```
RRRRRRRRRRR      MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM     SSSSSSSSSSS
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
RRRRRRRRRRR      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  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
RRR         RRR  MMM          MMM  SSSSSSSSSSS
```

```
RRRRRRRR    MM      MM    333333    RRRRRRRR      000000      000000    TTTTTTTTTT
RRRRRRRR    MM      MM    333333    RRRRRRRR      000000      000000    TTTTTTTTTT
RR      RR  MMMM  MMMM  33      33  RR      RR  00      00  00      00      TT
RR      RR  MMMM  MMMM  33      33  RR      RR  00      00  00      00      TT
RR      RR  MM  MM  MM        3333  RR      RR  00      00  00      00      TT
RR      RR  MM  MM  MM          33  RR      RR  00      00  00      00      TT
RRRRRRRR    MM      MM        33    RRRRRRRR    00      00  00      00      TT
RRRRRRRR    MM      MM      33      RRRRRRRR    00      00  00      00      TT
RR  RR      MM      MM        33    RR  RR      00      00  00      00      TT
RR  RR      MM      MM        33    RR  RR      00      00  00      00      TT
RR    RR    MM      MM  33    33    RR    RR    00      00  00      00      TT        ....
RR    RR    MM      MM  33    33    RR    RR    00      00  00      00      TT        ....
RR      RR  MM      MM    333333    RR      RR    000000      000000      TT        ....
RR      RR  MM      MM    333333    RR      RR    000000      000000      TT        ....

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
```

```
;     1          0001  0 MODULE RM3ROOT (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 !                Create new root bucket for index file organization
;    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:        Christian Saether          CREATION DATE:            8-AUG-78  20:30
;    45          0045  1 !
;    46          0046  1 ! Modified by:
;    47          0047  1 !
;    48          0048  1 !       V03-005 RAS0284         Ron Schaefer            30-Mar-1984
;    49          0049  1 !               Fix STV value on error paths for RMS$_RPL and RMS$_WPL errors.
;    50          0050  1 !
;    51          0051  1 !       V03-004 MCN0003         Maria del C. Nasr       31-Mar-1983
;    52          0052  1 !               More linkages reorganization.
;    53          0053  1 !
;    54          0054  1 !       V03-003 MCN0002         Maria del C. Nasr       22-Feb-1983
;    55          0055  1 !               Reorganize linkages
;    56          0056  1 !
;    57          0057  1 !       V03-002 KBT0231         Keith B. Thompson       23-Aug-1982
```

```
;     58          0058  1 !                 Reorganize psects
;     59          0059  1 !
;     60          0060  1 !       V03-001 MCN0001         Maria del C. Nasr       24-Mar-1982
;     61          0061  1 !                 Use macro to compute key buffer address.
;     62          0062  1 !
;     63          0063  1 !       V02-00  TMK0001         Todd M. Katz    01-Feb-1982
;     64          0064  1 !                 Take out all references to rear-end truncation. Support for
;     65          0065  1 !                 rear end truncation of prolog 3 compressed index keys does
;     66          0066  1 !                 not require any modification to the routines in this module.
;     67          0067  1 !
;     68          0068  1 !       V02-007 PSK0003         P Knibbe        16-Nov-1981
;     69          0069  1 !                 Add support for compressed indexes.
;     70          0070  1 !
;     71          0071  1 !       V02-006 PSK0002         P Knibbe        26-Oct-1981
;     72          0072  1 !                 Fix problem with BKT_ADDR not being CURBDB
;     73          0073  1 !                 for RECORD_SIZE.
;     74          0074  1 !
;     75          0075  1 !       V02-005 PSK0001         P Knibbe        09-Aug-1981
;     76          0076  1 !                 Add support for prologue three files.
;     77          0077  1 !
;     78          0078  1 !       V02-004 REFORMAT        C Saether       01-Aug-1980     22:32
;     79          0079  1 !
;     80          0080  1 !
;     81          0081  1 ! REVISION HISTORY:
;     82          0082  1 !
;     83          0083  1 !   Wendy Koenig,       24-OCT-78  14:03
;     84          0084  1 !   X0002 - MAKE CHANGES CAUSED BY SHARING CONVENTIONS
;     85          0085  1 !
