


```

RRRRRRRR      MM      MM      333333      PPPPPPPP      000000      SSSSSSSS      RRRRRRRR      FFFFFFFFFF      AAAAAA
RRRRRRRR      MM      MM      333333      PPPPPPPP      000000      SSSSSSSS      RRRRRRRR      FFFFFFFFFF      AAAAAA
RR      RR      MMMM      MMMM      33      33      PP      PP      00      00      SS      RRR      RR      FF      AA      AA
RR      RR      MMMM      MMMM      33      33      PP      PP      00      00      SS      RRR      RR      FF      AA      AA
RR      RR      MM      MM      33      33      PP      PP      00      00      SS      RRR      RR      FF      AA      AA
RR      RR      MM      MM      33      33      PP      PP      00      00      SS      RRR      RR      FF      AA      AA
RRRRRRRR      MM      MM      33      33      PPPPPPPP      00      00      SSSSSS      RRRRRRRR      FFFFFFFF      AA      AA
RRRRRRRR      MM      MM      33      33      PPPPPPPP      00      00      SSSSSS      RRRRRRRR      FFFFFFFF      AA      AA
RR      RR      MM      MM      33      33      PP      00      00      SS      RR      RR      FF      AAAAAAAAAA
RR      RR      MM      MM      33      33      PP      00      00      SS      RR      RR      FF      AAAAAAAAAA
RR      RR      MM      MM      33      33      PP      00      00      SS      RR      RR      FF      AA      AA
RR      RR      MM      MM      33      33      PP      00      00      SS      RR      RR      FF      AA      AA
RR      RR      MM      MM      333333      PP      000000      SSSSSSSS      RR      RR      FF      AA      AA
RR      RR      MM      MM      333333      PP      000000      SSSSSSSS      RR      RR      FF      AA      AA

```

```

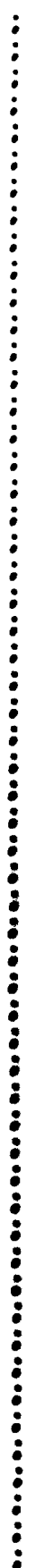
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

```

....
....
....
....

```



```

1 0001 0 MODULE RM3POSRFA (LANGUAGE (BLISS32) ,
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1
31 0031 1 FACILITY: RMS32 index sequential file organization
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1 This module positions to a record by RFA.
35 0035 1
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1
39 0039 1 VAX/VMS Operating System
40 0040 1
41 0041 1 --
42 0042 1
43 0043 1
44 0044 1 AUTHOR: Todd M. Katz CREATION DATE: 14-Jan-83
45 0045 1
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 V03-003 TSK0001 Tamar Krichevsky 15-Jun-1983
50 0050 1 Change addressing mode for RMSRU_RECLAIM to log relative.
51 0051 1
52 0052 1 V03-002 TMK0001 Todd M. Katz 11-Mar-1983
53 0053 1 If it possible that some reclamation will be required (the
54 0054 1 file is write accessed and RU Jouranallable), then make sure
55 0055 1 the primary data bucket is accessed exclusively.
56 0056 1
57 0057 1 V03-001 MCN0001 Maria del C. Nasr 24-Feb-1983

```

```

: 58      0058 1  | Reorganize Linkages
: 59      0059 1  |
: 60      0060 1  |
: 61      0061 1  | *****
: 62      0062 1  |
: 63      0063 1  | LIBRARY 'RMSLIB:RMS';
: 64      0064 1  |
: 65      0065 1  | REQUIRE 'RMSSRC:RMSIDXDEF';
: 66      0130 1  |
: 67      0131 1  | ! Define default PSECTS for code.
: 68      0132 1  |
: 69      0133 1  | PSECT
: 70      0134 1  |     CODE = RMSRMS3(PSECT_ATTR),
: 71      0135 1  |     PLIT = RMSRMS3(PSECT_ATTR);
: 72      0136 1  |
: 73      0137 1  | ! Linkages.
: 74      0138 1  |
: 75      0139 1  | LINKAGE
: 76      0140 1  |     L_RABREG_67,
: 77      0141 1  |     L_PRESERVE1;
: 78      0142 1  |
: 79      0143 1  | ! External Routines.
: 80      0144 1  |
: 81      0145 1  | EXTERNAL ROUTINE
: 82      0146 1  |     RMSFIND_BY_RRV      : RLSRABREG_67,
: 83      0147 1  |     RMSRLSBRT           : RLSPRESERVE1,
: 84      0148 1  |     RMSRU_RECLAIM       : RLSRABREG_67 ADDRESSING_MODE( LONG_RELATIVE );
: 85      0149 1  |

```

