00062  3$:     MOVL    STATUS, R0                               ; 1195
                                              04 00065              RET                                              ; 1196

; Routine Size:  102 bytes,    Routine Base:  EXCH$RTNAM_CODE + 08A6
```

```
;  1114          1197  1 GLOBAL ROUTINE rtnam_rename_action (res_len, res_buf) = %SBTTL 'rtnam_rename_action (res_len, res_buf)'
;  1115          1198  2 BEGIN
;  1116          1199  2 !++
;  1117          1200  2 !
;  1118          1201  2 ! FUNCTIONAL DESCRIPTION:
;  1119          1202  2 !
;  1120          1203  2 !       Secondary action routine for the RENAME verb for RT11 only
;  1121          1204  2 !
;  1122          1205  2 ! INPUTS:
;  1123          1206  2 !
;  1124          1207  2 !       res_len - length of result name
;  1125          1208  2 !       res_buf - address of result name
;  1126          1209  2 !
;  1127          1210  2 ! IMPLICIT INPUTS:
;  1128          1211  2 !
;  1129          1212  2 !       Command parameters and qualifiers as returned from CLI$ routines.  Global environment ref'd by exch$
;  1130          1213  2 !
;  1131          1214  2 ! OUTPUTS:
;  1132          1215  2 !
;  1133          1216  2 !       none
;  1134          1217  2 !
;  1135          1218  2 ! IMPLICIT OUTPUTS:
;  1136          1219  2 !
;  1137          1220  2 !       none
;  1138          1221  2 !
;  1139          1222  2 ! ROUTINE VALUE:
;  1140          1223  2 !
;  1141          1224  2 !       Success or worst error encountered.
;  1142          1225  2 !
;  1143          1226  2 ! SIDE EFFECTS:
;  1144          1227  2 !
;  1145          1228  2 !       Files may be renamed.
;  1146          1229  2 !--
;  1147          1230  2
;  1148          1231  2 $dbgtrc_prefix ('rtnam_rename_action> ');
;  1149          1232  2
;  1150          1233  2 LOCAL
;  1151          1234  2     ctx2 : $ref_bblock,                         ! context block to look for duplicates
;  1152          1235  2     rfp : $bblock [nam$c_bln+nam$c_maxrss],     ! related file parse - an RMS '. M block plus expanded string
;  1153          1236  2     fin_buf : $bvector [filb$s_result_name],    ! a buffer in which to build the actual final name
;  1154          1237  2     fin_len,
;  1155          1238  2     nam_len,
;  1156          1239  2     status
;  1157          1240  2     ;
;  1158          1241  2
;  1159          1242  2 BIND
;  1160          1243  2     rtnam = exch$a_gbl [excg$a_rtnam_work] : $ref_bblock,       ! Pointer to work area
;  1161          1244  2     out_name = rtnam [rtnam$q_output_filename] : $desc_block,
;  1162          1245  2     namb = rtnam [rtnam$a_out_namb] : $ref_bblock,
;  1163          1246  2     filb = rtnam [rtnam$a_inp_filb] : $ref_bblock,
;  1164          1247  2     volb = filb [filb$a_assoc_volb] : $ref_bblock,
;  1165          1248  2     ctx = filb [filb$a_context] : $ref_bblock,
;  1166          1249  2     ent = ctx [rt11ctx$a_ent_address] : $ref_bblock
;  1167          1250  2     ;
;  1168          1251  2
;  1169          1252  2 $block_check (2, .filb, filb, 450);
;  1170          1253  2 $block_check (2, .volb, volb, 460);
```

