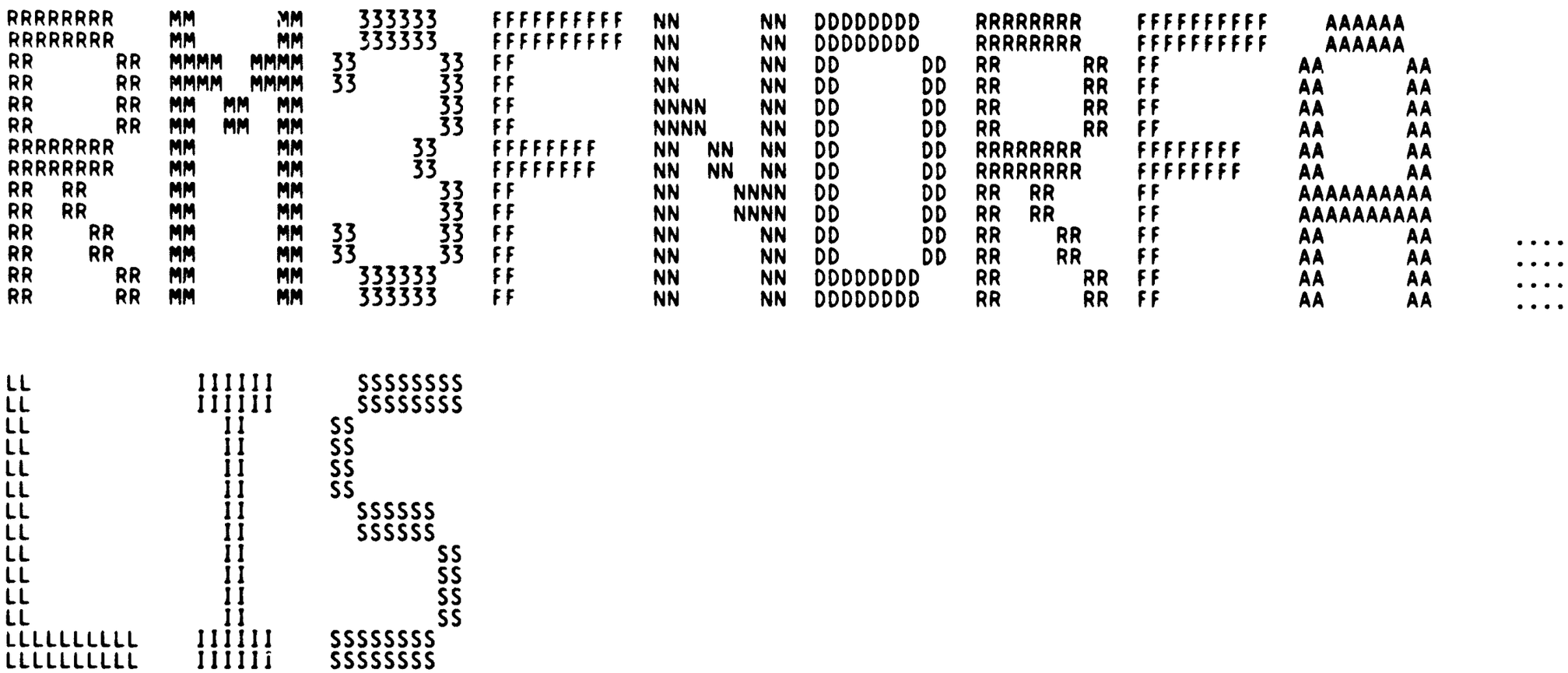
```
RRRRRRRRRRR      MMM          MMM      SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   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               SSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
```

FILEID**RM3FNDRFA



```
;  1          0001  0
;  2          0002  0 MODULE RM3FNDRFA (LANGUAGE (BLISS32) ,
;  3          0003  0                   IDENT = 'V04-000'
;  4          0004  0                   ) =
;  5          0005  1 BEGIN
;  6          0006  1 !
