```
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**RM3DISCON

RM
VO

```
RRRRRRRR    MM      MM    333333    DDDDDDDD     IIIIII      SSSSSSSS    CCCCCCCC    OOOOOO    NN      NN
RRRRRRRR    MM      MM    333333    DDDDDDDD     IIIIII      SSSSSSSS    CCCCCCCC    OOOOOO    NN      NN
RR      RR  MMMM  MMMM  33      33  DD      DD     II      SS          CC          OO      OO  NN      NN
RR      RR  MMMM  MMMM  33      33  DD      DD     II      SS          CC          OO      OO  NN      NN
RR      RR  MM  MM  MM        33    DD      DD     II      SS          CC          OO      OO  NNNN    NN
RR      RR  MM  MM  MM        33    DD      DD     II      SS          CC          OO      OO  NNNN    NN
RRRRRRRR    MM      MM      33      DD      DD     II        SSSSSS    CC          OO      OO  NN  NN  NN
RRRRRRRR    MM      MM      33      DD      DD     II        SSSSSS    CC          OO      OO  NN  NN  NN
RR  RR      MM      MM        33    DD      DD     II              SS  CC          OO      OO  NN    NNNN
RR  RR      MM      MM        33    DD      DD     II              SS  CC          OO      OO  NN    NNNN
RR    RR    MM      MM  33      33  DD      DD     II              SS  CC          OO      OO  NN      NN      ....
RR    RR    MM      MM  33      33  DD      DD     II              SS  CC          OO      OO  NN      NN      ....
RR      RR  MM      MM    333333    DDDDDDDD     IIIIII    SSSSSSSS      CCCCCCCC    OOOOOO    NN      NN      ....
RR      RR  MM      MM    333333    DDDDDDDD     IIIIII    SSSSSSSS      CCCCCCCC    OOOOOO    NN      NN      ....


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE RM3DISCON (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 !                INDEXED SPECIFIC CODE FOR $DISCONNECT
;    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:        Wendy Koenig     CREATION DATE:        10-APR-78  15:10
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! MODIFIED BY:
;    48          0048  1 !
;    49          0049  1 !       V03-010 MCN0005         Maria del C. Nasr       15-Mar-1983
;    50          0050  1 !               More linkages reorganization
;    51          0051  1 !
;    52          0052  1 !       V03-009 MCN0004         Maria del C. Nasr       24-Feb-1983
;    53          0053  1 !               Reorganize linkages.
;    54          0054  1 !
;    55          0055  1 !       V03-008 TMK0005         Todd M. Katz            22-Dec-1982
;    56          0056  1 !               If the file allowed update access, and is either a prologue 3
;    57          0057  1 !               file or defines alternate keys, then there is a new record
```

