```
RRRRRRRRRRR      MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMMMMM    MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRR         RRR  MMM   MMM   MMM   SSS
RRRRRRRRRRR      MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR   RRR        MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR      RRR     MMM          MMM             SSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
```

FILEID**RMSOCREAT

```
RRRRRRRR    MM      MM   SSSSSSSS   OOOOOO     CCCCCCCC RRRRRRRR   EEEEEEEEEE    AAAAAA   TTTTTTTTTT
RRRRRRRR    MM      MM   SSSSSSSS   OOOOOO     CCCCCCCC RRRRRRRR   EEEEEEEEEE    AAAAAA   TTTTTTTTTT
RR      RR  MMMM  MMMM  SS         OO    OO   CC       RR     RR  EE           AA    AA      TT
RR      RR  MMMM  MMMM  SS         OO    OO   CC       RR     RR  EE           AA    AA      TT
RR      RR  MM MM MM MM  SS         OO    OO   CC       RR     RR  EE           AA    AA      TT
RR      RR  MM MM MM MM  SS         OO    OO   CC       RR     RR  EE           AA    AA      TT
RRRRRRRR    MM  MM  MM    SSSSSS    OO    OO   CC       RRRRRRRR   EEEEEEEE     AA    AA      TT
RRRRRRRR    MM  MM  MM    SSSSSS    OO    OO   CC       RRRRRRRR   EEEEEEEE     AA    AA      TT
RR  RR      MM      MM          SS   OO    OO   CC       RR  RR     EE           AAAAAAAAAA    TT
RR  RR      MM      MM          SS   OO    OO   CC       RR  RR     EE           AAAAAAAAAA    TT
RR    RR    MM      MM          SS   OO    OO   CC       RR    RR   EE           AA    AA      TT
RR    RR    MM      MM          SS   OO    OO   CC       RR    RR   EE           AA    AA      TT
RR      RR  MM      MM  SSSSSSSS     OOOOOO     CCCCCCCC RR      RR EEEEEEEEEE   AA    AA      TT
RR      RR  MM      MM  SSSSSSSS     OOOOOO     CCCCCCCC RR      RR EEEEEEEEEE   AA    AA      TT

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
```

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

Table of contents

```
0000     1          $BEGIN  RMSOCREAT,000,RM$RMS,<DISPATCH FOR CREATE OPERATION>
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 ; Facility: rms32
0000    29 ;
0000    30 ; Abstract:
0000    31 ;                   this module is the highest level control routine
0000    32 ;                   to perform the $create function.
0000    33 ;
0000    34 ; Environment:
0000    35 ;                   star processor running starlet exec.
0000    36 ;
0000    37 ; Author: L F LAVERDURE,          Creation Date: 8-MAY-1977
0000    38 ;
0000    39 ; Modified By:
0000    40 ;
0000    41 ;       V03-008 RAS0286         Ron Schaefer                3-Apr-1984
0000    42 ;               Revise the searchlist continuation rules/errors.
0000    43 ;
0000    44 ;       V03-007 RAS0244         Ron Schaefer               23-Jan-1984
0000    45 ;               Add $DEVDEF macro.
0000    46 ;
0000    47 ;       V03-006 RAS0198         Ron Schaefer                6-Oct-1983
0000    48 ;               Merge $OPEN and $CREATE searchlist looping logic into
0000    49 ;               a single routine.   Update error checking logic to only
0000    50 ;               loop for certain errors, not for all.
0000    51 ;
0000    52 ;       V03-005 RAS0197         Ron Schaefer                5-Oct-1983
0000    53 ;               Add support for searchlists with create-if.   The
0000    54 ;               method is to try straight-forward $OPEN on ALL elements
0000    55 ;               of the search list.   If no file is found anywhere, then
0000    56 ;               do a create-if ACP operation on the first location.