;  7          0007  1 !**********************************************************************
;  8          0008  1 !*                                                                    *
;  9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                            *
; 12          0012  1 !*                                                                    *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 18          0018  1 !*  TRANSFERRED.                                                       *
; 19          0019  1 !*                                                                    *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 22          0022  1 !*  CORPORATION.                                                       *
; 23          0023  1 !*                                                                    *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
; 26          0026  1 !*                                                                    *
; 27          0027  1 !*                                                                    *
; 28          0028  1 !**********************************************************************
; 29          0029  1
; 30          0030  1 !++
; 31          0031  1 !
; 32          0032  1 ! FACILITY:     RMS32 INDEX SEQUENTIAL FILE ORGANIZATION
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !                Search given bucket for record with given ID
; 36          0036  1 !
; 37          0037  1 !
; 38          0038  1 ! ENVIRONMENT:
; 39          0039  1 !
; 40          0040  1 !                VAX/VMS OPERATING SYSTEM
; 41          0041  1 !
; 42          0042  1 !--
; 43          0043  1 !
; 44          0044  1 !
; 45          0045  1 ! AUTHOR:        Christian Saether     CREATION DATE:          24-APR-78  14:46
; 46          0046  1 !
; 47          0047  1 ! MODIFIED BY:
; 48          0048  1 !
; 49          0049  1 !       V03-004 MCN0008         Maria del C. Nasr       22-Mar-1983
; 50          0050  1 !               More linkages reorganization
; 51          0051  1 !
; 52          0052  1 !       V03-003 MCN0007         Maria del C. Nasr       24-Feb-1983
; 53          0053  1 !               Reorganize linkages
; 54          0054  1 !
; 55          0055  1 !       V03-002 KBT0166         Keith B. Thompson       23-Aug-1982
; 56          0056  1 !               Reorganize psects
; 57          0057  1 !
```

```
;    58          0058  1 ;       V03-001 TMK0001         Todd M. Katz            19-Jun-1982
;    59          0059  1 ;               !mplement cluster RMS solution for next record positioning.
;    60          0060  1 ;               This means that it no longer makes sense to call the routines
;    61          0061  1 ;               in this module to position to a SIDR record because the IDs
;    62          0062  1 ;               in SIDR records are no longer used for positioning purposes.
;    63          0063  1 ;               Therefore, these routines will contain an implicit assumption
;    64          0064  1 ;               that the record being located lies in a primary data bucket.
;    65          0065  1 ;
;    66          0066  1 ;       V02-010 KPL0002         Peter Lieberwirth       10-Jan-1982
;    67          0067  1 ;               If file is PLG 3, and find_by_id fails, return DEL if
;    68          0068  1 ;               id is less than BKT next record id (instead of RNF).
;    69          0069  1 ;               This is because we're completely deleting RRVs in PLG
;    70          0070  1 ;               3 files.  (This should have been done some time ago.)
;    71          0071  1 ;
;    72          0072  1 ;       V02-009 KPL0001         Peter Lieberwirth       23-Nov-1981
;    73          0073  1 ;               If RFA points to an index bucket for a PLG 3 file, return
;    74          0074  1 ;               DEL and not RFA because more likely error is record deleted
;    75          0075  1 ;               and bucket then reclaimed.  (less likely error is RFA trashed
;    76          0076  1 ;               by user)
;    77          0077  1 ;
;    78          0078  1 ;       V02-008 PSK0002         Paulina S. Knibbe       20-Apr-1981
;    79          0079  1 ;               Change last 'record not compressed' info to keep track
;    80          0080  1 ;               of the address instead of the ID.
;    81          0081  1 ;
;    82          0082  1 ;       V02-007 PSK0001         Paulina S. Knibbe       15-Apr-1981
;    83          0083  1 ;               Make some changes to fix prologue one and two searches
;    84          0084  1 ;
;    85          0085  1 ;       V02-006 MCN0006         Maria del C. Nasr       16-Mar-1981
;    86          0086  1 ;               Increase size of record identifier to a word.
;    87          0087  1 ;               Modify RM$FIND_BY_ID to handle new data level structure
;    88          0088  1 ;               (base level 1).