;     86          0086  1 !   Wendy Koenig,       26-JAN-79  9:19
;     87          0087  1 !   X0003 - GET RID OF SETTING VALID
;     88          0088  1 !
;     89          0089  1 !*****
;     90          0090  1
;     91          0091  1 LIBRARY 'RMSLIB:RMS';
;     92          0092  1
;     93          0093  1 REQUIRE 'RMSSRC:RMSIDXDEF';
;     94          0158  1
;     95          0159  1 ! define default psects for code
;     96          0160  1 !
;     97          0161  1 PSECT
;     98          0162  1     CODE = RM$RMS3(PSECT_ATTR),
;     99          0163  1     PLIT = RM$RMS3(PSECT_ATTR);
;    100          0164  1
;    101          0165  1 ! Linkages
;    102          0166  1 !
;    103          0167  1 LINKAGE
;    104          0168  1     L_CHKSUM,
;    105          0169  1     L_PRESERVE1,
;    106          0170  1     L_RABREG_4567,
;    107          0171  1     L_RABREG_567,
;    108          0172  1     L_RABREG_67,
;    109          0173  1     L_RABREG_7,
;    110          0174  1     L_RELEASE;
;    111          0175  1
;    112          0176  1 !
;    113          0177  1 ! External Routines
;    114          0178  1 !
```

```
;  115          0179  1
;  116          0180  1 EXTERNAL ROUTINE
;  117          0181  1     RM$MAK_IDX_REC      : RL$RABREG_67 NOVALUE,
;  118          0182  1     RM$MAKSUM           : RL$CHKSUM,
;  119          0183  1     RM$MOVE             : RL$PRESERVE1,
;  120          0184  1     RM$RECORD_SIZE      : RL$RABREG_567,
;  121          0185  1     RM$RELEASE          : RL$RELEASE ADDRESSING_MODE( GENERAL ),
;  122          0186  1     RM$KEY_DESC         : RL$RABREG_7,
;  123          0187  1     RM$INS_REC          : RL$RABREG_67;
;  124          0188  1
```

```
;  126     0189  1 GLOBAL ROUTINE RM$NEW_ROOT : RL$RABREG_4567 NOVALUE =
;  127     0190  1
;  128     0191  1 !++
;  129     0192  1 !
;  130     0193  1 ! RM$NEW_ROOT
;  131     0194  1 !
;  132     0195  1 ! Create new record and high key record for new root bucket.
;  133     0196  1 !
;  134     0197  1 ! CALLING SEQUENCE:
;  135     0198  1 !       RM$NEW_ROOT()
;  136     0199  1 !
;  137     0200  1 ! INPUT PARAMETERS:
;  138     0201  1 !       NONE
;  139     0202  1 !
;  140     0203  1 ! IMPLICIT INPUTS:
;  141     0204  1 !       BKT_ADDR - address of new root bucket buffer
;  142     0205  1 !
;  143     0206  1 ! OUTPUT PARAMETERS:
;  144     0207  1 !       NONE
;  145     0208  1 !
;  146     0209  1 ! IMPLICIT OUTPUTS:
;  147     0210  1 !       NONE
;  148     0211  1 !
;  149     0212  1 ! ROUTINE VALUE:
;  150     0213  1 !       NONE
;  151     0214  1 !
;  152     0215  1 ! SIDE EFFECTS:
;  153     0216  1 !       NONE
;  154     0217  1 !
