

Syr
NT
NT
NT
NT
NT

RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS	
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS	
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSSSS	
RRR		RRR	MMMMM	MMMMM	SSS	
RRR		RRR	MMMMM	MMMMM	SSS	
RRR		RRR	MMMMM	MMMMM	SSS	
RRR		RRR	MMM	MMM	MMM	SSS
RRR		RRR	MMM	MMM	MMM	SSS
RRR		RRR	MMM	MMM	MMM	SSS
RRR		RRR	MMM	MMM	SSS	
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSS	
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSS	
RRRRRRRRRRRR		MMM		MMM	SSSSSSSSSS	
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	SSSSSSSSSSSS	
RRR		RRR	MMM	MMM	SSSSSSSSSSSS	
RRR		RRR	MMM	MMM	SSSSSSSSSSSS	

NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT
NT

NT
NT
NT
NT
NT
PI

```

RRRRRRRR      MM      MM      11      CCCCCCCC  RRRRRRRR  EEEEEEEEEEE  AAAAAA  TTTTTTTTTTT  EEEEEEEEEEE
RRRRRRRR      MM      MM      11      CCCCCCCC  RRRRRRRR  EEEEEEEEEEE  AAAAAA  TTTTTTTTTTT  EEEEEEEEEEE
RR      RR      MMMM  MMMM  1111      CC      RR      RR  EE      AA      AA  TT      EE
RR      RR      MMMM  MMMM  1111      CC      RR      RR  EE      AA      AA  TT      EE
RR      RR      MM  MM  MM  11      CC      RR      RR  EE      AA      AA  TT      EE
RR      RR      MM  MM  MM  11      CC      RRRRRRRR  EEEEEEEEEEE  AAAAAA  AA      AA  TT      EEEEEEEEEEE
RRRRRRRR      MM      MM      11      CC      RRRRRRRR  EEEEEEEEEEE  AAAAAA  AA      AA  TT      EEEEEEEEEEE
RRRRRRRR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      11      CC      RR      RR  EE      AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      111111  CCCCCCCC  RRRRRRRR  EEEEEEEEEEE  AAAAAA  AA      AA  TT      EEEEEEEEEEE
RR      RR      MM      MM      111111  CCCCCCCC  RR      RR  EEEEEEEEEEE  AA      AA  TT      EEEEEEEEEEE

```

```

LL      111111  SSSSSSSS
LL      111111  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
LLLLLLLLLLL  111111  SSSSSSSS
LLLLLLLLLLL  111111  SSSSSSSS

```

RM1CREATE
Table of contents

SEQUENTIAL-SPECIFIC CREATE

J 9

16-SEP-1984 00:46:26 VAX/VMS Macro V04-00

Page 0

(3) 77
(4) 103

DECLARATIONS
RMSCREATE1 - SEQUENTIAL CREATE ROUTINE

```
0000 1          $BEGIN RM1CREATE,000,RMSRMS1,<SEQUENTIAL-SPECIFIC CREATE>
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 28 :++
0000 29 : Facility: rms32
0000 30 :
0000 31 : Abstract:
0000 32 :
0000 33 :     this routine performs the sequential file
0000 34 :     organization-specific create processing.
0000 35 :
0000 36 : Environment:
0000 37 :     star processor running starlet exec.
0000 38 :
0000 39 : Author: L F Laverdure,      Creation Date: 11-MAY-1977
0000 40 :
0000 41 : Modified By:
0000 42 :
0000 43 :     V03-004 KBT0428      Keith B. Thompson      3-Dec-1982
0000 44 :     Change ifb$w_devbufsiz to ifb$l_devbufsiz
0000 45 :
0000 46 :     V03-003 MCN0001      Maria del C. Nasr      30-Nov-1982
0000 47 :     Do not round up device buffer size to multiple of four
0000 48 :     if device is foreign magtape.
0000 49 :
0000 50 :     V03-002 KBT0137      Keith B. Thompson      20-Aug-1982
0000 51 :     Reorganize psects
0000 52 :
0000 53 :     V03-001 KBT0094      Keith B. Thompson      13-Jul-1982
0000 54 :     Clean up psects
0000 55 :
0000 56 :     V02-017 CDS0001      C Saether      31-Aug-1981
0000 57 :     Always set NORECLK if not disk (RND).
0000 58 :
0000 59 :     V02-016 RAS0020      Ron Schaefer      6-Aug-1981
0000 60 :     Correct fixed record length limit for magtape.
0000 61 :     ANY size record is permitted as long as the blocks
0000 62 :     are larger than 18 bytes.
0000 63 :
0000 64 :     V02-015 RAS0016      Ron Schaefer      6-Aug-1981
0000 65 :     allow stream files to any device; actual GET/PUi logic
0000 66 :     is prepared to ignore the format if necessary.
0000 67 :
0000 68 :     V014  REFORMAT      D M WALP      24-JUL_1980
0000 69 :
0000 70 :     V013  CDS0021      C SAETHER      26-JUL-1979      22:30
0000 71 :     take file sharing checks out of rm$create1
0000 72 :
0000 73 : --
0000 74 :
0000 75 :

