



```

RRRRRRRR  MM      MM      SSSSSSSS  000000  CCCCCCCC  RRRRRRRR  EEEEEEEEE  AAAAAA  TTTTTTTTTT
RRRRRRRR  MM      MM      SSSSSSSS  000000  CCCCCCCC  RRRRRRRR  EEEEEEEEE  AAAAAA  TTTTTTTTTT
RR      RR  MMMM  MMMM  SS          00      00  CC          RR      RR  EE          AA      AA  TT
RR      RR  MMMM  MMMM  SS          00      00  CC          RR      RR  EE          AA      AA  TT
RR      RR  MM   MM   MM  SS          00      0000  CC          RR      RR  EE          AA      AA  TT
RR      RR  MM   MM   MM  SS          00      0000  CC          RR      RR  EE          AA      AA  TT
RRRRRRRR  MM      MM      SSSSSS    00  00  00  CC          RRRRRRRR  EEEEEEEE  AA      AA  TT
RRRRRRRR  MM      MM      SSSSSS    00  00  00  CC          RRRRRRRR  EEEEEEEE  AA      AA  TT
RR  RR    MM      MM      SS          0000  00  00  CC          RR  RR    EEE          AAAAAAAAAA  TT
RR  RR    MM      MM      SS          0000  00  00  CC          RR  RR    EEE          AAAAAAAAAA  TT
RR      RR  MM      MM      SS          00      00  00  CC          RR      RR  EE          AA      AA  TT
RR      RR  MM      MM      SS          00      00  00  CC          RR      RR  EE          AA      AA  TT
RR      RR  MM      MM      SSSSSSSS  000000  CCCCCCCC  RR      RR  EEEEEEEEE  AA      AA  TT
RR      RR  MM      MM      SSSSSSSS  000000  CCCCCCCC  RR      RR  EEEEEEEEE  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

```

(2) 105  
(3) 133

DECLARATIONS  
RMS\$CREATE - \$CREATE ROUTINE

```

0000 1          $BEGIN RMSOCREAT,000,RMSRMS,<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 psepts and make entry point single 'S'
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      $STSTPT CREATE
FFF7' 30 0006 189      BSBW      RMS$SETI      ; do common setup
0009 190
03 69 38 E1 0009 191      BBC      #IFBSV_RESTART,(R9),3$ ; See if this is a restart operation.
FFF0' 31 000D 192      BRW      RMS$OPEN_ALT ; Yes, just make it an open call.
0010 193
0010 194
0010 195      ; NOTE: DOES NOT RETURN ON ERROR
0010 196
0010 197      ; an ifab has been set up
0010 198      ; perform various checks
0010 199
0010 200
0010 201 3$:      SSB      #IFBSV_CREATE,(R9) ; remember we're doing create
16 A8 01 89 0014 202      BISB3   #FABSM_PUT,FABSB_FAC(R8),-
22 A9 02 05 ED 0018 203      IFBSB_FAC(R9) ; imply put access
001A 204      ASSUME   FABSV_BRO EQ FABSV_BIO+1
22 A9 02 05 ED 001A 205      CMPZV   #FABSV_BIO,#2,IFBSB_FAC(R9),-
001F 206      #3 ; both bro & bio set?
0020 207      BNEQ    5$ ; branch if not
22 A9 20 8A 0022 208      BICB2   #FABSM_BIO,IFBSB_FAC(R9); clear bio (implied
0026 209      ; by bro without restrictions)
0026 210 5$:
0026 211
0026 212      ; validate rfm
0026 213
0026 214
0026 215
0026 216      ASSUME   IFBSV_RFM EQ 0
0026 217      ASSUME   IFBSB_RFM EQ 4
1F A8 90 0026 218      MOVB    FABSB_RFM(R8),-
50 A9 90 0029 219      IFBSB_RFMORG(R9) ; save rfm in rfmorg
002B 220
002B 221      ;
002B 222      ; check for rfm in supported range
002B 223
002B 224
002B 225      BBS      #IFBSV_BIO,-
09 22 A9 05 E0 002D 226      IFBSB_FAC(R9),10$ ; don't check for bio
06 1F A8 91 0030 227      CMPB    FABSB_RFM(R8),#FABSC_MAXRFM
003 18 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
0039 235 10$:      MOVW    FABSW_MRS(R8),-
60 A9 80 003C 236      IFBSW_MRS(R9) ; save mrs
1F A8 91 003E 237      CMPB    FABSB_RFM(R8),#FABSC_FIX; fixed length rfm?
0A 12 0042 238      BNEQ    20$ ; branch if not
36 A8 80 0044 239      MOVW    FABSW_MRS(R8),-
52 A9 80 0047 240      IFBSW_LRL(R9) ; save fixed length
003 14 0049 241      BGTR    20$
00A6 31 004B 242      BRW      ERRMRS ; branch if not > 0
48 A8 80 004E 243 20$:      MOVW    FABSW_GBC(R8),- ; store gbc