```
;  1171          1254  2 $block_check (2, .ctx, rt11ctx, 444);                          ! Make sure that it is what we think it is
;  1172          1255  2
;  1173          1256  2 ! If we haven't fetched the output name, do so now.  This gives us a chance to carry the sticky device over
;  1174          1257  2 ! output name
;  1175          1258  2 !
;  1176          1259  2 IF NOT .rtnam [rtnam$v_out_fetched]
;  1177          1260  2 THEN
;  1178          1261  3     BEGIN
;  1179          1262  3     LOCAL
;  1180          1263  3         desc : VECTOR [2, LONG];
;  1181          1264  3
;  1182          1265  3     desc [0] = .res_len;
;  1183          1266  3     desc [1] = .res_buf;
;  1184          1267  3
;  1185          1268  3     ! Fetch the desired filename and a pointer to a namb
;  1186          1269  3     !
;  1187          1270  4     IF NOT (status = exch$cmd_parse_filespec (%ASCID 'OUTPUT', desc, 0, rtnam [rtnam$q_output_filename], nam
;  1188          1271  3     THEN
;  1189          1272  3         $exch_signal_return (exch$_parseerr, 1, rtnam [rtnam$q_output_filename], .status);
;  1190          1273  3     rtnam [rtnam$a_out_namb] = .namb;           ! Save the namb pointer
;  1191          1274  3     $trace_print_fao ("output "!AS", expanded "!AS"",  rtnam [rtnam$q_output_filename], namb [namb$q_fullnam
;  1192          1275  3
;  1193          1276  3     ! If we have been given an invalid filename, then choke
;  1194          1277  3     !
;  1195          1278  3     IF .namb [namb$v_bad_pdp_char]
;  1196          1279  3       OR
;  1197          1280  3        .namb [namb$v_rt_truncate]
;  1198          1281  3     THEN
;  1199          1282  3         $exch_signal_return (exch$_badfilename, 3, out_name, .volb [volb$l_vol_type_len], volb [volb$t_vol_t
;  1200          1283  3
;  1201          1284  3     rtnam [rtnam$v_out_fetched] = true;
;  1202          1285  3
;  1203          1286  2     END;
;  1204          1287  2
;  1205          1288  2 ! Perform a related file parse to get the final name
;  1206          1289  2 !
;  1207          1290  3 IF NOT (status = exch$cmd_related_file_parse (
;  1208          1291  3                  .out_name [dsc$w_length], .out_name [dsc$a_pointer],          ! Command line name string
;  1209          1292  3                  .res_len, .res_buf,                                           ! Related file name
;  1210          1293  3                  rfp))
;  1211          1294  2 THEN
;  1212          1295  2     $exch_signal_return (exch$_badfilename, 3, out_name, .volb [volb$l_vol_type_len], volb [volb$t_vol_type]
;  1213          1296  2
;  1214          1297  2 nam_len = .rfp [nam$b_name] + .rfp [nam$b_type];
;  1215          1298  2 fin_len = .volb [volb$l_vol_ident_len] + .nam_len;              ! Length of volume ident plus file name
;  1216          1299  2 $logic_check (2, (.fin_len [EQU filb$s_result_name), 161);
;  1217          1300  2 CH$COPY (.volb [volb$l_vol_ident_len], volb [volb$t_vol_ident],               ! Volume name
;  1218          1301  2                              .nam_len, .rfp [nam$l_name], 0, filb$s_result_name, fin_buf);
;  1219          1302  2
;  1220          1303  2 $trace_print_fao ('related "!AF", final "!AF"',  .res_len, .res_buf, .fin_len, fin_buf);
;  1221          1304  2
;  1222          1305  2 ! Make sure he isn't trying to move it between devices
;  1223          1306  2 !
;  1224          1307  2 IF .volb NEQ .namb [namb$a_assoc_volb]
;  1225          1308  2 THEN
;  1226          1309  2     $exch_signal_return (exch$_norendev);
;  1227          1310  2
```

