


```

RRRRRRRR      MM      MM      SSSSSSSS      000000      RRRRRRRR      EEEEEEEEEEE      NN      NN      AAAAAA      MM      MM
RRRRRRRR      MM      MM      SSSSSSSS      000000      RRRRRRRR      EEEEEEEEEEE      NN      NN      AAAAAA      MM      MM
RR      RR      MMMM      MMMM      SS      00      00      RR      RR      EE      NN      NN      AA      AA      MMMM      MMMM
RR      RR      MM-MM      MMMM      SS      00      00      RR      RR      EE      NN      NN      AA      AA      MMMM      MMMM
RR      RR      MM      MM      SS      00      0000      RR      RR      EE      NNNN      NN      AA      AA      MM      MM
RR      RR      MM      MM      SS      00      0000      RR      RR      EE      NNNN      NN      AA      AA      MM      MM
RRRRRRRR      MM      MM      SSSSSS      00      00      00      RRRRRRRR      EEEEEEEEE      NN      NN      NN      AA      AA      MM      MM
RRRRRRRR      MM      MM      SSSSSS      00      00      00      RRRRRRRR      EEEEEEEEE      NN      NN      NN      AA      AA      MM      MM
RR      RR      MM      MM      SS      0000      00      RR      RR      EE      NN      NN      NN      AA      AA      MM      MM
RR      RR      MM      MM      SS      0000      00      RR      RR      EE      NN      NNNN      AAAAAAAAAA      MM      MM
RR      RR      MM      MM      SS      0000      00      RR      RR      EE      NN      NNNN      AAAAAAAAAA      MM      MM
RR      RR      MM      MM      SS      00      00      RR      RR      EE      NN      NN      AA      AA      MM      MM
RR      RR      MM      MM      SSSSSSSS      000000      RR      RR      EEEEEEEEEEE      NN      NN      AA      AA      MM      MM
RR      RR      MM      MM      SSSSSSSS      000000      RR      RR      EEEEEEEEEEE      NN      NN      AA      AA      MM      MM

```

```

....
....
....
....

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      I      SS
LL      I!      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

(2) 143
(3) 175

DECLARATIONS
RMS\$RENAME - RENAME FILE ROUTINE

```

0000 1          $BEGIN RMSORENAM,000,RMSRMS,<RENAME FILE SERVICE>
0000 2
0000 3
0000 4 *****
0000 5 *
0000 6 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 *   ALL RIGHTS RESERVED.
0000 9 *
0000 10 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 *  TRANSFERRED.
0000 16 *
0000 17 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 *  CORPORATION.
0000 20 *
0000 21 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 *
0000 24 *
0000 25 *****
0000 26
0000 27 ++
0000 28
0000 29 Abstract:
0000 30     this routine performs the $rename file rms function.
0000 31
0000 32 Environment:
0000 33     star processor running starlet exec.
0000 34
0000 35 Author: L F Laverdure,          creation date: 24-MAY-1978
0000 36
0000 37 Modified By:
0000 38
0000 39 V03-014 RAS0324          Ron Schaefer          10-Jul-1984
0000 40     Fix up possible accvio path if new fab is bogus;
0000 41     clear R10 before doing second network parse;
0000 42     generally clean up the flows and registers.
0000 43
0000 44 V03-013 JWT0173          Jim Teague          1-Apr-1984
0000 45     Clear last longword on ATR list since the list is
0000 46     now dynamically allocated.
0000 47
0000 48 V03-012 DGB0031          Donald G. Blair          22-Mar-1984
0000 49     Implement the XAB$V_PROPAGATE bit.
0000 50
0000 51 V03-011 RAS0277          Ron Schaefer          20-Mar-1984
0000 52     Fix RAS0246 for network $RENAME operations.
0000 53
0000 54 V03-010 JWT0166          Jim Teague          20-Mar-1984
0000 55     Use dynamically-allocated scratch page for accumulating
0000 56     ATRs for QIOs. FWA$T_ATR_LIST no longer exists.
0000 57

```

0000	58	:	V03-009	DGB0006	Donald G. Blair	01-Mar-1984
0000	59	:				
0000	60	:				
0000	61	:				
0000	62	:				
0000	63	:	V03-008	RAS0242,RAS0246	Ron Schaefer	23-Jan-1984
0000	64	:				
0000	65	:				
0000	66	:				
0000	67	:				
0000	68	:				
0000	69	:				
0000	70	:				
0000	71	:	V03-007	RAS0201	Ron Schaefer	17-Oct-1983
0000	72	:				
0000	73	:				
0000	74	:				
0000	75	:				
0000	76	:				
0000	77	:	V03-006	KPL0001	Peter Lieberwirth	20-Jun-1983
0000	78	:				
0000	79	:				
0000	80	:	V03-005	TSK0002	Tamar Krichevsky	12-Jun-1983
0000	81	:				
0000	82	:				
0000	83	:	V03-004	KRM0079	Karl Malik	10-Jan-1983
0000	84	:				
0000	85	:				
0000	86	:	V03-003	TSK0001	Tamar Krichevsky	28-Dec-1982
0000	87	:				
0000	88	:				
0000	89	:				
0000	90	:				
0000	91	:				
0000	92	:				
0000	93	:				
0000	94	:				
0000	95	:				
0000	96	:				
0000	97	:				
0000	98	:				
0000	99	:				
0000	100	:				
0000	101	:				
0000	102	:				
0000	103	:				
0000	104	:				
0000	105	:				
0000	106	:				
0000	107	:				
0000	108	:				
0000	109	:	V03-002	KRM0060	Karl Malik	10-Sep-1982
0000	110	:				
0000	111	:				
0000	112	:				
0000	113	:	V03-001	KBT0171	Keith B. Thompson	23-Aug-1982
0000	114	:				

