

[illegible]

— 52

Syn

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS

NT
NT

NT

NT
NT

PI

100

10

10

10

10

11. *Journal of the American Medical Association*, 273, 1995, 1000-1001.

1

```

RRRRRRRR      MM      MM      000000      FFFFFFFFFF      SSSSSSSS      EEEEEEEEEEE      TTTTTTTTTT      IIIIIII
RRRRRRRR      MM      MM      000000      FFFFFFFFFF      SSSSSSSS      EEEEEEEEEEE      TTTTTTTTTT      IIIIIII
RR      RR      MMMM      MMMM      00      00      FF      SS      EE      TT      II
RR      RR      MMMM      MMMM      00      00      FF      SS      EE      TT      II
RR      RR      MM      MM      00      0000      FF      SS      EE      TT      II
RR      RR      MM      MM      00      0000      FF      SS      EE      TT      II
RRRRRRRR      MM      MM      00      00      0000      FFFFFFFF      SSSSSS      EEEEEEEEE      TT      II
RRRRRRRR      MM      MM      00      00      00      FFFFFFFF      SSSSSS      EEEEEEEEE      TT      II
RR      RR      MM      MM      0000      00      FF      SS      EE      TT      II
RR      RR      MM      MM      0000      00      FF      SS      EE      TT      II
RR      RR      MM      MM      00      00      FF      SS      EE      TT      II
RR      RR      MM      MM      00      00      FF      SS      EE      TT      II
RR      RR      MM      MM      000000      FF      SSSSSSSS      EEEEEEEEEEE      TT      IIIIIII
RR      RR      MM      MM      000000      FF      SSSSSSSS      EEEEEEEEEEE      TT      IIIIIII
    
```

```

LL      IIIIIII      SSSSSSSS
LL      IIIIIII      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      IIIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIIII      SSSSSSSS
    
```


(2) 67
(3) 95

DECLARATIONS
RMSFSETI - \$OPEN AND \$CREATE COMMON SETUP ROUTINE

```
0000 1          $BEGIN RMOFSETI,000,RM$RMS0,<OPEN AND CREATE COMMON SETUP>
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 ++
0000 29 Facility: RMS32
0000 30
0000 31 Abstract:
0000 32 routine to perform setup common to $open and $create.
0000 33
0000 34 Environment:
0000 35 star processor running starlet exec.
0000 36
0000 37 Author: L F Laverdure, creation date: 4-JAN-1977
0000 38
0000 39 Modified By:
0000 40
0000 41 V03-004 DGB0014 Donald G. Blair 02-Mar-1984
0000 42 Set FIB$B_AGENT_MODE field; used for access mode protected
0000 43 files.
0000 44
0000 45 V03-003 LJA0025 Laurie Anderson 11-Oct-1982
0000 46 No longer need to pass R0 to RM$GTSLT. Cleans up call slightly.
0000 47
0000 48 V03-002 LJA0017 Laurie Anderson 02-Sep-1982
0000 49 Check to see if this is a restart open or create operation.
0000 50
0000 51 V03-001 KBT0304 Keith B. Thompson 28-Aug-1982
0000 52 Reorganize psects
0000 53
0000 54 V02-014 CDS0002 C D Saether 4-Sep-1981
0000 55 Always set NO$ECLK initially for ifab.
0000 56
0000 57 V02-013 REFORMAT Keith B. Thompson 29-Jul-1980
```


