


```

RRRRRRRR      MM      MM      SSSSSSSS      000000      RRRRRRRR      EEEEEEEEEEE      WW      WW      IIIIII      NN      NN
PRRRRRRR      MM      MM      SSSSSSSS      000000      RRRRRRRR      EEEEEEEEEEE      WW      WW      IIIIII      NN      NN
RR      RR      MMMM      MMMM      SS      00      00      RR      RR      EE      WW      WW      II      NN      NN
RR      RR      MMMM      MMMM      SS      00      00      RR      RR      EE      WW      WW      II      NN      NN
RR      RR      MM      MM      SS      00      0000      RR      RR      EE      WW      WW      II      NNNN      NN
RR      RR      MM      MM      SS      00      0000      RR      RR      EE      WW      WW      II      NNNN      NN
RRRRRRRR      MM      MM      SSSSSS      00      00      00      RRRRRRRR      EEEEEEEEEEE      WW      WW      II      NN      NN
RRRRRRRR      MM      MM      SSSSSS      00      00      00      RRRRRRRR      EEEEEEEEEEE      WW      WW      II      NN      NN
RR      RR      MM      MM      SS      0000      00      RR      RR      EE      WW      WW      II      NN      NN
RR      RR      MM      MM      SS      0000      00      RR      RR      EE      WW      WW      II      NN      NN
RR      RR      MM      MM      SS      00      00      RR      RR      EE      WWW      WWW      II      NN      NN
RR      RR      MM      MM      SSSSSSSS      000000      RR      RR      EEEEEEEEEEE      WW      WW      IIIIII      NN      NN
RR      RR      MM      MM      SSSSSSSS      000000      RR      RR      EEEEEEEEEEE      WW      WW      IIIIII      NN      NN

```

```

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

(2) 76
(3) 108

DECLARATIONS
RMS\$REWIND - \$REWIND FILE ROUTINE

```

0000 1          $BEGIN RMSOREWIN,000,RMSRMS,<FILE REWIND 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 routine is the highest level control
0000 32 :           routine to perform the $rewind function.
0000 33
0000 34 : Environment:
0000 35 :           star processor running starlet exec.
0000 36
0000 37 : Author: l f laverdure,           creation date: 24-OCT-1977
0000 38
0000 39 : Modified By:
0000 40
0000 41 :           V03-004 KPL0001           Peter Lieberwirth           20-Jun-1983
0000 42 :           Change some references to JNLFLG to JNLFLG2.
0000 43
0000 44 :           V03-003 JWH0182           Jeffrey W. Horn           10-Feb-1983
0000 45 :           If we are doing 'fake' record locking and we're in a
0000 46 :           recovery unit, then make sure we make call to RMSUNLOCKALL,
0000 47 :           regardless of the state of IFBSV_NORECLK.
0000 48
0000 49 :           V03-002 KBT0190           Keith B. Thompson           23-Aug-1982
0000 50 :           Reorganize psects and rename entry point to single '$'
0000 51
0000 52 :           V03-001 CDS0002           C Saether           17-Feb-1982
0000 53 :           Don't clear IRBSV_GBLBUFF on rewind.
0000 54
0000 55 :           V02-014 CDS0001           C Saether           03-Aug-1981
0000 56 :           Remove SQO check around PPF check.
0000 57

```

0000	58	:	V02-013	KPL0001	P Lieberwirth	39-Jul-1981	
0000	59	:			Change entry point to \$\$ form so we can get here via		
0000	60	:			RMSOBRNCH.		
0000	61	:					
0000	62	:	V02-012	KRM0002	K R Malik	09-DEC-1980	
0000	63	:			modify BKPMASK to prevent clearing IRBSV_DAP_CON.		
0000	64	:					
0000	65	:	V02-011	REFORMAT	P S Knibbe	25-Jul-1980	
0000	66	:					
0000	67	:	V010	JAK0020	J A Krycka	05-SEP-1979	09:00
0000	68	:			release 2.0 work.		
0000	69	:					
0000	70	:	V008	RAN0002	R A Newell	06-SEP-1978	08:53
0000	71	:			rms32 isam modification.		
0000	72	:					
0000	73	---					
0000	74	:					