;    89          0089  1 ;
;    90          0090  1 ;       V02-005 REFORMAT        K. E. Kinnear           23-Jul-1980     10:03
;    91          0091  1 ;
;    92          0092  1 ; REVISION HISTORY:
;    93          0093  1 ;
;    94          0094  1 ;       V01-004                 C. D. Saether           16-Jan-1979     10:11
;    95          0095  1 ;               FIND_BY_ID no longer requires CURBDB as input (calls
;    96          0096  1 ;               REC_OVHD instead of GETNEXTREC).
;    97          0097  1 ;
;    98          0098  1 ;       V01-003                 C. D. Saether           9-Jan-1979      11:59
;    99          0099  1 ;               NORLS_RNF must be cleard always (fix bug where it wasn't).
;   100          0100  1 ;
;   101          0101  1 ;       V01-002                 W. Koenig               24-Oct-1978     14:01
;   102          0102  1 ;               Make changes caused by sharing conventions.
;   103          0103  1 ;
;   104          0104  1 ;*****
;   105          0105  1
;   106          0106  1 LIBRARY 'RMSLIB:RMS';
;   107          0107  1
;   108          0108  1 REQUIRE 'RMSSRC:RMSIDXDEF';
;   109          0173  1
;   110          0174  1 ! Define default psects for code
;   111          0175  1 !
;   112          0176  1 PSECT
;   113          0177  1     CODE = RM$RMS3(PSECT_ATTR),
;   114          0178  1     PLIT = RM$RMS3(PSECT_ATTR);
```

```
;  115          0179  1
;  116          0180  1 ! Linkages
;  117          0181  1 !
;  118          0182  1
;  119          0183  1 LINKAGE
;  120          0184  1     L_RABREG_457,
;  121          0185  1     L_RABREG_567,
;  122          0186  1     L_RABREG_67,
;  123          0187  1     L_REC_OVHD,
;  124          0188  1     L_PRESERVE1;
;  125          0189  1
;  126          0190  1 ! External Routines
;  127          0191  1 !
;  128          0192  1
;  129          0193  1 EXTERNAL ROUTINE
;  130          0194  1     RM$GETBKT    : RL$RABREG_457,
;  131          0195  1     RM$REC_OVHD  : RL$REC_OVHD,
;  132          0196  1     RM$RLSBKT    : RL$_RESERVE1;
;  133          0197  1
```

```
;  135          0198  1 %SBTTL 'RM$FIN  BY_ID'
;  136          0199  1 GLOBAL ROUTINE RM$FIND_BY_ID : RL$RABREG_567 =
;  137          0200  1
;  138          0201  1 !++
;  139          0202  1 !
;  140          0203  1 ! FUNCTIONAL DESCRIPTION:
;  141          0204  1 !
;  142          0205  1 ! Searches a primary data bucket for a record with the given ID,
;  143          0206  1 ! returns REC_ADDR
;  144          0207  1 !
;  145          0208  1 ! CALLING SEQUENCE:
;  146          0209  1 !
;  147          0210  1 !       RM$FIND_BY_ID(,
;  148          0211  1 !
;  149          0212  1 ! INPUT PARAMETERS:
;  150          0213  1 !
;  151          0214  1 !       NONE
;  152          0215  1 !
;  153          0216  1 ! IMPLICIT INPUTS:
;  154          0217  1 !
;  155          0218  1 !       AP - ID of record to find
;  156          0219  1 !       BKT_ADDR - address of bucket to search in
;  157          0220  1 !       IFAB - for prologue version
;  158          0221  1 !       IDX_DFN - compression flags and key of reference
;  159          0222  1 !
;  160          0223  1 ! OUTPUT PARAMETERS:
;  161          0224  1 !
;  162          0225  1 !       NONE
;  163          0226  1 !
;  164          0227  1 ! IMPLICIT OUTPUTS:
;  165          0228  1 !
;  166          0229  1 !       REC_ADDR - address of record found
;  167          0230  1 !