```
;    58          0058  1 !               buffer which also must now be returned. Its address is stored
;    59          0059  1 !               in IRB$L_OLDBUF, and the size of the buffer is the same as the
;    60          0060  1 !               size of IRB$L_RECBUF. In some cases I have increased the size
;    61          0061  1 !               of these buffers by 2 to allow for sufficient room to store the
;    62          0062  1 !               size of the record together with the record itself within the
;    63          0063  1 !               buffer.
;    64          0064  1 !
;    65          0065  1 !       V03-007 TMK0004         Todd M. Katz            06-Dec-1982
;    66          0066  1 !               If the file is a prologue 1 or 2 file, contains alternate keys,
;    67          0067  1 !               and is open for write access then there will be a record buffer
;    68          0068  1 !               to deallocate whose address is in IRB$L_RECBUF. This record
;    69          0069  1 !               buffer will be used to store the primary data record being
;    70          0070  1 !               deleted during a $DELETE. The use of this buffer has become
;    71          0071  1 !               necessary because of the changes made in the $DELETE algorithm.
;    72          0072  1 !               The user data record is now deleted before the alternate keys,
;    73          0073  1 !               instead of the other way around which is how it has been done in
;    74          0074  1 !               the past. Thus, the primary data record must be saved off to
;    75          0075  1 !               the side so that after the user data record has been deleted it
;    76          0076  1 !               is still available for the extraction of the alternate keys as
;    77          0077  1 !               part of their deletion. This record buffer is used to save the
;    78          0078  1 !               primary data record.
;    79          0079  1 !
;    80          0080  1 !       V03-006 TMK0003         Todd M. Katz            02-Dec-1982
;    81          0081  1 !               The number of keybuffers is now represented by the symbol
;    82          0082  1 !               IFB$C_KBUFNUM.
;    83          0083  1 !
;    84          0084  1 !       V03-005 TMK0002         Todd M. Katz            24-Nov-1982
;    85          0085  1 !               There are now seven keybuffers to be deallocated.
;    86          0086  1 !
;    87          0087  1 !       V03-004 KBT0337         Keith B. Thompson       10-Sep-1982
;    88          0088  1 !               Remove ref. to RM$RETSOSPC routine
;    89          0089  1 !
;    90          0090  1 !       V03-003 KBT0164         Keith B. Thompson       21-Aug-1982
;    91          0091  1 !               Reorganize psects
;    92          0092  1 !
;    93          0093  1 !       V03-002 TMK0001         Todd M. Katz            01-Jul-1982
;    94          0094  1 !               Implement the RMS cluster solution for next record positioning.
;    95          0095  1 !               This involves no longer deallocation the NRP cell because
;    96          0096  1 !               the next record positioning context is now kept locally in the
;    97          0097  1 !               IRAB and there isn't any NRP cell to deallocate. Also add the
;    98          0098  1 !               deallocation of a sixth keybuffer.
;    99          0099  1 !
;   100          0100  1 !       V03-001 LJA0007         Laurie Anderson         25-Mar-1982
;   101          0101  1 !               Change KBUFSZ to reference a macro when computing buffer
;   102          0102  1 !               size and make IFB$B_KBUFSZ a word, now: IFB$W_KBUFSZ.
;   103          0103  1 !
;   104          0104  1 !       V02-017 CDS0004         C Saether               10-Dec-1981
;   105          0105  1 !               Rename the psect yet again.
;   106          0106  1 !
;   107          0107  1 !       V02-016 CDS0003         C Saether               29-Sep-1981
;   108          0108  1 !               Deallocate 5 keybuffers always.
;   109          0109  1 !
;   110          0110  1 !       V02-015 PSK0002         Paulina S. Knibbe       08-Aug-1981
;   111          0111  1 !               Remove support for truncated index keys on prologue
;   112          0112  1 !               three files. (SPLCTX)
;   113          0113  1 !
;   114          0114  1 !       V02-014 KPL0001         Peter Lieberwirth       24-Jul-1981
```

```
;  115          0115  1 !               Rename the PSECT again to fix broken barches.
;  116          0116  1 !
;  117          0117  1 !       V02-013 CDS0002         C D Saether             16-Jul-1981
;  118          0118  1 !               Do not deallocate anything on indirect rundown.
;  119          0119  1 !
;  120          0120  1 !       V02-012 MCN0003         Maria del C. Nasr       19-Jun-1981
;  121          0121  1 !               Deallocate the fifth key buffer for prologue 3 files.
;  122          0122  1 !
;  123          0123  1 !               PSK0001         Paulina S. Knibbe       12-Jun-1981
;  124          0124  1 !               Dellocate the split context block in prologue three files
;  125          0125  1 !
;  126          0126  1 !               MCN0002         Maria del C. Nasr       15-May-1981
;  127          0127  1 !               Deallocate the fourth key buffer in prologue 3 files.
;  128          0128  1 !
;  129          0129  1 !               MCN0001         Maria del C. Nasr       20-Apr-1981
;  130          0130  1 !               Add code to deallocate record buffer for prologue 3 files.
;  131          0131  1 !
;  132          0132  1 !       V03-011 CDS0001         C D Saether             6-Feb-1981      23:57
;  133          0133  1 !               Rename psect.
;  134          0134  1 !
;  135          0135  1 !       V02-010 R_FORMAT        K. E. Kinnear           23-Jul-1980     10:06
;  136          0136  1 !
;  137          0137  1 ! REVISION HISTORY:
;  138          0138  1 !
;  139          0139  1 !       V01-009                 C. D. Saether           1-Nov-1979      6:00
;  140          0140  1 !               RM$DISCONNOM returns lock BDB and BCB now.
;  141          0141  1 !
;  142          0142  1 !       V01-008                 W. Koenig               6-Feb-1979      17:22
;  143          0143  1 !               Decrement EXTRABUF IF BCNT GTR 2.
;  144          0144  1 !
;  145          0145  1 !       V01-007                 W. Koenig               22-Jan-1979     13:09
;  146          0146  1 !               Deallocate one lock BDB per stream, not per file.
;  147          0147  1 !               (Was needed for UPDATE RRV.)
;  148          0148  1 !
;  149          0149  1 !       V01-006                 W. Koenig               20-Nov-1978     11:19
;  150          0150  1 !               More changes for sharing.
;  151          0151  1 !
;  152          0152  1 !       V01-005                 W. Koenig               13-Nov-1978     13:36
;  153          0153  1 !               Sharing implementation changes.
;  154          0154  1 !
;  155          0155  1 !       V01-004                 W. Koenig               3-Nov-1978      9:12
;  156          0156  1 !               Size of UPDBUF is NUM_KEYS, instead of UBUFSZ.
;  157          0157  1 !
;  158          0158  1 !       V01-003                 W. Koenig               24-Oct-1978     14:01
;  159          0159  1 !               Make changes caused by sharing conventions.
;  160          0160  1 !
;  161          0161  1 !       V01-002                 W. Koenig               24-Jul-1978     14:00
;  162          0162  1 !               Deallocate 3rd key buffer used to store primary key on put.
;  163          0163  1 !
;  164          0164  1 !*****
;  165          0165  1
;  166          0166  1 LIBRARY 'RMSLIB:RMS';
;  167          0167  1
;  168          0168  1 REQUIRE 'RMSSRC:RMSIDXDEF';
;  169          0233  1
;  170          0234  1 ! define default psects for code
;  171          0235  1 !
```

