C 13

```
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
```

```
**FILE**ID**RM3PUTERR                          F  4                                                  RM3
                                                                                                     V04
```

```
RRRRRRRR    MM      MM    333333    PPPPPPPP    UU      UU  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR    RRRRRRRR
RRRRRRRR    MM      MM    333333    PPPPPPPP    UU      UU  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR    RRRRRRRR
RR      RR  MMMM  MMMM  33      33  PP      PP  UU      UU      TT      EE          RR      RR  RR      RR
RR      RR  MMMM  MMMM  33      33  PP      PP  UU      UU      TT      EE          RR      RR  RR      RR
RR      RR  MM  MM  MM          33  PP      PP  UU      UU      TT      EE          RR      RR  RR      RR
RR      RR  MM  MM  MM          33  PP      PP  UU      UU      TT      EE          RR      RR  RR      RR
RRRRRRRR    MM      MM        33    PPPPPPPP    UU      UU      TT      EEEEEEEE    RRRRRRRR    RRRRRRRR
RRRRRRRR    MM      MM        33    PPPPPPPP    UU      UU      TT      EEEEEEEE    RRRRRRRR    RRRRRRRR
RR  RR      MM      MM          33  PP          UU      UU      TT      EE          RR  RR      RR  RR
RR  RR      MM      MM          33  PP          UU      UU      TT      EE          RR  RR      RR  RR
RR    RR    MM      MM  33      33  PP          UU      UU      TT      EE          RR    RR    RR    RR         ....
RR    RR    MM      MM  33      33  PP          UU      UU      TT      EE          RR    RR    RR    RR         ....
RR      RR  MM      MM    333333    PP          UUUUUUUUUU      TT      EEEEEEEEEE  RR      RR  RR      RR       ....
RR      RR  MM      MM    333333    PP          UUUUUUUUUU      TT      EEEEEEEEEE  RR      RR  RR      RR       ....


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
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
MODULE RM3PUTERR (LANGUAG_ (BLISS32) ,
                IDENT = 'V04-000'
                ) =
BEGIN
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
!*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!*                                                                          *
!****************************************************************************

!++
!
! FACILITY:     RMS32 INDEX SEQUENTIAL FILE ORGANIZATION
!
! ABSTRACT:
!                $PUT and $UPDATE specific error cleanup routines
!
!
! ENVIRONMENT:
!
!               VAX/VMS OPERATING SYSTEM
!
!--

!
! AUTHOR:        Todd M. Katz        CREATION DATE:    17-Jul-82
!
!
! Modified by:
!
!       V03-014 MCN0003         Maria del C. Nasr       04-Apr-1983
!               Change linkage of RM$NULLKEY to RL$JSB.
!
!       V03-013 MCN0002         Maria del C. Nasr       15-Mar-1983
!               More linkages reorganization
!
!       V03-012 MCN0001         Maria del C. Nasr       28-Feb-1983
!               Reorganize linkages
```

```
1 !
1 !        V03-011 TMK0004          Todd M. Katz             15-Feb-1983
1 !                If the deletion of the RRV fails in RM$PUTUPD_ERROR, do not
1 !                delete the primary data record completely, and then create a
1 !                pointerless RRV at the end of the bucket. This is what is
1 !                done currently. Just delete the primary data record by calling
1 !                RM$DELETE_UDR so that it is deleted according to the normal
1 !                rules for primary data record deletion. This does leave the
1 !                possibility of having a RRV point to nothing (if the RRV
1 !                deletion fails and the primary data record is completely
1 !                deleted), but such a occurance would also exist if the
1 !                pointerless RRV were deleted as part of a CONVERT/RECLAIM.
1 !
1 !        V03-010 TMK0003          Todd M. Katz             05-Jan-1983
1 !                The routine RM$PUTUPD_ERROR was saving, zeroing, and then
1 !                restoring the current NRP key of reference while all newly
1 !                added SIDRs were being deleted. This is no longer necessary.
1 !
1 !        V03-009 TMK0002          Todd M. Katz             19-Sep-1982
1 !                Add support for prologue 3 SIDRs. This involves setting AP to
1 !                3 instead of to 1 each time RM$RECORD_KEY is called to extract
1 !                the key of the SIDR which is to be located and deleted.
1 !
1 !        V03-008 KBT0229          Keith B. Thompson        24-Aug-1982
1 !                Reorganize psects
1 !
