```
I 3
```

```
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

K 9

```
**FILE**ID**RENAME
```

```
RRRRRRRR    EEEEEEEEEE  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
RRRRRRRR    EEEEEEEEEE  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
RR       RR EE          NN      NN  AA      AA  MMMM  MMMM  EE
RR       RR EE          NN      NN  AA      AA  MMMM  MMMM  EE
RR       RR EE          NNNN    NN  AA      AA  MM  MM  MM  EE
RR       RR EE          NNNN    NN  AA      AA  MM  MM  MM  EE
RRRRRRRR    EEEEEEEE    NN  NN  NN  AA      AA  MM      MM  EEEEEEEE
RRRRRRRR    EEEEEEEE    NN  NN  NN  AA      AA  MM      MM  EEEEEEEE
RR   RR     EE          NN    NNNN  AAAAAAAAAA  MM      MM  EE
RR   RR     EE          NN    NNNN  AAAAAAAAAA  MM      MM  EE
RR    RR    EE          NN      NN  AA      AA  MM      MM  EE              ::::
RR    RR    EE          NN      NN  AA      AA  MM      MM  EE              ::::
RR      RR  EEEEEEEEEE  NN      NN  AA      AA  MM      MM  EEEEEEEEEE      ::::
RR      RR  EEEEEEEEEE  NN      NN  AA      AA  MM      MM  EEEEEEEEEE      ::::

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

```
0 MODULE RENAME ( %TITLE 'VAX/VMS RENAME Utility'
0                 IDENT = 'V04-000',
0                 MAIN = RENAME_MAIN
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:      VAX/VMS RENAME Utility
1 !
1 ! ABSTRACT:
1 !
1 !       This utility renames files and directories.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System, non-privileged user mode
1 !
1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 04-Oct-1982
1 !
1 !         Adapted from RENAME V03-000, written by Greg Robert
1 !
1 ! MODIFIED BY:
1 !
1 !       03-004  BLS0282         Benn Schreiber          5-MAR-1984
1 !               Use new argument to lib$delete_file for related file
1 !               processing.
1 !
1 !       3B-003  SBL3003         Steven B. Lionel        11-Feb-1983
1 !               /UIC is now /BYOWNER.
1 !
1 !       3B-002  SBL3002         Steven B. Lionel        19-Jan-1983
1 !               Remove special case RMS$_FEX code since $RENAME now
1 !               correctly handles this.
1 !
1 !       3B-001  SBL3001         Steven B. Lionel        4-Oct-1982
```

```
;    58         0058  1 !               Rewritten from RENAME V03-000 to use new CLI interface,
;    59         0059  1 !               LIB$RENAME_FILE, and file selection qualification routines.
;    60         0060  1 !
;    61         0061  1 !--
;    62         0062  1
```

```
  1 %SBTTL 'Declarations'
  1 !
  1 ! SWITCHES:
  1 !
  1
  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
  1
  1 !
  1 ! LINKAGES:
  1 !
  1 !      NONE
  1 !
  1 ! TABLE OF CONTENTS:
  1 !
  1
  1 FORWARD ROUTINE
  1      RENAME_MAIN,                                        ! Main routine
  1      CONFIRM_ROUTINE,                                    ! Confirm file to be renamed
  1      LOG_ROUTINE: NOVALUE,                               ! Log successful renames
  1      ERROR_ROUTINE,                                      ! Report errors
  1      ERROR_HANDLER;                                      ! Intercept signalled errors
  1
  1 !
  1 ! INCLUDE FILES:
  1 !
  1
  1 LIBRARY 'SYS$LIBRARY:STARLET';                           ! System symbols
  1
  1 !
  1 ! MACROS:
  1 !
  1 !      NONE
  1 !
  1 ! EQUATED SYMBOLS:
  1 !
  1
  1 !+
  1 ! Define shared messages needed by RENAME.
  1 !-
  1
P 1 $SHR_MSGDEF (MSG, 144, LOCAL,
P 1      (NOFILES, INFO),        ! No files selected
P 1      (OPENIN, ERROR),        ! Error opening input file
P 1      (OPENOUT, ERROR),       ! Error opening output file
P 1      (RENAMED, INFO),        ! File renamed
P 1      (SEARCHFAIL, ERROR)     ! Could not find file
  1      );
  1
  1 !
  1 ! FIELDS:
  1 !
  1
  1 !+
  1 ! Data structure LOCAL_BLOCK (defined by LB_FIELDS) is passed as the
  1 ! "user_argument" to LIB$RENAME_FILE, and thus to each of the action
  1 ! routines.
  1 !-
```

```
1
1 FIELD
1     LB_FIELDS =
1         SET
1
1         LB_L_WORST_ERROR        = [0,0,32,0],    ! Worst error so far
1         LB_L_QUAL_CONTEXT       = [4,0,32,0],    ! Context for LIB$QUAL_xxx
1         LB_L_FLAGS              = [8,0,32,0],    ! Flags
1         LB_V_LOG                = [8,0,1,0],     ! Set if /LOG specified
1         LB_V_FILE_SELECTED      = [8,1,1,0]      ! At least one file was renamed
1
1         TES;
1 LITERAL
1     LB_K_SIZE = 12;                              ! Size of LOCAL_BLOCK in bytes
1
1 !
1 ! PSECTS:
1 !