```
0000 76      .SBTTL  DECLARATIONS
0000 77
0000 78      :
0000 79      : Include Files:
0000 80      :
0000 81      :
0000 82      :
0000 83      : Macros:
0000 84      :
0000 85      :
0000 86      $DEVDEF
0000 87      $IFBDEF
0000 88      $IRBDEF
0000 89      $RMSDEF
0000 90
0000 91      :
0000 92      : Equated Symbols:
0000 93      :
0000 94      :
00000020 0000 95      BKP=IRB$L_BKPBITS*8 ; bit offset to bookkeeping bits
0000 96      BKPmask=^C<<1@<IRB$V_BUSY-BKP>>!--
0000 97      <1@<IRB$V_ASYNC-BKP>>!--
0000 98      <1@<IRB$V_ASYNCWAIT-BKP>>!--
0000 99      <1@<IRB$V_RAHWBH-BKP>>!--
0000 100     <1@<IRB$V_GBLBUFF-BKP>>!--
EFBFFBE6 0000 101     <1@<IRB$V_DAP_CONN-BKP>>> ; mask to reset bits
0000 102
0000 103     :
0000 104     : Own Storage:
0000 105     :
0000 106     :
```

```

0000 108      .SBTTL RMS$REWIND - $REWIND FILE ROUTINE
0000 109
0000 110      :++
0000 111      : RMS$REWIND - Rewind file
0000 112      :
0000 113      : this routine performs common rab function setup followed by an
0000 114      : implicit $free and $flush and then by dispatch to organization-dependent
0000 115      : $rewind code
0000 116      :
0000 117      : Calling sequence:
0000 118      :
0000 119      :     entered from exec as a result of user's calling sys$rewind
0000 120      :     (e.g., by using the $rewind macro)
0000 121      :
0000 122      : Input Parameters:
0000 123      :
0000 124      :     ap      user's argument list addr
0000 125      :
0000 126      : Implicit Inputs:
0000 127      :
0000 128      :     the contents of the rab and related irab and ifab.
0000 129      :
0000 130      : Output Parameters:
0000 131      :
0000 132      :     r1      destroyed
0000 133      :     r0      status code
0000 134      :
0000 135      : Implicit Outputs:
0000 136      :
0000 137      :     all locked records for the stream are freed, and the stream is
0000 138      :     repositioned at the start of the file.  the file attributes are also
0000 139      :     rewritten.
0000 140      :     for the sequential org., all bdb's and buffers are released.
0000 141      :     rab$l_sts and rab$l_stv reflect the status of the rewind.
0000 142      :
0000 143      :     a completion ast is queued if specified in the user arglist.
0000 144      :
0000 145      : Completion Codes:
0000 146      :
0000 147      :     standard rms (see functional spec for list).
0000 148      :
0000 149      : Side Effects:
0000 150      :
0000 151      :     none
0000 152      :
0000 153      :--
0000 154

```