```
; 1228          1311  2 ! We need to make sure that the output name doesn't already exist
; 1229          1312  2 !
; 1230          1313  2 ctx2 = exch$util_rt11ctx_allocate (.volb, 0);   ! Get a context block so we can look
; 1231          1314  2 IF exch$rtacp_find_file (.ctx2, .rfp [nam$l_name], .nam_len)
; 1232          1315  2   AND
; 1233          1316  2     .ent NEQA .ctx2 [rt11ctx$a_ent_address]      ! Let him rename it to the same, necessary for RENAME /PROTE
; 1234          1317  2 THEN
; 1235          1318  2     $exch_signal (exch$_norenexists, 4, .res_len, .res_buf, .fin_len, fin_buf)
; 1236          1319  2
; 1237          1320  2 ! Everything is ok, do the rename
; 1238          1321  2 !
; 1239          1322  2 ELSE
; 1240          1323  3     BEGIN
; 1241          1324  3
; 1242          1325  3     ! Convert the name to radix50 and put in into the directory buffer
; 1243          1326  3     !
; 1244          1327  3     exch$util_radix50_from_ascii (.rfp [nam$b_name], .rfp [nam$l_name],
; 1245          1328  3                                   rt11ctx$s_exp_name, ent [rt11ent$l_filename]);
; 1246          1329  3     exch$util_radix50_from_ascii (.rfp [nam$b_type] - 1, .rfp [nam$l_type] + 1,
; 1247          1330  3                                   rt11ctx$s_exp_type, ent [rt11ent$w_filetype]);
; 1248          1331  3
; 1249          1332  3     ! Also change the name in the context buffer, since it will be used by EXCH$RTACP_CHECK_POSITION
; 1250          1333  3     !
; 1251          1334  3     ctx [rt11ctx$l_filename] = .ent [rt11ent$l_filename];
; 1252          1335  3     ctx [rt11ctx$w_filetype] = .ent [rt11ent$w_filetype];
; 1253          1336  3
; 1254          1337  3     ! If a protection change was requested, do it here
; 1255          1338  3     !
; 1256          1339  3     IF .rtnam [rtnam$v_q_protect_explicit]
; 1257          1340  3     THEN
; 1258          1341  3         ent [rt11ent$v_typ_protected] = ctx [rt11ctx$v_typ_protected] = .rtnam [rtnam$v_q_protect];
; 1259          1342  3
; 1260          1343  3     ! Update the directory
; 1261          1344  3     !
; 1262          1345  3     exch$rt11_dirseg_put (.volb, .ctx [rt11ctx$l_seg_number]);          ! Write the directory segment
; 1263          1346  3
; 1264          1347  3     ! Log the action if requested
; 1265          1348  3     !
; 1266          1349  3     IF .rtnam [rtnam$v_q_log]
; 1267          1350  3     THEN
; 1268          1351  3         $exch_signal (exch$_renamed, 4, .res_len, .res_buf, .fin_len, fin_buf);
; 1269          1352  2     END;
; 1270          1353  2
; 1271          1354  2 ! Release the context block
; 1272          1355  2 !
; 1273          1356  2 exch$util_rt11ctx_release (.ctx2);
; 1274          1357  2
; 1275          1358  2 RETURN true;
; 1276          1359  1 END;


                                                                .PSECT  EXCH$RTNAM_PLIT,NOWRT,2

                     00  00  54  55  50  54  55  4F  00060 P.AAP:  .ASCII  \OUTPUT\<0><0>                    ;
                                                 010E0006  00068 P.AAO:  .LONG   17694726                    ;
                                                 00000000' 0006C         .ADDRESS P.AAP                      ;
```

```
                                                        .EXTRN  EXCH$_NORENEXISTS
                                                        .EXTRN  EXCH$_RENAMED