1
1 PSECT
1     CODE = $CODE (READ, NOWRITE, EXECUTE, SHARE, PIC,
1         ADDRESSING_MODE (WORD_RELATIVE)),
1     PLIT = $CODE (READ, NOWRITE, EXECUTE, SHARE, PIC,
1         ADDRESSING_MODE (WORD_RELATIVE)),
1     OWN  = $DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC,
1         ADDRESSING_MODE (LONG_RELATIVE)),
1     GLOBAL = $DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC,
1         ADDRESSING_MODE (LONG_RELATIVE));
1
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     CLI$GET_VALUE,                               ! Get value from command line
1     CLI$PRESENT,                                 ! Is qualifier present?
1     LIB$RENAME_FILE,                             ! Rename files
1     LIB$QUAL_FILE_MATCH,                         ! Match selection qualifiers
1     LIB$QUAL_FILE_PARSE;                         ! Parse selection qualifiers
1
1 EXTERNAL LITERAL
1     LIB$_FILFAIMAT,                              ! File failed to match qualifiers
1     LIB$_QUICONACT,                              ! Quit confirm action
1     LIB$_QUIPRO,                                 ! Quit processing
1     RENAME$_NOTRENAMED,                          ! File not renamed
1     RENAME$_NOTSAMEDEV;                          ! Can't rename across devices
1
```

```
%SBTTL 'RENAME_MAIN - Main routine'
ROUTINE RENAME_MAIN =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This is the main routine for RENAME.  It calls LIB$QUAL_FILE_PARSE
!       to parse the file selection qualifiers, sets up the local storage
!       block, and calls LIB$RENAME_FILE to do the work.  LIB$RENAME_FILE
!       will call other routines in this module as action routines to
!       process the command.  Upon return from LIB$RENAME_FILE, the most
!       severe error status will be returned as the exit status of RENAME
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = RENAME ()
!
! FORMAL PARAMETERS:
!
!       NONE
!
! IMPLICIT INPUTS:
!
!       Command line
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!       RENAME$_NOFILES No files selected
!       Other           Worst error encountered
!
! SIDE EFFECTS:
!
!       Renames one or more files.
!
!--

    BEGIN

    LOCAL
        LOCAL_BLOCK: VOLATILE BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS),
        QUAL_FLAGS,                             ! Qualifier select flags
        RENAME_FLAGS,                           ! Flags for LIB$RENAME_FILE
        OLD_NAME_DSC: BLOCK [8, BYTE],          ! Old name descriptor
        NEW_NAME_DSC: BLOCK [8, BYTE],          ! New name descriptor
        OLD_RSN_DSC: BLOCK [8, BYTE],           ! Old RSN descriptor
        NEW_RSN_DSC: BLOCK [8, BYTE],           ! New RSN descriptor
        CLI_STRING_DSC: BLOCK [8, BYTE],        ! Qualifier and parameter names
        FILESCAN_CONTEXT,                       ! Related file context pointer
        STATUS1;                                ! Return statuse

    !+
    ! Enable error handler which will keep track of the worst error
! encountered so far.  It will also unwind to the caller of RENAME
! (the CLI) if the status LIB$_QUIPRO is signalled.  Note that because
! LOCAL_BLOCK is a VOLATILE enable argument, BLISS zeroes it for us.
!-

ENABLE
    ERROR_HANDLER (LOCAL_BLOCK);

!+
! Set up strings and descriptors.  All strings are dynamic.
!-

FILESCAN_CONTEXT = 0;
OLD_RSN_DSC [DSC$W_LENGTH] = 0;
OLD_RSN_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
OLD_RSN_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;
OLD_RSN_DSC [DSC$A_POINTER] = 0;
NEW_RSN_DSC [DSC$W_LENGTH] = 0;
NEW_RSN_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
NEW_RSN_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;
NEW_RSN_DSC [DSC$A_POINTER] = 0;
OLD_NAME_DSC [DSC$W_LENGTH] = 0;                ! Initially zero length
OLD_NAME_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;     ! Text
OLD_NAME_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;     ! Dynamic
OLD_NAME_DSC [DSC$A_POINTER] = 0;               ! No allocation
NEW_NAME_DSC [DSC$W_LENGTH] = 0;                ! Initially zero length
NEW_NAME_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;     ! Text
NEW_NAME_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;     ! Dynamic
NEW_NAME_DSC [DSC$A_POINTER] = 0;               ! No allocation
CLI_STRING_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
CLI_STRING_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;

!+
! Indicate which qualifiers we want parsed.
!-

QUAL_FLAGS = LIB$M_CQF_CONFIRM +
             LIB$M_CQF_EXCLUDE +
             LIB$M_CQF_BEFORE +
             LIB$M_CQF_SINCE +
             LIB$M_CQF_CREATED +
             LIB$M_CQF_MODIFIED +
             LIB$M_CQF_EXPIRED +
             LIB$M_CQF_BACKUP +
             LIB$M_CQF_BYOWNER;

!+
! Parse file selection qualifiers and save context.
!-

LIB$QUAL_FILE_PARSE (QUAL_FLAGS, LOCAL_BLOCK [LB_L_QUAL_CONTEXT]);

!+
! See if /LOG was specified.
!-

CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('LOG');
CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('LOG');
IF CLI$PRESENT (CLI_STRING_DSC)
THEN
    LOCAL_BLOCK [LB_V_LOG] = 1;

!+
! See if /NONEW_VERSION was specified.
!-

CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('NEW_VERSION');
CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('NEW_VERSION');
RENAME_FLAGS = 0;
IF NOT CLI$PRESENT (CLI_STRING_DSC)
THEN
    RENAME_FLAGS = 1;           ! SET V_NONEW_VERSION bit

!+
! Get new filename, since there can only be one.
!-

CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('P2');
CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('P2');
CLI$GET_VALUE (CLI_STRING_DSC, NEW_NAME_DSC);

!+
! Until all "old" filespecs are processed, get the next old filespec
! and call LIB$RENAME_FILE for each one.  Use the previous old resultant
! filename as the related filename for each iteration.
