



```

RRRRRRRR      MM      MM      SSSSSSSS      000000      TTTTTTTTTT      RRRRRRRR      UU      UU      NN      NN      CCCCCCCC
RRRRRRRR      MM      MM      SSSSSSSS      000000      TTTTTTTTTT      RRRRRRRR      UU      UU      NN      NN      CCCCCCCC
RR      RR      MMMM      MMMM      SS      00      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MMMM      MMMM      SS      00      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MM      MM      MM      SS      00      0000      TT      RR      RR      UU      UU      NNNN      NN      CC
RR      RR      MM      MM      MM      SS      00      0000      TT      RR      RR      UU      UU      NNNN      NN      CC
RRRRRRRR      MM      MM      SSSSSS      00      00      00      TT      RRRRRRRR      UU      UU      NN      NN      CC
RRRRRRRR      MM      MM      SSSSSS      00      00      00      TT      RRRRRRRR      UU      UU      NN      NN      CC
RR      RR      MM      MM      SS      0000      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MM      MM      SS      0000      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MM      MM      SS      00      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MM      MM      SS      00      00      TT      RR      RR      UU      UU      NN      NN      CC
RR      RR      MM      MM      SSSSSSSS      000000      TT      RR      RR      UUUUUUUUUU      NN      NN      CCCCCCCC
RR      RR      MM      MM      SSSSSSSS      000000      TT      RR      RR      UUUUUUUUUU      NN      NN      CCCCCCCC

```

```

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

RMSOTRUNC  
Table of contents

TRUNCATE SEQUENTIAL FILE

C 3

16-SEP-1984 01:33:25 VAX/VMS Macro V04-00

Page 0

RI  
V(

(3) 78  
(4) 105

DECLARATIONS  
STRUNCATE ROUTINE

```
0000 1          $BEGIN RMSOTRUNC,000,RMSRMS,<TRUNCATE SEQUENTIAL FILE>
0000 2
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :*  ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :*  TRANSFERRED.
0000 17 :*
0000 18 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :*  CORPORATION.
0000 21 :*
0000 22 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
```

```
0000 29 : **
0000 30 : Facility: rms32
0000 31 :
0000 32 : Abstract:
0000 33 :     this module provides the $truncate rms service for magtape and
0000 34 :     sequential disk files.
0000 35 :
0000 36 : Environment:
0000 37 :     star processor running starlet exec.
0000 38 :
0000 39 : Author: l f laverdure,          Creation Date: 9-JAN-1977
0000 40 :
0000 41 : Modified By:
0000 42 :
0000 43 :     V03-004 TSK0002          Tamar Krichevsky          27-Jun-1983
0000 44 :     Fix bug introduced in V03-003.
0000 45 :
0000 46 :     V03-003 TSK0001          Tamar Krichevsky          21-Jun-1983
0000 47 :     Add support for journaling $TRUNCATE operations.
0000 48 :
0000 49 :     V03-002 KBT0408          Keith B. Thompson          24-Nov-1982
0000 50 :     Replace a movc with a movl and movw
0000 51 :
0000 52 :     V03-001 KBT0195          Keith B. Thompson          23-Aug-1982
0000 53 :     Reorganize psects and rename entry point to single '$'
0000 54 :
0000 55 :     V02-010 CDS0001          C Saether                  5-Nov-1981
0000 56 :     Set ifb$v_rw_attr flag in truncate routine.
0000 57 :
0000 58 :     V02-009 PSK0002          Paulina Knibbe             26-Aug-1981
0000 59 :     Fix a broken branch by changing psects (again).
0000 60 :
0000 61 :     V02-008 RAS0018          Ron Schaefer               7-Aug-1981
0000 62 :     Fix another broken branch for stream files by
0000 63 :     changing psects.
0000 64 :
0000 65 :     V02-007 PSK0001          Paulina S. Knibbe         24-Jul-1981
0000 66 :     Fix another broken branch.
0000 67 :
0000 68 :     V03-006 KPL0001          Peter Lieberwirth         8-Jun-1981
0000 69 :     Fix a broken branch.
0000 70 :
0000 71 :     V02-005 REFORMAT          Ron Schaefer              30-Jul-1980    09:54
0000 72 :     Reformat the source.
0000 73 :
0000 74 : --
0000 75 :
0000 76 :
```