                                                        .PSECT  EXCH$RTNAM_CODE,NOWRT,2

                                        0FFC 00000      .ENTRY  RTNAM_RENAME_ACTION, Save R2,R3,R4,R5,R6,-      ; 1197
                                                                R7,R8,R9,R10,R11
                                5E      FD90    CE  9E 00002    MOVAB   -624(SP), SP
                50 00000000G    EF              18  C1 00007    ADDL3   #24, EXCH$A_GBL, R0                     ; 1243
                                56              60  D0 0000F    MOVL    (R0), R6                                ; 1244
                                53        0C    A6  9E 00012    MOVAB   12(R6), R3
                54        24    A6              1C  C1 00016    ADDL3   #28, 36(R6), R4                         ; 1247
                50        24    A6              20  C1 0001B    ADDL3   #32, 36(R6), R0                         ; 1248
                                58              60  D0 00020    MOVL    (R0), R8                                ; 1249
                                52 035B00FA     8F  D0 00023    MOVL    #56295674, R2                           ; 1252
                                51      01C2    8F  3C 0002A    MOVZWL  #450, R1
                                50        24    A6  D0 0002F    MOVL    36(R6), R0
                             00000000G          EF  16 00033    JSB     EXCH$UTIL_BLOCK_CHECK
                                57              64  D0 00039    MOVL    (R4), R7                                ; 1253
                                52 041B00F3     8F  D0 0003C    MOVL    #68878579, R2
                                51      01CC    8F  3C 00043    MOVZWL  #460, R1
                                50              57  D0 00048    MOVL    R7, R0
                             00000000G          EF  16 0004B    JSB     EXCH$UTIL_BLOCK_CHECK
                                52 008200F4     8F  D0 00051    MOVL    #8519924, R2                            ; 1254
                                51      01BC    8F  3C 00058    MOVZWL  #444, R1
                                50              58  D0 0005D    MOVL    R8, R0
                             00000000G          EF  16 00060    JSB     EXCH$UTIL_BLOCK_CHECK
                47        20    A6              03  E0 00066    BBS     #3, 32(R6), 2$                          ; 1259
                          08    AE        04    AC  7D 0006B    MOVQ    RES_LEN, DESC                           ; 1265
                                          30    A6  9F 00070    PUSHAB  48(R6)                                  ; 1270
                                                53  DD 00073    PUSHL   R3
                                                7E  D4 00075    CLRL    -(SP)
                                          14    AE  9F 00077    PUSHAB  DESC
                                        0000'   CF  9F 0007A    PUSHAB  P.AAO
                00000000G       EF              05  FB 0007E    CALLS   #5, EXCH$CMD_PARSE_FILESPEC
                                54              50  D0 00085    MOVL    R0, STATUS
                                16              54  E8 00088    BLBS    STATUS, 1$
                                52 00000000G    8F  D0 0008B    MOVL    #EXCH$_PARSEERR, TEMP                   ; 1272
                                                18  BB 00092    PUSHR   #^M<R3,R4>
                                                01  DD 00094    PUSHL   #1
                                                52  DD 00096    PUSHL   TEMP
                00000000G       00              04  FB 00098    CALLS   #4, LIB$SIGNAL
                                                46  11 0009F    BRB     4$
                                50        30    A6  D0 000A1 1$: MOVL    48(R6), R0                              ; 1273
                                24        6E    A0  E8 000A5    BLBS    110(R0), 3$                             ; 1278
                1F        6E    A0              02  E0 000A9    BBS     #2, 110(R0), 3$                         ; 1280
                          20    A6              08  88 000AE    BISB2   #8, 32(R6)                              ; 1284
                                          0110  CE  9F 000B2 2$: PUSHAB  RFP                                    ; 1290
                                7E        04    AC  7D 000B6    MOVQ    RES_LEN, -(SP)                          ; 1292
                                          04    A3  DD 000BA    PUSHL   4(R3)                                   ; 1291
                                7E              63  3C 000BD    MOVZWL  (R3), -(SP)
                00000000G       EF              05  FB 000C0    CALLS   #5, EXCH$CMD_RELATED_FILE_PARSE
                                54              50  D0 000C7    MOVL    R0, STATUS
                                1C              54  E8 000CA    BLBS    STATUS, 5$                              ; 1290
                                52 00000000G    8F  D0 000CD 3$: MOVL    #EXCH$_BADFILENAME, TEMP               ; 1295
                                          5D    A7  9F 000D4    PUSHAB  93(R7)
```