```

: 87 0150 1 GLOBAL ROUTINE RMSPOS_RFA : RL$RABREG_67 =
: 88 0151 1
: 89 0152 1 ++
: 90 0153 1
: 91 0154 1 FUNCTIONAL DESCRIPTION:
: 92 0155 1
: 93 0156 1 This routine positions to a primary data record by its RFA.
: 94 0157 1
: 95 0158 1 If RMS finds that the record with the appropriate RFA is marked
: 96 0159 1 RU_DELETE and the Recovery Unit in which the record was deleted is
: 97 0160 1 still active, then RMS returns positioned to this record and lets a
: 98 0161 1 higher level routine decide what to do. If the Recovery Unit in which
: 99 0162 1 the record was deleted has successfully terminated, then RMS will
100 0163 1 return an error of DEL after deleting this RU_DELETED record (if it has
101 0164 1 write access to the file), and releasing the primary data bucket.
102 0165 1
103 0166 1 If RMS encounters a record that is marked RU_UPDATE and is in a special
104 0167 1 format then RMS will return positioned to this record after
105 0168 1 reformatting it. The reformatting is done if RMS has write access to
106 0169 1 the file, and the Recovery Unit in which it was updated has
107 0170 1 successfully terminated.
108 0171 1
109 0172 1 CALLING SEQUENCE:
110 0173 1
111 0174 1 RMSPOS_RFA()
112 0175 1
113 0176 1 INPUT PARAMETERS:
114 0177 1 NONE
115 0178 1
116 0179 1 IMPLICIT INPUTS:
117 0180 1
118 0181 1 IFAB - address of IFAB
119 0182 1 IFBSV_RU - if set, the file is Recovery Unit Journallable
120 0183 1 IFBSV_WRTACC - if set, file is open for write access
121 0184 1
122 0185 1 RAB - address of RAB
123 0186 1 RABSL_RFA0 - RFA VBN of primary data record to be retrieved
124 0187 1 RABSW_RFA4 - RFA IF of primary data record to be retrieved
125 0188 1
126 0189 1 OUTPUT PARAMETERS:
127 0190 1 NONE
128 0191 1
129 0192 1 IMPLICIT OUTPUTS:
130 0193 1
131 0194 1 IRAB - address of IRAB
132 0195 1 IRBSL_CURBDB - address of BDB for primary data bucket
133 0196 1 IRBSW_RFA_ID - ID of current primary data record
134 0197 1 IRBSL_RFA_VBN - VBN of current primary data bucket
135 0198 1
136 0199 1 REC_ADDR - address of primary data record
137 0200 1
138 0201 1 ROUTINE VALUE:
139 0202 1
140 0203 1 SUC - the primary data record with this RFA has been found.
141 0204 1 DEL - the primary data record with this RFA has been deleted.
142 0205 1 RNF - the primary data record with this RFA has not been found.
143 0206 1

```

```

: 144 0207 1 : SIDE EFFECTS:
: 145 0208 1
: 146 0209 1 : On success, REC_ADDR points to the non-deleted primary data record
: 147 0210 1 : and the BDB of the primary data bucket maybe found in IRB$L_CURBDB.
: 148 0211 1 : On failures, all accessed buckets are released.
: 149 0212 1 : If the record is marked RU_DELETED, it might have been deleted.
: 150 0213 1 : If the record is marked RU_UPDATED, it might have been reformatted.
: 151 0214 1
: 152 0215 1
: 153 0216 1
: 154 0217 2 BEGIN
: 155 0218 2
: 156 0219 2 BUILTIN
: 157 0220 2 AP;
: 158 0221 2
: 159 0222 2 EXTERNAL REGISTER
: 160 0223 2 COMMON RAB_STR,
: 161 0224 2 R_IDX_DFN_STR,
: 162 0225 2 R_REC_ADDR_STR;
: 163 0226 2
: 164 0227 2 LOCAL
: 165 0228 2 STATUS;
: 166 0229 2
: 167 0230 2 : If it possible that some reclamation will be required (the file is write
: 168 0231 2 : accessed and RU Jouranallable), then make sure the primary data bucket is
: 169 0232 2 : accessed exclusively.
: 170 0233 2
: 171 0234 2 IF .IFAB[IFBSV_WRTACC]
: 172 0235 2 AND
: 173 0236 2 .IFAB[IFBSV_RU]
: 174 0237 2 THEN
: 175 0238 2 IRAB[IRBSB_CACHEFLGS] = CSHM_LOCK;
: 176 0239 2
: 177 0240 2 : RMS is successful at positioning to the primary data record by its RFA.
: 178 0241 2
: 179 0242 2 IF (STATUS = RMS$FIND_BY_RRV (.RAB[RAB$L_RFA0], .RAB[RAB$W_RFA4], 0))
: 180 0243 2 THEN
: 181 0244 2 BEGIN
: 182 0245 2
: 183 0246 2 : If RMS finds that the record with this RFA has been deleted within a
: 184 0247 2 : Recovery Unit, then it subjects this record to further processing
: 185 0248 2 : before deciding whether to return this record, or to return an error
: 186 0249 2 : of record deleted.
: 187 0250 2
: 188 0251 2 IF .REC_ADDR[IRC$V_RU_DELETE]
: 189 0252 2 THEN
: 190 0253 2 BEGIN
: 191 0254 2
: 192 0255 2 LOCAL
: 193 0256 2 TEMP_STATUS;
: 194 0257 2
: 195 0258 2 : If the Recovery Unit in which the record was deleted has
: 196 0259 2 : not completed or if the file has not been opened for write
: 197 0260 2 : access then no space can be reclaimed. Return an error of
: 198 0261 2 : RMS$DEL unless the record itself could not be locked in which
: 199 0262 2 : case return the record so that caller may decide what to do with
: 200 0263 2 : it.

```