0000    57 ;
```

```
0000     58 ;        V03-004 JWH0215         Jeffrey W. Horn         13-Apr-1983
0000     59 ;                Fix broken branch.
0000     60 ;
0000     61 ;        V03-003 LJA0021         Laurie Anderson         3-Sep-1982
0000     62 ;                If this is a Restart create operation, then just go and
0000     63 ;                open the file, using OPEN.
0000     64 ;
0000     65 ;        V03-002 KBT0175         Keith B. Thompson       23-Aug-1982
0000     66 ;                Reorganize psects and make entry point single '$'
0000     67 ;
0000     68 ;        V03-001 RAS0084         Ron Schaefer    2-Apr-1982
0000     69 ;                Force stream format files to have RAT non null.
0000     70 ;
0000     71 ;        V02-018 CDS0001         C Saether       5-Feb-1982
0000     72 ;                Store GBC in ifab from fab.
0000     73 ;
0000     74 ;        V02-017 JAK0068         J A Krycka      29-DEC-1981
0000     75 ;                Preserve alternate success code on network exit path.
0000     76 ;
0000     77 ;        V02-016 MCN0005         Maria del C. Nasr       10-Feb-1981
0000     78 ;                Make sure magtape is mounted before creating file.
0000     79 ;
0000     80 ;        V02-015 JAK0048         J A KRYCKA      25-SEP-1980
0000     81 ;                Move network specific code from RMOCRECOM to this module and
0000     82 ;                add support for the create-if (CIF) option.
0000     83 ;
0000     84 ;        V014    REFORMAT        D M WALP        25-JUL-1980
0000     85 ;
0000     86 ;        V013    RAS0003         R A SCHAEFER    27-NOV-1979
0000     87 ;                allow any record format for bio access.
0000     88 ;
0000     89 ;        V012    TMH0001         TIM HALVORSEN   13-OCT-1979
0000     90 ;                fix wildcard check to use the wildcard summary bit
0000     91 ;                rather than checking explicitly for wc name,type,ver.
0000     92 ;
0000     93 ;        V011    RAN0002         R A NEWELL      1-SEP-1978      12:00
0000     94 ;                rms32 isam modification.  redefinition of entry points to
0000     95 ;                resolve out of range branches.
0000     96 ;
0000     97 ;        V010    JAK0001         J A KRYCKA      24-AUG-1978     07:53
0000     98 ;                miscellaneous clean-up prior to decnet v1.0 code freeze.
0000     99 ;                skip organization dependent code on network operations.
0000    100 ;
0000    101 ;--
0000    102 ;
0000    103
```

```
                     0000   105          .SBTTL  DECLARATIONS
                     0000   106
                     0000   107 ;
                     0000   108 ; Include Files:
                     0000   109 ;
                     0000   110
                     0000   111 ;
                     0000   112 ; Macros:
                     0000   113 ;
                     0000   114
                     0000   115          $DEVDEF
                     0000   116          $IFBDEF
                     0000   117          $FIBDEF
                     0000   118          $FABDEF
                     0000   119          $FWADEF
                     0000   120          $RMSDEF
                     0000   121
                     0000   122 ;
                     0000   123 ; Equated Symbols:
                     0000   124 ;
                     0000   125
00000020             0000   126          FOP=    FAB$L_FOP*8             ; bit offset to file options longword
                     0000   127
                     0000   128 ;
                     0000   129 ; Own Storage:
                     0000   130 ;
                     0000   131
```

```
0000   133          .SBTTL  RMS$CREATE - $CREATE ROUTINE
0000   134
0000   135 ;++
0000   136 ; RMS$CREATE
0000   137 ;
0000   138 ;  RMS$CREATE -
0000   139 ;
0000   140 ;       this routine performs the highest level $create processing.
0000   141 ;       its functions include:
0000   142 ;
0000   143 ;       1. common setup and checks
0000   144 ;       2. dispatch to organization-dependent code
0000   145 ;          which calls common create routines
0000   146 ;
0000   147 ;
0000   148 ; Calling sequence:
0000   149 ;
0000   150 ;       entered from exec as a result of user's calling sys$create
0000   151 ;       (e.g., by using the $create macro).