!-

CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('P1');
CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('P1');
WHILE (CLI$GET_VALUE (CLI_STRING_DSC, OLD_NAME_DSC)) DO
    BEGIN
    STATUS1 = LIB$RENAME_FILE (
        OLD_NAME_DSC,                   ! Old name
        NEW_NAME_DSC,                   ! New name
        0,                              ! No default name
        0,                              ! Related name - previous old RSN
        RENAME_FLAGS,                   ! Rename flags
        LOG_ROUTINE,                    ! Log routine
        ERROR_ROUTINE,                  ! Error routine
        CONFIRM_ROUTINE,                ! Confirm routine
        LOCAL_BLOCK,                    ! User argument
        OLD_RSN_DSC,                    ! Old resultant name
        NEW_RSN_DSC,                    ! New resultant name
        FILESCAN_CONTEXT);              ! Related file context

    IF NOT .STATUS1
    THEN
        SIGNAL_STOP (.STATUS1);
    END;

!+
! If an error occurred, return the worst error.
! If no files were selected, return an informational to that effect.
!-
```

```
;  344      0342 2
;  345      0343 2        IF .LOCAL_BLOCK [LB_L_WORST_ERROR] NEQ 0
;  346      0344 2        THEN
;  347      0345 2            RETURN .LOCAL_BLOCK [LB_L_WORST_ERROR];
;  348      0346 2
;  349      0347 2        IF NOT .LOCAL_BLOCK [LB_V_FILE_SELECTED]
;  350      0348 2        THEN
;  351      0349 3            BEGIN
;  352      0350 3            SIGNAL (MSG$_NOFILES);  ! Display message "no files selected"
;  353      0351 3            RETURN MSG$_NOFILES;    ! Won't get displayed again
;  354      0352 2            END;
;  355      0353 2
;  356      0354 2        RETURN SS$_NORMAL;
;  357      0355 2
;  358      0356 1        END;                                    ! End of routine RENAME_MAIN


                                                                .TITLE  RENAME VAX/VMS RENAME Utility
                                                                .IDENT  \V04-000\

                                                                .PSECT  $CODE,NOWRT,  SHR,  PIC,2

                                      47  4F  4C  00000 P.AAA:  .ASCII  \LOG\                          ;
          4E  4F  49  53  52  45  56  5F  57  45  4E  00003 P.AAB:  .ASCII  \NEW_VERSION\                  ;
                                          32  50  0000E P.AAC:  .ASCII  \P2\                           ;
                                          31  50  00010 P.AAD:  .ASCII  \P1\                           ;

                                                                .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                                                                .EXTRN  LIB$RENAME_FILE
                                                                .EXTRN  LIB$QUAL_FILE_MATCH
                                                                .EXTRN  LIB$QUAL_FILE_PARSE
                                                                .EXTRN  LIB$_FILFAIMAT, LIB$_QUICONACT
                                                                .EXTRN  LIB$_QUIPRO, RENAME$_NOTRENAMED
                                                                .EXTRN  RENAME$_NOTSAMEDEV

                                        001C 00000 RENAME_MAIN:
                                                                .WORD   Save R2,R3,R4                  ; 0172
                    54 00000000G  00  9E 00002                  MOVAB   CLI$GET_VALUE, R4              ;
                    53 00000000G  00  9E 00009                  MOVAB   CLI$PRESENT, R3                ;
                    5E            38  C2 00010                  SUBL2   #56, SP                        ;
                              2C  AE  7C 00013                  CLRQ    LOCAL_BLOCK                    ; 0212
                              34  AE  D4 00016                  CLRL    LOCAL_BLOCK                    ;
                    6D      0103  CF  DE 00019                  MOVAL   7$, (FP)                       ;
                              7E  D4  0001E                     CLRL    FILESCAN_CONTEXT               ; 0240
              18    AE 020E0000   8F  D0 00020                  MOVL    #34471936, OLD_RSN_DSC         ; 0241
                              1C  AE  D4 00028                  CLRL    OLD_RSN_DSC+4                  ; 0244
              10    AE 020E0000   8F  D0 0002B                  MOVL    #34471936, NEW_RSN_DSC         ; 0245
                              14  AE  D4 00033                  CLRL    NEW_RSN_DSC+4                  ; 0248
              28    AE 020E0000   8F  D0 00036                  MOVL    #34471936, OLD_NAME_DSC        ; 0249
                              2C  AE  D4 0003E                  CLRL    OLD_NAME_DSC+4                 ; 0252
              20    AE 020E0000   8F  D0 00041                  MOVL    #34471936, NEW_NAME_DSC        ; 0253
                              24  AE  D4 00049                  CLRL    NEW_NAME_DSC+4                 ; 0256
              0A    AE      010E  8F  B0 0004C                  MOVW    #270, CLI_STRING_DSC+2         ; 0257
                    7E      01FF  8F  3C 00052                  MOVZWL  #511, QUAL_FLAGS               ; 0271
                              38  AE  9F 00057                  PUSHAB  LOCAL_BLOCK+4                  ; 0278
                              04  AE  9F 0005A                  PUSHAB  QUAL_FLAGS                     ;
          00000000G  00           02  FB 0005D                  CALLS   #2, LIB$QUAL_FILE_PARSE        ;
```

```
            0C AE          03 B0 00064        MOVW    #3, CLI_STRING_DSC         ; 0284
            10 AE       83 AF 9E 00068        MOVAB   P.AAA, CLI_STRING_DSC+4    ; 0285
                        0C AE 9F 0006D        PUSHAB  CLI_STRING_DSC             ; 0286
               63          01 FB 00070        CALLS   #1, CLI$PRESENT            ;
               04          50 E9 00073        BLBC    R0, 1$                     ;
