```
FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFFFFFFF.FFF            111              111              XXX
FFFFFFFFFFFF            111              111              XXX
FFFFFFFFFFFF            111              111              XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                  111111111        111111111     XXX         XXX
FFF                  111111111        111111111     XXX         XXX
FFF                  111111111        111111111     XXX         XXX
```

```
_$25

Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

```
MM      MM    AAAAAA    PPPPPPPP    VV      VV  BBBBBBBB    NN      NN
MM      MM    AAAAAA    PPPPPPPP    VV      VV  BBBBBBBB    NN      NN
MMMM  MMMM  AA      AA  PP      PP  VV      VV  BB      BB  NN      NN
MMMM  MMMM  AA      AA  PP      PP  VV      VV  BB      BB  NN      NN
MM  MM  MM  AA      AA  PP      PP  VV      VV  BB      BB  NNNN    NN
MM  MM  MM  AA      AA  PP      PP  VV      VV  BB      BB  NNNN    NN
MM      MM  AA      AA  PPPPPPPP    VV      VV  BBBBBBBB    NN  NN  NN
MM      MM  AA      AA  PPPPPPPP    VV      VV  BBBBBBBB    NN  NN  NN
MM      MM  AAAAAAAAAA  PP          VV      VV  BB      BB  NN    NNNN
MM      MM  AAAAAAAAAA  PP          VV      VV  BB      BB  NN    NNNN
MM      MM  AA      AA  PP            VV  VV    BB      BB  NN      NN      ....
MM      MM  AA      AA  PP            VV  VV    BB      BB  NN      NN      ....
MM      MM  AA      AA  PP              VV      BBBBBBBB    NN      NN      ....
MM      MM  AA      AA  PP              VV      BBBBBBBB    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 MAPVBN (
;     2          0002  0                 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 !*                                                                           *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                      *
;    13          0013  1 !*                                                                           *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    19          0019  1 !*  TRANSFERRED.                                                              *
;    20          0020  1 !*                                                                           *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    23          0023  1 !*  CORPORATION.                                                              *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*****************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 1
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine maps the specified virtual blocks to their
;    38          0038  1 !       corresponding logical blocks using the supplied window.
;    39          0039  1 !       The window is turned if necessary.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       STARLET operating system, including privileged system services
;    44          0044  1 !       and internal exec routines.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  3-Mar-1977  12:20
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-006 CDS0005         Christian D. Saether    20-Aug-1984
;    54          0054  1 !               Modify test for no lock.
;    55          0055  1 !
;    56          0056  1 !       V03-005 CDS0004         Christian D. Saether    14-Aug-1984
;    57          0057  1 !               Modify handling of fcb rebuilding.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-004 CDS0003         Christian D. Saether    25-Apr-1984
;    60          0060  1 !               Use longword addressing on some routines.
;    61          0061  1 !
;    62          0062  1 !       V03-003 CDS0002         Christian D. Saether    30-Dec-1983
;    63          0063  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    64          0064  1 !
;    65          0065  1 !       V03-002 CDS0001         Christian D. Saether     2-Feb-1983
;    66          0066  1 !               Changes for distributed file system.  Don't believe
;    67          0067  1 !               FCB$L_FILESIZE anymore, always check the header.
;    68          0068  1 !
;    69          0069  1 !       V03-001 ACG0297         Andrew C. Goldstein,    5-Aug-1982  18:26
;    70          0070  1 !               Fix maintenance of UCB context in updating cathedral windows
;    71          0071  1 !
;    72          0072  1 !       V02-004 ACG0229         Andrew C. Goldstein,    23-Dec-1981  21:08
;    73          0073  1 !               Move updating of PMS$GL_TURN from TURN_WINDOW
;    74          0074  1 !
;    75          0075  1 !       V02-003 LMP0003         L. Mark Pilant,          9-Dec-1981  14:07
;    76          0076  1 !               Added support for cathedral windows.
;    77          0077  1 !