0000   152 ;
0000   153 ; Input Parameters:
0000   154 ;
0000   155 ;       ap      user's argument list addr
0000   156 ;
0000   157 ; Implicit Inputs:
0000   158 ;
0000   159 ;       the contents of the fab and possible related user interface
0000   160 ;       blocks.
0000   161 ;
0000   162 ; Output Parameters:
0000   163 ;
0000   164 ;       r0      status code
0000   165 ;       r1      destroyed
0000   166 ;
0000   167 ; Implicit Outputs:
0000   168 ;
0000   169 ;       the various fields of the fab are filled in to reflect
0000   170 ;       the status of the created open file. (see rms functional spec for
0000   171 ;       a complete list.)
0000   172 ;       an ifab is initialized to reflect the created file.
0000   173 ;
0000   174 ;       a completion ast is queued if so specified by the user.
0000   175 ;
0000   176 ; Completion Codes:
0000   177 ;
0000   178 ;       standard rms (see functional spec for list).
0000   179 ;
0000   180 ; Side Effects:
0000   181 ;
0000   182 ;       none
0000   183 ;
0000   184 ;--
0000   185
```

```
                     0000   187          $ENTRY  RMS$CREATE
                     0000   188          $TSTPT  CREATE
            FFF7'  30  0006   189          BSBW    RM$FSETI                ; do common setup
                     0009   190
      03 69   3B   E1  0009   191          BBC     #IFB$V_RESTART,(R9),3$  ; See if this is a restart operation.
            FFF0'  31  000D   192          BRW     RM$OPEN_ALT             ; Yes, just make it an open call.
                     0010   193
                     0010   194 ;
                     0010   195 ;NOTE: DOES NOT RETURN ON ERROR
                     0010   196 ;
                     0010   197 ;  an ifab has been set up
                     0010   198 ;  perform various checks
                     0010   199 ;
                     0010   200
                     0010   201 3$:      SSB     #IFB$V_CREATE,(R9)      ; remember we're doing create
      16 A8   01   89  0014   202          BISB3   #FAB$M_PUT,FAB$B_FAC(R8),-
            22 A9          0018   203                  IFB$B_FAC(R9)           ; imply put access
                     001A   204          ASSUME  FAB$V_BRO EQ FAB$V_BIO+1
22 A9   02   05   ED  001A   205          CMPZV   #FAB$V_BIO,#2,IFB$B_FAC(R9),-
                03   001F   206                  #3                      ; both bro & bio set?
                04   12  0020   207          BNEQ    5$                      ; branch if not
      22 A9   20   8A  0022   208          BICB2   #FAB$M_BIO,IFB$B_FAC(R9); clear bio (implied
                     0026   209                                          ; by bro without restrictions)
                     0026   210 5$:
                     0026   211
                     0026   212 ;
                     0026   213 ;  validate rfm
                     0026   214 ;
                     0026   215
                     0026   216          ASSUME  IFB$V_RFM EQ 0
                     0026   217          ASSUME  IFB$S_RFM EQ 4
      1F A8   90     0026   218          MOVB    FAB$B_RFM(R8),-
      50 A9          0029   219                  IFB$B_RFMORG(R9)        ; save rfm in rfmorg
                     002B   220
                     002B   221 ;
                     002B   222 ;  check for rfm in supported range
                     002B   223 ;
                     002B   224
         05   E0     002B   225          BBS     #IFB$V_BIO,-
   09 22 A9          002D   226                  IFB$B_FAC(R9),10$       ; don't check for bio
06    1F A8   91     0030   227          CMPB    FAB$B_RFM(R8),#FAB$C_MAXRFM
         03   1B     0034   228          BLEQU   10$
       00B3   31     0036   229          BRW     ERRRFM
                     0039   230
                     0039   231 ;
                     0039   232 ;  set mrs, gbc, lrl, rat, and fsz from fab
                     0039   233 ;
                     0039   234
      36 A8   B0     0039   235 10$:     MOVW    FAB$W_MRS(R8),-
      60 A9          003C   236                  IFB$W_MRS(R9)           ; save mrs
01    1F A8   91     003E   237          CMPB    FAB$B_RFM(R8),#FAB$C_FIX; fixed length rfm?
