



```

RRRRRRRR      MM      MM      SSSSSSSS      000000      EEEEEEEEEEE      RRRRRRRR      AAAAAA      SSSSSSSS      EEEEEEEEEEE
RRRRRRRR      MM      MM      SSSSSSSS      000000      EEEEEEEEEEE      RRRRRRRR      AAA^AA      SSSSSSSS      EEEEEEEEEEE
RR      RR      MMMM      MMMM      SS      00      00      EE      RR      RR      AA      ^A      SS      EE
RR      RR      MMMM      MMMM      SS      00      00      EE      RR      RR      AA      AA      SS      EE
RR      RR      MM      MM      SS      00      0000      EE      RR      RR      AA      AA      SS      EE
RR      RR      MM      MM      SS      00      0000      EE      RR      RR      AA      AA      SS      EE
RRRRRRRR      MM      MM      SSSSSS      00      00      00      EEEEEEEEE      RRRRRRRR      AA      AA      SSSSSS      EEEEEEEEE
RRRRRRRR      MM      MM      SSSSSS      00      00      00      EEEEEEEEE      RRRRRRRR      AA      AA      SSSSSS      EEEEEEEEE
RR      RR      MM      MM      SS      0000      00      EE      RR      RR      AAAAAAAAAA      SS      EE
RR      RR      MM      MM      SS      0000      00      EE      RR      RR      AAAAAAAAAA      SS      EE
RR      RR      MM      MM      SS      00      00      EE      RR      RR      AA      AA      SS      EE
RR      RR      MM      MM      SS      00      00      EE      RR      RR      AA      AA      SS      EE
RR      RR      MM      MM      SSSSSSSS      000000      EEEEEEEEEEE      RR      RR      AA      AA      SSSSSSSS      EEEEEEEEEEE
RR      RR      MM      MM      SSSSSSSS      000000      EEEEEEEEEEE      RR      RR      AA      AA      SSSSSSSS      EEEEEEEEEEE

```

```

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

```

(3) 72  
(4) 99

DECLARATIONS  
RMS\$ERASE - ERASE FILENAME STRING ROUTINE

```
0000 1          $BEGIN RMSOERASE,000,RMSRMS,<DELETE FILE 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 28 :++
0000 29 :
0000 30 : Facility: rms32
0000 31 :
0000 32 : Abstract:
0000 33 :         this routine is the highest level control
0000 34 :         routine to perform the $erase function.
0000 35 :
0000 36 : Environment:
0000 37 :         star processor running starlet exec.
0000 38 :
0000 39 : Author:      L F Laverdure,   creation date: 27-SEP-1977
0000 40 :
0000 41 : Modified By:
0000 42 :
0000 43 :         V03-006 JEJ0041      J E Johnson      20-Jun-1984
0000 44 :         Don't map errors from network deletes (NT$ERASE) as
0000 45 :         they have already been processed.
0000 46 :
0000 47 :         V03-005 RAS0239      Ron Schaefer    20-Jan-1984
0000 48 :         Don't RMSMAPERR errors from RMSASSIGN or RM$SETDID
0000 49 :         as they have already been mapped if possible.
0000 50 :         Fixup to RAS0224.
0000 51 :
0000 52 :         V03-004 RAS0224      Ron Schaefer    19-Dec-1983
0000 53 :         Clean up search list implementation (KBT0539),
0000 54 :         by basically backing it out. It is inappropriate
0000 55 :         to look all thru the list for a file to delete.
0000 56 :
0000 57 :         V03-003 KBT0539      Keith B. Thompson  7-Jun-1983
0000 58 :         Add search list support
0000 59 :
0000 60 :         V03-002 KBT0518      Keith B. Thompson  23-May-1983
0000 61 :         RM$XPFN moved so ref to it changed to JSB
0000 62 :
0000 63 :         V03-001 KBT0180      Keith B. Thompson  23-Aug-1982
0000 64 :         Reorganize psects and rename entry point to single '$'
0000 65 :
0000 66 :         V02-007 MCN0001      Maria del C. Nasr   29-Jul-1981
0000 67 :         Rename entry point to RM$$$ to support long branches.
0000 68 :
0000 69 : --
0000 70 :
```

```
0000 72          .SBTTL  DECLARATIONS
0000 73
0000 74  :
0000 75  : Include Files:
0000 76  :
0000 77  :
0000 78  :
0000 79  : Macros:
0000 80  :
0000 81
0000 82          $IFBDEF
0000 83          $DEVDEF
0000 84          $NAMDEF
0000 85          $FWADEF
0000 86          $FIBDEF
0000 87          $IODEF
0000 88          $RMSDEF
0000 89
0000 90  :
0000 91  : Equated Symbols:
0000 92  :
0000 93  :
0000 94  :
0000 95  : Own Storage:
0000 96  :
0000 97
```

```

0000 99          .SBTTL RMS$ERASE - ERASE FILENAME STRING ROUTINE
0000 100
0000 101 :++
0000 102 :
0000 103 : RMS$ERASE
0000 104 :
0000 105 :     The filename, issues the acp qio function to delete the file, and
0000 106 :     then deletes the ifab.
0000 107 :
0000 108 : Calling sequence:
0000 109 :
0000 110 :     Entered from exec as a result of user's calling sys$erase
0000 111 :
0000 112 : Input Parameters:
0000 113 :
0000 114 :     ap      user's argument list addr
0000 115 :
0000 116 : Implicit Inputs:
0000 117 :
0000 118 :     The contents of the fab (fna, fns, dna, dns) and related nam block.
0000 119 :
0000 120 : Output Parameters:
0000 121 :
0000 122 :     r1      destroyed
0000 123 :     r0      status code
0000 124 :
0000 125 : Implicit Outputs:
0000 126 :
0000 127 :     The sts and stv fields of the fab are output along with the
0000 128 :     various fields of the nam block (dvi, fid, did, esl and the buffer at esa,
0000 129 :     rsl and the buffer at rsa)
0000 130 :     to reflect the status of the $erase operation. (see rms functional
0000 131 :     spec for a complete list.)
0000 132 :
0000 133 :     A completion ast is queued if specified in the user arglist.
0000 134 :
0000 135 : Completion Codes:
0000 136 :
0000 137 :     Standard rms (see functional spec for list).
0000 138 :
0000 139 : Side Effects:
0000 140 :     none
0000 141 :
0000 142 :--
0000 143 :
0000 144 :     $ENTRY RMS$ERASE
0000 145 :     $STPT  ERASE
0000 146 :     BSBW   RMSFSETI           ; create an ifab
0000 147 :                                     ; does not return on error
0000 148 :     JSB   RMS$XPFN           ; expand file name
0000 149 :     BLBC  RO,CLEANUP        ; continue if ok
0000 150 :
0000 151 :
0000 152 :     We parse a file name ok
0000 153 :
0000 154 :
0000 155 :     TSTB  FWASB_ESCFLG(R10)   ; ppf indicated?
  