0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :--
0000 65

V0012 CDS0001 C D Saether 11-MAR-1980
fix errnoslt error to call RET1PAG instead of RETPAG
V0011 TMH0001 T. Halvorsen 28-SEP-1979
add FSETI_ALT entry point for SETDDIR


```
0000 67      .SBTTL  DECLARATIONS
0000 68
0000 69  ::
0000 70  :: Include Files:
0000 71  ::
0000 72  ::
0000 73  ::
0000 74  :: Macros:
0000 75  ::
0000 76
0000 77      $IFBDEF
0000 78      $FABDEF
0000 79      $IMPDEF
0000 80      $PIODEF
0000 81      $PSLDEF
0000 82      $RMSDEF
0000 83
0000 84  ::
0000 85  :: Equated Symbols:
0000 86  ::
0000 87
00000020 0000 88      FOP=FAB$L_FOP*8          ; bit offset to fop
0000 89
0000 90  ::
0000 91  :: Own Storage:
0000 92  ::
0000 93
```



```
0000 95      .SBTTL RMSFSETI - $OPEN AND $CREATE COMMON SETUP ROUTINE
0000 96
0000 97      :++
0000 98      :
0000 99      : RMSFSETI -
0000 100     : RMSFSETI_ALT -
0000 101     :
0000 102     : this routine calls RMSFABCHK to check the basic parameters
0000 103     : for a fab function, allocates and inits an ifab, and
0000 104     : then dispatches to rmsfset (at its alternate entry point)
0000 105     : to finish the setup processing.
0000 106     :
0000 107     :
0000 108     : Calling sequence:
0000 109     :
0000 110     :     BSBW    RMSFSETI
0000 111     :
0000 112     : Input Parameters:
0000 113     :
0000 114     :     AP      parameter list addr
0000 115     :
0000 116     : Implicit Inputs:
0000 117     :
0000 118     :     the various fields of the fab.
0000 119     :
0000 120     : Output Parameters:
0000 121     :
0000 122     :     R11      impure area address
0000 123     :     R10      ifab address
0000 124     :     R9       ifab address
0000 125     :     R8       fab address
0000 126     :     R7       caller's mode
0000 127     :     R0 thru R5 destroyed
0000 128     :
0000 129     : Implicit Outputs:
0000 130     :
0000 131     :     various fields of the ifab are initialized as
0000 132     :     well as the ifi, sts, and stv fields of the fab.
0000 133     :
0000 134     : Completion Codes:
0000 135     :
0000 136     :     none. if an error is detected, returns to user
0000 137     :     (not caller) after cleaning up and returning a standard
0000 138     :     rms error code, in particular, dme, ifi, or any of
0000 139     :     the codes returned by RMSFABCHK.
0000 140     :
0000 141     : Side Effects:
0000 142     :
0000 143     :     none
0000 144     :
0000 145     :--
0000 146
```



```
FFFD' 30 0000 148 RMSFSETI::
0000 149 BSBW RMSFABCHK ; valid fab?
0003 150 ; returns only if o.k.
0003 151 ; NOTE:
0003 152 ; note: sets R11 to impure addr
0003 153 ; R9 to IFI value
0003 154 ; R8 to fab addr
03 13 0003 155 BEQL RMSFSETI_ALT
00A7 31 0005 156 BRW ERRIFI ; error if IFI non-zero
0008 157
0008 158 ;
0008 159 ; alternate entry point - fab is already checked
0008 160 ; R7 = caller's access mode
0008 161 ; R8 = fab address
0008 162 ; R11 = impure area address
0008 163 ;
0008 164
0008 165 RMSFSETI_ALT::
0008 166
0008 167 ;
0008 168 ; so far, so good - time to allocate an ifab
0008 169 ;
0008 170 ;!!!!
0008 171 ; \may need to add code to share a page.\
0008 172 ;!!!!
0008 173 ;
0008 174 ; check for process permanent file setup, and
0008 175 ; if so, switch the impure area pointer
0008 176 ;
0008 177
OF 68 32 E1 0008 178 BBC #FAB$V_PPF+FOP,(R8),10$ ; branch if image i/o file
03 57 91 000C 179 CMPB R7,#PS[$C_USER ; attempt from user mode?
03 12 000F 180 BNEQ 8$
00A2 31 0011 181 BRW ERRPRV ; caught you!
5B 00000000'9F DE 0014 182 8$: MOVAL @#PIO$GW_PIOIMPA,R11 ; no, so it's ok - switch impure pointer
FFE2' 30 001B 183 10$: BSBW RM$GET1PAG ; go get a free page
03 50 E8 001E 184 BLBS R0,12$
00A4 31 0021 185 BRW ERROR ; branch on failure
0024 186
0024 187 ;
0024 188 ; set up the page/file free space header at the start of the page
0024 189 ;
0024 190
83 51 53 D0 0024 191 12$: MOVL R3,R1 ; save address of page
83 08 A3 9E 0027 192 MOVAB 8(R3),(R3)+ ; set free space address (flink)
83 04 A3 9E 002B 193 MOVAB 4(R3),(R3)+ ; and blink