         0A   12     0042   238          BNEQ    20$                     ; branch if not
      36 A8   B0     0044   239          MOVW    FAB$W_MRS(R8),-
      52 A9          0047   240                  IFB$W_LRL(R9)           ; save fixed length
         03   14     0049   241          BGTR    20$
       00A6   31     004B   242          BRW     ERRMRS                  ; branch if not > 0
      48 A8   B0     004E   243 20$:     MOVW    FAB$W_GBC(R8),-         ; store gbc
```

```
                64 A9          0051   244                      IFB$W_GBC(R9)
                   03    18     0053   245              BGEQ    21$                     ; ok if non-negative
                 00A0    31     0055   246              BRW     ERRGBC                  ; else it's an error
                                0058   247 21$:
                                0058   248 ;
                                0058   249 ; force stream format files to have RAT non-null.
                                0058   250 ;
                                0058   251              ASSUME  FAB$C_STM LT FAB$C_STMLF
                                0058   252              ASSUME  FAB$C_STM LT FAB$C_STMCR
                                0058   253
                1F A8           0058   254              CMPB    FAB$B_RFM(R8),-         ; stream format?
                   04           005B   255                      #FAB$C_STM
                   0A    1F     005C   256              BLSSU   22$                     ; nope
                   07    93     005E   257              BITB    #<FAB$M_CR!FAB$M_FTN!FAB$M_PRN>,-
                1E A8           0060   258                      FAB$B_RAT(R8)           ; carriage control already set?
                   04    12     0062   259              BNEQ    22$                     ; ok
             1E A8    02    88  0064   260              BISB2   #FAB$M_CR,FAB$B_RAT(R8) ; force RAT=CR
                1E A8    90    0068   261 22$:          MOVB    FAB$B_RAT(R8),-
                51 A9           006B   262                      IFB$B_RAT(R9)           ; save rat
                                006D   263
                                006D   264 ;
                                006D   265 ;  check that only one of 'prn', 'cr' and 'ftn' rat bits are on
                                006D   266 ;
                                006D   267
                   51    D4     006D   268              CLRL    R1                      ; zero-extend the field
          51    1E A8    F8 8F  006F   269              BICB3   #255\<FAB$M_CR!FAB$M_FTN!FAB$M_PRN>,FAB$B_RAT(R8),R1
                06 51    02    0075   270              BBC     #FAB$V_PRN,R1,23$       ; branch if not prn
                03   1F A8    91  0079   271              CMPB    FAB$B_RFM(R8),#FAB$C_VFC; rfm = vfc?
                   69    12     007D   272              BNEQ    ERRRAT                  ; branch if not
                                007F   273              ASSUME  FAB$V_FTN EQ 0
                                007F   274              ASSUME  FAB$V_CR EQ FAB$V_FTN+1
                                007F   275              ASSUME  FAB$V_PRN EQ FAB$V_CR+1
64 17'8F        51    E1        007F   276 23$:         BBC     R1,I^#^B00010111,ERRRAT ; branch if yes
                                0084   277
                                0084   278 ;
                                0084   279 ;  if vfc record format, check for 0 fixed header size and if
                                0084   280 ;  found make it 2 bytes
                                0084   281 ;
                                0084   282
03    1F A8     91              0084   283              CMPB    FAB$B_RFM(R8),#FAB$C_VFC
                   16    12     0088   284              BNEQ    30$                     ; omit check if not vfc
                3F A8    90     008A   285              MOVB    FAB$B_FSZ(R8),-
                5F A9           008D   286                      IFB$B_FSZ(R9)           ; save fsz value
                   04    12     008F   287              BNEQ    29$                     ; branch if value specified
5F A9              02    90     0091   288              MOVB    #2,IFB$B_FSZ(R9)        ; else set default value
                   02    E1     0095   289 29$:          BBC     #FAB$V_PRN,-
  06 51 A9                     0097   290                      IFB$B_RAT(R9),30$       ; branch if not prn
02    5F A9     91              009A   291              CMPB    IFB$B_FSZ(R9),#2        ; fsz > 1?