```
0000 105 .SBTTL $TRUNCATE ROUTINE
0000 106
0000 107 :++
0000 108 : RM$TRUNCATE:
0000 109 : this routine truncates a magtape or disk file of sequential organization,
0000 110 : declaring eof at the current record pointer. truncate must immediately
0000 111 : follow a successful $get, $find, or $update.
0000 112 : the file must not be accessed for block i/o.
0000 113
0000 114 : Calling sequence:
0000 115
0000 116 : entered as a result of user's calling sys$truncate
0000 117
0000 118 : Input Parameters:
0000 119
0000 120 : ap user's argument list
0000 121
0000 122 : Implicit Inputs:
0000 123
0000 124 : the contents of the rab
0000 125
0000 126 : Output Parameters:
0000 127
0000 128 : r0 status code
0000 129 : r1 destroyed
0000 130
0000 131 : Implicit Outputs:
0000 132
0000 133 : the sts field of the rab
0000 134
0000 135 : Completion Codes:
0000 136
0000 137 : standard rms
0000 138
0000 139 : Side Effects:
0000 140
0000 141 : there is no current record following a truncate
0000 142
0000 143 :--
0000 144
0000 145
0000 146 :
0000 147 : verify operation allowed (accessed for record i/o to a magtape or disk,
0000 148 : having a current record)
0000 149 : and set eof to the record pointer, backspacing 1 block if magtape
0000 150 :
0000 151
0000 152 $ENTRY RMS$TRUNCATE
0000 153 $STPT TRUNCATE
0006 154 $RABSET FAC=IFB$V_TRN,BIO=0,CFLG=1 ; set up stream
000A 155
000A 156 :
000A 157 : (must be truncate accessed,
000A 158 : not doing block i/o,
000A 159 : clear find last)
000A 160 :
000A 161
```

```

76 6A 3E E0 000A 162      BBS      #IFBSV DAP (R10),NTTRUNC ; branch if network operation
                000E 163      ASSUME   IFBSC_SEQ EQ 0
23 AA 95 000E 164      TSTB    IFBSB_ORGCASE(R10) ; seq. file org.?
79 12 0011 165      BNEQ    ERRIOP ; branch if not
                0013 166      ASSUME   DEV$V_REC EQ 0
76 6A E8 0013 167      BLBS    IFBSL_PRIM_DEV(R10),ERRIOP ; branch if unit record dev.
62 A9 B5 0016 168      TSTW    IRBSW_CSIZTR9) ; do we have a current record?
78 13 0019 169      BEQL    ERRCUR ; branch if not
                001B 170
                001B 171 ;
                001B 172 ; validity checks all o.k.
                001B 173 ;
                001B 174 ; if this is a magtape, backspace to rewrite the block just read.
                001B 175 ;
                001B 176 ;
09 6A 05 E1 001B 177      BBC      #DEV$V_SQD,IFBSL_PRIM_DEV(R10),20$ ; branch if not magtape
56 01 CE 001F 178      MNEGL   #1,R6 ; backspace 1 block
FFDB' 30 0022 179      BSBW    RM$SPACE MT ; go space it
56 50 E9 0025 180      BLBC    R0,ERRXIT ; get out on error
                0028 181
00A0 CA 95 0028 182 20$: TSTB    IFBSB_JNLFLG(R10) ; Is the file being journaled?
                OF 13 002C 183      BEQL    30$ ; No, set eof and nrp pointers
                1C BB 002E 184      PUSHR   #*M<R2, R3, R4 > ; Save the contents of registers
                1B DD 0030 185      PUSHL   #RJR$ TRUNCATE ; Yes, journal the $TRUNCATE operation
00000000'EF 16 0032 186      JSB     RM$SEQJNL ; Constructed and write journal entry
                1C BA 0038 187      POPR    #*M<R2, R3, R4 > ; Restore registers
41 50 E9 003A 188      BLBC    R0, ERRXIT ; Get out on error
                003D 189 ;
                003D 190 ; set nrp and eof from rp
                003D 191 ;
                003D 192 ;
40 A9 48 A9 7D 003D 193 30$: MOVQ    IRBSL_RP(R9),IRBSL_NRP(R9) ; set nrp
74 AA 48 A9 D0 0042 194      MOVL    IRBSL_RP(R9),IFBSL_EBK(R10) ; and eof
5C AA 4C A9 B0 0047 195      MOVW    IRBSW_RP_OFF(R9),IFBSW_FFB(R10)
                004C 196      SSB     #IFBSV_RQ_ATTR,(R10) ; note to rewrite attributes.
                0050 197 ;
                0050 198 ;
                0050 199 ; clean up current bdb.
                0050 200 ;
                0050 201 ; if the new eof vbn is in the buffer, point to it and say it's dirty,
                0050 202 ; thus causing it to get rewritten with appropriate padding,
                0050 203 ; else just release the buffer.
                0050 204 ;
                0050 205 ;
54 20 A9 D0 0050 206      MOVL    IRBSL_CURBDB(R9),R4 ; is there a current bdb?
                21 13 0054 207      BEQL    SUCXIT ; branch if not
51 48 A9 1C A4 C3 0056 208      SUBL3   BDB$B_VBN(R4),IRBSL_RP_VBN(R9),R1 ; get relative vbn
51 51 F6 005C 209      CVTLB   R1,R1 ; make byte value
                10 1D 005F 210      BVS     40$ ; branch if not in range
48 A4 51 91 0061 211      CMPB    R1,BDB$B_REL_VBN(R4) ; is it in buffer?
                OA 1A 0065 212      BGTRU   40$ ; branch if not
48 A4 51 90 0067 213      MOVB    R1,BDB$B_REL_VBN(R4) ; point to this block
OA A4 02 88 006B 214      BISB2   #BDB$M_DRT,BDB$B_FLGS(R4) ; say dirty
                06 11 006F 215      BRB     SUCXIT
                0071 216 ;
                0071 217 ;
                0071 218 ; release unneeded bdb

```