```

201 0264 4
202 0265 5
203 0266 4
204 0267 5
205 0268 5
206 0269 6
207 0270 5
208 0271 5
209 0272 5
210 0273 5
211 0274 5
212 0275 5
213 0276 5
214 0277 5
215 0278 5
216 0279 4
217 0280 4
218 0281 4
219 0282 4
220 0283 4
221 0284 4
222 0285 4
223 0286 4
224 0287 4
225 0288 3
226 0289 3
227 0290 3
228 0291 3
229 0292 3
230 0293 3
231 0294 3
232 0295 3
233 0296 3
234 0297 3
235 0298 4
236 0299 3
237 0300 4
238 0301 4
239 0302 4
240 0303 4
241 0304 4
242 0305 5
243 0306 4
244 0307 4
245 0308 4
246 0309 4
247 0310 4
248 0311 3
249 0312 4
250 0313 4
251 0314 4
252 0315 3
253 0316 3
254 0317 3
255 0318 3
256 0319 3
257 0320 3

```

```

!
IF NOT (TEMP_STATUS = RMSRU_RECLAIM())
THEN
  BEGIN
    IF .TEMP_STATUS<0,16> NEQU RMSERR(RLK)
    THEN
      STATUS = RMSERR(DEL);
    END
    ! If the Recovery Unit in which the record was deleted has
    ! successfully terminated, then RMS will be able to delete the
    ! record at this time. In this case, change the return status
    ! to RMS$DEL.
  ELSE
    STATUS = RMSERR(DEL);
  END
  ! If the primary data record RMS has positioned was updated within a
  ! Recovery Unit, then re-format the record at this time provided
  ! the file is opened for write access, and the Recovery Unit has
  ! terminated successfully.
ELSE
  IF .REC_ADDR[IR($V_RU_UPDATE)]
  AND
  .IFAB[IFBSV_WRTACC]
  THEN
    RMSRU_RECLAIM();
    ! If the record originally positioned to has been found to be deleted,
    ! then release the primary data bucket in which it was found.
  IF .STATUS<0,16> EQLU RMSERR(DEL)
  THEN
    BEGIN
      GLOBAL REGISTER
      R_BDB_STR;
      RELEASE (IRAB[IRB$$_CURBDB])
    END
    ! Otherwise, setup to update the NRP context with the current address
    ! in the file of this primary data record.
  ELSE
    BEGIN
      IRAB[IRB$$_RFA_VBN] = .BBLOCK[IRAB[IRB$$_CURBDB], BDB$$_VBN];
      IRAB[IRB$$_RFA_ID] = IRC$_ID(REC_ADDR);
    END;
  END
  ! RMS was not successful at positioning to the primary data record by its
  ! RFA. Setup to return the reason why.
!

```


	50	01	A6	9A	00076		MOVZBL	1(REC_ADDR), R0
			04	11	0007A		BRB	7\$
	50	01	A6	3C	0007C	6\$:	MOVZWL	1(REC_ADDR), R0
74	A9		50	B0	00080	7\$:	MOVW	R0, 1T6(IRAB)
			0F	11	00084		BRB	9\$
		20	A9	D4	00086	8\$:	CLRL	32(IRAB)
827A	8F		52	B1	00089		CMPW	STATUS, #33402
			05	12	0008E		BNEQ	9\$
	52	82B2	8F	3C	00090		MOVZWL	#33458, STATUS
	50		52	D0	00095	9\$:	MOVL	STATUS, R0
			14	BA	00098		POPR	#*M<R2,R4>
			05	0009A			RSB	

.....
0242
0323
0325
.....
0327
0332
0333
.....

: Routine Size: 155 bytes, Routine Base: RMSRMS3 + 0000

```

: 271      0334 1
: 272      0335 1 END
: 273      0336 1
: 274      0337 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
RMSRMS3	155	NOVEC,NOWRT, RD, EXE,NOSHR, GBL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[RMS.OBJ]RMS.L32;1	3109	43	1	154	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RM3POSRFA/OBJ=OBJ\$:RM3POSRFA MSRC\$:RM3POSRFA/UPDATE=(ENH\$:RM3POSRFA)

```

: Size:      155 code + 0 data bytes
: Run Time:  00:05.2
: Elapsed Time: 00:16.5
: Lines/CPU Min: 3866
: Lexemes/CPU-Min: 14627

```

RM3POSFPA
V04-000

M 13
16-Sep-1984 01:55:44 VAX-11 Bliss-32 V4.0-742

Page 8

RM3
V04

: Memory Used: 72 pages
: Compilation Complete

.....