```



```

64 A9      0051      244      IFBSW_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$:
           0058      248      ::
           0058      249      :: force stream format files to have RAT non-null.
           0058      250      ::
           0058      251      ASSUME    FABSC_STM LT FABSC_STMLF
           0058      252      ASSUME    FABSC_STM LT FABSC_STMCR
           0058      253      ::
  1F A8     91 0058      254      CMPB     FABSB_RFM(R8),-      ; stream format?
   04      005B      255      #FABSC_STM
   0A      1F 005C      256      BLSSU   22$      ; nope
   07      93 005E      257      BITB     <FABSM_CR!FABSM_FTN!FABSM_PRN>,-
  1E A8     0060      258      FABSB_RAT(R8)      ; carriage control already set?
   04      12 0062      259      BNEQ     22$      ; ok
  1E A8     02 88 0064      260      BISB2   #FABSM_CR,FABSB_RAT(R8) ; force RAT=CR
  1E A8     02 90 0068      261      22$:    MOVB     FABSB_RAT(R8),-
  51 A9     006B      262      IFBSB_RAT(R9)      ; save rat
           006D      263      ::
           006D      264      ::
           006D      265      :: check that only one of 'prn', 'cr' and 'ftn' rat bits are on
           006D      266      ::
           006D      267      ::
  51 1E A8   F8 8F 8B 006D      268      CLRL     R1      ; zero-extend the field
  06 51     02 E1 006F      269      BICB3   #255<FABSM_CR!FABSM_FTN!FABSM_PRN>,FABSB_RAT(R8),R1
  03 1F A8   91 0075      270      BBC      #FABSV_PRN,R1,23$      ; branch if not prn
           0079      271      CMPB     FABSB_RFM(R8),#FABSC_VFC ; rfm = vfc?
           007D      272      BNEQ     ERRRAT      ; branch if not
           007F      273      ASSUME   FABSV_FTN EQ 0
           007F      274      ASSUME   FABSV_CR EQ FABSV_FTN+1
           007F      275      ASSUME   FABSV_PRN EQ FABSV_CR+1
  64 17'8F 51 E1 007F      276      23$:    BBC      R1,1^#B00010111,ERRRAT ; branch if yes
           0084      277      ::
           0084      278      ::
           0084      279      :: if vfc record format, check for 0 fixed header size and if
           0084      280      :: found make it 2 bytes
           0084      281      ::
           0084      282      ::
  03 1F A8   91 0084      283      CMPB     FABSB_RFM(R8),#FABSC_VFC
           16      12 0088      284      BNEQ     30$      ; omit check if not vfc
           3F A8   90 008A      285      MOVB     FABSB_FSZ(R8),-
           5F A9   008D      286      IFBSB_FSZ(R9)      ; save fsz value
           04      12 008F      287      BNEQ     29$      ; branch if value specified
  5F A9     02 90 0091      288      MOVB     #2,IFBSB_FSZ(R9)      ; else set default value
           02      E1 0095      289      29$:    BBC      #FABSV_PRN,-
  06 51     02 91 0097      290      IFBSB_RAT(R9),30$      ; branch if not prn
  02 5F A9   91 009A      291      CMPB     IFBSB_FSZ(R9),#2      ; fsz > 1?
           50      1F 009E      292      BLSSU   ERRFSZ      ; branch if not
           FF5D' 30 00A0      293      30$:    BSBW     RMSPRFLNMALT      ; process file name
  65 50     E9 00A3      294      BLBC     RO,EXIT      ; exit on error
  6A 18     E0 00A6      295      BBS      #FWASV_WILDCARD,(R10),-
           56      00A9      296      ERRSYN      ; no wildcards allowed
  6A 38     E1 00AA      297      CIF:    BBC      #FWASV_SLPRESENT,(R10),- ; no problem if no searchlist
           04      00AD      298      SETORG
  68 39     E0 00AE      299      BBS      #FABSV_CIF+FOP,(R8),- ; special loop if CIF
           300