```
0000 115 :
0000 116 :
0000 117 :
0000 118 :
0000 119 :
0000 120 :
0000 121 :
0000 122 :
0000 123 :
0000 124 :
0000 125 :
0000 126 :
0000 127 :
0000 128 :
0000 129 :
0000 130 :
0000 131 :
0000 132 :
0000 133 :
0000 134 :
0000 135 :
0000 136 :
0000 137 :
0000 138 :
0000 139 :
0000 140 :
0000 141 :
```

V02-010 RAS0043 Ron Schaefer 29-Oct-1981
Implement root directory support for target filespec.
Must call RMS\$ASSIGN to assign the channel for the
target device in order to get the root directory
evaluated.

V02-009 KPL0001 Peter Lieberwirth 29-Jul-1981
Rename entry point to \$\$ form so we can dispatch here
via RMSOBRNCH.

V02-008 REFORMAT Maria del C. Nasr 24-Jul-1980

v007 KEK001 K E Kinnear 11-MAR-1980 18:40
Change check for same device on old and new file name
from string check to check of ucb addresses.

v006 RAS0009 R A Schaefer 20-DEC-1979 15:40
Set-up for invalid directory rename operation check
in rms\$setdid.

v005 JAK0001 J A Krycka 27-AUG-1978 09:50
Miscellaneous clean-up prior to decnet v1.0 code freeze.
revise network error processing.

```
0000 143      .SBTTL  DECLARATIONS
0000 144
0000 145      ;
0000 146      ; Include Files:
0000 147      ;
0000 148
0000 149      ;
0000 150      ; Macros:
0000 151      ;
0000 152
0000 153      $RMSDEF
0000 154      $DEVDEF
0000 155      $FABDEF
0000 156      $NAMDEF
0000 157      $IFBDEF
0000 158      $FIBDEF
0000 159      $IODEF
0000 160      $FWADEF
0000 161      $IMPDEF
0000 162
0000 163      ;
0000 164      ; Equated Symbols:
0000 165      ;
0000 166
00000020 0000 167      FOP=FAB$L FOP*8      ; bit offset to fop bits
0000000A 0000 168      V_ACS = 10          ; field contains a nodename followed
0000 169                                          ; by an access control string
0000 170      ;
0000 171      ; Own Storage:
0000 172      ;
0000 173
```



```

0000 232 :
0000 233 : the sts and stv fields of the fab are output along with
0000 234 : the fields of the nam block
0000 235 :
0000 236 : a completion ast is queued if specified in the user arglist.
0000 237 :
0000 238 : Completion Codes:
0000 239 :
0000 240 : standard rms, in particular acc, bln, dev, dir, dnf, dnr, drn,
0000 241 : ent, esa, esl, fab, fnd, fnf, fnm, ifi, iop, nam, prv,
0000 242 : rmv, rst, rss, sys, typ, ver, wld.
0000 243 :
0000 244 : Side Effects:
0000 245 :
0000 246 : none
0000 247 :
0000 248 :--
0000 249 :

```