83 51 D0 002F 194 MOVL R1,(R3)+ ; set free space's f link
83 51 D0 0032 195 MOVL R1,(R3)+ ; and its b link
63 52 08 C3 0035 196 SUBL3 #8,R2,(R3) ; and its size (512 bytes)
52 2E 9A 0039 197 MOVZBL #IFB$C_BLN/4,R2 ; block size for ifab
FFC1' 30 003C 198 BSBW RM$GETBLK ; get ifab space (can't fail)
59 51 D0 003F 199 MOVL R1,R9 ; save ifab address
0042 200
0042 201 ;
0042 202 ; set ifab bid (R9 points to ifab), save user mode in ifab, store
0042 203 ; agent mode in ifab.
0042 204 ;
```



```

08 A9 0B 90 0042 205
0A A9 57 90 0042 206      MOVB    #IFB$C_BID,IFB$B_BID(R9)
04 EF 0046 207      MOVB    R7,IFB$B_MODE(R9)      ; store access mode of caller
50 4A A8 02 EF 004A 208      EXTZV    #FAB$V_FILE_MODE,-      ; r0 = user's request-mode field
57 50 91 004C 209      #FAB$S_FILE_MODE,FAB$B_ACMODES(R8),R0
03 1E 0050 210      CMPB    R0,R7      ; maximize request-mode (i.e. make it
50 57 90 0053 211      BGEQU    14$      ; less privileged) with caller's mode.
4F A9 50 90 0055 212      MOVB    R7,R0      ; place maximized mode in R0
0058 213 14$: MOVB    R0,IFB$B_AGENT_MODE(R9) ; store request mode in ifb
005C 214
005C 215 ;
005C 216 ; initialize bdb chain list header
005C 217 ;
005C 218 ;
40 A9 40 A9 DE 005C 219      MOVAL    IFB$L_BDB_FLNK(R9),IFB$L_BDB_FLNK(R9)
44 A9 40 A9 DE 0061 220      MOVAL    IFB$L_BDB_FLNK(R9),IFB$L_BDB_FLNK(R9)
0066 221      SSB    #IFB$V_NORECLK,(R9)      ; always init to no record locking.
006A 222 ;
006A 223 ; Check to see if this is a restart open or create operation by looking for
006A 224 ; a context XAB and whether the restart option bit is set.
006A 225 ;
5C DD 006A 226      PUSHL    AP      ; Save argument pointer
FF91' 30 006C 227      BSBW    RMS$OPNCREXAB      ; Look for Context XAB
5C 8E D0 006F 228      POPL    AP      ; Restore AP
48 50 E9 0072 229      BLBC    R0,ERRNOSLT      ; Error: return, after returning IFB pg
04 69 3B E1 0075 230      BBC     #IFB$V_RESTART,(R9),15$ ; If not restart operation, continue
56 02 A8 3C 0079 231      MOVZWL   FAB$W_IFI(R8),R6      ; Set up IFI value for get slot.
007D 232
007D 233 ;
007D 234 ; allocate and initialize a slot in the ifab table
007D 235 ;
007D 236 ;
55 18 AB D0 007D 237 15$: MOVL    IMP$L_IFABTBL(R11),R5      ; get table addr
FF7C' 30 0081 238      BSBW    RMS$GTSLT      ; and get a slot
36 50 E9 0084 239      BLBC    R0,ERRNOSLT      ; branch if none
02 A8 56 B0 0087 240      MOVW    R6,FAB$W_IFI(R8)      ; store ifi value
008B 241      ASSUME    IMP$W_RMSSTATUS EQ 0
008B 242      ASSUME    IMP$V_IIOS EQ 0
1E 6B E8 008B 243      BLBS     (R11),20$      ; branch if not pio segment
008E 244
008E 245 ;
008E 246 ; save rat in ifi for later input to connect
008E 247 ;
008E 248 ;
1E A8 F0 008E 249      INSV     FAB$B_RAT(R8),-
16 0091 250      #FAB$V_PPF_RAT+<FAB$W_IFI*8>,-
68 08 0092 251      #FAB$S_PPF_RAT,(R8)
14 68 33 E1 0094 252      BBC     #FAB$V_INP_FOP,(R8),20$ ; branch if not 'input'
0098 253      SSB     #IFB$V_PPF_INPUT,(R9)      ; set 'input' flag
009C 254
009C 255 ;
009C 256 ; initialize the end of data scan string to a single '$'
009C 257 ;
009C 258 ;
00000000'9F 2401 8F B0 009C 259      MOVW    #1+<^A/$/a8>,@#PIO$GT_ENDSTR
00000000'9F 02 8A 00A5 260      BICB2   #1@PIO$V_EOD,@#PIO$GW_STATUS; clear 'looking for eod' flag
FF51' 31 00AC 261 20$: BRW     RMSFSET_ALT      ; finish setup processing
```



```
00AF 263
00AF 264 :
00AF 265 : error handling
00AF 266 :
00AF 267 :
00AF 268 ERRIFI:
12 11 00AF 269 RMSERR IFI ; invalid ifi (must be zero)
00B4 270 BRB ERROR
00B6 271
00B6 272 :
00B6 273 : an ifab was allocated but no ifab table slot - return ifab block
00B6 274 :
00B6 275 :
00B6 276 : attempt to create a process-permanent file from user mode
00B6 277 :
00B6 278
00B6 279 ERRPRV:
0B 11 00B6 280 RMSERR PRV
00BB 281 BRB ERROR
00BD 282
00BD 283 ERRNOSLT:
54 50 DD 00BD 284 PUSHL R0 ; Save error code.
59 DO 00BF 285 MOVL R9,R4 ; ifab addr
FF3B' 30 00C2 286 BSBW RM$RET1PAG ; return the ifab page
50 8ED0 00C5 287 POPL R0 ; Restore the error code.
00C8 288
FF35' 31 00C8 289 ERROR: BRW RM$EX_NOSTR
00CB 290
00CB 291 .END
```