1 !        V03-007 TMK0001          Todd M. Katz             17-Jul-1982
1 !                Completely revised the routines in this module because of
1 !                changes in the routines they interface to.
1 !
1 !*****
1
1 LIBRARY 'RMSLIB:RMS';
1
1 REQUIRE 'RMSSRC:RMSIDXDEF';
1
1 ! Define default PSECTS for code
1 !
1 PSECT
1     CODE = RM$RMS3(PSECT_ATTR),
1     PLIT = RM$RMS3(PSECT_ATTR);
1
1 ! Linkages
1 !
1 LINKAGE
1     L_ERROR_LINK1,
1     L_ERROR_LINK2,
1     L_JSB,
1     L_LINK_7_10_11,
1     L_RABREG_4567,
1     L_RABREG_67,
1     L_RABREG_7,
1     L_PRESERVE1;
1
1 ! External Routines
1 !
```

```
;  115          0179  1 EXTERNAL ROUTINE
;  116          0180  1     RM$DELETE_RRV        : RL$RABREG_4567,
;  117          0181  1     RM$DELETE_SIDR       : RL$RABREG_7,
;  118          0182  1     RM$DELETE_UDR        : RL$RABREG_4567,
;  119          0183  1     RM$FIND_BY_RRV       : RL$RABREG_67,
;  120          0184  1     RM$GET_NEXT_KEY      : RL$LINK_7_10_11,
;  121          0185  1     RM$KEY_DESC          : RL$RABREG_7,
;  122          0186  1     RM$NOREAD_LONG       : RL$JSB,
;  123          0187  1     RM$NULLKEY           : RL$JSB,
;  124          0188  1     RM$RECORD_KEY        : RL$PRESERVE1,
;  125          0189  1     RM$RLSBKT            : RL$PRESERVE1;
```

```
1 %SBTTL  'RM$PUTUPD_ERROR'
1 GLOBAL ROUTINE RM$PUTUPD_ERROR : RL$ERROR_LINK2 NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine's responsibility is to delete SIDR entires and the user
1 !       data record on $PUT/$UPDATE errors.
1 !
1 !       If this routine is called with the index descriptor for the primary key
1 !       of reference then all SIDR entries are deleted, otherwise, the deletion
1 !       of SIDR entries begins with 1 less than the current index descriptor.
1 !
1 !       The user data record and any RRV associated with it are deleted only if
1 !       the error occurred on a $PUT. The user data record will not be deleted
1 !       if the error occurred on any type of a $UPDATE (a regular $UPDATE or a
1 !       $PUT converted into a $UPDATE).
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       RM$PUTUPD_ERROR()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       IDX_DFN                 - address of index decriptor
1 !           IDX$B_DESC_NO       - descriptor number (index into update buffer)
1 !           IDX$B_KEYREF        - key of reference
1 !           IDX$W_MINRECSZ      - minimum record size necessary to contain key
1 !
1 !       IFAB                    - address of IFAB
1 !           IFB$W_KBUFSZ        - size of a keybuffer
1 !           IFB$B_NUM_KEYS      - number of keys in the file
1 !           IFB$B_PLG_VER       - prologue version of the file
1 !
1 !       IRAB                    - address of IRAB
1 !           IRB$B_CACHEFLGS     - flags for bucket retrieval routines
1 !           IRB$L_CURBDB        - address of current buffer descriptor block
1 !           IRB$L_KEYBUF        - address of contigious keybuffers
1 !           IRB$B_MODE          - access mode of caller
1 !           IRB$L_NXTBDB        - address of a BDB (used to hold RRV bucket BDB)
1 !           IRB$W_PUTUP_ID      - ID of user data record
1 !           IRB$L_PUTUP_VBN     - VBN of user data record
1 !           IRB$W_UDR_ID        - ID of current primary data record
1 !           IRB$L_UDR_VBN       - VBN of current primary data record
1 !           IRB$V_UPDATE        - if set, current operation is an $UPDATE
1 !           IRB$L_UPDBUF        - address of internal update buffer
1 !
1 !       RAB                     - address of RAB
1 !           RAB$L_RBF           - record buffer containing user data record
1 !           RAB$W_RSZ           - size of user data record
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
```

```
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       IRB$B_CACHEFLGS          - the bit CSH$V_LOCK will be set
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       AP is trashed.