;  168          0231  1 ! ROUTINE VALUE:
;  169          0232  1 !
;  170          0233  1 !       SUC if found, RNF or DEL otherwise
;  171          0234  1 !
;  172          0235  1 ! SIDE EFFECTS:
;  173          0236  1 !
;  174          0237  1 !       NONE
;  175          0238  1 !
;  176          0239  1 !--
;  177          0240  1
;  178          0241  2     BEGIN
;  179          0242  2
;  180          0243  2     BUILTIN
;  181          0244  2         AP;
;  182          0245  2
;  183          0246  2     EXTERN  REGISTER
;  184          0247  2         R_  _ADDR_STR,
;  185          0248  2         R_  C_ADDR_STR,
;  186          0249  2         R_IDX_DFN_STR,
;  187          0250  2         R_IFAB_STR,
;  188          0251  2         R_IRAB_STR;
;  189          0252  2
;  190          0253  2     LOCAL
;  191          0254  2         REC_SIZE,
```

```
;  192     0255  2          EOB;                            . local for end of bucket address
;  193     0256  2
;  194     0257  2      MAP
;  195     0258  2          AP      : WORD;
;  196     0259  2
;  197     0260  2      ! Calculate beginning and end of search addresses
;  198     0261  2      !
;  199     0262  2      REC_ADDR = .BKT_ADDR + BKT$C_OVERHDSZ;
;  200     0263  2      EOB = .BKT_ADDR + .BKT_ADDR[BKT$W_FREESPACE];
;  201     0264  2
;  202     0265  2      ! Keep track of last key with no front compression in order to
;  203     0266  2      ! optimize rebuilding the primary key when we find the right record
;  204     0267  2      !
;  205     0268  2      IRAB[IRB$L_LST_NCMP] = .REC_ADDR;
;  206     0269  2
;  207     0270  2      ! If not a prologue 3 file, process as before
;  208     0271  2      !
;  209     0272  2
;  210     0273  2      IF .IFAB[IFB$B_PLG_VER] LSSU PLG$C_VER_3
;  211     0274  2      THEN
;  212     0275  2
;  213     0276  2          ! Step thru the bucket until the record id is matcned
;  214     0277  2          !
;  215     0278  2          WHILE .REC_ADDR LSSA .EOB
;  216     0279  2          DO
;  217     0280  2
;  218     0281  2              IF .RFC_ADDR[IRC$B_ID] EQL .AP
;  219     0282  2
;  220     02f3  2              THEN
;  221     0284  2                  ! This is it! We found the record. Return success.
;  222     0285  2                  !
;  223     0286  3                  RETURN RMSSUC(SUC)
;  224     0287  3
;  225     0288  2              ELSE
;  226     0289  2                  ! Haven't found it yet, keep looking.
;  227     0290  2                  !
;  228     0291  3                  BEGIN
;  229     0292  3                  REC_ADDR = .REC_ADDR + RM$REC_OVHD(0; REC_SIZE);
;  230     0293  3                  REC_ADDR = .REC_ADDR + .REC_SIZE;
;  231     0294  3                  END
;  232     0295  2      ELSE
;  233     0296  2
;  234     0297  2          ! Prologue version 3 file, loop until record found
;  235     0298  2          !
;  236     0299  2
;  237     0300  2          WHILE .REC_ADDR LSSA .EOB
;  238     0301  2          DO
;  239     0302  2
;  240     0303  2              IF .REC_ADDR[IRC$W_ID] EQL .AP
;  241     0304  2              THEN
;  242     0305  3                  RETURN RMSSUC(SUC)              ! We founu the record
;  243     0306  3
;  244     0307  2              ELSE
;  245     0308  3                  BEGIN
;  246     0309  3
;  247     0310  3                  LOCAL
;  248     0311  3                      REC_OVHD;
```

```
;  249          0312  3
;  250          0313  3                    REC_OVHD = RM$REC_OVHD(0; REC_SIZE);
;  251          0314  3
;  252          0315  3                    ! If compressed primary key, save address of record with a zero
;  253          0316  3                    ! compression count
;  254          0317  3                    !