            3C AE          01 88 00076        BISB2   #1, LOCAL_BLOCK+8          ; 0288
            0C AE          0B B0 0007A 1$:    MOVW    #11, CLI_STRING_DSC        ; 0294
            10 AE     FF6F CF 9E 0007E        MOVAB   P.AAB, CLI_STRING_DSC+4    ; 0295
                        08 AE D4 00084        CLRL    RENAME_FLAGS               ; 0296
                        0C AE 9F 00087        PUSHAB  CLI_STRING_DSC             ; 0297
               63          01 FB 0008A        CALLS   #1, CLI$PRESENT            ;
               04          50 E8 0008D        BLBS    R0, 2$                     ;
            08 AE          01 D0 00090        MOVL    #1, RENAME_FLAGS           ; 0299
            0C AE          02 B0 00094 2$:    MOVW    #2, CLI_STRING_DSC         ; 0305
            10 AE     FF60 CF 9E 00098        MOVAB   P.AAC, CLI_STRING_DSC+4    ; 0306
                        24 AE 9F 0009E        PUSHAB  NEW_NAME_DSC               ; 0307
                        10 AE 9F 000A1        PUSHAB  CLI_STRING_DSC             ;
               64          02 FB 000A4        CALLS   #2, CLI$GET_VALUE          ;
            0C AE          02 B0 000A7        MOVW    #2, CLI_STRING_DSC         ; 0315
            10 AE     FF4F CF 9E 000AB        MOVAB   P.AAD, CLI_STRING_DSC+4    ; 0316
                        2C AE 9F 000B1 3$:    PUSHAB  OLD_NAME_DSC               ; 0317
                        10 AE 9F 000B4        PUSHAB  CLI_STRING_DSC             ;
               64          02 FB 000B7        CALLS   #2, CLI$GET_VALUE          ;
               3B          50 E9 000BA        BLBC    R0, 4$                     ;
                        04 AE 9F 000BD        PUSHAB  FILESCAN_CONTEXT           ; 0319
                        18 AE 9F 000C0        PUSHAB  NEW_RSN_DSC                ;
                        24 AE 9F 000C3        PUSHAB  OLD_RSN_DSC                ;
                        40 AE 9F 000C6        PUSHAB  LOCAL_BLOCK                ;
                     0000V CF 9F 000C9        PUSHAB  CONFIRM_ROUTINE            ;
                     0000V CF 9F 000CD        PUSHAB  ERROR_ROUTINE              ;
                     0000V CF 9F 000D1        PUSHAB  LOG_ROUTINE                ;
                        24 AE 9F 000D5        PUSHAB  RENAME_FLAGS               ;
                           7E 7C 000D8        CLRQ    -(SP)                      ;
                        4C AE 9F 000DA        PUSHAB  NEW_NAME_DSC               ;
                        58 AE 9F 000DD        PUSHAB  OLD_NAME_DSC               ;
     00000000G 00          0C FB 000E0        CALLS   #12, LIB$RENAME_FILE       ;
               52          50 D0 000E7        MOVL    R0, STATUS1                ;
               C4          52 E8 000EA        BLBS    STATUS1, 3$                ; 0333
                           52 DD 000ED        PUSHL   STATUS1                    ; 0335
     00000000G 00          01 FB 000EF        CALLS   #1, LIB$STOP               ;
                           B9 11 000F6        BRB     3$                         ; 0317
                        34 AE D5 000F8 4$:    TSTL    LOCAL_BLOCK                ; 0343
                           05 13 000FB        BEQL    5$                         ;
               50       34 AE D0 000FD        MOVL    LOCAL_BLOCK, R0            ; 0345
                              04 00101        RET                                ;
  15        3C AE          01 E0 00102 5$:    BBS     #1, LOCAL_BLOCK+8, 6$      ; 0347
                  00901333 8F DD 00107        PUSHL   #9442099                   ; 0350
     00000000G 00          01 FB 0010D        CALLS   #1, LIB$SIGNAL             ;
               50 00901333 8F D0 00114        MOVL    #9442099, R0               ; 0351
                              04 0011B        RET                                ;
               50          01 D0 0011C 6$:    MOVL    #1, R0                     ; 0354
                              04 0011F        RET                                ; 0356
                            0000 00120 7$:    .WORD   Save nothing               ; 0212
               50       08 AC D0 00122        MOVL    8(AP), R0                  ;
               50       04 A0 D0 00126        MOVL    4(R0), R0                  ;
                        F4 A0 9F 0012A        PUSHAB  LOCAL_BLOCK                ;
                           01 DD 0012D        PUSHL   #1                         ;
```

```
                                          5E  DD 0012F          PUSHL   SP                                          ;
                     7E          04       AC  7D 00131          MOVQ    4(AP), -(SP)                                ;
               0000V  CF                  03  FB 00135          CALLS   #3, ERROR_HANDLER                           ;
                                              04 0013A          RET                                                 ;

; Routine Size:  315 bytes,    Routine Base:  $CODE + 0012
```

```
%SBTTL 'CONFIRM_ROUTINE - Confirm that rename should be done'
ROUTINE CONFIRM_ROUTINE (
    OLD_RSN: REF BLOCK [, BYTE],            ! Old resultant string descriptor
    NEW_RSN: REF BLOCK [, BYTE],            ! New resultant string descriptor
    FAB: REF BLOCK [, BYTE],                ! File's FAB
    LOCAL_BLOCK: REF BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS)
    ) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine is called by LIB$RENAME_FILE to confirm that the
!       file described by FAB should be renamed.
!
! CALLING SEQUENCE:
!
!       confirm.wv.v = CONFIRM_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
!                           FAB.rr.r, LOCAL_BLOCK.mr.r)
!
! FORMAL PARAMETERS:
!
!       OLD_RSN         - Old resultant name.
!
!       NEW_RSN         - New resultant name.
!
!       FAB             - FAB that describes file to be renamed.
!
!       LOCAL_BLOCK     - Block of local information.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
!