```

0000 251
0000 252 :++
0000 253 :
0000 254 : entry point for $rename function
0000 255 :
0000 256 :--
0000 257 :
0000 258 $ENTRY RMS$RENAME
0000 259 $TSTPT RENAME ; flag rename done
FFF7' 30 0006 260 BSBW RMS$SETI ; create ifab
0009 261 :
0009 262 :
0009 263 : Return to user on error
0009 264 : r11=impure area addr
0009 265 : r9=ifab addr
0009 266 : r8=fab addr
0009 267 :
0009 268 :
51 58 DD 0009 269 PUSHL R8 ; save old fab address
10 AC DO 000B 270 MOVL 16(AP),R1 ; get the new fab address
01CD 30 000F 271 BSBW NEWFAB1 ; checkout new fab
58 58 DD 0012 272 PUSHL R8 ; save newfab addr
04 AE DO 0014 273 MOVL 4(SP),R8 ; restore old fab addr
0018 274 :
0018 275 :++
0018 276 :
0018 277 : Parse the old file spec and look up the old file.
0018 278 :
0018 279 :--
57 28 A8 DO 0018 280 :
09 13 001C 281 MOVL FAB$$_NAM(R8),R7 ; locate the old NAM block
FFDF' 30 001E 282 BEQL 10$ ; ok without one
59 50 E9 0021 283 BSBW RMS$CHKNAM ; is the NAM block valid?
30 A7 DD 0024 284 BLBC R0,100$ ; if not, then return an error
01D1 30 0027 285 PUSHL NAM$$_WCC(R7) ; save the current wild card context,
; RMS$PARSE_FILE overwrites this field
04 13 002A 286 10$: BSBW PARSE_FILE ; parse the old file spec
30 A7 8ED0 002C 287 BEQL 20$ ; and perform various checks
4F 6A 19 E0 0030 288 POPL NAM$$_WCC(R7) ; okay if no NAM block
0034 289 BBS #FWAS$_NODE,(R10),120$ ; restore wild card context.
0034 290 ; Branch if node
0034 291 :++
0034 292 :
0034 293 :
0034 294 :
0034 295 : The following access of the file is similar to FILFNC, but has to be done
0034 296 : in line. The journaling attributes need to be retrieved and there is no
0034 297 : provision in FILFNC to do so.
0034 298 :
0034 299 :--
58 AA FFC9' 30 0034 300 BSBW RMS$GET1PAG ; Grab a scratch page
55 53 DO 0037 301 MOVL R3,FWAS$_ATR_WORK(R10) ; Save scratch page address
003E 302 MOVL R3,R5 ; and put it in R5
00000000'EF 16 003E 303 JSB RMS$RTVJNL ; add the descriptor for journaling
0044 304 ; attributes to end of the list
65 D4 0044 305 CLRL (R5) ; indicate end of ATR list
0044 306
0044 307

```

```

    7E D4 0046 308      CLRL  -(SP)                ; P6 => 0
58 AA DD 0048 309      PUSHL FWASL_ATR_WORK(R10)      ; P5 => descriptors for file attributes
FFB2' 30 004B 310      BSBW  RMSFCP_P4_P2          ; set up P4 through P2
50 32 9A 004E 311      MOVZBL #IOS_ACCESS,R0      ; function code for file lookup
FFAC' 30 0051 312      BSBW  RMSFCPFNC           ; lookup the old file
      0054 313
      50 DD 0054 314      PUSHL  R0                ; Save status
54 58 AA D0 0056 315      MOVL  FWASL_ATR_WORK(R10),R4  ; Pass address of scratch page
FFA3' 30 005A 316      BSBW  RMSRET1PAG          ; Return scratch page
58 AA D4 005D 317      CLRL  FWASL_ATR_WORK(R10)    ; Indicate no work area now
      50 8ED0 0060 318      POPL  R0                ; Restore status
      0063 319
1A 50 E9 0063 320      BLBC  R0,110$             ; continue if successful
FF97' 30 0066 321      BSBW  RMS$FILLNAM        ; fill in the old NAM blk, if any
11 50 E9 0069 322      BLBC  R0,100$             ; get out on error
      006C 323
      006C 324 :++
      006C 325 :
      006C 326 : save the old file name context and process new
      006C 327 :
      006C 328 :--
      006C 329
      5A DD 006C 330      PUSHL  R10                ; save old FWA address
016A 30 006E 331      BSBW  NEUFAB              ; Probe the new FAB's location, check
      0071 332      ; FAB's validity, load R8 with its adr
57 28 A8 D0 0071 333      MOVL  FAB$S_NAM(R8),R7      ; locate the new NAM block
      18 13 0075 334      BEQL  40$                ; okay if none
FF86' 30 0077 335      BSBW  RMSCHKNAM        ; is the NAM block valid?
OF 50 E8 007A 336      BLBS  R0,30$             ; if yes, continue renaming the file
      007D 337
      01AE 31 007D 338 100$: BRW  CLEAN              ; return an error
      0080 339
      01C5 31 0080 340 110$: BRW  ERRACC            ; otherwise, return an error
      0083 341
      00D5 31 0083 342 120$: BRW  NETRENAM         ; Branch (network operation)
      0086 343
      01D3 31 0086 344 130$: BRW  ERRRMV          ; otherwise, return an error
      0089 345
      01C0 31 0089 346 140$: BRW  ERRDEV
      008C 347
      30 A7 DD 008C 348 30$: PUSHL  NAM$S_WCC(R7)    ; save the current wild card context,
      008F 349      ; RMS$PARSE_FILE overwrites this field
      0169 30 008F 350 40$: BSBW  PARSE_FILE      ; parse new file name (but doesn't set
      0092 351      ; DID, therefore it doesn't stall,
      0092 352      ; so user's blks don't need probing.)
      04 13 0092 353      BEQL  50$                ; okay if no NAM block
5C 0000000'EF 9E 0C98 354 50$: POPL  NAM$S_WCC(R7)  ; restore wild card context.
      FF5E' 30 009F 355      MOVAB  RMS$ENTXAB_ARGS,AP  ; ap = arg for xabscan
      DB 50 E9 00A2 356      BSBW  RMS$XAB_SCAN      ; handle pro xab
      00A5 357      BLBC  R0,100$             ; exit on error
      00A5 358
      00A5 359
      00A5 360 :++
      00A5 361 :
      00A5 362 : verify that old and new device names are the same.
      00A5 363 : Must first assign channel for new file. Old file has channel in IFAB.
      00A5 364 :

```