```
;  172          0236  1 PSECT
;  173          0237  1     CODE = RM$RMS3(PSECT_ATTR),
;  174          0238  1     PLIT = RM$RMS3(PSECT_ATTR);
;  175          0239  1
;  176          0240  1 ! Linkages
;  177          0241  1 !
;  178          0242  1
;  179          0243  1 LINKAGE
;  180          0244  1     L_RABREG,
;  181          0245  1     L_RABREG_7,
;  182          0246  1     L_RETSPC;
;  183          0247  1
;  184          0248  1
;  185          0249  1 ! External Routines
;  186          0250  1 !
;  187          0251  1 EXTERNAL ROUTINE
;  188          0252  1     RM$KEY_DESC : RL$RABREG_7,
;  189          0253  1     RM$RETSPC1  : RL$RETSPC;
;  190          0254  1
```

```
;  192          0255  1 %SBTTL 'RM$DISCONNECT3B'
;  193          0256  1 GLOBAL ROUTINE RM$DISCONNECT3B : RL$RABREG =
;  194          0257  1
;  195          0258  1 !++
;  196          0259  1 !
;  197          0260  1 ! FUNCTIONAL DESCRIPTION:
;  198          0261  1 !
;  199          0262  1 !  This module performs the following functions required for
;  200          0263  1 !       disconnecting indexed files:
;  201          0264  1 !
;  202          0265  1 !    1 -- return key buffers and update buffer
;  203          0266  1 !    2 -- return old record buffer
;  204          0267  1 !    3 -- return record output buffer
;  205          0268  1 !
;  206          0269  1 ! CALLING SEQUENCE:
;  207          0270  1 !
;  208          0271  1 !       BSBW    RM$DISCONNECT3B
;  209          0272  1 !        Entered via case branch from RMS$DISCONNECT or RMS$CLOSE and
;  210          0273  1 !        BSBW from RM$DISCONNECT3
;  211          0274  1 !
;  212          0275  1 !       Returns to RM$DISCONNECT3
;  213          0276  1 !        and then branches to RM$DISCOMMON to finish up.
;  214          0277  1 !
;  215          0278  1 ! INPUT PARAMETERS:
;  216          0279  1 !
;  217          0280  1 !       NONE
;  218          0281  1 !
;  219          0282  1 ! IMPLICIT INPUTS:
;  220          0283  1 !
;  221          0284  1 !       R8 -- RAB address
;  222          0285  1 !       R9 -- IRAB address
;  223          0286  1 !       R10 -- IFAB address
;  224          0287  1 !       R11 -- IMPURE AREA address
;  225          0288  1 !
;  226          0289  1 ! OUTPUT PARAMETERS:
;  227          0290  1 !
;  228          0291  1 !       NONE
;  229          0292  1 !
;  230          0293  1 ! IMPLICIT OUTPUTS:
;  231          0294  1 !
;  232          0295  1 !       NONE
;  233          0296  1 !
;  234          0297  1 ! ROUTINE VALUE:
;  235          0298  1 !
;  236          0299  1 !       Usual RMS status codes.
;  237          0300  1 !
;  238          0301  1 ! SIDE EFFECTS:
;  239          0302  1 !
;  240          0303  1 !       Returns:        Key buffer
;  241          0304  1 !                       Old record buffer
;  242          0305  1 !                       Update buffer
;  243          0306  1 !                       BCB & the lock BDB if they existed
;  244          0307  1 !
;  245          0308  1 !--
;  246          0309  1
;  247          0310  2     BEGIN
;  248          0311  2
```