!       TRUE if file should be renamed
!       FALSE if file should not be renamed
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    LOCAL
        PROMPT_ARGS: VECTOR [2, LONG],              ! FAO argument list for prompt
        PROMPT_DSC: BLOCK [8, BYTE],                ! Descriptor for prompt string
        MATCH_STATUS;                               ! Status from QUAL_FILE_MATCH

    !+
    ! Set up PROMPT_ARGS and PROMPT_DSC.
```

```
;  417      0414  2      !-
;  418      0415  2
;  419      0416  2      PROMPT_ARGS [0] = OLD_RSN [0,0,0,0];          ! Descriptor address
;  420      0417  2      PROMPT_ARGS [1] = NEW_RSN [0,0,0,0];          ! Descriptor address
;  421      0418  2      PROMPT_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  422      0419  2      PROMPT_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  423      0420  2      PROMPT_DSC [DSC$W_LENGTH] = %CHARCOUNT ('!AS --> !AS ? (Y,N,Q,All): ');
;  424      0421  2      PROMPT_DSC [DSC$A_POINTER] = UPLIT BYTE ('!AS --> !AS ? (Y,N,Q,All): ');
;  425      0422  2
;  426      0423  2      !+
;  427      0424  2      ! Call LIB$QUAL_FILE_MATCH to see if this file meets the criteria.
;  428      0425  2      !-
;  429      0426  2
;  430      0427  2      MATCH_STATUS = LIB$QUAL_FILE_MATCH (
;  431      0428  2          LOCAL_BLOCK [LB_L_QUAL_CONTEXT],
;  432      0429  2          FAB [0,0,0,0],
;  433      0430  2          0,                  ! File name omitted
;  434      0431  2          PROMPT_DSC,
;  435      0432  2          PROMPT_ARGS,
;  436      0433  2          0);                 ! Prompt routine omitted
;  437      0434  2
;  438      0435  2      !+
;  439      0436  2      ! If a failure status other than "failed to match" was returned,
;  440      0437  2      ! signal it.  If it is LIB$_QUIPRO, ERROR_HANDLER will unwind back to
;  441      0438  2      ! the CLI with a success status.
;  442      0439  2      !-
;  443      0440  2
;  444      0441  2      IF NOT .MATCH_STATUS
;  445      0442  2      THEN
;  446      0443  2          IF .MATCH_STATUS NEQU LIB$_FILFAIMAT
;  447      0444  2          THEN
;  448      0445  3              IF (.MATCH_STATUS EQLU LIB$_QUIPRO) OR (.FAB [FAB$L_STS])
;  449      0446  3              THEN
;  450      0447  2                  SIGNAL (.MATCH_STATUS)
;  451      0448  2              ELSE
;  452      0449  3                  BEGIN
;  453      0450  3                  !+
;  454      0451  3                  ! LIB$QUAL_FILE_MATCH couldn't open the file.
;  455      0452  3                  ! Signal "error opening file".
;  456      0453  3                  !-
;  457      0454  3
;  458      0455  3                  LOCAL
;  459      0456  3                      NAM: REF BLOCK [, BYTE],
;  460      0457  3                      RSN_DESCR: BLOCK [8, BYTE];
;  461      0458  3                  NAM = .FAB [FAB$L_NAM];
;  462      0459  3                  RSN_DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  463      0460  3                  RSN_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  464      0461  3                  RSN_DESCR [DSC$W_LENGTH] = .NAM [NAM$B_RSL];
;  465      0462  3                  RSN_DESCR [DSC$A_POINTER] = .NAM [NAM$L_RSA];
;  466      0463  3                  SIGNAL (MSG$_OPENIN, 1, RSN_DESCR,
;  467      0464  3                      .FAB [FAB$L_STS], .FAB [FAB$L_STV]);
;  468      0465  2                  END;
;  469      0466  2
;  470      0467  2      !+
;  471      0468  2      ! If this file was selected, indicate that in LOCAL_BLOCK.
;  472      0469  2      !-
;  473      0470  2
;  474          0471  2        IF .MATCH_STATUS
;  475          0472  2        THEN
;  476          0473  2            LOCAL_BLOCK [LB_V_FILE_SELECTED] = 1;
;  477          0474  2
;  478          0475  2        !+
;  479          0476  2        ! Return to LIB$RENAME_FILE with match status.