```

```
0000 77      .SBTTL DECLARATIONS
0000 78
0000 79      :
0000 80      : Include Files:
0000 81      :
0000 82      :
0000 83      :
0000 84      : Macros:
0000 85      :
0000 86      :
0000 87      : $FABDEF
0000 88      : $IFBDEF
0000 89      : $DEVDEF
0000 90      : $RMSDEF
0000 91      :
0000 92      :
0000 93      : Equated Symbols:
0000 94      :
0000 95      :
00000020 0000 96      : FOP=FAB$L_FOP*8      ; bit offset to fop
0000 97      :
0000 98      :
0000 99      : Own Storage:
0000 100     :
0000 101     :
```

RM
S)
S)
S)
S)
S)
DE
DE
DE
IF
IF
IF
IF
IF
IF
IF
IF
IF
IF
P)
RM
RM
TF

PS
-
:
RM
SA

PH
-
IF
CO
PA
S)
PA
S)
PS
CI
AS

TI
2)
TI
1)
1)

```
0000 103 .SBTTL RMS$CREATE1 - SEQUENTIAL CREATE ROUTINE
0000 104
0000 105 :++
0000 106 : RMS$CREATE1
0000 107 :
0000 108 : RMS$CREATE1
0000 109 :
0000 110 : this routine performs all of the file create
0000 111 : functions that are specific to the sequential
0000 112 : file organization, including:
0000 113 :
0000 114 : 1. checking that write-sharing has not been specified
0000 115 : 2. allowing user to override default block size for magtape
0000 116 : 3. performing eof positioning to start of file
0000 117 : 4. checking xab chain validity
0000 118 : 5. calling the common create routine
0000 119 :
0000 120 : Calling sequence:
0000 121 :
0000 122 : entered via case branch from rm$open
0000 123 : returns by jumping to rm$createxit
0000 124 :
0000 125 : Input Parameters:
0000 126 :
0000 127 : r11 impure area address
0000 128 : r10 fwa address
0000 129 : r9 ifab address
0000 130 : r8 fab address
0000 131 :
0000 132 : Implicit Inputs:
0000 133 :
0000 134 : the contents of the fab, ifab, & fwa.
0000 135 :
0000 136 : Output Parameters:
0000 137 :
0000 138 : r0 status code
0000 139 : r1-r7 destroyed
0000 140 :
0000 141 : Implicit Outputs:
0000 142 :
0000 143 : various fields in the ifab & fab are initialized.
0000 144 :
0000 145 : Completion Codes:
0000 146 :
0000 147 : standard rms
0000 148 :
0000 149 : Side Effects:
0000 150 :
0000 151 : none
0000 152 :
0000 153 :--
0000 154
```

```
0000 156 RMSCREATE1::
0000 157     $STPT  CREATE1
0006 158
0006 159 ;
0006 160 ; providing interlocking
0006 161 ;
0006 162
69 1C  E0 0006 163     BBS     #DEV$V_RND,IFB$S_PRIM_DEV(R9),-
2E      0009 164     SETBL1      ; branch if disk
000A 165
51 A9  08 8A 000A 166     SSB     #IFB$V_NORECLK, (R9) ; never do locking if not disk.
69 05  E1 000E 167     BICB2    #FAB$M_BLK,IFB$B_RAT(R9); clear blk for unit rec devices
16      0012 168     BBC     #DEV$V_SQD,IFB$S_PRIM_DEV(R9),-
0015 169     SETBLS      ; branch if not mt
0016 170
0016 171 ;
0016 172 ; force 'records can't cross block boundaries' and
0016 173 ; check for default block size override for magtape
0016 174 ;