                   50    1F     009E   292              BLSSU   ERRFSZ                  ; branch if not
                                00A0   293 30$:
                FF5D'    30     00A0   294              BSBW    RM$PRFLNMALT            ; process file name
             65 50       E9     00A3   295              BLBC    R0,EXIT                 ; exit on error
          6A    18       E0     00A6   296              BBS     #FWA$V_WILDCARD,(R10),-
                   56           00A9   297                      ERRSYN                  ; no wildcards allowed
          6A    38       E1     00AA   298 CIF:          BBC     #FWA$V_SLPRESENT,(R10),-; no problem if no searchlist
                   04           00AD   299                      SETORG
          68    39       E0     00AE   300              BBS     #FAB$V_CIF+FOP,(R8),-   ; special loop if CIF
```

```
                 62        00B1   301                     DOCIF
                           00B2   302 ;
                           00B2   303 ;  dispatch to organization-dependent create code
                           00B2   304 ;
                           00B2   305
      04   04    EF         00B2   306 SETORG: EXTZV   #FAB$V_ORG,#FAB$S_ORG,-
50      1D A8                00B5   307                FAB$B_ORG(R8),R0          ; get organization code
23 A9    50    90            00B8   308         MOVB   R0,IFB$B_ORGCASE(R9)      ; and save for dispatching
25 6A    19    E0            00BC   309         BBS    #FWA$V_NODE,(R10),20$     ; branch if network operation
08 69    05    E1            00C0   310         BBC    #DEV$V_SQD,IFB$L_PRIM_DEV(R9),10$    ; branch if not magtape
34 69    13    E1            00C4   311         BBC    #DEV$V_MNT,IFB$L_PRIM_DEV(R9),ERRDNR ; error, if magtape not mounted
30 69    15    E0            00C8   312         BBS    #DEV$V_DMT,IFB$L_PRIM_DEV(R9),ERRDNR ; error, if magtape marked for
                           00CC   313 10$:    CASE   TYPE=B,SRC=R0,-
                           00CC   314                DISPLIST=<RM$CREATE1,C2_BR,RM$CREATE3>
                           00D6   315                                           ; pick up correct routine
                           00D6   316
                           00D6   317 ;
                           00D6   318 ;  error returns
                           00D6   319 ;