;  480          0477  2        !-
;  481          0478  2
;  482          0479  2        RETURN .MATCH_STATUS;
;  483          0480  2
;  484          0481  1        END;                                   ! End of routine CONFIRM_ROUTINE



28  20  3F  20  53  41  21  20  3E  2D  2D  20  53  41  21  0014D P.AAE:   .ASCII  \!AS --> !AS ? (Y,N,Q,All): \         ;
            20  3A  29  6C  6C  41  2C  51  2C  4E  2C  59  0015C                                                          ;



                                             003C 00000 CONFIRM_ROUTINE:
                                                                   .WORD   Save R2,R3,R4,R5                              ; 0358
                         55 00000000G  00  9E 00002                MOVAB   LIB$SIGNAL, R5                                ;
                         5E            18  C2 00009                SUBL2   #24, SP                                       ;
                    10   AE        04  AC  7D 0000C                MOVQ    OLD_RSN, PROMPT_ARGS                          ; 0416
                    08   AE 010E001B  8F  D0 00011                MOVL    #17694747, PROMPT_DSC                         ; 0420
                    0C   AE        C9  AF  9E 00019                MOVAB   P.AAE, PROMPT_DSC+4                           ; 0421
                                       7E  D4 0001E                CLRL    -(SP)                                         ; 0429
                                   14  AE  9F 00020                PUSHAB  PROMPT_ARGS                                   ;
                                   10  AE  9F 00023                PUSHAB  PROMPT_DSC                                    ;
                                       7E  D4 00026                CLRL    -(SP)                                         ;
                         52        0C  AC  7D 00028                MOVQ    FAB, R2                                       ;
                                       52  DD 0002C                PUSHL   R2                                            ;
                                   04  A3  9F 0002E                PUSHAB  4(R3)                                         ; 0428
               00000000G 00            06  FB 00031                CALLS   #6, LIB$QUAL_FILE_MATCH                       ; 0429
                         54            50  D0 00038                MOVL    R0, MATCH_STATUS                              ;
                         45            54  E8 0003B                BLBS    MATCH_STATUS, 4$                              ; 0441
               00000000G 8F            54  D1 0003E                CMPL    MATCH_STATUS, #LIB$_FILFAIMAT                 ; 0443
                                       39  13 00045                BEQL    3$                                            ;
               00000000G 8F            54  D1 00047                CMPL    MATCH_STATUS, #LIB$_QUIPRO                    ; 0445
                                       04  13 0004E                BEQL    1$                                            ;
                         07        08  A2  E9 00050                BLBC    8(R2), 2$                                     ;
                                       54  DD 00054 1$:            PUSHL   MATCH_STATUS                                  ; 0447
                         65            01  FB 00056                CALLS   #1, LIB$SIGNAL                                ;
                                       25  11 00059                BRB     3$                                            ;
                         50        28  A2  D0 0005B 2$:            MOVL    40(R2), NAM                                   ; 0458
                    02   AE      010E  8F  B0 0005F                MOVW    #270, RSN_DESCR+2                             ; 0459
                         6E        03  A0  9B 00065                MOVZBW  3(NAM), RSN_DESCR                             ; 0461
                    04   AE        04  A0  D0 00069                MOVL    4(NAM), RSN_DESCR+4                           ; 0462
                         7E        08  A2  7D 0006E                MOVQ    8(R2), -(SP)                                  ; 0464
                                   08  AE  9F 00072                PUSHAB  RSN_DESCR                                     ; 0463
                                       01  DD 00075                PUSHL   #1                                            ;
                            0090109A  8F  DD 00077                PUSHL   #9441434                                      ;
                         65            05  FB 0007D                CALLS   #5, LIB$SIGNAL                                ;
                         04            54  E9 00080 3$:            BLBC    MATCH_STATUS, 5$                              ; 0471
                    08   A3            02  88 00083 4$:            BISB2   #2, 8(R3)                                     ; 0473
                         50            54  D0 00087 5$:            MOVL    MATCH_STATUS, R0                              ; 0479
                                           04 0008A                RET                                                   ; 0481
```

```
; Routine Size:  139 bytes,    Routine Base:  $CODE + 0168
```

```
1 %SBTTL 'LOG_ROUTINE - Log successful rename'
1 ROUTINE LOG_ROUTINE (
1     OLD_RSN: REF BLOCK [, BYTE],          ! Old resultant name
1     NEW_RSN: REF BLOCK [, BYTE],          ! New resultant name
1     LOCAL_BLOCK: REF BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS)
1     ): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called by LIB$RENAME_FILE after a successful
1 !       rename.  If /LOG was specified, a message indicating the names
1 !       of the files renamed is displayed.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       CALL LOG_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
1 !                         LOCAL_BLOCK.mr.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OLD_RSN         - Old resultant name.
1 !
1 !       NEW_RSN         - New resultant name.
1 !
1 !       LOCAL_BLOCK     - Block of local information.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     !+
2     ! If /LOG was specified, log the rename.
2     !-
2
2     IF .LOCAL_BLOCK [LB_V_LOG]
2     THEN
2         SIGNAL (MSG$_RENAMED, 2, OLD_RSN [0,0,0,0], NEW_RSN [0,0,0,0]);
2
2     RETURN;
2
```

```
;   543         0539  1     END;                                          ! End of routine LOG_ROUTINE



                                              0000 00000 LOG_ROUTINE:
                                                                   .WORD   Save nothing                                    ; 0483
                        50       0C    AC  D0 00002                MOVL    LOCAL_BLOCK, R0                                 ; 0533
                        13       08    A0  E9 00006                BLBC    8(R0), 1$                                       ;
                        7E       04    AC  7D 0000A                MOVQ    OLD_RSN, -(SP)                                  ; 0535
                                       02  DD 0000E                PUSHL   #2                                              ;
                          009012A3     8F  DD 00010                PUSHL   #9441955                                        ;
               00000000G  00           04  FB 00016                CALLS   #4, LIB$SIGNAL                                  ;
                                           04 0001D 1$:            RET                                                     ; 0539

; Routine Size:  30 bytes,     Routine Base:  $CODE + 01F3
```

```
1 %SBTTL 'ERROR_ROUTINE - Report failed rename'
1 ROUTINE ERROR_ROUTINE (
1     OLD_RSN: REF BLOCK [, BYTE],        ! Old resultant name
1     NEW_RSN: REF BLOCK [, BYTE],        ! New resultant name
1     RMS_STS: REF BLOCK [, BYTE],        ! RMS STS status code
1     RMS_STV: REF BLOCK [, BYTE],        ! RMS STV status code
1     ERROR_SOURCE: REF VECTOR [, LONG],  ! Source of error code
1     LOCAL_BLOCK: REF BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS)
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called by LIB$RENAME_FILE after a failed
1 !       rename.  The error message is signalled.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       status.wlc.v = ERROR_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
1 !                       RMS_STS.rlc.r, RMS_STV.rlc.r,
1 !                       ERROR_SOURCE.rl.r, LOCAL_BLOCK.mr.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OLD_RSN         - Old resultant name.
1 !
1 !       NEW_RSN         - New resultant name.
1 !
1 !       RMS_STS         - The RMS STS condition code
1 !
1 !       RMS_STV         - The RMS STV condition code
1 !
1 !       ERROR_SOURCE    - An integer that indicates where the error occurred.
1 !                               0 = Error from LIB$FILE_SCAN
1 !                               1 = Error from $PARSE
1 !                               2 = Error from $RENAME
1 !       LOCAL_BLOCK     - Block of local information.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       SS$_NORMAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals the error
1 !