0016 175
51 3C  A8  B0 0016 176     MOVW    FAB$W_BLS(R8),R1 ; get block size
0C      001A 177     BEQL    20$ ; branch if not speced
12 51  B1 001C 178     CMPW    R1,#18 ; is block size ge min.
03      001F 179     BGEQU   10$ ; branch if yes
51 12  B0 0021 180     MOVW    #18,R1 ; minimum record size
48 A9  51 3C 0024 181 10$:  MOVZWL  R1,IFB$S_DEVBUFSIZ(R9) ; set device buffer size
0028 182
0028 183 20$:  SSB     #IFB$V_ANSI_D,(R9) ; flag as ansi
002C 184
002C 185 ;
002C 186 ; round up bls & return it
002C 187 ;
002C 188
69 18  E0 002C 189 SETBLS: BBS     #DEV$V_FOR,IFB$S_PRIM_DEV(R9),-
08      002F 190     SETBLT      ; do not round up if foreign magtape
48 A9  03  C0 0030 191     ADDL2   #3,IFB$S_DEVBUFSIZ(R9)
48 A9  03  CA 0034 192     BICL2   #3,IFB$S_DEVBUFSIZ(R9)
3C A8  48 A9 B0 0038 193 SETBL1: MOVW    IFB$S_DEVBUFSIZ(R9),FAB$W_BLS(R8)
008C C9  06  E1 003D 194     BBC     #DEV$V_SPL,IFB$S_AS_DEV(R9),-
06      0042 195     SETEOF      ; branch if device not spooled
0094 C9  B0 0043 196     MOVW    IFB$S_ASDEVBSIZ(R9),-
3C A8      0047 197     FAB$W_BLS(R8) ; return assigned device buf size
0049 198
0049 199 ;
0049 200 ; set eof block to vbn 1 and first free byte to 0.
0049 201 ;
0049 202
74 A9  D6 0049 203 SETEOF: INCL   IFB$S_EBK(R9) ; eof at vbn 1
004C 204
004C 205 ;
004C 206 ; if fixed record format and 'blk' set in rat,
004C 207 ; make sure record size is less than 1 block
004C 208 ;
004C 209
50 A9  91 004C 210     CMPB    IFB$B_RFMORG(R9),-
01      004F 211     #FAB$C_FIX ; fixed rfm?
0C      12 0050 212     BNEQ    CHKXAB ; branch if not
```



```

51 A9 03 E1 0052 213 BBC #FABS$V_BLK,IFBS$B_RAT(R9),-
      07 0056 214 CHKXAB ; branch if 'blk' not speced
60 A9 B1 0057 215 CMPW IFBS$W_MRS(R9),-
48 A9 005A 216 IFBS$L_DEVBUFSIZ(R9) ; does record fit?
      14 1A 005C 217 BGTRU RMS$CRE_ERRMRS ; branch if not
      005E 218
      005E 219 :++
      005E 220 :
      005E 221 : entry point for block i/o create for unknown file org
      005E 222 :
      005E 223 :--
      005E 224
      005E 225 RMS$BIO_CREATE::
      005E 226
      005E 227 :
      005E 228 : process allocation xab if present, and set deq and rtdeq
      005E 229 :
      005E 230
      005E 231 CHKXAB:
      FF9F' 30 005E 232 BSB# RMS$SETALLOC
      07 50 E9 0061 233 BLBC RO,EXIT
      0064 234
      0064 235 :
      0064 236 : call the common create file routine
      0064 237 :
      0064 238
      FF99' 30 0064 239 BSBW RMS$CREATECOM
      0067 240
      0067 241 :
      0067 242 : note that file is positioned at eof in the ifab
      0067 243 : so that connect will be set up for $put.
      0067 244 :
      0067 245
      0067 246
      FF92' 31 006B 247 EXIT: SSB #IFBS$V_EOF,(R9) ; flag eof
      BRW RMS$CREATEXIT