;  255          0318  3                    IF  .IDX_DFN[IDX$V_KEY_COMPR]
;  256          0319  3                      AND .(.REC_ADDR + .REC_OVHD + 1)<0,8> EQL 0
;  257          0320  3                    THEN
;  258          0321  3                        IRAB[IRB$L_LST_NCMP] = .REC_ADDR;
;  259          0322  3                    !
;  260          0323  3                    REC_ADDR = .REC_ADDR + .REC_OVHD + .REC_SIZE;
;  261          0324  2                    END;
;  262          0325  2
;  263          0326  2
;  264          0327  2        ! Can't find this ID at all. Return record not found code.
;  265          0328  2        !
;  266          0329  2        IF .IFAB[IFB$B_PLG_VER] LSSU PLG$C_VER_3
;  267          0330  2        THEN
;  268          0331  2
;  269          0332  2            ! Prolog 1 or 2 file, return RNF.
;  270          0333  2            !
;  271          0334  3            RETURN RMSERR(RNF)
;  272          0335  2        ELSE
;  273          0336  2
;  274          0337  2            ! Prolog 3 file, return DEL if RRV has been deleted
;  275          0338  2            !
;  276          0339  2            IF .AP GEQU .BKT_ADDR[ BKT$W_NXTRECID ]
;  277          0340  2            THEN
;  278          0341  2
;  279          0342  2                ! id being searched for is not less than next record id,
;  280          0343  2                ! so it can't correspond to a deleted RRV, this return RNF
;  281          0344  2                !
;  282          0345  3                RETURN RMSERR(RNF)
;  283          0346  2            ELSE
;  284          0347  2
;  285          0348  2                ! RRV has been deleted, therefore the record must have been deleted
;  286          0349  2                !
;  287          0350  2                RETURN RMSERR(DEL);
;  288          0351  2
;  289          0352  1        END;

                                                        .TITLE  RM3FNDRFA
                                                        .IDENT  \V04-000\

                                                        .EXTRN  RM$GETBKT, RM$REC_OVHD
                                                        .EXTRN  RM$RLSBKT

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

                                    0C   BB 00000 RM$FIND_BY_ID::
                                                        PUSHR   #^M<R2,R3>                      ; 0199
                          56        0E   A5  9E 00002         MOVAB   14(R5), REC_ADDR           ; 0262
                          52        04   A5  3C 00006         MOVZWL  4(BKT_ADDR), EOB           ; 0263
                          52             55  C0 0000A         ADDL2   BKT_ADDR, EOB              ;
                    0098  C9             56  D0 0000D         MOVL    REC_ADDR, 152(IRAB)        ; 0268
```

```
                                03     00B7   CA  91 00012          CMPB    183(IFAB), #3                   ; 0273
                                       1E   1E 00017              BGEQU   2$
                                52           56  D1 00019 1$:     CMPL    REC_ADDR, EOB                   ; 0278
                                             4A  1E 0001C         BGEQU   6$
                                50       01  A6  9A 0001E         MOVZBL  1(REC_ADDR), R0                 ; 0281
                                5C           50  B1 00022         CMPW    R0, AP
                                             1B  13 00025         BEQL    3$
                                             51  D4 00027         CLRL    R1                              ; 0292
                                         0000G   30 00029         BSBW    RM$REC_OVHD
                                53           51  D0 0002C         MOVL    R1, R3
                                56           50  C0 0002F         ADDL2   R0, REC_ADDR
                                56           53  C0 00032         ADDL2   REC_SIZE, REC_ADDR              ; 0293
                                             E2  11 00035         BRB     1$                              ; 0281
                                52           56  D1 00037 2$:     CMPL    REC_ADDR, EOB                   ; 0300
                                             2C  1E 0003A         BGEQU   6$