```

```

62          00B1 301          DOCIF
           00B2 302          :
           00B2 303          : dispatch to organization-dependent create code
           00B2 304          :
           00B2 305          :
50  04 04 EF 00B2 306 SETORG: EXTZV #FAB$V_ORG,#FAB$S_ORG,-
23  A9 1D A8 00B5 307          FAB$B_ORG(R8),RO ; get organization code
25  6A 50 90 00B8 308          RO,IFB$B_ORGCASE(R9) ; and save for dispatching
08  69 19 E0 00BC 309          #FWA$V_NODE,(R10),20$ ; branch if network operation
34  69 05 E1 00C0 310          BBC #DEV$V_SQD,IFB$S_PRIM_DEV(R9),10$ ; branch if not magtape
30  69 13 E1 00C4 311          BBC #DEV$V_MNT,IFB$S_PRIM_DEV(R9),ERRDNR ; error, if magtape not mounted
           00C8 312          BBS #DEV$V_DMT,IFB$S_PRIM_DEV(R9),ERRDNR ; error, if magtape marked for
           00CC 313 10$: CASE TYPE=B,SRC=RO,-
           00CC 314          DISPLIST=<RMS$CREATE1,C2_BR,RMS$CREATE3>
           00D6 315          ; pick up correct routine
           00D6 316          :
           00D6 317          :
           00D6 318          : error returns
           00D6 319          :
           00D6 320          :
60  8F 93 00DB 321          RMSERR ORG ; org not supported
22  A9 00DE 322          BITB #FAB$M_BIO!FAB$M_BRO,-
           29 13 00E0 323          IFB$B_FAC(R9) ; either bio or bro set?
           FF1B' 31 00E2 324          BEQL EXIT ; branch if not (error)
           00E5 325          BRW RMS$BIO_CREATE ; go do block i/o create
0050 31 00E5 326          20$: BRW NTCRE ; helper branch
           00E8 327          :
           00E8 328          :
           1E 10 00E8 329 ERRRAT: BSBB ERROR
           00EA 330          RMSERR_WORD RAT ; both 'cr' and 'ftn' bits on
           00EC 331          :
           1A 10 00EC 332          ERRRFM: BSBB ERROR
           00EE 333          RMSERR_WORD RFM ; bad rfm field
           00F0 334          :
           16 10 00F0 335          ERRFSZ: BSBB ERROR
           00F2 336          RMSERR_WORD FSZ ; fsz doesn't specify 'put'
           00F4 337          :
           12 10 00F4 338          ERRMRS: BSBB ERROR
           00F6 339          RMSERR_WORD MRS ; mrs zero for fixed length rfm
           00F8 340          :
           0E 10 00F8 341          ERRGBC: BSBB ERROR
           00FA 342          RMSERR_WORD GBC ; GBC must be non-negative
           00FC 343          :
           0A 10 00FC 344          ERRDNR: BSBB ERROR
           00FE 345          RMSERR_WORD DNR ; device not mounted
68  3D E0 0100 346          ERRSYN: BBS #FAB$V_OFP+FOP,(R8),-
           A6 0103 347          CIF ; wild card o.k. if ofp set
           02 10 0104 348          BSBB ERROR
           0106 349          RMSERR_WORD SYN ; wild card seen
           0108 350          :