1 !       All new SIDR entries inserted during the current operation before the
1 !           error occurred are deleted as is the user data record and any RRV
1 !           pointing to it if a new data record was inserted during the course
1 !           of the operation before the error occurred (ie - the operation was
1 !           a $PUT).
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         R_IDX_DFN_STR,
2         COMMON_RAB_STR;
2
2     GLOBAL REGISTER
2         R_REC_ADDR_STR;
2
2     ! If the file allows secondary keys then delete any of those that had been
2     ! newly inserted before the error occurred.
2     !
2     IF .IFAB[IFB$B_NUM_KEYS] GTRU 0
2     THEN
3         BEGIN
3
3         LABEL
3             ENTRY;
3
3         LOCAL
3             KREF                  : BYTE,
3             SAVE_UDR_ID           : WORD,
3             SAVE_UDR_VBN          : LONG;
3
3         ! The routine which is called to deleted each SIDR entry,
3         ! RMS$DELETE_SIDR, operates only on the current primary data record.
3         ! However, this routine maybe called to delete SIDR entries of a record
3         ! other than the current primary data record. Therefore, in order to
3         ! make use of RMS$DELETE_SIDR, RMS must fool it into believing that it
3         ! is operating on the current primary data record. This is done by
3         ! saving the RFA address of the current primary data record, if there
3         ! is one, and replacing it with the RFA address of the user data record
3         ! whose SIDR entries are to be deleted.
3         !
3         SAVE_UDR_VBN  = .IRAB[IRB$L_UDR_VBN];
3         SAVE_UDR_ID   = .IRAB[IRB$W_UDR_ID];
3
3         IRAB[IRB$L_UDR_VBN]  = .IRAB[IRB$L_PUTUP_VBN];
```

```
3              IRAB[IRB$W_UDR_ID]   = .IRAB[IRB$W_PUTUP_ID];
3
3              ! The user data record whose $PUT/$UPDATE resulted in the error maybe
3              ! found in the user's record buffer. The keys of the SIDR entries to
3              ! be deleted maybe extracted from it.
3              !
3              REC_ADDR = .RAB[RAB$L_RBF];
3
3              ! If this routine was called with the index descriptor for the primary
3              ! key then all SIDR entries for the user data record are deleted;
3              ! otherwise, all SIDR entries up until the entry that was being
3              ! inserted when the error ocurred are deleted.
3              !
3              KREF = .IDX_DFN[IDX$B_KEYREF];
3              RM$KEY_DESC(0);
3
4              WHILE (RM$GET_NEXT_KEY()
4                     AND
4                   (.IDX_DFN[IDX$B_KEYREF] NEQU .KREF))
3              DO
3
3                  ! Delete each SIDR entry that had been inserted before the error
3                  ! occurred.
3                  !
4 ENTRY:           BEGIN
4
4                  BUILTIN
4                      AP;
4
4                  ! Under the following circumstances, the SIDR entry for the current
4                  ! index descriptor being processed will not be deleted:
4                  !
4                  ! 1. The operation being performed when the error occurred was an
4                  !    $UPDATE and no new SIDR was inserted for this key of reference.
4                  ! 2. RMS does not have read access to the user's record buffer.
4                  ! 3. No new SIDR was inserted for this key of reference because the
4                  !    user's data record was too short  ) contain such a key.
4                  !
5                  IF   (.IRAB[IRB$V_UPDATE]
5                          AND
6                        (NOT .BBLOCK[.IRAB[IRB$L_UPDBUF] + .IDX_DFN[IDX$B_DESC_NO],
5                               UPD$V_INS_NEW]))
4                       OR
4                      RM$NOREAD_LONG (.RAB[RAB$W_RSZ], .REC_ADDR, .IRAB[IRB$B_MODE])
4                       OR
5                      (.RAB[RAB$W_RSZ] LSSU .IDX_DFN[IDX$W_MINRECSZ])
4                  THEN
4                      LEAVE ENTRY;
4
4                  ! Extract into keybuffer 2, the secondary key for the key of
4                  ! reference being processed from the user's record buffer. Check
4                  ! whether the key is null and only delete the SIDR entry for this
4                  ! key of reference if it is not.
4                  !