1 !--
1
2     BEGIN
```

```
;  602          0597  2
;  603          0598  2      !+
;  604          0599  2      ! Case on the error source code to signal the proper message.
;  605          0600  2      !-
;  606          0601  2
;  607          0602  2      CASE .ERROR_SOURCE [0] FROM 0 TO 2 OF
;  608          0603  2          SET
;  609          0604  2
;  610          0605  2          [0]:    ! Error searching for file
;  611          0606  3              BEGIN
;  612          0607  3              SIGNAL (MSG$_SEARCHFAIL, 1, OLD_RSN [0,0,0,0],
;  613          0608  3                  .RMS_STS [0,0,32,0], .RMS_STV [0,0,32,0]);
;  614          0609  2              END;
;  615          0610  2
;  616          0611  2          [1]:    ! Error during $PARSE of new filespec
;  617          0612  3              BEGIN
;  618          0613  3              SIGNAL (MSG$_OPENOUT, 1, NEW_RSN [0,0,0,0],
;  619          0614  3                  .RMS_STS [0,0,32,0], .RMS_STV [0,0,32,0]);
;  620          0615  2              END;
;  621          0616  2
;  622          0617  2          [2]:    ! Error during $RENAME
;  623          0618  3              BEGIN
;  624          0619  3              !+
;  625          0620  3              ! If error is RMS$_DEV then assume the user is trying to
;  626          0621  3              ! rename across devices and give a specific message.
;  627          0622  3              !-
;  628          0623  3
;  629          0624  3              IF .RMS_STS [0,0,32,0] EQL RMS$_DEV
;  630          0625  3              THEN
;  631          0626  3                  SIGNAL (RENAME$_NOTRENAMED, 1, OLD_RSN [0,0,0,0],
;  632          0627  3                      RENAME$_NOTSAMEDEV)
;  633          0628  3              ELSE
;  634          0629  3                  SIGNAL (MSG$_OPENIN, 1, OLD_RSN [0,0,0,0],
;  635          0630  3                      .RMS_STS [0,0,32,0], .RMS_STV [0,0,32,0]);
;  636          0631  2              END;
;  637          0632  2          TES;
;  638          0633  2
;  639          0634  2      RETURN SS$_NORMAL;              ! Continue renaming
;  640          0635  2
;  641          0636  1      END;                                    ! End of routine ERROR_ROUTINE



                                    0004 00000 ERROR_ROUTINE:
                                                         .WORD   Save R2                                   ; 0541
                  52 00000000G  00   9E 00002            MOVAB   LIB$SIGNAL, R2                            ;
        02        00        14  BC   CF 00009            CASEL   @ERROR_SOURCE, #0, #2                     ; 0602
      002C      0019       0006         0000E 1$:        .WORD   2$-1$,=                                   ;
                                                                 3$-1$,-                                   ;
                                                                 4$-1$                                     ;
                            10  BC   DD 00014 2$:        PUSHL   @RMS_STV                                  ; 0608
                            0C  BC   DD 00017            PUSHL   @RMS_STS                                  ;
                            04  AC   DD 0001A            PUSHL   OLD_RSN                                   ; 0607
                                01   DD 0001D            PUSHL   #1                                        ;
                    0090123A    8F   DD 0001F            PUSHL   #9441850                                  ;
```

```
                                        44  11 00025       BRB     6$                                      ;
                               10       BC  DD 00027  3$:  PUSHL   @RMS_STV                                ; 0614
                               0C       BC  DD 0002A       PUSHL   @RMS_STS                                ;
                               08       AC  DD 0002D       PUSHL   NEW_RSN                                 ; 0613
                                        01  DD 00030       PUSHL   #1                                      ;
                         009010A2       8F  DD 00032       PUSHL   #9441442                                ;
                                        31  11 00038       BRB     6$                                      ;
         000184C4    8F        0C       BC  D1 0003A  4$:  CMPL    @RMS_STS, #99524                        ; 0624
                                        16  12 00042       BNEQ    5$                                      ;
                         00000000G      8F  DD 00044       PUSHL   #RENAME$_NOTSAMEDEV                     ; 0626
                               04       AC  DD 0004A       PUSHL   OLD_RSN                                 ;
                                        01  DD 0004D       PUSHL   #1                                      ;
                         00000000G      8F  DD 0004F       PUSHL   #RENAME$_NOTRENAMED                     ;
                     62                 04  FB 00055       CALLS   #4, LIB$SIGNAL                          ;
                                        14  11 00058       BRB     7$                                      ;
                               10       BC  DD 0005A  5$:  PUSHL   @RMS_STV                                ; 0630
                               0C       BC  DD 0005D       PUSHL   @RMS_STS                                ;
                               04       AC  DD 00060       PUSHL   OLD_RSN                                 ; 0629
                                        01  DD 00063       PUSHL   #1                                      ;
                         0090109A       8F  DD 00065       PUSHL   #9441434                                ;
                     62                 05  FB 0006B  6$:  CALLS   #5, LIB$SIGNAL                          ;
                     50                 01  D0 0006E  7$:  MOVL    #1, R0                                  ; 0634
                                            04 00071       RET                                             ; 0636

; Routine Size:  114 bytes,    Routine Base:  $CODE + 0211
```

```
1 %SBTTL 'ERROR_HANDLER - Condition handler'
1 ROUTINE ERROR_HANDLER(
1       SIGARGS: REF BLOCK [, BYTE],            ! Signal argument list
1       MCHARGS: REF BLOCK [, BYTE] ,           ! Mechanism argument list
1       ENBARGS: REF VECTOR [, LONG]            ! Enable argument list
1       ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This is the condition handler for RENAME_MAIN.  It stores the
1 !       "worst error" in the LOCAL_BLOCK.  If the error is LIB$_QUIPRO,
1 !       caused by the user typing Quit to the confirm prompt, an unwind
1 !       with status SS$_NORMAL is done to the caller of RENAME_MAIN.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = ERROR_HANDLER (sigargs.rbu.ra, mchargs.mbu.ra,
1 !                                         enbargs.mbu.ra)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIGARGS         - The signal arguments list
1 !