RMOFSETI
Symbol table

OPEN AND CREATE COMMON SETUP

H 13

16-SEP-1984 00:23:12
5-SEP-1984 16:21:50

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

Page 8
(6)

\$\$PSECT_EP	=	00000000		
\$\$RMSTEST	=	0000001A		
\$\$RMS_PBUGCHK	=	00000010		
\$\$RMS_TBUGCHK	=	00000008		
\$\$RMS_UMODE	=	00000004		
ERRIFT		000000AF	R	01
ERRNOSLT		000000BD	R	01
ERROR		000000C8	R	01
ERRPRV		000000B6	R	01
FAB\$B_ACMODES	=	0000004A		
FAB\$B_RAT	=	0000001E		
FAB\$B_FOP	=	00000004		
FAB\$S_FILE_MODE	=	00000002		
FAB\$S_PPF_RAT	=	00000008		
FAB\$V_FILE_MODE	=	00000004		
FAB\$V_INP	=	00000013		
FAB\$V_PPF	=	00000012		
FAB\$V_PPF_RAT	=	00000006		
FAB\$W_IFI	=	00000002		
FOP	=	00000020		
IFB\$B_AGENT_MODE	=	0000004F		
IFB\$B_BID	=	00000008		
IFB\$B_MODE	=	0000000A		
IFB\$C_BID	=	0000000B		
IFB\$C_BLN	=	000000B8		
IFB\$B_BDB_BLNK	=	00000044		
IFB\$B_BDB_FLNK	=	00000040		
IFB\$V_NORECLK	=	00000033		
IFB\$V_PPF_INPUT	=	0000002E		
IFB\$V_RESTART	=	0000003B		
IMPSL_IFABTBL	=	00000018		
IMPSV_ILOS	=	00000000		
IMPSW_RMSSTATUS	=	00000000		
PIOSGT_ENDSTR	*****		X	01
PIOSGW_PIOIMPA	*****		X	01
PIOSGW_STATUS	*****		X	01
PIOSV_EOD	=	00000001		
PSL\$C_USER	=	00000003		
RMSEX_NOSTR	*****		X	01
RMSFABCHK	*****		X	01
RMSFSETI	00000000		RG	01
RMSFSETI_ALT	00000008		RG	01
RMSFSETI_ALT	*****		X	01
RMSGET1PAG	*****		X	01
RMSGETBLK	*****		X	01
RMSGTSLT	*****		X	01
RMSOPNCREXAB	*****		X	01
RMSRET1PAG	*****		X	01
RMS\$IFI	=	00018564		
RMS\$PRV	=	0001829A		

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR
RM\$RMS0	000000CB (203.)	01 (1.)	PIC USR
\$ABS\$	00000000 (0.)	02 (2.)	NOPIC USR

	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE
	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:01.20
Command processing	122	00:00:00.64	00:00:04.10
Pass 1	247	00:00:06.42	00:00:20.02
Symbol table sort	0	00:00:00.70	00:00:01.18
Pass 2	62	00:00:01.30	00:00:03.40
Symbol table output	8	00:00:00.07	00:00:00.12
Psect synopsis output	2	00:00:00.03	00:00:00.13
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	474	00:00:09.25	00:00:30.15

The working set limit was 1350 pages.

35101 bytes (69 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 663 non-local and 8 local symbols.

291 source lines were read in Pass 1, producing 13 object records in Pass 2.

19 pages of virtual memory were used to define 18 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	8
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	14

761 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

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

0318 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