```
;  249          0312  2      EXTERNAL REGISTER
;  250          0313  2          COMMON_RAB_STR;
;  251          0314  2
;  252          0315  2      GLOBAL REGISTER
;  253          0316  2          R_IDX_DFN_STR;
;  254          0317  2
;  255          0318  2      ! If block I/O then return successfully, nothing needs doing.
;  256          0319  2      ! Also do nothing on indirect rundown.
;  257          0320  2      !
;  258          0321  2      IF .IFAB[IFB$V_BIO] OR .IFAB[IFB$V_PPF_IMAGE]
;  259          0322  2      THEN
;  260          0323  2          RETURN RMSSUC();
;  261          0324  2
;  262          0325  2      ! Decrement the number of EXTRABUFS if we had extra buffers.
;  263          0326  2      !
;  264          0327  2      IF .IRAB[IRB$B_BCNT] GTR 2
;  265          0328  2      THEN
;  266          0329  2          IFAB[IFB$B_EXTRABUF] = .IFAB[IFB$B_EXTRABUF] - (.IRAB[IRB$B_BCNT] - 2);
;  267          0330  2
;  268          0331  2      ! Return key buffer, ignore errors
;  269          0332  2      !
;  270          0333  3      BEGIN
;  271          0334  3
;  272          0335  3      LOCAL
;  273          0336  3          SIZE,
;  274          0337  3          BUFFER_SIZE;
;  275          0338  3
;  276          0339  3      SIZE = .IFAB[IFB$W_KBUFSZ] * IFB$C_KBUFNUM;
;  277          0340  3
;  278          0341  3      IF .SIZE LSS 12
;  279          0342  3      THEN
;  280          0343  3          SIZE = 12;
;  281          0344  3
;  282          0345  3      RM$RETSPC1 (.SIZE, 0, KEYBUF_ADDR(1));
;  283          0346  3
;  284          0347  3      ! Return update buffer ignoring errors
;  285          0348  3      !
;  286          0349  3      IF .IFAB[IFB$V_UPD]
;  287          0350  3      THEN
;  288          0351  4          BEGIN
;  289          0352  4          SIZE = .IFAB[IFB$B_NUM_KEYS];
;  290          0353  4
;  291          0354  4          IF .SIZE LSS 12
;  292          0355  4          THEN
;  293          0356  4              SIZE = 12;
;  294          0357  4
;  295          0358  4          RM$RETSPC1(.SIZE, 0, .IRAB[IRB$L_UPDBUF]);
;  296          0359  3          END;
;  297          0360  3
;  298          0361  3      ! Determine the size of a record buffer, in case there is one to be
;  299          0362  3      ! returned. Make sure to include the two two bytes reserved for containing
;  300          0363  3      ! the record size.
;  301          0364  3      !
;  302          0365  3      RM$KEY_DESC(0);
;  303          0366  3
;  304          0367  3      IF .IFAB[IFB$W_MRS] EQLU 0
;  305          0368  3      THEN
```