                           00D6   320
                           00D6   321         RMSERR  ORG                       ; org not supported
   60 8F    93            00DB   322         BITB   #FAB$M_BIO!FAB$M_BRO,-
   22 A9                   00DE   323                IFB$B_FAC(R9)              ; either bio or bro set?
      29    13            00E0   324         BEQL   EXIT                       ; branch if not (error)
       FF1B'  31           00E2   325         BRW    RM$BIO_CREATE              ; go do block i/o create
                           00E5   326
       0050   31           00E5   327 20$:    BRW    NTCRE                      ; helper branch
                           00E8   328
                           00E8   329 ERRRAT:
       1E     10           00E8   330         BSBB   ERROR
                           00EA   331         RMSERR_WORD    RAT                ; both 'cr' and 'ftn' bits on
                           00EC   332
                           00EC   333 ERRRFM:
       1A     10           00EC   334         BSBB   ERROR
                           00EE   335         RMSERR_WORD    RFM                ; bad rfm field
                           00F0   336
                           00F0   337 ERRFSZ:
       16     10           00F0   338         BSBB   ERROR
                           00F2   339         RMSERR_WORD    FSZ                ; fsz doesn't specify 'put'
                           00F4   340
                           00F4   341 ERRMRS:
       12     10           00F4   342         BSBB   ERROR
                           00F6   343         RMSERR_WORD    MRS                ; mrs zero for fixed length rfm
                           00F8   344
                           00F8   345 ERRGBC:
       0E     10           00F8   346         BSBB   ERROR
                           00FA   347         RMSERR_WORD    GBC                ; GBC must be non-negative
                           00FC   348
                           00FC   349 ERRDNR:
       0A     10           00FC   350         BSBB   ERROR
                           00FE   351         RMSERR_WORD    DNR                ; device not mounted
                           0100   352
   68  3D    E0            0100   353 ERRSYN: BBS    #FAB$V_OFP+FOP,(R8),-
       A6                  0103   354                CIF                        ; wild card o.k. if ofp set
       02     10           0104   355         BSBB   ERROR
                           0106   356         RMSERR_WORD    SYN                ; wild card seen
                           0108   357
```

```
                     0108   358
          50   9E    3C   0108   359 ERROR:  MOVZWL  @(SP)+,R0                ; pick up in-line error code
                     010B   360
            FEF2'    31   010B   361 EXIT:   BRW     RM$CLSCU                 ; clean up and return to user
                     010E   362
00000000'EF          17   010E   363 C2_BR:  JMP     RM$CREATE2
                     0114   364
                     0114   365 ;++
                     0114   366 ;  Process searchlists for a create-if operation.
                     0114   367 ;  The method is to loop thru the entire searchlist trying to access
                     0114   368 ;  the file, if we find one anywhere, we proceed to RM$OPEN_CIF to finish
                     0114   369 ;  the normal $OPEN.   If no file is ever found, then we discard the
                     0114   370 ;  exhausted FWA, re-parse the filespec from scratch and proceed to do
                     0114   371 ;  an ACP create-if operation on the first element of the list.
                     0114   372 ;--
                     0114   373
            FEE9'    30   0114   374 DOCIF:  BSBW    RM$SETDID                ; set up directory
          06 50      E9   0117   375         BLBC    R0,10$                   ; this element failed
            FEE3'    30   011A   376         BSBW    RM$ACCESS                ; try to access the file
          15 50      E8   011D   377         BLBS    R0,20$                   ; found it, continue with OPEN
            FEDD'    30   0120   378 10$:    BSBW    RM$CHK_SLIST1            ; get next element in list
          EE 50      E8   0123   379         BLBS    R0,DOCIF                 ; and try it
          E2 51      E9   0126   380         BLBC    R1,EXIT                  ; don't continue either
            FED4'    30   0129   381         BSBW    RM$DEALLOCATE_FWA        ; release exhausted searchlist
            FED1'    30   012C   382         BSBW    RM$PRFLNMALT             ; get first element again
          D9 50      E9   012F   383         BLBC    R0,EXIT                  ; failed on second attempt!
            FF7D     31   0132   384         BRW     SETORG                   ; and proceed with create-if
                     0135   385
            FEC8'    31   0135   386 20$:    BRW     RM$OPEN_CIF              ; continue with open
                     0138   387
                     0138   388 ;++
                     0138   389 ; Network specific code for $CREATE.