;  155     0218  1 !--
;  156     0219  1
;  157     0220  2     BEGIN
;  158     0221  2
;  159     0222  2     EXTERNAL REGISTER
;  160     0223  2         COMMON_RAB_STR,
;  161     0224  2         COMMON_IO_STR,
;  162     0225  2         R_REC_ADDR_STR,
;  163     0226  2         R_IDX_DFN_STR;
;  164     0227  2
;  165     0228  2     BKT_ADDR[BKT$L_NXTBKT] = .BDB[BDB$L_VBN];
;  166     0229  2     BKT_ADDR[BKT$B_LEVEL] = .BKT_ADDR[BKT$B_LEVEL] + 1;
;  167     0230  2     BKT_ADDR[BKT$V_LASTBKT] = 1;
;  168     0231  2     BKT_ADDR[BKT$V_ROOTBKT] = 1;
;  169     0232  2     REC_ADDR = .BKT_ADDR + BKT$C_OVERHDSZ;
;  170     0233  2     IRAB[IRB$L_LST_REC] = .REC_ADDR;
;  171     0234  2
;  172     0235  2     IRAB [IRB$L_MIDX_TMP1] = .IRAB [IRB$L_VBN_LEFT];
;  173     0236  2     RM$MAK_IDX_REC(.BKT_ADDR);
;  174     0237  2
;  175     0238  2     BKT_ADDR[BKT$W_FREESPACE] = .REC_ADDR - .BKT_ADDR;
;  176     0239  2     REC_ADDR = .IRAB[IRB$L_LST_REC];
;  177     0240  2     IRAB[IRB$W_POS_INS] = BKT$C_OVERHDSZ;
;  178     0241  2     IRAB[IRB$B_SPL_BITS] = 0;
;  179     0242  2     IRAB[IRB$L_REC_COUNT] = 0;
;  180     0243  2
;  181     0244  3     BEGIN
;  182     0245  3
```

```
;  183          0246  3     LOCAL
;  184          0247  3         TEMP_SIZE,
;  185          0248  3         SAVE_BDB;
;  186          0249  3
;  187          0250  3     SAVE_BDB = .IRAB [IRB$L_CURBDB];
;  188          0251  3     IRAB [IRB$L_CURBDB] = .IRAB [IRB$L_NXTBDB];
;  189          0252  3
;  190          0253  3     IF .IDX_DFN [IDX$V_IDX_COMPR]
;  191          0254  3     THEN
;  192          0255  4         BEGIN
;  193          0256  4
;  194          0257  4         MACRO
;  195          0258  4             KEYLEN      = 0,0,8,0 %,
;  196          0259  4             COMPR       = 0,8,8,0 %;
;  197          0260  4
;  198          0261  4         LOCAL
;  199          0262  4             KEYBUF : REF BBLOCK;
;  200          0263  4
;  201          0264  4         KEYBUF = KEYBUF_ADDR(2);
;  202          0265  4
;  203          0266  4         RM$MOVE (.IDX_DFN [IDX$B_KEYSZ], .KEYBUF, .KEYBUF + 2);
;  204          0267  4         KEYBUF [KEYLEN] = .IDX_DFN [IDX$B_KEYSZ];
;  205          0268  4         KEYBUF [COMPR] = 0;
;  206          0269  3         END;
;  207          0270  3
;  208          0271  3     TEMP_SIZE = RM$RECORD_SIZE();
;  209          0272  3
;  210          0273  3     IF NOT RM$INS_REC(.BKT_ADDR, .TEMP_SIZE)
;  211          0274  3     THEN
;  212          0275  3         BUG_CHECK;
;  213          0276  3
;  214          0277  3     IRAB [IRB$L_CURBDB] = .SAVE_BDB;
;  215          0278  2     END; ! Of local definition of temp_size and save_bdb
;  216          0279  2
;  217          0280  2     IRAB[IRB$L_VBN_LEFT] = .BDB[BDB$L_VBN];
;  218          0281  1     END;


                                                .TITLE  RM3ROOT
                                                .IDENT  \V04-000\

                                                .EXTRN  RM$MAK_IDX_REC, RM$MAKSUM
                                                .EXTRN  RM$MOVE, RM$RECORD_SIZE
                                                .EXTRN  RM$RELEASE, RM$KEY_DESC
                                                .EXTRN  RM$INS_REC, RM$BUG3

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

                            52  DD 00000 RM$NEW_ROOT::
                                                PUSHL   R2                              ; 0189
                08  A5      1C  A4  D0 00002   MOVL    28(BDB), 8(BKT_ADDR)            ; 0228
                            0C  A5  96 00007   INCB    12(BKT_ADDR)                    ; 0229
                0D  A5          03  88 0000A   BISB2   #3, 13(BKT_ADDR)                ; 0231
                    56      0E  A5  9E 0000E   MOVAB   14(R5), REC_ADDR                ; 0232
                4C  A9          56  D0 00012   MOVL    REC_ADDR, 76(IRAB)              ; 0233
                                55  DD 00016   PUSHL   BKT_ADDR                        ; 0236
                              0000G 30 00018   BSBW    RM$MAK_IDX_REC                 ;
```

```
                        5E              04  C0 0001B           ADDL2   #4, SP                                                  ;
                04  A5                  56  A3 0001E           SUBW3   BKT_ADDR, REC_ADDR, 4(BKT_ADDR)                  ; 0238
                                        56  A9 00023           MOVL    76(IRAB), REC_ADDR                               ; 0239