```

00A5 365 :--
00A5 366
7E 20 A9 B0 00A5 367      MOVW  IFBSW_CHNL(R9),-(SP)  ; save old file channel
    FF54 30 00A9 368      BSBW  RMS$ASSIGN          ; assign channel for new file
    DA 50 E9 00AC 369      BLBC  RO,140$           ; if error, say bad device name
                                $DASSGN_S CHAN=IFBSW_CHNL(R9) ; remove the channel
20 A9 8E B0 00BA 371      MOVW  -(SP)+,IFBSW_CHNL(R9) ; restore the old channel
    50 6E D0 00BE 372      MOVL  (SP),RO           ; get old FWA pointer
    0198 C0 2D 00C1 373      CMPC5  FWASQ_SHRFIL_LCK(R0),- ; check if canonical device names
    019C D0 00C5 374      @FWASQ_SHRFIL_LCK+4(R0),- ; as returned by $GETDVI
    FF 8F 00C8 375      #-1- ; funny fill to prevent matches
    0198 CA 00CA 376      FWASQ_SHRFIL_LCK(R10),- ; are the same
    019C DA 00CD 377      @FWASQ_SHRFIL_LCK+4(R10)
                                BNEQ  140$ ; if not, then return an error
    50 6E D0 00D2 379      MOVL  (SP),RO           ; get old FWA pointer
    01F8 C0 D0 00D5 380      MOVL  FFAST_FIBBUF+IFBSW_FID(R0),- ; copy FID to special place
    0240 CA 00D9 381      FFAST_RNM_FID(R10),- ; for invalid directory rename check
    01FC C0 B0 00DC 382      MOVW  4+FFAST_FIBBUF+IFBSW_FID(R0),-
    0244 CA 00E0 383      4+FFAST_RNM_FID(R10),- ; all 6 bytes
    FF1A 30 00E3 384      BSBW  RMS$SETFIB        ; set up FIB descriptor
    FF17 30 00E6 385      BSBW  RMS$SETDID_ALT     ; look up new directory id
    6C 50 E9 00E9 386      BLBC  RO,90$           ; get out on error
    00EC 387
    00EC 388 :++
    00EC 389
    00EC 390 : both old and new names parsed o.k., the device and unit are the same,
    00EC 391 : and the old file has been looked up. Now, do the remove of the old name
    00EC 392 : and then the enter of the new name. If the enter fails, go back and reenter
    00EC 393 : the old name.
    00EC 394
    00EC 395 :--
    00EC 396
51 08 AE D0 00EC 397      MOVL  8(SP),R1          ; restore old FAB addr
    00EC 30 00F0 398      BSBW  NEWFAB1          ; check it out
    5A 5A DD 00F3 399      PUSHL R10             ; save the new FWA addr
    5A 04 AE D0 00F5 400      MOVL  4(SP),R10        ; restore old FWA addr
    55 35 9A 00F9 401      MOVZBL #IOS_DELETE,R5 ; qio function code for remove
    00D0 30 00FC 402      BSBW  FILFNC           ; remove the old file name
    84 50 E9 00FF 403      BLBC  RO,130$         ; if successful, enter the new name
                                0102 404
    50 5A D0 0102 405      MOVL  R10,RO           ; make room for the new FWA addr
    5A 8ED0 0105 406      POPL  R10             ; retrieve the new FWA addr
    00D0 30 0108 407      BSBW  NEWFAB          ; Probe the new FAB's location, check
                                ; FAB's validity and load R8 with it's add
    01F8 C0 D0 0108 409      MOVL  FFAST_FIBBUF+IFBSW_FID(R0),-
    01F8 CA 010F 410      FFAST_FIBBUF+IFBSW_FID(R10); copy old FID to new FIB
    01FC C0 B0 0112 411      MOVW  4+FFAST_FIBBUF+IFBSW_FID(R0),-
    01FC CA 0116 412      4+FFAST_FIBBUF+IFBSW_FID(R10)
    55 33 9A 0119 413      MOVZBL #IOS_CREATE,R5 ; qio function code for enter
    00B0 30 011C 414      BSBW  FILFNC           ; enter the new file name
    21 50 E9 011F 415      BLBC  RO,70$         ; if successful, finish up chores
                                0122 416
    00B6 30 0122 417      BSBW  NEWFAB          ; Probe the new FAB's location, check
                                ; FAB's validity and load R8 with it's add
    FED8 30 0125 418      BSBW  RMS$FILLNAM       ; fill in new nam blk, if any
    2D 50 E9 0128 420      BLBC  RO,90$         ; get out on error
    57  D5 012B 421      TSTL  R7             ; do we have a nam blk?

```