1 !       MCHARGS         - The mechanism arguments list
1 !
1 !       ENBARGS         - The enable arguments list.  It is an array of
1 !                         two longwords, the second of which is the
1 !                         address of LOCAL_BLOCK.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       SS$_RESIGNAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2       LOCAL_BLOCK: REF BLOCK [, BYTE] FIELD (LB_FIELDS),
2       OLD_CODE: BLOCK [4, BYTE],      ! Old status
2       NEW_CODE: BLOCK [4, BYTE];      ! New status
2
2     !+
2     ! Check to see if this is an unwind.  If so, return.
```

```
0694 2      ! Also return if the message is a success status.
0695 2      !-
0696 2
0697 2      NEW_CODE = .SIGARGS [CHF$L_SIG_NAME];
0698 3      IF (.NEW_CODE) OR (.NEW_CODE EQLU SS$_UNWIND)
0699 2      THEN
0700 2          RETURN SS$_RESIGNAL;
0701 2
0702 2      !+
0703 2      ! Check to see if this is LIB$_QUIPRO.  If so, unwind.
0704 2      !-
0705 2
0706 2      IF .NEW_CODE EQLU LIB$_QUIPRO
0707 2      THEN
0708 3          BEGIN
0709 3          MCHARGS [CHF$L_MCH_SAVR0] = SS$_NORMAL; ! Restore status of R0
0710 3          $UNWIND ();     ! Unwind to caller of establisher
0711 3          RETURN SS$_RESIGNAL;
0712 2          END;
0713 2
0714 2      !+
0715 2      ! If this status is "worse" than the previous worst status, save it.
0716 2      !-
0717 2
0718 2      LOCAL_BLOCK = .ENBARGS [1];
0719 2      OLD_CODE = .LOCAL_BLOCK [LB_L_WORST_ERROR];
0720 2      IF .NEW_CODE [STS$V_SEVERITY] GTRU .OLD_CODE [STS$V_SEVERITY]
0721 2      THEN
0722 2          LOCAL_BLOCK [LB_L_WORST_ERROR] = .NEW_CODE + STS$M_INHIB_MSG;
0723 2
0724 2      RETURN SS$_RESIGNAL;
0725 2
0726 1      END;                                    ! End of routine ERROR_HANDLER


                                                         .EXTRN  SYS$UNWIND

                                         000C 00000 ERROR_HANDLER:
                                                         .WORD   Save R2,R3                          ; 0638
                          50      04  AC  D0 00002       MOVL    SIGARGS, R0                         ; 0697
                          52      04  A0  D0 00006       MOVL    4(R0), NEW_CODE                     ;
                          43          52  E8 0000A       BLBS    NEW_CODE, 2$                        ; 0698
              00000920    8F          52  D1 0000D       CMPL    NEW_CODE, #2336                     ;
                                      3A  13 00014       BEQL    2$                                  ;
              00000000G   8F          52  D1 00016       CMPL    NEW_CODE, #LIB$_QUIPRO              ; 0706
                                      13  12 0001D       BNEQ    1$                                  ;
                          50      08  AC  D0 0001F       MOVL    MCHARGS, R0                         ; 0709
                    0C    A0          01  D0 00023       MOVL    #1, 12(R0)                          ;
                                      7E  7C 00027       CLRQ    -(SP)                               ; 0710
              00000000G   00          02  FB 00029       CALLS   #2, SYS$UNWIND                      ;
                                      1E  11 00030       BRB     2$                                  ; 0711
                          50      0C  AC  D0 00032 1$:   MOVL    ENBARGS, R0                         ; 0718
                          50      04  A0  D0 00036       MOVL    4(R0), LOCAL_BLOCK                  ;
                          51          60  D0 0003A       MOVL    (LOCAL_BLOCK), OLD_CODE             ; 0719
  53          51          03          00  EF 0003D       EXTZV   #0, #3, OLD_CODE, R3                ; 0720
  53          52          03          00  ED 00042       CMPZV   #0, #3, NEW_CODE, R3                ;
```

```
                                      07  1B 00047         BLEQU   2$                                      ;
                         60 10000000  E2  9E 00049         MOVAB   268435456(R2), (LOCAL_BLOCK)            ; 0722
                         50     0918  8F  3C 00050 2$:     MOVZWL  #2328, R0                               ; 0724
                                          04 00055         RET                                             ; 0726

; Routine Size:  86 bytes,    Routine Base:  $CODE + 0283
;  734          0727  1 END                                                     ! End of module RENAME
;  735          0728  1
;  736          0729  0 ELUDOM




                                                                  .EXTRN  LIB$SIGNAL, LIB$STOP

;                                    PSECT SUMMARY
;
;       Name                     Bytes                  Attributes
;
;   $CODE                            729  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                             Library Statistics
;
;                                          -------- Symbols --------        Pages        Processing
;       File                               Total     Loaded   Percent      Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1       9776        41          0        581          00:01.0




;                                    COMMAND QUALIFIERS

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

; Size:          684 code + 45 data bytes
; Run Time:         00:14.9
; Elapsed Time:     00:50.0
; Lines/CPU Min:      2931
; Lexemes/CPU-Min: 19025
; Memory Used:  103 pages
; Compilation Complete
```

0051 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SETACNTNG
LIS

RENAMEMSG
LIS

RENAME
LIS

RUNDET
LIS

RUNMSG
LIS

RUNCVTUIC
LIS

SET
LIS