```

0071 219 :
0071 220 :
FF8C' 30 0071 221 40$: BSBW RMSRLNERR
20 A9 D4 0074 222 CLRL IRBSL_CURBDB(R9) ; say no current bdb
0077 223 :
0077 224 :
0077 225 : successful exit (fall thru to error exit)
0077 226 :
0077 227 :
0077 228 SUCXIT: SSB #IRBSV_EOF,(R9) ; say at eof
007B 229 RMSSUC
007E 230 :
007E 231 :
007E 232 : error exit - say no current record
007E 233 :
007E 234 :
62 A9 B4 007E 235 ERRXIT: CLRW IRBSW CSIZ(R9)
FF7C' 31 0081 236 EXIT: BRW RMSEX RMS
0084 237 :
0084 238 :
0084 239 : perform network truncate function
0084 240 :
0084 241 :
0084 242 NTRUNC:
FF79' 30 0084 243 BSBW NTSTRUNCATE ; truncate file via remote fal
F4 50 E9 0087 244 BLBC RO,ERRXIT ; branch on error
EB 11 008A 245 BRB SUCXIT ; branch aid
008C 246 :
008C 247 :
008C 248 : handle invalid device or organization error.
008C 249 :
008C 250 :
EE 11 008C 251 ERRIOP: RMSERR IOP
0091 252 BRB EXIT
0093 253 :
0093 254 :
0093 255 : handle no current record error.
0093 256 :
0093 257 :
E7 11 0093 258 ERRCUR: RMSERR CUR
0098 259 BRB EXIT
009A 260 :
009A 261 .END

```

```

$$PSECT EP = 00000000
$$RMSTEST = 0000001A
$$RMS_PBUGCHK = 00000010
$$RMS_TBUGCHK = 00000008
$$RMS_UMODE = 00000004
BDBSB_FLGS = 0000000A
BDBSB_REL_VBN = 00000048
BDBSL_VBN = C000001C
BDBSM_DRT = 00000002
DEVSV_REC = 00000000
DEVSV_SQD = 00000005
ERRCUR 00000093 R 01
ERRIOP 0000008C R R 01
ERRXIT 0000007E R R 01
EXIT 00000081 R 01
IFBSB_JNLFLG = 000000A0
IFBSB_ORGCASE = 00000023
IFBSC_SEQ = 00000000
IFBSL_EBK = 00000074
IFBSL_PRIM_DEV = 00000000
IFBSV_DAP = 0000003E
IFBSV_RW_ATTR = 00000034
IFBSV_TPN = 00000004
IFBSW_FFB = 0000005C
IRBSL_CURBDB = 00000020
IRBSL_NRP = 00000040
IRBSL_RP = 00000048
IRBSL_RP_VBN = 00000048
IRBSV_EOF = 00000021
IRBSW_CSIZ = 00000062
IRBSW_RP_OFF = 0000004C
NTSTRUNCATE ***** X 01
NTRUNC 00000084 R 01
PIOA_TRACE ***** X 01
RJRS_TRUNCATE = 0000001B
RMSEXAMS ***** X 01
RMSRLNERR ***** X 01
RMSRSET ***** X 01
RMSSEQJNL ***** X 01
RMSSPACE_MT ***** X 01
RMSSTRUNCATE = FFFFFFFE RG 01
RMS_CUR = 000184B4
RMS_IOP = 00018574
SUCXIT 00000077 R 01
TPTSL_TRUNCATE ***** 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
RMSRMS	0000009A ( 154.)	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	35	00:00:00.09	00:00:00.82
Command processing	133	00:00:00.75	00:00:09.40
Pass 1	295	00:00:09.15	00:00:27.42
Symbol table sort	0	00:00:01.32	00:00:02.20
Pass 2	58	00:00:01.58	00:00:04.70
Symbol table output	7	00:00:00.09	00:00:00.12
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	531	00:00:13.00	00:00:44.68

The working set limit was 1350 pages.  
51310 bytes (101 pages) of virtual memory were used to buffer the intermediate code.  
There were 60 pages of symbol table space allocated to hold 1026 non-local and 5 local symbols.  
261 source lines were read in Pass 1, producing 13 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	14
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	19

1159 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

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