4                  AP = 3;
4
5                  BEGIN
```

```
5                   GLOBAL REGISTER
5                       R_BDB;
5
5                   RM$RECORD_KEY (KEYBUF_ADDR(2));
4                   END;
4
4                   AP = 1;
4                   IF RM$NULLKEY (KEYBUF_ADDR(2))
4                   THEN
4                       RM$DFLETE_SIDR();
4
3                   END;
3
3               ! Restore the RFA of the cur'ent primary data record (if there is one)
3               ! to its corresponding locat on in the IRAB as part of the next record
3               ! positioning context as RMS has finished deleting SIDR entries.
3               !
3               IRAB[IRB$L_UDR_VBN]  = .SAVE_UDR_VBN;
3               IRAB[IRB$W_UDR_ID]   = .SAVE_UDR_ID;
2               END;
2
2           ! If the error occurred during a $PUT, then a user data record was
2           ! inserted before any SIDRs and must be deleted. If the error occurred
2           ! during an $UPDATE then just the deletion of any new SIDR entries required
2           ! by the $UPDATE is sufficient to restore the record to the state it
2           ! occupied prior to the $UPDATE. None of the SIDR entries for the user data
2           ! record existing in the file before the $UPDATE are deleted until all the
2           ! new SIDR entries are inserted so there are no SIDR entries to re-insert,
2           ! and of course, the user data record itself can never be deleted because
2           ! it existed in the file prior to the $UPDATE.
2           !
2           ! NOTE that it is possible that RMS will also have to delete an RRV for this
2           ! new user data record even though RRVs are never created during the
2           ! insertion of a new primary data record. This is because RMS will release
2           ! the primary data bucket containing the new record during index updates
2           ! and SIDR entry insertions, and the action of some other stream may cause
2           ! the bucket containing this new primary data bucket to split and an RRV
2           ! created for it.
2           !
2           IF NOT .IRAB[IRB$V_UPDATE]
2           THEN
3               BEGIN
3
3               RM$KEY_DESC(0);
3
3               ! Attempt to position to the user data record, and delete it if able to
3               ! successfully position to it. Perform the FIND_BY_RRV in a way such
3               ! that if an RRV was created for the new primary data record between
3               ! the time this stream released the primary data bucket and reclaims it
3               ! below, the RRV bucket will be locked during the positioning to the
3               ! user data record, and the address of the BDB for it placed in
3               ! IRB$L_NXTBDB.
3               !
3               IRAB[IRB$B_CACHEFLGS] = CSH$M_LOCK;
3               IRAB[IRB$L_NXTBDB] = 0;
3
3               IF RM$FIND_BY_RRV (.IRAB[IRB$L_PUTUP_VBN], .IRAB[IRB$W_PUTUP_ID], 1)
```

```
:   355         0418  3         THEN
:   356         0419  4             BEGIN
:   357         0420  4
:   358         0421  4             GLOBAL REGISTER
:   359         0422  4                 R_BDB_STR,
:   360         0423  4                 R_BKT_ADDR;
:   361         0424  4
:   362         0425  4             ! If the new user data record is found not to be in its original
:   363         0426  4             ! bucket, then the RRV for it must be deleted.
:   364         0427  4             !
:   365         0428  4             IF (BDB = .IRAB[IRB$L_NXTBDB]) NEQU 0
:   366         0429  4             THEN
:   367         0430  5                 BEGIN
:   368         0431  5                 IRAB[IRB$L_NXTBDB] = 0;
:   369         0432  5                 RM$DELETE_RRV();
:   370         0433  4                 END;
:   371         0434  4
:   372         0435  4             ! Delete the new user data record.
:   373         0436  4             !
:   374         0437  4             BDB = .IRAB[IRB$L_CURBDB];
:   375         0438  4             IRAB[IRB$L_CURBDB] = 0;
:   376         0439  4
:   377         0440  4             RM$DELETE_UDR();
:   378         0441  4
:   379         0442  4             ! Mark the primary data bucket that contained the new user data
:   380         0443  4             ! record dirty, and release it.
:   381         0444  4             !