                                48  A9  0E  B0 00027           MOVW    #14, 72(IRAB)                                    ; 0240
                                    44  A9  94 0002B           CLRB    68(IRAB)                                         ; 0241
                                0094    C9  D4 0002E           CLRL    148(IRAB)                                        ; 0242
                                        52  20 A9 D0 00032     MOVL    32(IRAB), SAVE_BDB                               ; 0250
                                    20  A9  3C  A9 D0 00036     MOVL    60(IRAB), 32(IRAB)                               ; 0251
                        1C          1C  A7  03  E1 0003B       BBC     #3, 28(IDX_DFN), 1$                              ; 0253
                                        51  00B4    CA 3C 00040 MOVZWL  180(IFAB), KEYBUF                                ; 0264
                                        51  60  A9  C0 00045   ADDL2   96(IRAB), KEYBUF                                 ;
                                            02  A1  9F 00049   PUSHAB  2(KEYBUF)                                        ; 0266
                                            51  DD 0004C       PUSHL   KEYBUF                                           ;
                                        7E  20  A7  9A 0004E   MOVZBL  32(IDX_DFN), -(SP)                               ;
                                          0000G 30 00052       BSBW    RM$MOVE                                          ;
                                        5E  0C  C0 00055       ADDL2   #12, SP                                          ;
                                        61  20  A7  9B 00058   MOVZBW  32(IDX_DFN), (KEYBUF)                            ; 0267
                                          0000G 30 0005C 1$:   BSBW    RM$RECORD_SIZE                                   ; 0271
                                            50  DD 0005F       PUSHL   TEMP_SIZE                                        ; 0273
                                            55  DD 00061       PUSHL   BKT_ADDR                                         ;
                                          0000G 30 00063       BSBW    RM$INS_REC                                       ;
                                        5E  08  C0 00066       ADDL2   #8, SP                                           ;
                                        03  50  E8 00069       BLBS    R0, 2$                                           ;
                                          0000G 30 0006C       BSBW    RM$BUG3                                          ; 0274
                            20  A9          52  D0 0006F 2$:   MOVL    SAVE_BDB, 32(IRAB)                               ; 0277
                        0088    C9      1C  A4  D0 00073       MOVL    28(BDB), 136(IRAB)                               ; 0280
                                            04  BA 00079       POPR    #^M<R2>                                          ; 0281
                                            05  0007B         RSB                                                      ;

; Routine Size:  124 bytes,    Routine Base:  RM$RMS3 + 0000

;  219          0282  1
```

```
;   221          0283  1 GLOBAL ROUTINE RM$UPD_PLG : RL$RABREG_7 =
;   222          0284  1
;   223          0285  1 !++
;   224          0286  1 !
;   225          0287  1 ! FUNCTIONAL DESCRIPTION:
;   226          0288  1 !
;   227          0289  1 ! Update prologue to reflect new root level and root vbn
;   228          0290  1 !
;   229          0291  1 ! CALLING SEQUENCE:
;   230          0292  1 !       RM$UPD_PLG ()
;   231          0293  1 !
;   232          0294  1 ! INPUT PARAMETERS:
;   233          0295  1 !       NONE
;   234          0296  1 !
;   235          0297  1 ! IMPLICIT INPUTS:
;   236          0298  1 !       NONE
;   237          0299  1 !
;   238          0300  1 ! OUTPUT PARAMETERS:
;   239          0301  1 !       NONE
;   240          0302  1 !
;   241          0303  1 ! IMPLICIT OUTPUTS:
;   242          0304  1 !       NONE
;   243          0305  1 !
;   244          0306  1 ! ROUTINE VALUE:
;   245          0307  1 !       success
;   246          0308  1 !       tre - if rootlevel attempts to go beyond 255
;   247          0309  1 !       wpl - prologue write error
;   248          0310  1 !
;   249          0311  1 ! SIDE EFFECTS:
;   250          0312  1 !       NONE
;   251          0313  1 !