```
;  306          0369  3          BUFFER_SIZE = .IDX_DFN[IDX$B_DATBKTSZ] * 512
;  307          0370  3      ELSE
;  308          0371  3          IF .IFAB[IFB$B_PLG_VER] GEQU PLG$C_VER_3
;  309          0372  3          THEN
;  310          0373  3              BUFFER_SIZE = .IFAB[IFB$W_MRS] + IRC$C_KEYCMPOVH
;  311          0374  3                                            + IRC$C_DATCMPOVH
;  312          0375  3                                            + 2
;  313          0376  3          ELSE
;  314          0377  3              BUFFER_SIZE = .IFAB[IFB$W_MRS] + 2;
;  315          0378  3
;  316          0379  3      IF .BUFFER_SIZE LSS 12
;  317          0380  3      THEN
;  318          0381  3          BUFFER_SIZE = 12;
;  319          0382  3
;  320          0383  3      ! Return record buffer if prologue 3 file, or if prologue 1 or 2 file,
;  321          0384  3      ! secondary keys are defined, and the file is write accessed. Ignore all
;  322          0385  3      ! errors.
;  323          0386  3      !
;  324          0387  4      IF  (.IFAB[IFB$B_PLG_VER] EQLU PLG$C_VER_3)
;  325          0388  3       OR
;  326          0389  4          (.IFAB[IFB$B_NUM_KEYS] GTRU 1
;  327          0390  4                  AND
;  328          0391  4                 .IFAB[IFB$V_WRTACC])
;  329          0392  3      THEN
;  330          0393  4          BEGIN
;  331          0394  4          RM$RETSPC1 (.BUFFER_SIZE, 0, .IRAB[IRB$L_RECBUF]);
;  332          0395  4
;  333          0396  4          ! Return the old record buffer, ignoring all errors, if the file allows
;  334          0397  4          ! update access and is either a prologue 3 file or defines alternate
;  335          0398  4          ! keys.
;  336          0399  4          !
;  337          0400  4          IF .IFAB[IFB$V_UPD]
;  338          0401  4          THEN
;  339          0402  4              RM$RETSPC1(.BUFFER_SIZE, 0, .IRAB[IRB$L_OLDBUF]);
;  340          0403  3          END;
;  341          0404  2      END;
;  342          0405  3      RETURN RMSSUC()
;  343          0406  3
;  344          0407  1      END;


                                                .TITLE  RM3DISCON
                                                .IDENT  \V04-000\

                                                .EXTRN  RM$KEY_DESC, RM$RETSPC1

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

                                00BC    8F  BB 00000 RM$DISCONNECT3B::
                                                PUSHR   #^M<R2,R3,R4,R5,R7>             ; 0256
                      03    22  AA          05  E1 00004         BBC     #5, 34(IFAB), 2$    ; 0321
                                            00B7  31 00009 1$:      BRW     12$             ;
                      F8    04  AA          02  E0 0000C 2$:      BBS     #2, 4(IFAB), 1$     ;
                            02    54        A9  91 00011          CMPB    84(IRAB), #2        ; 0327
                                            12  1B 00015          BLEQU   3$              ;
                            50  00B6        CA  9A 00017          MOVZBL  182(IFAB), R0       ; 0329
                            51    54        A9  9A 0001C          MOVZBL  84(IRAB), R1        ;
```