           0108 351          :
           0108 352          :
           0108 353          :
           0108 354          :
           0108 355          :
           0108 356          :
           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 RMSCLSCU ; clean up and return to user
010E 362
00000000'EF 17 010E 363 C2_BR: JMP RMSCREATE2
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 RMSOPEN_CIF to finish
0114 369 ; the normal $OPEN. If no file is ever found, then we discard the
0114 370 ; exhausted FWA, re-parse the filespec from scratch and proceed to do
0114 371 ; an ACP create-if operation on the first element of the list.
0114 372 ;--
0114 373
FEE9' 30 0114 374 DOCIF: BSBW RMSSETDID ; set up directory
06 50 E9 0117 375 BLBC RO,10$ ; this element failed
FEE3' 30 011A 376 BSBW RMSACCESS ; try to access the file
15 50 E8 011D 377 BLBS RO,20$ ; found it, continue with OPEN
FEDD' 30 0120 378 10$: BSBW RMSCHK_SLIST1 ; get next element in list
EE 50 E8 0123 379 BLBS RO,DOCIF ; and try it
E2 51 E9 0126 380 BLBC R1,EXIT ; don't continue either
FED4' 30 0129 381 BSBW RMSDEALLOCATE_FWA ; release exhausted searchlist
FED1' 30 012C 382 BSBW RMSPRFLNMALT ; get first element again
D9 50 E9 012F 383 BLBC RO,EXIT ; failed on second attempt!
FF7D 31 0132 384 BRW SETORG ; and proceed with create-if
0135 385
FEC8' 31 0135 386 20$: BRW RMSOPEN_CIF ; continue with open
0138 387
0138 388 ;++
0138 389 ; Network specific code for $CREATE.
0138 390 ;--
0138 391
0138 392 NTCRE:
FEC5' 30 0138 393 BSBW NTSACCESS ; Perform network access function
04 68 26 E1 013E 394 BLBC RO,60$ ; Branch on failure
10 69 3F E0 0142 395 BBC #FABS$V_SQO+FOP,(R8),10$ ; Branch if SQO not specified
06 68 39 E1 014A 396 SSB #IFBS$V_SQO,(R9) ; and save bit in IFAB
FEAF' 30 014E 397 10$: BBS #IFBS$V_NSP,(R9),3 $ ; Branch if task-to-task
19 50 E8 0151 398 BBC #FABS$V_CIF+FOP,(R8),20$ ; Branch if CIF not specified
0154 400 BSBW NTSOPEN ; Try to open file via remote FAL
0154 401 BLBS RO,50$ ; Branch on successful open
0154 402 ; Note that we could check for errors
0154 403 ; such as RMS$ DNF, DNR, PRV, WLK, etc.
FEA9' 30 0154 404 20$: BSBW NTSCREATE ; Create file via remote FAL
10 50 E9 0157 405 BLBC RO,40$ ; Branch on failure
01 8B 015A 406 30$: PUSHR #*M<R0> ; Save possible alternate success code
FEA1' 30 015C 407 BSBW RMSFILLNAM ; Fill in NAM block with resultant name
01 8A 015F 408 POPR #*M<R0> ; Restore status code
05 68 39 E1 0161 409 BBC #FABS$V_CIF+FOP,(R8),40$ ; Branch if CIF not specified
FE93' 31 016A 410 RMSERR CREATED ; Return alternate success code
FE90' 31 016D 411 40$: BRW RMSCREATEXIT ; Join common create code
0170 412 50$: BRW RMSOPEN_CIF ; Join common open code
0170 413 ; (which will call RMSFILLNAM)
0170 414 60$: RMSERR CRE,R1 ; Establish default error code