```
                                                59   A7  DD 000D7          PUSHL   89(R7)                                          ;
                                                     53  DD 000DA          PUSHL   R3                                              ;
                                                     03  DD 000DC          PUSHL   #3                                              ;
                                                     52  DD 000DE          PUSHL   TEMP                                            ;
                                00000000G    00          05  FB 000E0          CALLS   #5, LIB$SIGNAL                                  ;
                                                     71  11 000E7 4$:       BRB     8$                                              ;
                                             50      FEDB    CD  9A 000E9 5$:       MOVZBL  RFP+59, R0                              ; 1297
                                             51      FEDC    CD  9A 000EE           MOVZBL  RFP+60, R1                              ;
                                6E           50              51  C1 000F3           ADDL3   R1, R0, NAM_LEN                         ;
                            04  AE     65   A7              6E  C1 000F7           ADDL3   NAM_LEN, 101(R7), FIN_LEN               ; 1298
                                00000100     8F      04      AE  D1 000FD           CMPL    FIN_LEN, #256                           ; 1299
                                                     13  1B 00105          BLEQU   6$                                              ;
                                             7E      A1      8F  9A 00107          MOVZBL  #161, -(SP)                             ;
                                                     01  DD 0010B          PUSHL   #1                                              ;
                                         00000000G   8F  DD 0010D          PUSHL   #EXCH$_BADLOGIC                                 ;
                                00000000G    00          03  FB 00113          CALLS   #3, LIB$STOP                                    ;
                                             5B      FEEC    CD  D0 0011A 6$:       MOVL    RFP+76, R11                             ; 1301
                                             5A      0100    8F  3C 0011F           MOVZWL  #256, R10                               ; 1300
                                             59        10    AE  9E 00124           MOVAB   FIN_BUF, R9                             ;
                        5A          00       69  A7    65  A7  2C 00128          MOVC5   101(R7), 105(R7), #0, R10, (R9)         ;
                                                     69          0012F                                                              ;
                                                     0E  18 00130          BGEQ    7$                                              ;
                                             59      65  A7  C0 00132          ADDL2   101(R7), R9                             ;
                                             5A      65  A7  C2 00136          SUBL2   101(R7), R10                            ;
                        5A          00           6B          6E  2C 0013A          MOVC5   NAM_LEN, (R11), #0, R10, (R9)           ;
                                                     69          0013F                                                              ;
                                             50      30  A6  D0 00140 7$:       MOVL    48(R6), R0                              ; 1307
                                         74  A0              57  D1 00144          CMPL    R7, 116(R0)                             ;
                                                     14  13 00148          BEQL    9$                                              ;
                                         52  00000000G   8F  D0 0014A          MOVL    #EXCH$_NORENDEV, TEMP                   ; 1309
                                                     52  DD 00151          PUSHL   TEMP                                            ;
                                00000000G    00          01  FB 00153          CALLS   #1, LIB$SIGNAL                                  ;
                                             50          52  D0 00158 8$:       MOVL    TEMP, R0                                ;
                                                     04  00015D          RET                                                     ;
                                                     7E  D4 0015E 9$:       CLRL    -(SP)                                   ; 1313
                                                     57  DD 00160          PUSHL   R7                                              ;
                                00000000G    EF          02  FB 00162          CALLS   #2, EXCH$UTIL_RT11CTX_ALLOCATE                  ;
                                             53          50  D0 00169          MOVL    R0, CTX2                                ;
                                                     6E  DD 0016C          PUSHL   NAM_LEN                                 ; 1314
                                         0808        8F  BB 0016E          PUSHR   #^M<R3,R11>                             ;
                                00000000G    EF          03  FB 00172          CALLS   #3, EXCH$RTACP_FIND_FILE                        ;
                                             1B          50  E9 00179          BLBC    R0, 10$                                 ;
                                    7E   A3      7E      A8  D1 0017C          CMPL    126(R8), 126(CTX2)                      ; 1316
                                                     14  13 00181          BEQL    10$                                             ;
                                             10      AE  9F 00183          PUSHAB  FIN_BUF                                 ; 1318
                                             08      AE  DD 00186          PUSHL   FIN_LEN                                 ;
                                         7E  04      AC  7D 00189          MOVQ    RES_LEN, -(SP)                          ;
                                                     04  DD 0018D          PUSHL   #4                                              ;
                                             00000000G   8F  DD 0018F          PUSHL   #EXCH$_NORENEXISTS                              ;
                                                     73  11 00195          BRB     12$                                             ;
                                         52      7E      A8  D0 00197 10$:      MOVL    126(R8), R2                             ; 1328
                                             02      A2  9F 0019B          PUSHAB  2(R2)                                   ;
                                                     06  DD 0019E          PUSHL   #6                                              ;
                                                     5B  DD 001A0          PUSHL   R11                                             ;
                                         7E      FEDB    CD  9A 001A2          MOVZBL  RFP+59, -(SP)                           ;
                                00000000G    EF          04  FB 001A7          CALLS   #4, EXCH$UTIL_RADIX50_FROM_ASCII                ;
                                             06      A2  9F 001AE          PUSHAB  6(R2)                                   ; 1330
```