                                5C       01  A6  B1 0003C         CMPW    1(REC_ADDR), AP                 ; 0303
                                             05  12 00040         BNEQ    4$
                                50           01  D0 00042 3$:     MOVL    #1, R0                          ; 0305
                                             3A  11 00045         BRB     9$
                                             51  D4 00047 4$:     CLRL    R1                              ; 0313
                                         0000G   30 0004          BSBW    RM$REC_OVHD
                                53           51  D0 0004C         MOVL    R1, R3
                          0B        1C   A7      06  E1 0004F         BBC     #6, 28(IDX_DFN), 5$             ; 0318
                                       01 A046   95  00054         TSTB    1(REC_OVHD)[REC_ADDR]           ; 0319
                                             05  12 00058         BNEQ    5$
                                0098     C9      56  D0 0005A         MOVL    REC_ADDR, 152(IRAB)             ; 0321
                                50           56  C0 0005F 5$:     ADDL2   REC_ADDR, R0                    ; 0323
                                56           50  C1 00062         ADDL3   REC_SIZE, R0, REC_ADDR
                                             CF  11 00066         BRB     2$                              ; 0303
                                03     00B7   CA  91 00068 6$:     CMPB    183(IFAB), #3                   ; 0329
                                             06  1F 0006D         BLSSU   7$
                                   06        A5      5C  B1 0006F         CMPW    AP, 6(BKT_ADDR)                 ; 0339
                                             07  1F 00073         BLSSU   8$
                                50       82B2   8F  3C 00075 7$:     MOVZWL  #33458, R0                      ; 0345
                                             05  11 0007A         BRB     9$
                                50       8262   8F  3C 0007C 8$:     MOVZWL  #33378, R0                      ; 0350
                                             0C  BA 00081 9$:     POPR    #^M<R2,R3>                      ; 0352
                                                 05 00083         RSB

; Routine Size:  132 bytes,    Routine Base:  RM$RMS3 + 0000

;  290          0353  1
```

```
;  292     0354  1 %SBTTL 'RM$FIND_BY_RFA'
;  293     0355  1 GLOBAL ROUTINE RM$FIND_BY_RFA (VBN) : RL$RABREG_67 =
;  294     0356  1
;  295     0357  1 !++
;  296     0358  1 !
;  297     0359  1 ! FUNCTIONAL DESCRIPTION.
;  298     0360  1 !
;  299     0361  1 ! Get primary data bucket, search for record with matching ID,
;  300     0362  1 ! and return address of record found
;  301     0363  1 !
;  302     0364  1 ! CALLING SEQUENCE:
;  303     0365  1 !
;  304     0366  1 !       RM$FIND_BY_RFA (VBN)
;  305     0367  1 !
;  306     0368  1 ! INPUT PARAMETERS:
;  307     0369  1 !
;  308     0370  1 !       VBN - VBN of bucket to search
;  309     0371  1 !
;  310     0372  1 ! IMPLICIT INPUTS:
;  311     0373  1 !
;  312     0374  1 !       IDX_DFN - address of index descriptor
;  313     0375  1 !       IRAB - address of internal RAB
;  314     0376  1 !       IRAB [ NORLS_RNF ] - do release bucket on RNF error
;  315     0377  1 !       IRAB [ CACHEFLAGS ] - passed through to GETBKT
;  316     0378  1 !       AP - ID of record to search for
;  317     0379  1 !
;  318     0380  1 ! OUTPUT PARAMETERS:
;  319     0381  1 !
;  320     0382  1 !       NONE
;  321     0383  1 !
;  322     0384  1 ! IMPLICIT OUTPUTS:
;  323     0385  1 !
;  324     0386  1 !       REC_ADDR - address of record with ID match
;  325     0387  1 !       IRAB[IRB$L_CURBDB] = BDB of bucket searched if success
;  326     0388  1 !                          = 0 if any error
;  327     0389  1 !       IRAB[ NORLS_RNF ] = 0 always
;  328     0390  1 !
;  329     0391  1 ! ROUTINE VALUE:
;  330     0392  1 !
;  331     0393  1 !       SUC - if search successful
;  332     0394  1 !       RNF - if no match
;  333     0395  1 !       RFA - bad level (neg 0)
;  334     0396  1 !       error codes passed back if GETBKT error
;  335     0397  1 !