```

OD 13 012D 422 BEQL 60$ ; branch if not
      012F 423
      012F 424 :
      012F 425 : set the lower and highver flags in the nam block
      012F 426 :
      012F 427 :
      012F 428 ASSUME FIBSV_HIGHVER EQ FIBSV_LOWVER+1
      012F 429 ASSUME NAMSV_HIGHVER EQ NAMSV_LOWVER+1
51 02 0E EF 012F 430 EXTZV #FIBSV_LOWVER,#2,- ; get version bits from FIB
02 0208 CA 0132 431 FIBSV_NMCTL+FWAST,FIBBUF(R10),R1
      OE 51 FO 0136 432 R1,#NAMSV_LOWVER,#2,- ; copy them into user's NAM block
      34 A7 013A 433 NAMSL_FNB(R7)
58 08 AE DO 013C 434 60$: MOVL 8(SP),R8 ; retrieve the old FAB addr
      FEBD' 31 0140 435 BRW RMSCLSCU ; go evaporate ifab & return to user
      0143 436
5A 50 DD 0143 437 70$: PUSHL R0 ; save the error code for later use
      04 AE DO 0145 438 MOVL 4(SP),R10 ; restore old FWA address
      0083 30 0149 439 BSBW FILFNC ; do the re-enter *ASSUMPTION -- R5 HAS
      014C 440 ; NOT CHANGED SINCE LAST ENTER.
      03 50 E8 014C 441 BLBS R0,80$ ; if enter worked, return an error
      0102 31 014F 442 BRW ERRREENT ; if not, give up - file has been lost
      0152 443
      50 8ED0 0152 444 80$: POPL R0 ; retrieve ACP error from enter
      0100 31 0155 445 BRW ERRENT ; enter failed, but re-enter succeeded,
      0158 446 ; old file is still intact
      0158 447
      00D3 31 0158 448 90$: BRW CLEAN

```