                     0138   390 ;--
                     0138   391
                     0138   392 NTCRE:
              FEC5'  30   0138   393         BSBW    NT$ACCESS                ; Perform network access function
            32 50    E9   013B   394         BLBC    R0,60$                   ; Branch on failure
04 68         26     E1   013E   395         BBC     #FAB$V_SQO+FOP,(R8),10$  ; Branch if SQO not specified
                     0142   396         SSB     #IFB$V_SQO,(R9)          ;  and save bit in IFAB
10 69         3F     E0   0146   397 10$:    BBS     #IFB$V_NSP,(R9),30$      ; Branch if task-to-task
06 68         39     E1   014A   398         BBC     #FAB$V_CIF+FOP,(R8),20$  ; Branch if CIF not specified
              FEAF'  30   014E   399         BSBW    NT$OPEN                  ; Try to open file via remote FAL
            19 50    E8   0151   400         BLBS    R0,50$                   ; Branch on successful open
                     0154   401                                              ; Note that we could check for errors
                     0154   402                                              ;  such as RMS$_DNF, DNR, PRV, WLK, etc.
                     0154   403                                              ;  to avoid getting same error on create
              FEA9'  30   0154   404 20$:    BSBW    NT$CREATE                ; Create file via remote FAL
            10 50    E9   0157   405         BLBC    R0,40$                   ; Branch on failure
               01    BB   015A   406 30$:    PUSHR   #^M<R0>                  ; Save possible alternate success code
              FEA1'  30   015C   407         BSBW    RM$FILLNAM               ; Fill in NAM block with resultant name
               01    BA   015F   408         POPR    #^M<R0>                  ; Restore status code
05 68         39     E1   0161   409         BBC     #FAB$V_CIF+FOP,(R8),40$  ; Branch if CIF not specified
                     0165   410         RMSERR  CREATED                  ; Return alternate success code
              FE93'  31   016A   411 40$:    BRW     RM$CREATEXIT             ; Join common create code
              FE90'  31   016D   412 50$:    BRW     RM$OPEN_CIF              ; Join common open code
                     0170   413                                              ;  (which will call RM$FILLNAM)
                     0170   414 60$:    RMSERR  CRE,R1                   ; Establish default error code
```

```
FE88'  30  0175   415          BSBW    RM$MAPERR                ; Try to map SS code to RMS code
FE85'  31  0178   416          BRW     RM$CREATEXIT             ; join common create code
           017B   417
           017B   418          .END
```

```
$$.PSECT_EP                     = 00000000
$$RMSTEST                       = 0000001A
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
C2_BR                             0000010E R      01
CIF                               000000AA R      01
DEV$V_DMT                       = 00000015
DEV$V_MNT                       = 00000013
DEV$V_SQD                       = 00000005
DOCIF                             00000114 R      01
ERRDNR                            000000FC R      01
ERRFSZ                            000000F0 R      01
ERRGBC                            000000F8 R      01
ERRMRS                            000000F4 R      01
ERROR                             00000108 R      01
ERRRAT                            000000E8 R      01
ERRRFM                            000000EC R      01
ERRSYN                            00000100 R      01
EXIT                              0000010B R      01
FAB$B_FAC                       = 00000016
FAB$B_FSZ                       = 0000003F
FAB$B_ORG                       = 0000001D
FAB$B_RAT                       = 0000001E
FAB$B_RFM                       = 0000001F
FAB$C_FIX                       = 00000001
FAB$C_MAXRFM                    = 00000006
FAB$C_STM                       = 00000004
FAB$C_STMCR                     = 00000006
FAB$C_STMLF                     = 00000005
FAB$C_VFC                       = 00000003
FAB$L_FOP                       = 00000004
FAB$M_BIO                       = 00000020
FAB$M_BRO                       = 00000040
FAB$M_CR                        = 00000002
FAB$M_FTN                       = 00000001
FAB$M_PRN                       = 00000004
FAB$M_PUT                       = 00000001