```
                                           03  DD 001B1          PUSHL   #3                                   ;
                  7E      FEF0   CD        01  C1 001B3          ADDL3   #1, RFP+80, -(SP)                    ; 1329
                                 7E  FEDC  CD  9A 001B9          MOVZBL  RFP+60, -(SP)                        ;
                                           6E  D7 001BE          DECL    (SP)                                 ;
                    00000000G    EF        04  FB 001C0          CALLS   #4, EXCH$UTIL_RADIX50_FROM_ASCII     ; 1330
                           3A    A8    02  A2  D0 001C7          MOVL    2(R2), 58(R8)                        ; 1334
                           3E    A8    06  A2  B0 001CC          MOVW    6(R2), 62(R8)                        ; 1335
                  12       1C    A6        02  E1 001D1          BBC     #2, 28(R6), 11$                      ; 1339
        50        1C      A6     01        01  EF 001D6          EXTZV   #1, #1, 28(R6), R0                   ; 1341
     39 A8                01     07        50  F0 001DC          INSV    R0, #7, #1, 57(R8)                   ;
     01 A2                01     07        50  F0 001E2          INSV    R0, #7, #1, 1(R2)                    ;
                                       76  A8  DD 001E8 11$:     PUSHL   118(R8)                              ; 1345
                                           57  DD 001EB          PUSHL   R7                                   ;
                    00000000G    EF        02  FB 001ED          CALLS   #2, EXCH$RT11_DIRSEG_PUT             ;
                                 19    1C  A6  E9 001F4          BLBC    28(R6), 13$                          ; 1349
                                       10  AE  9F 001F8          PUSHAB  FIN_BUF                              ; 1351
                                       08  AE  DD 001FB          PUSHL   FIN_LEN                              ;
                                 7E    04  AC  7D 001FE          MOVQ    RES_LEN, -(SP)                       ;
                                           04  DD 00202          PUSHL   #4                                   ;
                                00000000G  8F  DD 00204          PUSHL   #EXCH$_RENAMED                       ;
                    00000000G    00        06  FB 0020A 12$:     CALLS   #6, LIB$SIGNAL                       ;
                                           53  DD 00211 13$:     PUSHL   CTX2                                 ; 1356
                    00000000G    EF        01  FB 00213          CALLS   #1, EXCH$UTIL_RT11CTX_RELEASE        ;
                                 50        01  D0 0021A          MOVL    #1, R0                               ; 1358
                                               04 0021D          RET                                          ; 1359

; Routine Size:  542 bytes,    Routine Base:  EXCH$RTNAM_CODE + 090C
```

```
; 1278          1360  1 END
; 1279          1361  0 ELUDOM


                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                               PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  EXCH$RTNAM_PLIT                112  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  EXCH$RTNAM_CODE               2858  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        17         0        1000        00:01.9
;  _$255$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1  1151       139        12          79        00:01.3


;                                   COMMAND QUALIFIERS

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

; Size:         2858 code + 112 data bytes
; Run Time:       00:49.1
; Elapsed Time:   02:32.2
; Lines/CPU Min:   1662
; Lexemes/CPU-Min: 19137
; Memory Used:  217 pages
; Compilation Complete
```

EXCSHOW
LIS

F11A

F11AACP
MAP

EXCRTNAM
LIS

EXCRTACP
LIS

EXCUTIL
LIS