:   382         0445  4             BDB[BDB$V_DRT] = 1;
:   383         0446  4             RM$RLSBKT(0);
:   384         0447  4
:   385         0448  3             END;
:   386         0449  3
:   387         0450  2         END;
:   388         0451  2
:   389         0452  1     END;


                                                                  .TITLE  RM3PUTERR
                                                                  .IDENT  \V04-000\

                                                                  .EXTRN  RM$DELETE_RRV, RM$DELETE_SIDR
                                                                  .EXTRN  RM$DELETE_UDR, RM$FIND_BY_RRV
                                                                  .EXTRN  RM$GET_NEXT_KEY
                                                                  .EXTRN  RM$KEY_DESC, RM$NOREAD_LONG
                                                                  .EXTRN  RM$NULLKEY, RM$RECORD_KEY
                                                                  .EXTRN  RM$RLSBKT

                                                                  .PSECT  RM$RMS3,NOWRT,  GBL,  PIC,2

                                        007D   8F  BB 00000 RM$PUTUPD_ERROR::
                                                                  PUSHR   #^M<R0,R2,R3,R4,R5,R6>                    ; 0191
                                        00B2   CA  95 00004       TSTB    178(IFAB)                                 ; 0278
                                               03  12 00008       BNEQ    1$                                        ;
                                            0091   31 0000A       BRW     5$                                        ;
                                55      00B0   C9  D0 0000D 1$:   MOVL    176(IRAB), SAVE_UDR_VBN                   ; 0300
                                52      00BC   C9  B0 00012       MOVW    188(IRAB), SAVE_UDR_ID                    ; 0301
                        00B0    C9        78   A9  D0 00017       MOVL    120(IRAB), 176(IRAB)                      ; 0303
```

```
                 00BC   C9   0080   C9  B0 0001D         MOVW     128(IRAB), 188(IRAB)            ; 0304  ;
                        56     28   A8  D0 00024         MOVL     40(RAB), REC_ADDR               ; 0310  ;
                        53     21   A7  90 00028         MOVB     33(IDX_DFN), KREF               ; 0317  ;
                                    7E  D4 0002C         CLRL     -(SP)                           ; 0318  ;
                                 0000G  30 0002E         BSBW     RM$KEY_DESC                     ;       ;
                        5E          04  C0 00031         ADDL2    #4, SP                          ;       ;
                                 0000G  30 00034 2$:     BSBW     RM$GET_NEXT_KEY                 ; 0320  ;
                        5A          50  E9 00037         BLBC     R0, 4$                          ;       ;
                        53     21   A7  91 0003A         CMPB     33(IDX_DFN), KREF               ; 0322  ;
                                    54  13 0003E         BEQL     4$                              ;       ;
          0B       06   A9          03  E1 00040         BBC      #3, 6(IRAB), 3$                 ; 0342  ;
                        50     10   A7  9A 00045         MOVZBL   16(IDX_DFN), R0                 ; 0345  ;
                        50     64   A9  C0 00049         ADDL2    100(IRAB), R0                   ;       ;
                        E4          60  E9 0004D         BLBC     (R0), 2$                        ;       ;
                        7E     0A   A9  9A 00050 3$:     MOVZBL   10(IRAB), -(SP)                 ; 0347  ;
                                    56  DD 00054         PUSHL    REC_ADDR                        ;       ;
                        7E     22   A8  3C 00056         MOVZWL   34(RAB), -(SP)                  ;       ;
                                 0000G  30 0005A         BSBW     RM$NOREAD_LONG                  ;       ;
                        5E          0C  C0 0005D         ADDL2    #12, SP                         ;       ;
                        D1          50  E8 00060         BLBS     R0, 2$                          ;       ;
                   22   A7     22   A8  B1 00063         CMPW     34(RAB), 34(IDX_DFN)            ; 0349  ;
                                    CA  1F 00068         BLSSU    2$                              ;       ;
                        5C          03  D0 0006A         MOVL     #3, AP                          ; 0358  ;
                        50   00B4   CA  3C 0006D         MOVZWL   180(IFAB), R0                   ; 0364  ;
                               60 B940  9F 00072         PUSHAB   @96(IRAB)[R0]                   ;       ;
                                 0000G  30 00076         BSBW     RM$RECORD_KEY                   ;       ;
                        5C          01  D0 00079         MOVL     #1, AP                          ; 0367  ;