```

0000 156      $ENTRY RMS$REWIND
0000 157      $TSTPT REWIND
0006 158      $RABSET ; do common setup with no specific checks
000A 159
000A 160
000A 161      ; returns to user on error
000A 162      ;
000A 163
63 6A 3E E0 000A 164      BBS #IFBSV_DAP,(R10),NTRWND ; branch if network operation
        22 E0 000E 165      BBS #IRBSV_PPF_IMAGE,-
6B 69 0E E1 0010 166      (R9),ERRIOP ; branch if indirect ppf
        6A E1 0012 167      BBC #DEV$V_FOD,-
        67 0014 168      IFBSL PRIM_DEV(R10),-
06 00A2 CA E1 0015 169      ERRIOP ; or if unit rec device
        02 E1 0016 170      BBC #IFBSV_RUP,-
04 00A2 C9 E0 0018 171      IFBSB JNLFLG2(R10),10$ ; branch if not in RU
03 6A 33 E0 001C 172      BBS #IFBSV_RU_RLK,-
        FFD7' 30 001E 173      IFBSB JNLFLG2(R9),20$ ; branch if 'fake' record locking
        0022 174 10$: BBS #IFBSV_NORECLK,(R10),30$ ; omit record unlock if no locking
        0026 175 20$: BSBW RMSUNLOCKALL ; free any locked records
        0029 176      ; (ignore possible error)
        0029 177 30$:
        0029 178
        0029 179      ;
        0029 180      ; dispatch to org-dependent code
        0029 181      ;
        0029 182
        0029 183      CASE TYPE=B,-
        0029 184      SRC=IFBSB_ORGCASE(R10),-; and dispatch
        0029 185      DISPLIST=ZRWSEQ,RWREL,RWIDX>
        0034 186
        0034 187      ;
        0034 188      ;*****
        0034 189      ;
        0034 190      ; indexed specific rewind
        0034 191      ;
        0034 192      ;*****
        0034 193      ;
        0034 194
00000000'EF 16 0034 195 RWIDX: JSB RMS$REWIND3 ; do indexed specific rewind
        27 11 003A 196      BRB DONE
        003C 197
        003C 198      ;*****
        003C 199      ;
        003C 200      ; sequential file org rewind
        003C 201      ;
        003C 202      ;*****
        003C 203
        FFC1' 30 003C 204 RWSEQ: BSBW RMS$FLUSH ; write and release any bdb
        003F 205      ; rewriting file attributes
        2C 50 E9 003F 206      BLBC RO,EXIT ; get out on error
        62 A9 B4 0042 207      CLRW IRBSW_CSIZ(R9) ; reset to no current record
        05 E1 0045 208      BBC #DEV$V_SQD,-
        6A 0047 209      IFBSL PRIM_DEV(R10),-
        0D 0048 210      RWREL ; branch if not magtape
        0049 211
        0049 212      ;

```