;  336     0398  1 ! SIDE EFFECTS:
;  337     0399  1 !
;  338     0400  1 !       Bucket searched is accessed if success,
;  339     0401  1 !       released on any error unless NORLS_RNF is set,
;  340     0402  1 !               then do not release on RNF error
;  341     0403  1 !
;  342     0404  1 !--
;  343     0405  1
;  344     0406  2     BEGIN
;  345     0407  2
;  346     0408  2     LOCAL
;  347     0409  2         ST;
;  348     0410  2
```

```
;  349          0411  2      BUILTIN
;  350          0412  2          AP;
;  351          0413  2
;  352          0414  2      EXTERNAL REGISTER
;  353          0415  2          R_REC_ADDR_STR,
;  354          0416  2          R_IDX_DFN_STR,
;  355          0417  2          COMMON_RAB_STR;
;  356          0418  2
;  357          0419  2      GLOBAL REGISTER
;  358          0420  2          COMMON_IO_STR;
;  359          0421  2
;  360          0422  2      LABEL
;  361          0423  2          BLOCK;
;  362          0424  2
;  363          0425  2      REC_ADDR = .AP;                                 ! save AP across IO
;  364          0426  2
;  365          0427  2        Get access to bucket, save status, and check for success.
;  366          0428  2      !
;  367          0429  2
;  368          0430  3      IF NOT (S. = RM$GETBKT(.VBN, .IDX_DFN[IDX$B_DATBKTSZ]*512))
;  369          0431  2      THEN
;  370          0432  3          BEGIN
;  371          0433  3          IRAB[IRB$L_CURBDB] = 0;
;  372          0434  3          IRAB[IRB$V_NORLS_RNF] = 0;
;  373          0435  3          RETURN .ST;
;  374          0436  3
;  375          0437  3          END;
;  376          0438  2
;  377          0439  2      IRAB[IRB$L_CURBDB] = .BDB;                      ! set up CURBDB
;  378          0440  2
;  379          0441  2      ! Got the bucket. Now make sure we're at level 0.
;  380          0442  2      !
;  381          0443  2      AP = .REC_ADDR;                                 ! restore AP
;  382          0444  2
;  383          0445  2  BLOCK :
;  384          0446  3      BEGIN
;  385          0447  3
;  386          0448  3      IF .BKT_ADDR[BKT$B_LEVEL] NEQ 0
;  387          0449  3      THEN
;  388          0450  3
;  389          0451  3          IF .IFAB[ IFB$B_PLG_VER ] LSSU PLG$C_VER_3
;  390          0452  3          THEN
;  391          0453  4              BEGIN
;  392          0454  4              ! PLG 2 file, return RFA error
;  393          0455  4              !
;  394          0456  4              ST = RMSERR(RFA);
;  395          0457  4              LEAVE BLOCK;
;  396          0458  4              END
;  397          0459  3          ELSE
;  398          0460  4              BEGIN
;  399          0461  4              ! PLG 3 file, more likely error is record deleted and then bucket
;  400          0462  4              ! reclaimed
;  401          0463  4              !
;  402          0464  4              ST = RMSERR(DEL);
;  403          0465  4              LEAVE BLOCK;
;  404          0466  3              END;
;  405          0467  3
```

```
;  406          0468  3     ! Bucket level OK. Now search for ID.
;  407          0469  3     !
;  408          0470  3
;  409          0471  4     IF (ST = RM$FIND_BY_ID())
;  410          0472  3     THEN
;  411          0473  4         (IRAB[IRB$V_NORLS_RNF] = 0;
;  412          0474  3         RETURN .ST);
;  413          0475  3
;  414          0476  3     IF .IRAB[IRB$V_NORLS_RNF]
;  415          0477  3     THEN
;  416          0478  4         (IRAB[IRB$V_NORLS_RNF] = 0;
;  417          0479  3         RETURN .ST);
;  418          0480  3
;  419          0481  2     END;                                    ! end of block BLOCK
;  420          0482  2
;  421          0483  2     ! Common error code - zero current BDB and return error code.