FAB$S_ORG                       = 00000004
FAB$V_BIO                       = 00000005
FAB$V_BRO                       = 00000006
FAB$V_CIF                       = 00000019
FAB$V_CR                        = 00000001
FAB$V_FTN                       = 00000000
FAB$V_OFP                       = 0000001D
FAB$V_ORG                       = 00000004
FAB$V_PRN                       = 00000002
FAB$V_SQO                       = 00000006
FAB$W_GBC                       = 00000048
FAB$W_MRS                       = 00000036
FOP                             = 00000020
FWA$V_NODE                      = 00000019
FWA$V_SLPRESENT                 = 00000038
FWA$V_WILDCARD                  = 00000018
IFB$B_FAC                       = 00000022
IFB$B_FSZ                       = 0000005F
IFB$B_ORGCASE                   = 00000023
IFB$B_RAT                       = 00000051
IFB$B_RFMORG                    = 00000050
IFB$L_PRIM_DEV                  = 00000000
IFB$S_RFM                       = 00000004
IFB$V_BIO                       = 00000005
IFB$V_CREATE                    = 00000032
IFB$V_NSP                       = 0000003F
IFB$V_RESTART                   = 0000003B
IFB$V_RFM                       = 00000000
IFB$V_SQO                       = 0000002D
IFB$W_GBC                       = 00000064
IFB$W_LRL                       = 00000052
IFB$W_MRS                       = 00000060
NT$ACCESS                         ********    X   01
NT$CREATE                         ********    X   01
NT$OPEN                           ********    X   01
NTCRE                             00000138 R      01
PIO$A_TRACE                       ********    X   01
RM$ACCESS                         ********    X   01
RM$BIO_CREATE                     ********    X   01
RM$CHK_SLIST1                     ********    X   01
RM$CLSCU                          ********    X   01
RM$CREATE1                        ********    X   01
RM$CREATE2                        ********    X   01
RM$CREATE3                        ********    X   01
RM$CREATEXIT                      ********    X   01
RM$DEALLOCATE_FWA                 ********    X   01
RM$FILLNAM                        ********    X   01
RM$FSETI                          ********    X   01
RM$MAPERR                         ********    X   01
RM$OPEN_ALT                       ********    X   01
RM$OPEN_CIF                       ********    X   01
RM$PRFLNMALT                      ********    X   01
RM$SETDID                         ********    X   01
RMS$CREATE                      = FFFFFFFE RG     01
RMS$_CRE                        = 0001C00A
RMS$_CREATED                    = 00010619
RMS$_DNR                        = 00018272
RMS$_FSZ                        = 00018534
RMS$_GBC                        = 000187CC
RMS$_MRS                        = 000185D4
RMS$_ORG                        = 0001860C
RMS$_RAT                        = 0001864C
RMS$_RFM                        = 00018664
RMS$_SYN                        = 000186D4
SETORG                            000000B2 R      01
TPT$L_CREATE                      ********    X   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RM$RMS	0000017B (379.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
ABS	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	31	00:00:00.10	00:00:00.54
Command processing	110	00:00:00.64	00:00:06.15
Pass 1	326	00:00:10.79	00:00:30.20
Symbol table sort	0	00:00:01.54	00:00:02.41
Pass 2	87	00:00:02.09	00:00:04.44
Symbol table output	13	00:00:00.10	00:00:00.17
Psect synopsis output	1	00:00:00.05	00:00:00.25
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	570	00:00:15.31	00:00:44.16

The working set limit was 1500 pages.
60563 bytes (119 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1186 non-local and 23 local symbols.
418 source lines were read in Pass 1, producing 14 object records in Pass 2.
23 pages of virtual memory were used to define 22 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RMS.OBJ]RMS.MLB;1	11
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	18

1294 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

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

RMS0FSCN
LIS

RM3XALLO
LIS

RMS0CONN
LIS

RMS0ENTER
LIS

RMS0ERASE
LIS

RMS0CREAT
LIS

RMS0DELET
LIS

RMS0FIND
LIS

RM3XSUMO
LIS

RMS0DISPL
LIS

RM3XKEYO
LIS

RMS0GET
LIS

RMS0CLOSE
LIS

RMS0DISC
LIS

RMS0EXTEN
LIS

RMS0BLKIO
LIS