;   252          0314  1 !--
;   253          0315  1
;   254          0316  2     BEGIN
;   255          0317  2
;   256          0318  2     EXTERNAL REGISTER
;   257          0319  2         COMMON_RAB_STR,
;   258          0320  2         R_IDX_DFN_STR;
;   259          0321  2
;   260          0322  2     GLOBAL REGISTER
;   261          0323  2         R_BDB_STR;
;   262          0324  2
;   263          0325  2     LOCAL
;   264          0326  2         STATUS,
;   265          0327  2         KEY     : REF BBLOCK;
;   266          0328  2
;   267          0329  2     ! set search and cache flags to force read with lock of prologue
;   268          0330  2     !
;   269          0331  2     IRAB[IRB$V_NEW_IDX] = 1;
;   270          0332  2     IRAB[IRB$B_CACHEFLGS] = CSH$M_LOCK;
;   271          0333  2
;   272          0334  2     RETURN_ON_ERROR (RM$KEY_DESC(.IDX_DFN[IDX$B_KEYREF]));
;   273          0335  2
;   274          0336  2     BDB = .IRAB[IRB$L_LOCK_BDB];
;   275          0337  2     IRAB[IRB$L_LOCK_BDB] = 0;
;   276          0338  2     KEY = 0;
;   277          0339  2
```

```
;  278     0340  2     IF .IDX_DFN[IDX$B_KEYREF] NEQ 0
;  279     0341  2     THEN
;  280     0342  2         KEY = (.IDX_DFN[IDX$B_KEYREF] - 1) MOD 5;
;  281     0343  2
;  282     0344  2     KEY = .KEY*(KEY$C_BLN + KEY$C_SPARE);
;  283     0345  2     KEY = .KEY + .BDB[BDB$L_ADDR];
;  284     0346  2
;  285     0347  2     ! update level number and root vbn on the disk
;  286     0348  2     !
;  287     0349  2     KEY[KEY$B_ROOTLEV] = .KEY[KEY$B_ROOTLEV] + 1;
;  288     0350  2
;  289     0351  2     IF .KEY[KEY$B_ROOTLEV] EQL 0
;  290     0352  2     THEN
;  291     0353  3         BEGIN
;  292     0354  3         RM$RELEASE(0);
;  293     0355  3         RETURN RMSERR(TRE);
;  294     0356  2         END;
;  295     0357  2
;  296     0358  2     KEY[KEY$L_ROOTVBN] = .IRAB[IRB$L_VBN_LEFT];
;  297     0359  2
;  298     0360  2     ! recalculate checksum
;  299     0361  2     !
;  300     0362  2     RM$MAKSUM(.BDB[BDB$L_ADDR]);
;  301     0363  2     BDB[BDB$V_DRT] = 1;
;  302     0364  2
;  303     0365  2     ! force write of prologue
;  304     0366  2     !
;  305     0367  2     STATUS = RM$RELEASE(RLS$M_WRT_THRU);
;  306     0368  2
;  307     0369  2     IF NOT .STATUS
;  308     0370  2     THEN
;  309     0371  3         BEGIN
;  310     0372  3         IF .RAB [RAB$L_STV] EQL 0
;  311     0373  3         THEN
;  312     0374  3             RAB [RAB$L_STV] = .STATUS OR 1^16;
;  313     0375  3         RETURN RMSERR(WPL);
;  314     0376  2         END;
;  315     0377  2
;  316     0378  2     ! prologue has been updated successfully so now udpate in core index
;  317     0379  2     ! descriptor
;  318     0380  2     !