```

0049 213 ; magtape support
0049 214 ;
0049 215 ;
74 AA 01 CE 0049 216 MNEGL #1,IFB$ EBK(R10) ; set eof block to max value
SC AA B4 004D 217 CLRW IFB$W FFB(R10) ; reset eof offset
FFAD' 30 0050 218 BSBW RMSREWIND_MT ; do rewind
18 50 E9 0053 219 BLBC RO,EXIT ; get out on error
0056 220 ; else fall thru
0056 221 ;
0056 222 :*****
0056 223 ;
0056 224 ; relative file org rewind
0056 225 ;
0056 226 :*****
0056 227 ;
0056 228 ASSUME IRB$ _NRP_OFF EQ IRB$ _NRP_VBN+4
0056 229 ;
40 A9 7C 0056 230 RWREL: CLRQ IRB$ _NRP_VBN(R9) ; reset nrp
18 E0 0059 231 BBS #DEV$V FOR,-
03 6A 005B 232 IFB$ _PRIM_DEV(R10),10$ ; branch if non-file-struct.
40 A9 96 005D 233 INCB IRB$ _NRP_VBN(R9) ; nrp = vbn 1
0060 234 ;
0060 235 10$: ASSUME IRB$ _RP_OFF EQ IRB$ _RP_VBN+4
0060 236 ;
48 A9 7C 0060 237 CLRQ IRB$ _RP_VBN(R9)
0063 238 ;
0063 239 ;
0063 240 ; clear all irab bookkeeping bits except busy, async, asyncwait, rahwbh
0063 241 ; gblbuff & dap_con.
0063 242 ;
0063 243 ;
04 A9 EFBFFBE6 8F CA 0063 244 DONE: BICL #BKPMASK,IRB$ _BKPBITS(R9)
FF8F' 31 006B 245 RMSSUC
006E 246 EXIT: BRW RMSEX RMS ; return to caller
0071 247 ;
0071 248 ;++
0071 249 ;
0071 250 ; perform network rewind function
0071 251 ;
0071 252 ;--
0071 253 ;
0071 254 NTRWND:
FF8C' 30 0071 255 BSBW NTSREWIND ; rewind the file
F7 50 E9 0074 256 BLBC RO,EXIT ; branch on error
40 A9 01 D0 0077 257 MOVL #1,IRB$ _NRP_VBN(R9) ; reset next block pointer
E6 11 007B 258 BRB DONE ; rejoin mainline
007D 259 ;
007D 260 ;
007D 261 ; handle error
007D 262 ;
007D 263 ;
007D 264 ERRIOP:
007D 265 RMSERR IOP ; can't rewind unit rec dev. or
0082 266 ; process perm. file if indirectly
0082 267 ; connected
EA 11 0082 268 BRB EXIT
0084 269 ;

```

RMSOREWIN
V04-000

FILE REWIND OPERATION
RMS\$REWIND - \$REWIND FILE ROUTINE

B 13

16-SEP-1984 01:28:42 VAX/VMS Macro V04-00
5-SEP-1984 16:25:23 [RMS.SRC]RMSOREWIN.MAR;1

Page 7
(4)

RMS
V04

0084 270 .END

\$\$PSECT EP	=	00000000		
\$\$RMSTEST	=	0000001A		
\$\$RMS_PBUGCHK	=	00000010		
\$\$RMS_TBUGCHK	=	00000008		
\$\$RMS_UMODE	=	00000004		
BKP	=	00000020		
BKPMASK	=	EFBFFBE6		
DEVSV_FOD	=	0000000E		
DEVSV_FOR	=	00000018		
DEVSV_SQD	=	00000005		
DONE		00000063	R	01
ERRIOP		0000007D	R	01
EXIT		0000006E	R	01
IFBSB_JNLFLG2	=	000000A2		
IFBSB_ORGCASE	=	00000023		
IFBSL_EBK	=	00000074		
IFBSL_PRIM_DEV	=	00000000		
IFBSV_DAP	=	0000003E		
IFBSV_NORECLK	=	0C000033		
IFBSV_RUP	=	00000002		
IFBSV_RU_RLK	=	00000003		
IFBSW_FFB	=	0000005C		
IRBSL_BKPBITS	=	00000004		
IRBSL_NRP_OFF	=	00000044		
IRBSL_NRP_VBN	=	00000040		
IRBSL_RP_OFF	=	0000004C		
IRBSL_RP_VBN	=	00000048		
IRBSV_ASYNC	=	00000023		
IRBSV_ASYNCWAIT	=	00000024		
IRBSV_BUSY	=	00000020		
IRBSV_DAP_CONN	=	0000003C		
IRBSV_GBLBUFF	=	00000036		
IRBSV_PPF_IMAGE	=	00000022		
IRBSV_RAHOBH	=	0000002A		
IRBSW_CSIZ	=	00000062		
NT\$REWIND		*****	X	01
NTRWIND		00000071	R	01
PIO\$A TRACE		*****	X	01
RMSEX RMS		*****	X	01
RMSFLUSH		*****	X	01
RMSREWIND3		*****	X	01
RMSREWIND_MT		*****	X	01
RMSRSET		*****	X	01
RMSUNLOCKALL		*****	X	01
RMS\$REWIND	=	FFFFFFFFE	RG	01
RMS\$IOP	=	00018574		
RWIDR		00000034	R	01
RWREL		00000056	R	01
RWSEQ		0000003C	R	01
TPT\$REWIND		*****	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	00000084 (132.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
SABSS	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	29	00:00:00.09	00:00:01.07
Command processing	110	00:00:00.79	00:00:04.22
Pass 1	245	00:00:06.52	00:00:15.80
Symbol table sort	0	00:00:00.79	00:00:00.92
Pass 2	59	00:00:01.34	00:00:02.60
Symbol table output	8	00:00:00.08	00:00:00.15
Psect synopsis output	2	00:00:00.01	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	455	00:00:09.62	00:00:24.80

The working set limit was 1350 pages.
34665 bytes (68 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 678 non-local and 6 local symbols.
270 source lines were read in Pass 1, producing 13 object records in Pass 2.
21 pages of virtual memory were used to define 20 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	10
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	16

799 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

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

0330 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