```

```
006E 249
006E 250 ;
006E 251 ; handle errors
006E 252 ;
006E 253 ;
06 10 006E 254 RMSCRE_ERRRFM::
006E 255 BSBB ERROR ; rfm not udf,fix,var, or vfc
0070 256 RMSERR_WORD RFP ; for non-disk device
0072 257
02 10 0072 258 RMSCRE_ERRMRS::
0072 259 BSBB ERROR ; fixed record length <18
0074 260 RMSERR_WORD MRS ; for magtape or > bls and 'blk' set
0076 261
50 9E 3C 0076 262 ERROR: MOVZWL @(SP)+,R0 ; pick up error code
FF84' 31 0079 263 BRW RMSCREATEXIT ; and get out
007C 264
007C 265 .END
```

RMICREATE
Symbol table

SEQUENTIAL-SPECIFIC CREATE

E 10

16-SEP-1984 00:46:26
5-SEP-1984 16:27:15

VAX/VMS Macro V04-00
[RMS.SRC]RMICREATE.MAR;1

Page 8
(7)

```

$$PSECT EP           = 00000000
$$RMSTEST           = 0000001A
$$RMS_PBUGCHK      = 00000010
$$RMS_TBUGCHK      = 00000008
$$RMS_UMODE        = 00000004
CHKXAB             = 0000005E R    01
DEVSV_FOR          = 00000018
DEVSV_RND          = 0000001C
DEVSV_SPL          = 00000006
DEVSV_SQD          = 00000005
ERROR              = 00000076 R    01
EXIT               = 0000006B R    01
FABSC_FIX          = 00000001
FABSL_FOP          = 00000004
FABSM_BLK          = 00000008
FABSV_BLK          = 00000003
FABSW_BLS          = 0000003C
FOP                = 00000020
IFBSB_RAT          = 00000051
IFBSB_RFMORG       = 00000050
IFBSL_ASDEVBSIZ   = 00000094
IFBSL_AS_DEV       = 0000008C
IFBSL_DEVBUFSIZ   = 00000048
IFBSL_EBK          = 00000074
IFBSL_PRIM_DEV     = 00000000
IFBSV_ANSI_D       = 00000026
IFBSV_EOF          = 00000021
IFBSV_NORECLK     = 00000033
IFBSW_MRS          = 00000060
PICSA_TRACE        ***** X    01
RMSBIO_CREATE      0000005E RG   01
RMSCREATE1         00000000 RG   01
RMSCREATELOM       ***** X    01
RMSCREATEEXIT      ***** X    01
RMSCRE_ERRMRS      00000072 RG   01
RMSCRE_ERRRFM      0000006E RG   01
RMSSETALLOC        ***** X    01
RMS$MRS            = 000185D4
RMS$RFM            = 00018664
SETBL1             00000038 R    01
SETBLS             0000002C R    01
SETEOF             00000049 R    01
TPTSL_CREATE1     ***** 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
RMSRMS1	0000007C (124.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$AB\$\$	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	33	00:00:00.09	00:00:00.49
Command processing	114	00:00:00.71	00:00:04.11
Pass 1	236	00:00:05.92	00:00:18.44
Symbol table sort	0	00:00:00.68	00:00:01.06
Pass 2	57	00:00:01.23	00:00:03.30
Symbol table output	7	00:00:00.07	00:00:00.07
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	450	00:00:08.72	00:00:27.49

The working set limit was 1350 pages.
33677 bytes (66 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 665 non-local and 5 local symbols.
265 source lines were read in Pass 1, producing 13 object records in Pass 2.
16 pages of virtual memory were used to define 15 macros.

↑-----↑
! Macro library statistics !
↑-----↑

Macro library name	Macros defined
-----	-----
-\$255\$DUA28:[RMS OBJ]RMS.MLB;1	7
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	11

747 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RM1CREATE/OBJ=OBJ\$:RM1CREATE MSRC\$:RM1CREATE/UPDATE=(ENH\$:RM1CREATE)+EXECMLS/LIB+LIB\$:RMS/LIB

RM1CONN
LIS

RM1GET
LIS

RM1INPSON
LIS

RM1DISCON
LIS

RM1GETINT
LIS

RM1CREATE
LIS

RM1JOURNL
LIS