;  319     0381  2     IDX_DFN[IDX$B_ROOTLEV] = .IDX_DFN[IDX$B_ROOTLEV] + 1;
;  320     0382  2     IDX_DFN[IDX$L_ROOTVBN] = .IRAB[IRB$L_VBN_LEFT];
;  321     0383  2     RETURN RMSSUC(SUC);
;  322     0384  2
;  323     0385  1     END;


                                3C  BB 00000 RM$UPD_PLG::
                                                PUSHR   #^M<R2,R3,R4,R5>                ; 0283
              42   A9              08  88 00002          BISB2   #8, 66(IRAB)            ; 0331
              40   A9              01  90 00006          MOVB    #1, 64(IRAB)            ; 0332
                   7E        21    A7  9A 0000A          MOVZBL  33(IDX_DFN), -(SP)      ; 0334
                             0000G 30  0000E             BSBW    RM$KEY_DESC             ;
```

```
                          5E                04  C0 00011         ADDL2   #4, SP                              ;
                          7F                50  E9 00014         BLBC    STATUS, 5$                          ;
                          54      0084      C9  D0 00017         MOVL    132(IRAB), BDB                      ; 0336
                                  0084      C9  D4 0001C         CLRL    132(IRAB)                           ; 0337
                                            55  D4 00020         CLRL    KEY                                 ; 0338
                                    21      A7  95 00022         TSTB    33(IDX_DFN)                         ; 0340
                                            12  13 00025         BEQL    1$                                  ;
                          55        21      A7  9A 00027         MOVZBL  33(IDX_DFN), KEY                    ; 0342
        7E FFFFFFFF   8F  55                01  7A 0002B         EMUL    #1, KEY, #-1, -(SP)                 ;
        55            55  8E                05  7B 00034         EDIV    #5, (SP)+, KEY, KEY                 ;
                          55  00000066      8F  C4 00039 1$:     MULL2   #102, KEY                           ; 0344
                          55        18      A4  C0 00040         ADDL2   24(BDB), KEY                        ; 0345
                                    09      A5  96 00044         INCB    9(KEY)                              ; 0349
                                            0F  12 00047         BNEQ    2$                                  ; 0351
                                            53  D4 00049         CLRL    R3                                  ; 0354
                              00000000G     00  16 0004B         JSB     RM$RELEASE                          ;
                          50      86DC      8F  3C 00051         MOVZWL  #34524, R0                          ; 0355
                                            3E  11 00056         BRB     5$                                  ;
                   0C     A5      0088      C9  D0 00058 2$:     MOVL    136(IRAB), 12(KEY)                  ; 0358
                          55        18      A4  D0 0005E         MOVL    24(BDB), R5                         ; 0362
                                       0000G 30 00062            BSBW    RM$MAKSUM                           ;
                   0A     A4                02  88 00065         BISB2   #2, 10(BDB)                         ; 0363
                          53                02  D0 00069         MOVL    #2, R3                              ; 0367
                              00000000G     00  16 0006C         JSB     RM$RELEASE                          ;
                          15                50  E8 00072         BLBS    STATUS, 4$                          ; 0369
                                    0C      A8  D5 00075         TSTL    12(RAB)                             ; 0372
                                            09  12 00078         BNEQ    3$                                  ;
           0C     A8              50  00010000  8F  C9 0007A     BISL3   #65536, STATUS, 12(RAB)             ; 0374
                          50      C11C      8F  3C 00083 3$:     MOVZWL  #49436, R0                          ; 0375
                                            0C  11 00088         BRB     5$                                  ;
                                    15      A7  96 0008A 4$:     INCB    21(IDX_DFN)                         ; 0381
                   18     A7      0088      C9  D0 0008D         MOVL    136(IRAB), 24(IDX_DFN)              ; 0382
                          50                01  D0 00093         MOVL    #1, R0                              ; 0383
                                            3C  BA 00096 5$:     POPR    #^M<R2,R3,R4,R5>                    ; 0385
                                                05 00098         RSB                                         ;

; Routine Size:  153 bytes,    Routine Base:  RM$RMS3 + 007C


;  324          0386   1
;  325          0387   1 END
;  326          0388   1
;  327          0389   0 ELUDOM



                                  PSECT SUMMARY
;
;       Name                    Bytes                          Attributes
;
;   RM$RMS3                          277  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        72          2       154         00:00.3
```

```
;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RM3ROOT/OBJ=OBJ$:RM3ROOT MSRC$:RM3ROOT/UPDATE=(ENH$:RM3ROOT)

; Size:         277 code + 0 data bytes
; Run Time:       00:08.1
; Elapsed Time:   00:15.1
; Lines/CPU Min:   2892
; Lexemes/CPU-Min: 20802
; Memory Used:  84 pages
; Compilation Complete
```

RM3PROBE
LIS

RM3SIOXSP
LIS

RM3PUTERR
LIS

RM3PUTUPD
LIS

RM3SPLUDR
LIS

RM3RRV
LIS

RM3ROOT
LIS

RM3PUT
LIS