                        50   00B4   CA  3C 0007C         MOVZWL   180(IFAB), R0                   ; 0368  ;
                        6E     60 B940  9E 00081         MOVAB    @96(IRAB)[R0], (SP)             ;       ;
                                 0000G  30 00086         BSBW     RM$NULLKEY                      ;       ;
                        5E          04  C0 00089         ADDL2    #4, SP                          ;       ;
                        A5          50  E9 0008C         BLBC     R0, 2$                          ;       ;
                                 0000G  30 0008F         BSBW     RM$DELETE_SIDR                  ; 0370  ;
                                    A0  11 00092         BRB      2$                              ; 0320  ;
                 00B0   C9          55  D0 00094 4$:     MOVL     SAVE_UDR_VBN, 176(IRAB)         ; 0378  ;
                 00BC   C9          52  B0 00099         MOVW     SAVE_UDR_ID, 188(IRAB)          ; 0379  ;
          42       06   A9          03  E0 0009E 5$:     BBS      #3, 6(IRAB), 7$                 ; 0400  ;
                                    7E  D4 000A3         CLRL     -(SP)                           ; 0404  ;
                                 0000G  30 000A5         BSBW     RM$KEY_DESC                     ;       ;
                   40   A9          01  90 000A8         MOVB     #1, 64(IRAB)                    ; 0414  ;
                               3C   A9  D4 000AC         CLRL     60(IRAB)                        ; 0415  ;
                        6E          01  D0 000AF         MOVL     #1, (SP)                        ; 0417  ;
                        7E   0080   C9  3C 000B2         MOVZWL   128(IRAB), -(SP)                ;       ;
                               78   A9  DD 000B7         PUSHL    120(IRAB)                       ;       ;
                                 0000G  30 000BA         BSBW     RM$FIND_BY_RRV                  ;       ;
                        5E          0C  C0 000BD         ADDL2    #12, SP                         ;       ;
                        22          50  E9 000C0         BLBC     R0, 7$                          ;       ;
                        54     3C   A9  D0 000C3         MOVL     60(IRAB), BDB                   ; 0428  ;
                                    06  13 000C7         BEQL     6$                              ;       ;
                               3C   A9  D4 000C9         CLRL     60(IRAB)                        ; 0431  ;
                                 0000G  30 000CC         BSBW     RM$DELETE_RRV                   ; 0432  ;
                        54     20   A9  D0 000CF 6$:     MOVL     32(IRAB), BDB                   ; 0437  ;
                               20   A9  D4 000D3         CLRL     32(IRAB)                        ; 0438  ;
                                 0000G  30 000D6         BSBW     RM$DELETE_UDR                   ; 0440  ;
                   0A   A4          02  88 000D9         BISB2    #2, 10(BDB)                     ; 0445  ;
                                    7E  D4 000DD         CLRL     -(SP)                           ; 0446  ;
                                 0000G  30 000DF         BSBW     RM$RLSBKT                       ;       ;
```

```
                              5E          04  C0 000E2          ADDL2    #4, SP                           :
                                   007D   8F  BA 000E5 7$:      POPR     #^M<R0,R2,R3,R4,R5,R6>           : 0452
                                              05 000E9          RSB                                       :

; Routine Size:  234 bytes,    Routine Base:  RM$RMS3 + 0000
```

```
%SBTTL  'RM$CLEAN_BDB'
GLOBAL ROUTINE RM$CLEAN_BDB : RL$ERROR_LINK1 NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine's responsibility is to release any buckets that
!       are currently accessed.
!
!
! CALLING SEQUENCE:
!
!       BSBW RM$CLEAN_BDB()
!
! INPUT PARAMETERS:
!       NONE
!
! IMPLICIT INPUTS:
!
!       IRAB                        - address of IRAB
!           IPB$L_CURBDB
!           IRB$L_LOCKBDB
!           IRB$L_NXTBDB
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!
!       If there is a bucket associated with IRB$L_NXTBDB, it is released.
!       if there is a bucket associated with IRB$L_LOCK_BDB, it is released.
!       if there is a bucket associated with IRB$L_CURBDB, it is released.
!
!--

    BEGIN

    EXTERNAL REGISTER
        COMMON_RAB_STR;

    GLOBAL REGISTER
        R_REC_ADDR,
        R_IDX_DFN,
        R_BDB_STR;

    ! If there is an accessed bucket associated with IRB$L_NXTBDB,
    ! then release it.
    !
    IF (BDB = .IRAB[IRB$L_NXTBDB]) NEQ 0
    THEN
```

```
;  448        0510  3           BEGIN
;  449        0511  3           IRAB[IRB$L_NXTBDB] = 0;
;  450        0512  3           RM$RLSBKT(0);
;  451        0513  2           END;
;  452        0514  2
;  453        0515  2       ! If there is an accessed bucket associated with IRB$L_LOCK_BDB,
;  454        0516  2       ! then release it.