```

015B 450 :++
015B 451 :
015B 452 : The parse (& parse check) of the old filespec has been completed and a
015B 453 : nodename has been found. Now parse the new filespec, make sure the node
015B 454 : names match and issue the network $rename request.
015B 455 :
015B 456 :--
015B 457 :
015B 458 NETRENAM:
015B 459 :
FEA2' 30 015B 460 BSBW NTSNWA_FREE ; Get rid of old filespec
015E 461 ; NWA (no longer needed)
20 A9 B4 015E 462 $DASSGN_S IFBSW CHNL(R9) ; Deassign the old channel
0169 463 CLRW -IFBSW_CHNL(R9) ; Clear the old channel field
016C 464 :
016C 465 :++
016C 466 :
016C 467 : save the old file name context and process new
016C 468 :
016C 469 :--
016C 470 :
SA DD 016C 471 10$: PUSHL R10 ; save old fwa address
006A 30 016E 472 BSBW NEWFAB ; get new name fab addr in r8
0087 30 0171 473 BSBW PARSE_FILE ; parse new file name
51 08 AE D0 0174 474 MOVL 8(SP),R1 ; restore old fab address
0064 30 0178 475 BSBW NEWFAB1 ; and check it out
017B 476 :
017B 477 :+
017B 478 : Check the new filespec for a nodename.
017B 479 :-
4D 6A 19 E1 017B 480 BBC #FWASV_NODE,(R10),40$ ; Nodename on new file name?
017F 481 :
017F 482 :+
017F 483 : Make sure the nodenames match (local RMS will take care of the device
017F 484 : comparison on the remote node).
017F 485 :-
017F 486 :
52 52 6E D0 017F 487 20$: MOVL (SP),R2 ; Get old FWA adr in reg
52 01B4 C2 DE 0182 488 MOVAL FWASQ NODE1(R2),R2 ; Get old filename FWA descriptor adr
08 02 A2 0A E1 0187 489 MOVZWL (R2),R4 ; Get old nodename length
04 B2 54 22 3A 018A 490 BBC #V_ACS,2(R2),22$ ; Branch if no access string present
53 01B4 CA DE 018F 491 ; (high word of desc.)
08 02 A3 0A E1 018F 492 LOCC #^A/'/,R4,@4(R2) ; Find the beginning quote
04 B3 55 22 3A 0194 493 SUBL2 R0,R4 ; Compute nodename length (w/o acs)
55 63 3C 0197 494 22$: MOVAL FWASQ NODE1(R10),R3 ; Get new filename FWA descriptor adr
08 02 A3 0A E1 019C 495 MOVZWL (R3),R5 ; Get new nodename length
04 B3 55 22 3A 019F 496 BBC #V_ACS,2(R3),23$ ; Branch if no access string present
04 B3 55 22 3A 01A4 497 LOCC #^A/'/,R5,@4(R3) ; Find the beginning quote
04 B3 55 50 C2 01A9 498 SUBL2 R0,R5 ; Compute nodename length (w/o acs)
04 B3 55 00 04 B2 54 2D 01AC 499 23$: CMPC5 R4,@4(R2),#0,R5,@4(R3) ; Compare the nodenames
01B4 500 BNEQ 40$ ; Branch on error and exit
01B6 501 :
01B6 502 :+
01B6 503 : They match - This is a valid network $RENAME request
01B6 504 :-
01B6 505 :
FE47' 30 01B6 506 25$: BSBW NTSACCESS ; Assign the logical link

```

```
51 03 50 E9 01B9 507 BLBC R0,30$ ; Branch on error
    FE41' 30 01BC 508 BSBW NT$RENAME ; Issue the DAP $RENAME request
    SA 6E D0 01BF 509 30$: MOVL (SP),R10 ; Restore old FWA adr
    08 AE D0 01C2 510 MOVL 8(SP),R1 ; Restore old FAB adr
    0016 30 01C6 511 BSBW NEWFAB1 ; and check it out
    FE34' 31 01C9 512 BRW RMSCLSCU ; Cleanup and exit.
    0081 31 01CC 513 ;
    0081 31 01CC 514 40$: BRW ERRNOD ; Can't mix network and local rename
```

```

01CF 516
01CF 517 :++
01CF 518 :
01CF 519 : filfnc subroutine to issue a qio function.
01CF 520 :
01CF 521 : inputs:
01CF 522 :
01CF 523 :         r5         io function code
01CF 524 :         r8         fab address
01CF 525 :         r9         ifab address
01CF 526 :         r10        fwa address
01CF 527 :         r11        impure area address
01CF 528 :
01CF 529 : outputs:
01CF 530 :
01CF 531 :         r0         status code
01CF 532 :         r1-r4,ap   destroyed
01CF 533 :
01CF 534 :--
01CF 535 :
50  7E  7C 01CF 536 FILFNC: CLRQ   -(SP)           ; p6=p5=0
    FE2C' 30 01D1 537      BSBW   RMS$FCP_P4_P2       ; set up p4 thru p2 to
01D4 538                        ; process name from fwa
50  55  D0 01D4 539      MOVL   R5,R0           ; io function code to right reg.
    FE26' 30 01D7 540      BSBW   RMS$FCPFNC        ; do acp function
    05  01DA 541      RSB
01DB 542
01DB 543 :++
01DB 544 :
01DB 545 : subroutine to load address of new fab into r8 and check it for goodness.
01DB 546 :
01DB 547 : inputs:
01DB 548 :
01DB 549 :         r9         ifab address
01DB 550 :         8(sp)      new fab address for NEWFAB
01DB 551 :         r1         new fab address for NEWFAB1
01DB 552 :
01DB 553 : outputs:
01DB 554 :
01DB 555 :         r8         new fab address
01DB 556 :         r1         destroyed
01DB 557 :
01DB 558 : note: does not return if fab is bad.
01DB 559 :
01DB 560 :--
01DB 561 :
01DB 562 :         ASSUME   FAB$L_STS+4   EQ   FAB$L_STV
01DB 563 :
51  08  AE  D0 01DB 564 NEWFAB: MOVL   8(SP),R1           ; get new fab address
01DF 565 NEWFAB1:
01DF 566      IFNORD  #FAB$C_BLN,(R1),ERRFAB,IFB$B_MODE(R9)
01E8 567                        ; branch if new fab not readable
    03  61  91 01E8 568      CMPB   FAB$B_BID(R1),#FAB$C_BID ; is it a fab?
50  8F  01  A1  91 01EB 569      BNEQ   ERRFAB ; branch if not
    73  12
    70  1F 01ED 570      CMPB   FAB$B_BLN(R1),#FAB$C_BLN; is it long enough?
58  51  D0 01F2 571      BLSSU  ERRBLN ; branch if not
    51  D0 01F4 572      MOVL   R1,R8 ; put in right register

```


RMSORENAM
V04-000

RENAME FILE SERVICE
RMS\$RENAME - RENAME FILE ROUTINE

N 11

16-SEP-1984 01:27:42 VAX/VMS Macro V04-00
5-SEP-1984 16:25:20 [RMS.SRC]RMSORENAM.MAR;1

Page 14
(8)

RMSO
V04-

08 A8 7C 01F7 573 CLRQ FAB\$L_STS(R8) ; clear sts & stv
05 01FA 574 RSB