```

RMSOCREAT  
V04-000

DISPATCH FOR CREATE OPERATION  
RM<sup>c</sup> REATE - \$CREATE ROUTINE

N 8

16-SEP-1984 01:13:26 VAX/VMS Macro V04-00  
5-SEP-1984 16:24:43 [RMS.SRC]RMSOCREAT.MAR;1

Page 9  
(4)

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

RMSOCREAT  
Symbol table

DISPATCH FOR CREATE OPERATION

B 9

16-SEP-1984 01:13:26 VAX/VMS Macro V04-00  
5-SEP-1984 16:24:43 [RMS.SRC]RMSOCREAT.MAR;1

Page 10  
(4)

RMS  
V04

\$\$PSECT_EP	=	00000000			IFBSB_RAT	=	00000051		
\$\$RMSTEST	=	0000001A			IFBSB_RFMORG	=	00000050		
\$\$RMS_PBUGCHK	=	00000010			IFBSL_PRIM_DEV	=	00000000		
\$\$RMS_TBUGCHK	=	00000008			IFBSR_RFM	=	00000004		
\$\$RMS_UMODE	=	00000004			IFBSV_BIO	=	00000005		
C2_BR		0000010E	R	01	IFBSV_CREATE	=	00000032		
CIF		000000AA	R	01	IFBSV_NSP	=	0000003F		
DEVSV_DMT	=	00000015			IFBSV_RESTART	=	0000003B		
DEVSV_MNT	=	00000013			IFBSV_RFM	=	00000000		
DEVSV_SQD	=	00000005			IFBSV_SQD	=	0000002D		
DOCIF		00000114	R	01	IFBSW_GBC	=	00000064		
ERRDNR		000000FC	R	01	IFBSW_LRL	=	00000052		
ERRFSZ		000000F0	R	01	IFBSW_MRS	=	00000060		
ERRGBC		000000F8	R	01	NT\$ACCESS	*****	X	01	
ERRMRS		000000F4	R	01	NT\$CREATE	*****	X	01	
ERROR		00000108	R	01	NT\$OPEN	*****	X	01	
ERRRAT		000000E8	R	01	NT\$CRE	00000138	R	01	
ERRRFM		000000EC	R	01	PIOSA_TRACE	*****	X	01	
ERRSYN		00000100	R	01	RMSACCESS	*****	X	01	
EXIT		0000010B	R	01	RMSBIO_CREATE	*****	X	01	
FABSB_FAC	=	00000016			RMSCHK_SLIST1	*****	X	01	
FABSB_FSZ	=	0000003F			RMSCLSCU	*****	X	01	
FABSB_ORG	=	0000001D			RMSCREATE1	*****	X	01	
FABSB_RAT	=	0000001E			RMSCREATE2	*****	X	01	
FABSB_RFM	=	0000001F			RMSCREATE3	*****	X	01	
FABSC_FIX	=	00000001			RMSCREATEXIT	*****	X	01	
FABSC_MAXRFM	=	00000006			RMSDEALLOCATE_FWA	*****	X	01	
FABSC_STM	=	00000004			RMSFILLNAM	*****	X	01	
FABSC_STMCR	=	00000006			RMSFSETI	*****	X	01	
FABSC_STMLF	=	00000005			RMSMAPERR	*****	X	01	
FABSC_VFC	=	00000003			RMSOPEN_ALT	*****	X	01	
FABSL_FOP	=	00000004			RMSOPEN_CIF	*****	X	01	
FABSM_BIO	=	00000020			RMSPRFLNMALT	*****	X	01	
FABSM_BRO	=	00000040			RMSSETDID	*****	X	01	
FABSM_CR	=	00000002			RMS\$CREATE	=	FFFFFFFFE	RG	01
FABSM_FTN	=	00000001			RMS\$CRE	=	0001C00A		
FABSM_PRN	=	00000004			RMS\$CREATED	=	00010619		
FABSM_PUT	=	00000001			RMS\$DNR	=	00018272		
FABSS_ORG	=	00000004			RMS\$FSZ	=	00018534		
FABSV_BIO	=	00000005			RMS\$GBC	=	000187CC		
FABSV_BRO	=	00000006			RMS\$MRS	=	000185D4		
FABSV_CIF	=	00000019			RMS\$ORG	=	0001860C		
FABSV_CR	=	00000001			RMS\$RAT	=	0001864C		
FABSV_FTN	=	00000000			RMS\$RFM	=	00018664		
FABSV_OFP	=	0000001D			RMS\$SYN	=	000186D4		
FABSV_ORG	=	00000004			SETORG	000000B2	R	01	
FABSV_PRN	=	00000002			TPT\$L_CREATE	*****	X	01	
FABSV_SQD	=	00000006							
FABSW_GBC	=	00000048							
FABSW_MRS	=	00000036							
FOP	=	00000020							
FWASV_NODE	=	00000019							
FWASV_SLPRESENT	=	00000038							
FWASV_WILDCARD	=	00000018							
IFBSB_FAC	=	00000022							
IFBSB_FSZ	=	0000005F							
IFBSB_ORGCASE	=	00000023							

-----  
! 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	0000017B ( 379.)	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	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
_\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	11
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
-\$255\$DUA28:[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