```

FFF7' 30 0006 146

00000000'EF 16 0009 148

50 50 E9 000F 149

0012 150

0012 151

0012 152

0012 153

0012 154

OC AA 95 0012 155

```

4A 6A 55 12 0015 156 BNEQ ERRIOP ; branch if yes
      18 E0 0017 157 BBS #FWASV WILDCARD,(R10),ERRWLD ; branch if wild card seen
      FFE2' 30 0018 158 BSBW RMS$ASSIGN ; assign channel to device
      41 50 E9 001E 159 BLBC RO,CLEANUP ; failure
5F 69 3E E0 0021 160 BBS #IFBSV_DAP,(R9),NTERASE ; branch if network file oper
      69 1C E1 0025 16 BBC #DEV$V_RND,IFBSL_PRIM_DEV(R9),- ; branch if not disk
      43 0028 162 ERRIOP
      FFD4' 30 0029 163 BSBW RMS$SETDID ; lookup the directory name
      33 50 E9 002C 164 BLBC RO,CLEANUP ; failure
      002F 165
      002F 166 ;
      002F 167 ; The assign and set did succeeded
      002F 168 ;
      002F 169 ; Note: R7 set to NAM blk addr
      002F 170 ;
      002F 171 ;
01FE CA D5 002F 172 60$: TSTL FIBSW_DID+FWAST_FIBBUF(R10) ; did specified?
      1A 12 0033 173 BNEQ DELETE ; branch if yes (implies fid=0)
01F8 CA D5 0035 174 TSTL FIBSW_FID+FWAST_FIBBUF(R10) ; fid specified?
      42 13 0039 175 BEQL ERRBUG ; branch if none
      003B 176 ;
      003B 177 ;
      003B 178 ; since fid is set, there must be a nam blk. set did from the nam blk.
      003B 179 ;
      003B 180 ;
      2A A7 B0 003B 181 MOVW NAMSW_DID(R7),-
01FE CA 003E 182 FIBSW_DID_NUM+FWAST_FIBBUF(R10)
      0C 13 0041 183 BEQL DELETE ; branch if none
      2C A7 D0 0043 184 MOVL NAMSW_DID_SEQ(R7),-
0200 CA 0046 185 FIBSW_DID_SEQ+FWAST_FIBBUF(R10)
      0049 186 SSB #FIB$V_FINDFID,- ; flag find by file id
      0049 187 FIBSW_NMCTL+FWAST_FIBBUF(R10)
      004F 188 ;
      004F 189 ;
      004F 190 ; set up the qio parameters and issue the delete qio
      004F 191 ;
      004F 192 ;
      7E 7C 004F 193 DELETE: CLRQ -(SP) ; p6=p5=0
      FFAC' 30 0051 194 BSBW RMS$FCP_P4_P2 ; build p4 thru p2
      0054 195 ;
      0054 196 ;
      0054 197 ; (resultant name, name string)
      0054 198 ;
      0054 199 ;
50 0135 8F 3C 0054 200 MOVZWL #IOS_DELETE!IOSM_DELETE,RO ; function code
      FFA4' 30 0059 201 BSBW RMS$FCPFNC ; do the delete
      14 50 E9 005C 202 BLBC RO,ERRDLT ; branch on error
      FF9E' 30 005F 203 FILNAM: BSBW RMS$FILLNAM ; fill in nam blk, if any
      0062 204 ;
      0062 205 CLEANUP:
      FF9B' 31 0062 206 BRW RMS$CLSCU ; all done - evaporate ifab
      0065 207 ;
      0065 208 ;++
      0065 209 ;
      0065 210 ; process errors
      0065 211 ;
      0065 212 ;--