```
                50              51  C2 00020          SUBL2   R1, R0                                  ;
              00B6  CA          02  81 00023          ADDB3   #2, R0, 182(IFAB)                       ;
                57      00B4    CA  3C 00029 3$:      MOVZWL  180(IFAB), SIZE                         ; 0339
                57              06  C4 0002E          MULL2   #6, SIZE                                ;
                0C              57  D1 00031          CMPL    SIZE, #12                               ; 0341
                                03  18 00034          BGEQ    4$                                      ;
                57              0C  D0 00036          MOVL    #12, SIZE                               ; 0343
                54        60    A9  D0 00039 4$:      MOVL    96(IRAB), R4                            ; 0345
                                53  D4 0003D          CLRL    R3                                      ;
                52              57  D0 0003F          MOVL    SIZE, R2                                ;
                          0000G 30  00042             BSBW    RM$RETSPC1                              ;
              19        22    AA  03  E1 00045          BBC     #3, 34(IFAB), 6$                        ; 0349
                57      00B2    CA  9A 0004A          MOVZBL  178(IFAB), SIZE                         ; 0352
                0C              57  D1 0004F          CMPL    SIZE, #12                               ; 0354
                                03  18 00052          BGEQ    5$                                      ;
                57              0C  D0 00054          MOVL    #12, SIZE                               ; 0356
                54        64    A9  D0 00057 5$:      MOVL    100(IRAB), R4                           ; 0358
                                53  D4 0005B          CLRL    R3                                      ;
                52              57  D0 0005D          MOVL    SIZE, R2                                ;
                          0000G 30  00060             BSBW    RM$RETSPC1                              ;
                                7E  D4 00063 6$:      CLRL    -(SP)                                   ; 0365
                          0000G 30  00065             BSBW    RM$KEY_DESC                             ;
                5E              04  C0 00068          ADDL2   #4, SP                                  ;
                50        60    AA  3C 0006B          MOVZWL  96(IFAB), R0                            ; 0367
                                0A  12 0006F          BNEQ    7$                                      ;
                57        17    A7  9A 00071          MOVZBL  23(IDX_DFN), BUFFER_SIZE                ; 0369
              57                57  09 78 00075          ASHL    #9, BUFFER_SIZE, BUFFER_SIZE            ;
                                11  11 00079          BRB     9$                                      ;
                03      00B7    CA  91 0007B 7$:      CMPB    183(IFAB), #3                           ; 0371
                                06  1F 00080          BLSSU   8$                                      ;
                57        07    A0  9E 00082          MOVAB   7(R0), BUFFER_SIZE                      ; 0375
                                04  11 00086          BRB     9$                                      ; 0373
                57        02    A0  9E 00088 8$:      MOVAB   2(R0), BUFFER_SIZE                      ; 0377
                0C              57  D1 0008C 9$:      CMPL    BUFFER_SIZE, #12                        ; 0379
                                03  18 0008F          BGEQ    10$                                     ;
                57              0C  D0 00091          MOVL    #12, BUFFER_SIZE                        ; 0381
                03      00B7    CA  91 00094 10$:     CMPB    183(IFAB), #3                           ; 0387
                                0B  13 00099          BEQL    11$                                     ;
                01      00B2    CA  91 0009B          CMPB    178(IFAB), #1                           ; 0389
                                21  1B 000A0          BLEQU   12$                                     ;
                1D        06    AA  E9 000A2          BLBC    6(IFAB), 12$                            ; 0391
                54        68    A9  D0 000A6 11$:     MOVL    104(IRAB), R4                           ; 0394
                                53  D4 000AA          CLRL    R3                                      ;
                52              57  D0 000AC          MOVL    BUFFER_SIZE, R2                         ;
                          0000G 30  000AF             BSBW    RM$RETSPC1                              ;
              0C        22    AA  03  E1 000B2          BBC     #3, 34(IFAB), 12$                       ; 0400
                54        6C    A9  D0 000B7          MOVL    108(IRAB), R4                           ; 0402
                                53  D4 000BB          CLRL    R3                                      ;
                52              57  D0 000BD          MOVL    BUFFER_SIZE, R2                         ;
                          0000G 30  000C0             BSBW    RM$RETSPC1                              ;
                50              01  D0 000C3 12$:     MOVL    #1, R0                                  ; 0405
                      00BC      8F  BA 000C6          POPR    #^M<R2,R3,R4,R5,R7>                     ; 0407
                                    05 000CA          RSB                                             ;

; Routine Size:  203 bytes,    Routine Base:  RM$RMS3 + 0000
```

```
;  345          0408  1
;  346          0409  1 END
;  347          0410  1
;  348          0411  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;      Name                      Bytes                          Attributes
;
;  RM$RMS3                         203  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         38         1          154        00:00.4
```

```
;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RM3DISCON/OBJ=OBJ$:RM3DISCON MSRC$:RM3DISCON/UPDATE=(ENH$:RM3DISCON)

; Size:          203 code + 0 data bytes
; Run Time:          00:06.1
; Elapsed Time:      00:17.6
; Lines/CPU Min:      4042
; Lexemes/CPU-Min:  13711
; Memory Used:  83 pages
; Compilation Complete
```

RM3FACE
LIS

RM3DISCON
LIS

RM3CONN
LIS

RM3DELETE
LIS

RM3CREATE
LIS