```

01FB 576
01FB 577 :++
01FB 578
01FB 579 : parse_file subroutine to parse the filespec using RM$PARSE_FILE and
01FB 580 : to verify that the parsed file spec contained
01FB 581 : no wild cards, specified a disk device and did not specify a
01FB 582 : foreign device or process-permanent file.
01FB 583
01FB 584 : inputs:
01FB 585
01FB 586 :     r0      status code (checked for success)
01FB 587 :     r10     fwa address
01FB 588 :     r8      fab address
01FB 589
01FB 590 : outputs:
01FB 591
01FB 592 :     z-bit   set if no nam block exists
01FB 593 :     R7      0 if z-bit set
01FB 594 :     z-bit   clear if nam block exists
01FB 595 :     R7      ptr to nam block if z-bit clear
01FB 596
01FB 597 : note: does not return on error.
01FB 598
01FB 599 :--
01FB 600
01FB 601 PARSE_FILE:
    5A  D4 01FB 602 CLRL R10 ; signal no fwa
FE00' 30 01FD 603 BSBW RM$PARSE_FILE ; parse new file name (but doesn't set
    0200 604 ; DID, therefore it doesn't stall,
    0200 605 ; so user's blks don't need probing.)
    2B 50 E9 0200 606 BLBC R0,CLEAN ; get out if parse failed
    6A 18 E0 0203 607 BBS #FWASV_WILDCARD,(R10),- ; get out if wild card speced
    61 0206 608 ERRWLD
    0C AA 95 0207 609 TSTB FWASB ESCFLG(R10) ; ppf?
    60 12 020A 610 BNEQ ERRIOP ; branch if yes
    08 6A 19 E0 020C 611 BBS #FWASV_NODE,(R10),10$ ; don't check device if node
    1C E1 0210 612 BBC #DEVSV_RND,-
    58 69 0212 613 IFBSL PRIM_DEV(R9),ERRIOP ; error if not disk
    18 E0 0214 614 BBS #DEVSV_FOR,-
    54 69 0216 615 IFBSL PRIM_DEV(R9),ERRIOP ; error if mounted foreign
    57 28 AB D0 0218 616 10$: MOVL FABSL_NAM(R8),R7 ; NAM block present?
    08 13 021C 617 BEQL 20$ ; nope, but that's okay
    FDDF' 30 021E 618 BSBW RMSCHKNAM ; is it useable?
    OA 50 E9 0221 619 BLBC R0,CLEAN ; nope
    57 D5 0224 620 TSTL R7 ; set z-bit
    05 0226 621 20$: RSB

```


RMSORENAM
V04-000

RENAME FILE SERVICE
RMS\$RENAME - RENAME FILE ROUTINE

D 12

16-SEP-1984 01:27:42 VAX/VMS Macro V04-00
5-SEP-1984 16:25:20 [RMS.SRC]RMSORENAM.MAR;1

Page 17
(12)

026E	680	RMSERR_WORD	IOP	; attempt to rename ppf
0270	681			; foreign or non disk device
0270	682			
0270	683	.END		

RMS
Pse

PSE

RMS
\$AB

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
346
The
270
21

Mac

-\$2
-\$2
-\$2
TOT

799

The

MAC

RMSORENAM
Symbol table

RENAME FILE SERVICE

E 12

16-SEP-1984 01:27:42 VAX/VMS Macro V04-00
5-SEP-1984 16:25:20 [RMS.SRC]RMSORENAM.MAR;1

Page 18
(12)