;  455        0517  2       !
;  456        0518  2       IF (BDB = .IRAB[IRB$L_LOCK_BDB]) NEQ 0
;  457        0519  2       THEN
;  458        0520  3           BEGIN
;  459        0521  3           IRAB[IRB$L_LOCK_BDB] = 0;
;  460        0522  3           RM$RLSBKT(0);
;  461        0523  2           END;
;  462        0524  2
;  463        0525  2       ! If there is an accessed bucket associated with IRB$L_CURBDB,
;  464        0526  2       ! then release it.
;  465        0527  2       !
;  466        0528  2       IF (BDB = .IRAB[IRB$L_CURBDB]) NEQ 0
;  467        0529  2       THEN
;  468        0530  3           BEGIN
;  469        0531  3           IRAB[IRB$L_CURBDB] = 0;
;  470        0532  3           RM$RLSBKT(0);
;  471        0533  2           END;
;  472        0534  2
;  473        0535  1       END;



                              00D1    8F   BB 00000 RM$CLEAN_BDB::
                                                         PUSHR    #^M<R0,R4,R6,R7>                    ; 0454
                    54          3C    A9   D0 00004      MOVL     60(IRAB), BDB                       ; 0508
                                      0B   13 00008      BEQL     1$                                  ;
                                3C    A9   D4 0000A      CLRL     60(IRAB)                            ; 0511
                                      7E   D4 0000D      CLRL     -(SP)                               ; 0512
                                   0000G   30 0000F      BSBW     RM$RLSBKT                           ;
                    5E                04   C0 00012      ADDL2    #4, SP                              ;
                    54        0084    C9   D0 00015 1$:  MOVL     132(IRAB), BDB                      ; 0518
                                      0C   13 0001A      BEQL     2$                                  ;
                              0084    C9   D4 0001C      CLRL     132(IRAB)                           ; 0521
                                      7E   D4 00020      CLRL     -(SP)                               ; 0522
                                   0000G   30 00022      BSBW     RM$RLSBKT                           ;
                    5E                04   C0 00025      ADDL2    #4, SP                              ;
                    54          20    A9   D0 00028 2$:  MOVL     32(IRAB), BDB                       ; 0528
                                      0B   13 0002C      BEQL     3$                                  ;
                                20    A9   D4 0002E      CLRL     32(IRAB)                            ; 0531
                                      7E   D4 00031      CLRL     -(SP)                               ; 0532
                                   0000G   30 00033      BSBW     RM$RLSBKT                           ;
                    5E                04   C0 00036      ADDL2    #4, SP                              ;
                              00D1    8F   BA 00039 3$:  POPR     #^M<R0,R4,R6,R7>                    ; 0535
                                           05 0003D      RSB                                          ;

; Routine Size:  62 bytes,    Routine Base:  RM$RMS3 + 00EA
;  474          0536  1
;  475          0537  1 END
;  476          0538  1
;  477          0539  0 ELUDOM



;                                      PSECT SUMMARY

;       Name                          Bytes                           Attributes

;  RM$RMS3                                  296  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  GBL,  REL,  CON,  PIC,ALIGN(2)



;                                Library Statistics

;                                                 -------- Symbols --------       Pages        Processing
;       File                                      Total     Loaded   Percent       Mapped       Time

;  _$255$DUA28:[RMS.OBJ]RMS.L32;1                  3109        51         1          154          00:00.4



;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RM3PUTERR/OBJ=OBJ$:RM3PUTERR MSRC$:RM3PUTERR/UPDATE=(ENH$:RM3PUTERR)

; Size:          296 code + 0 data bytes
; Run Time:         00:09.4
; Elapsed Time:     00:19.5
; Lines/CPU Min:     3444
; Lexemes/CPU-Min: 15559
; Memory Used:  103 pages
; Compilation Complete
```

0327 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RM3PROBE
LIS

RM3SIDXSP
LIS

RM3PUTERR
LIS

RM3PUTUPD
LIS

RM3SPLUDR
LIS

RM3RRV
LIS

RM3ROOT
LIS

RM3PUT
LIS