;  422          0484  2     ! Clear NORLS_RNF flag always.
;  423          0485  2     !
;  424          0486  2     RM$RLSBKT(0);
;  425          0487  2     IRAB[IRB$L_CURBDB] = 0;
;  426          0488  2     IRAB[IRB$V_NORLS_RNF] = 0;
;  427          0489  2     RETURN .ST;
;  428          0490  2
;  429          0491  1     END;

                        30  BB 00000 RM$FIND_BY_RFA::
                                    PUSHR   #^M<R4,R5>                          ; 0355
                    56      5C  D0 00002            MOVL    AP, REC_ADDR        ; 0425
                    50  17  A7  9A 00005            MOVZBL  23(IDX_DFN), R0     ; 0430
7E                  50      09  78 00009            ASHL    #9, R0, -(SP)       ;
                        10  AE  DD 0000D            PUSHL   VBN                 ;
                          0000G 30 00010            BSBW    RM$GETBKT           ;
                    5E          08  C0 00013        ADDL2   #8, SP              ;
                    51          50  D0 00016        MOVL    R0, ST              ;
                    37          51  E9 00019        BLBC    ST, 4$              ;
                20  A9          54  D0 0001C        MOVL    BDB, 32(IRAB)       ; 0439
                    5C          56  D0 00020        MOVL    REC_ADDR, AP        ; 0443
                          0C    A5  95 00023        TSTB    12(BKT_ADDR)        ; 0448
                                15  13 00026        BEQL    2$                  ;
                    03  00B7    CA  91 00028        CMPB    183(IFAB), #3       ; 0451
                                07  1E 0002D        BGEQU   1$                  ;
                    51  865C    8F  3C 00030        MOVZWL  #34396, ST          ; 0456
                                15  11 00034        BRB     3$                  ; 0457
                    51  8262    8F  3C 00036 1$:    MOVZWL  #33378, ST          ; 0464
                                0E  11 0003B        BRB     3$                  ; 0465
                          FF3C  30  00 0003D 2$:    BSBW    RM$FIND_BY_ID       ; 0471
                    51          50  D0 00040        MOVL    R0, ST              ;
                    10          51  E8 00043        BLBS    ST, 5$              ;
0B                  42  A9      05  E0 00046        BBS     #5, 66(IRAB), 5$    ; 0476
                                7E  D4 0004B 3$:    CLRL    -(SP)               ; 0486
                          0000G 30  00 0004D        BSBW    RM$RLSBKT           ;
                    5E          04  C0 00050        ADDL2   #4, SP              ;
                          20    A9  D4 00053 4$:    CLRL    32(IRAB)            ; 0487
```

```
                     42  A9          20  BA 00056 5$:    BICB2   #32, 66(IRAB)                    ; 0488
                         50          51  D0 0005A         MOVL    ST, R0                          ; 0489
                                     30  BA 0005D         POPR    #^M<R4,R5>                      ; 0491
                                         05 0005F         RSB                                     ;

; Routine Size:  96 bytes,    Routine Base:  RM$RMS3 + 0084

;  430          0492  1
;  431          0493  1 END
;  432          0494  1
;  433          0495  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  RM$RMS3                       228  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        46         1       154          00:00.3



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RM3FNDRFA/OBJ-OBJ$:RM3FNDRFA MSRC$:RM3FNDRFA/UPDATE=(ENH$:RM3FNDRFA)

; Size:         228 code + 0 data bytes
; Run Time:       00:07.1
; Elapsed Time:   00:21.7
; Lines/CPU Min:   4165
; Lexemes/CPU-Min: 14945
; Memory Used:  72 pages
; Compilation Complete
```

RM3MAKIDX
LIS

RM3FNDRRV
LIS

RM3FNDRFA
LIS

RM3GET
LIS

RM3IUDR
LIS

RM3JOURNL
LIS

RM3KEYDSC
LIS

RM3MISC
LIS

RM3MISPUT
LIS