```

SS.PSECT EP      = 00000000
SSRMSTEST       = 0000001A
SSRMS_PBUGCHK   = 00000010
SSRMS_TBUGCHK   = 00000008
SSRMS_UMODE     = 00000004
CLEAN           = 0000022E R    01
CLEAN1         = 0000022B R    01
DEVSV_FOR      = 00000018
DEVSV_RND      = 0000001C
ERRACC         = 00000248 R    01
ERRBLN        = 00000264 R    01
ERRDEV        = 0000024C R    01
ERRENT        = 00000258 R    01
ERRFAB        = 00000260 R    01
ERRIOP        = 0000026C R    01
ERRNOD        = 00000250 R    01
ERRRENT       = 00000254 R    01
ERRRMV        = 0000025C R    01
ERRWLD        = 00000268 R    01
FABS_BID      = 00000000
FABS_BLN     = 00000001
FABS_C_BID   = 00000003
FABS_C_BLN  = 00000050
FABS_L_FOP   = 00000004
FABS_L_NAM  = 00000028
FABS_L_STS  = 00000008
FABS_L_STV  = 0000000C
FIBSV_HIGHVER = 0000000F
FIBSV_LOWER  = 0000000E
FIBSW_FID   = 00000004
FIBSW_NMCTL = 00000014
FILFNC      = 000001CF R    01
FOP         = 00000020
FWASB_ESCFLG = 0000000C
FWASL_ATR_WORK = 00000058
FWASQ_NODE1  = 000001B4
FWASQ_SHRFIL_LCK = 00000198
FWAST_FIBBUF = 000001F4
FWAST_RNM_FID = 00000240
FWASV_NODE  = 00000019
FWASV_WILDCARD = 00000018
IFBSB_MODE  = 0000000A
IFBSL_PRIM_DEV = 00000000
IFBSW_CHNL  = 00000020
IMPSL_SAVED_SP = 00000014
IOS_ACCESS  = 00000032
IOS_CREATE  = 00000033
IOS_DELETE  = 00000035
MCLEAN     = 00000227 R    01
NAMSL_FNB  = 00000034
NAMSL_WCC  = 00000030
NAMSV_HIGHVER = 0000000F
NAMSV_LOWER  = 0000000E
NETRENAM   = 0000015B R    01
NEWFAB     = 000001DB R    01
NEWFAB1    = 000001DF R    01
NTSACCESS  = ***** X    01
  
```

```

NTSNWA FREE      ***** X    01
NTSRENAME        ***** X    01
PARSE_FILE      000001FB R    01
PIOSA_TRACE     ***** X    01
RMSASSIGN       ***** X    01
RMSCHKNAM       ***** X    01
RMSCLSCU        ***** X    01
RMSENTXAB_ARGS ***** X    01
RMSFCPFNC       ***** X    01
RMSFCP_P4_P2    ***** X    01
RMSFILCNAM      ***** X    01
RMSFSETI        ***** X    01
RMSGET1PAG      ***** X    01
RMSPARSE_FILE   ***** X    01
RMSRET1PAG      ***** X    01
RMSRTVJNL       ***** X    01
RMSSETDID_ALT   ***** X    01
RMSSETFIB       ***** X    01
RMSXAB_SCAN     ***** X    01
RMS$RENAME      = FFFFFFFE RG   01
RMS$_ACC        = 0001C002
RMS$_BLN        = 0001842C
RMS$_DEV        = 000184C4
RMS$_ENT        = 0001C01A
RMS$_FAB        = 0001850C
RMS$_FACILITY   = 00000001
RMS$_IOP        = 00018574
RMS$_NOD        = 000185F4
RMS$_REENT      = 0001C15C
RMS$_RMV        = 0001C0FC
RMS$_WLD        = 00018744
SYSSDASSGN      ***** GX   01
TPTSL_RENAME    ***** X    01
V_ACS          = 0000000A
  
```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMSRMS	00000270 (624.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	39	00:00:00.12	00:00:00.47
Command processing	137	00:00:00.75	00:00:03.88
Pass 1	419	00:00:15.63	00:00:37.82
Symbol table sort	0	00:00:02.45	00:00:04.00
Pass 2	129	00:00:03.17	00:00:08.60
Symbol table output	11	00:00:00.12	00:00:00.12
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	739	00:00:22.26	00:00:54.91

The working set limit was 1650 pages.
89006 bytes (174 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1746 non-local and 23 local symbols.
683 source lines were read in Pass 1, producing 15 object records in Pass 2.
24 pages of virtual memory were used to define 23 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	9
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	19

1843 GEIS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSORENAM/OBJ=OBJ\$:RMSORENAM MSRC\$:RMSORENAM/UPDATE=(ENH\$:RMSORENAM)+EXECML\$/LIB+LIB\$:RMS/LIB