```



```

0065 213
0065 214 ERRWLD:
0065 215 RMSERR WLD
F6 11 006A 216 BRB CLEANUP
006C 217
006C 218 ERRIOP:
006C 219 RMSERR IOP
EF 11 0071 220 BRB CLEANUP
0073 221
0073 222 ERRDLT:
0073 223 RMSERR MKD,R1
FF85' 30 0078 224 BSBW RMS$MAPERR
E5 11 007B 225 BRB CLEANUP
007D 226
007D 227 ;
007D 228 ; Bad one!
007D 229 ;
007D 230
007D 231 ERRBUG: RMSTBUG FTLS_NODIDORFID ; no did or fid
0084 232
0084 233 ;++
0084 234 ;
0084 235 ; process network erase function
0084 236 ;
0084 237 ;--
0084 238
0084 239 NTERASE:
FF79' 30 0084 240 BSBW NT$ACCESS ; establish logical link with fal
E9 50 E9 0087 241 BLBC RO,ERRDLT ; branch on failure
FF73' 30 008A 242 BSBW NT$ERASE ; erase file at remote node
CF 50 E8 008D 243 BLBS RO,FILNAM ; fill in nam blk, if any
D0 11 0090 244 BRB CLEANUP ; otherwise report the error.
0092 245
0092 246 .END

```

```

$$PSECT_EP           = 00000000
$$RMSTEST            = 0000001A
$$RMS_PBUGCHK        = 00000010
$$RMS_TBUGCHK        = 00000008
$$RMS_UMODE          = 00000004
CLEANOP              = 00000062 R    01
DELETE                = 0000004F R    01
DEVSV_RND            = 0000001C
ERRBUG               = 0000007D R    01
ERRDLT               = 00000073 R    01
ERRIOP               = 0000006C R    01
ERRWLD               = 00000065 R    01
FIBSV_FINDFID        = 0000000B
FIBSW_DID             = 0000000A
FIBSW_DID_NUM        = 0000000A
FIBSW_DID_SEQ        = 0000000C
FIBSW_FID            = 00000004
FIBSW_NMCTL          = 00000014
FILNAM               = 0000005F R    01
FTLS_NODIDORFID      = FFFFFFFEB
FWASB_ESCFLG         = 0000000C
FWAST_FIBBUF         = 000001F4
FWASV_WILDCARD       = 00000018
IFBSL_PRIM_DEV       = 00000000
IFBSV_DAP            = 0000003E
IOSM_DELETE          = 00000100
IOS_DELETE           = 00000035
NAMSW_DID            = 0000002A
NAMSW_DID_SEQ        = 0000002C
NTSACCESS            ***** X    01
NTSERASE             ***** X    01
NTERASE              00000084 R    01
PIOA_TRACE           ***** X    01
RMSASSIGN            ***** X    01
RMSBUG               ***** X    01
RMSCLSCU             ***** X    01
RMSFCPFNC            ***** X    01
RMSFCP_P4_P2        ***** X    01
RMSFILENAM           ***** X    01
RMSFSETI             ***** X    01
RMSMAPERR            ***** X    01
RMSSETDID            ***** X    01
RMSXPFN              ***** X    01
RMSSERASE            = FFFFFFFE RG  01
RMSS_IOP              = 00018574
RMSS_MKD              = 0001C032
RMSS_WLD              = 00018744
TPT&L_ERASE         ***** X    01
    
```

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
RMSRMS	00000092 ( 146.)	01 ( 1.)	PIC USR	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE

RMSOERASE  
Psect synopsis

DELETE FILE OPERATION

L 13

16-SEP-1984 01:17:19 VAX/VMS Macro V04-00  
5-SEP-1984 16:24:53 [RMS.SRC]RMSOERASE.MAR;1

Page 8  
(4)

RM  
VO

SABSS 0000000 ( 0.) 02 ( 2.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.44
Command processing	107	00:00:00.69	00:00:06.10
Pass 1	377	00:00:13.50	00:00:29.19
Symbol table sort	0	00:00:02.17	00:00:03.10
Pass 2	58	00:00:02.18	00:00:04.66
Symbol table output	6	00:00:00.10	00:00:00.44
Psect synopsis output	2	00:00:00.03	00:00:00.10
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	581	00:00:18.76	00:00:44.14

The working set limit was 1350 pages.  
74388 bytes (146 pages) of virtual memory were used to buffer the intermediate code.  
There were 80 pages of symbol table space allocated to hold 1563 non-local and 2 local symbols.  
246 source lines were read in Pass 1, producing 13 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	12
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	18

1683 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

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