;    78          0078  1 !       V02-002 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
;    79          0079  1 !               Previous revision history moved to F11A.REV
;    80          0080  1 !**
;    81          0081  1
;    82          0082  1
;    83          0083  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    84          0084  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    86          1075  1 GLOBAL ROUTINE MAP_VBN (VBN, WINDOW, BLOCK_COUNT, UNMAPPED_BLOCKS) : L_NORM =
;    87          1076  1
;    88          1077  1 !++
;    89          1078  1 !
;    90          1079  1 ! FUNCTIONAL DESCRIPTION:
;    91          1080  1 !
;    92          1081  1 !       This routine maps the specified virtual blocks to their
;    93          1082  1 !       corresponding logical blocks using the supplied window.
;    94          1083  1 !       the window is turned if necessary.
;    95          1084  1 !
;    96          1085  1 ! CALLING SEQUENCE:
;    97          1086  1 !       MAP_VBN (ARG1, ARG2, ARG3, ARG4)
;    98          1087  1 !
;    99          1088  1 ! INPUT PARAMETERS:
;   100          1089  1 !       ARG1: desired VBN
;   101          1090  1 !       ARG2: address of window to use
;   102          1091  1 !       ARG3: number of blocks to map
;   103          1092  1 !               if not present, 1
;   104          1093  1 !
;   105          1094  1 ! IMPLICIT INPUTS:
;   106          1095  1 !       CURRENT_VCB: address of VCB in process
;   107          1096  1 !       CURRENT_UCB: address of UCB in process
;   108          1097  1 !
;   109          1098  1 ! OUTPUT PARAMETERS:
;   110          1099  1 !       ARG4: if present, addres to store number of unmapped blocks
;   111          1100  1 !
;   112          1101  1 ! IMPLICIT OUTPUTS:
;   113          1102  1 !       NONE
;   114          1103  1 !
;   115          1104  1 ! ROUTINE VALUE:
;   116          1105  1 !       starting LBN or -1 if no map
;   117          1106  1 !
;   118          1107  1 ! SIDE EFFECTS:
;   119          1108  1 !       window may be turned, header may be read, volume may be switched
;   120          1109  1 !
;   121          1110  1 !--
;   122          1111  1
;   123          1112  2 BEGIN
;   124          1113  2
;   125          1114  2 MAP
;   126          1115  2         WINDOW          : REF BBLOCK;
;   127          1116  2
;   128          1117  2 LOCAL
;   129          1118  2         COUNT,                                  ! number of blocks to map
;   130          1119  2         UNMAPPED,                               ! address to store unmapped block count
;   131          1120  2         DUMMY,                                  ! place for above by default
;   132          1121  2         UCB             : REF BBLOCK,           ! address of mapping UCB
;   133          1122  2         FCB             : REF BBLOCK,           ! address of FCB of file
;   134          1123  2         HEADER          : REF BBLOCK,           ! address of file header
;   135          1124  2         LBN;                                    ! resulting LBN of map
;   136          1125  2
;   137          1126  2 EXTERNAL
;   138          1127  2         CLU$GL_CLUB     : ADDRESSING_MODE (GENERAL),
;   139          1128  2         PMS$GL_TURN     : ADDRESSING_MODE (ABSOLUTE);
;   140          1129  2                                                 ! system count of window turns
;   141          1130  2
;   142          1131  2 BIND_COMMON;
```

```
;  143          1132  2
;  144          1133  2 EXTERNAL ROUTINE
;  145          1134  2         REBLD_PRIM_FCB  : L_NORM NOVALUE, ! rebuild a primary fcb from header
;  146          1135  2         BUILD_EXT_FCBS  : L_NORM NOVALUE, ! build extension fcb chain,
;  147          1136  2         SWITCH_VOLUME   : L_NORM,         ! switch context to specified volume
;  148          1137  2         MAP_WINDOW      : L_NORM,         ! scan window map
;  149          1138  2         READ_HEADER     : L_NORM,         ! read file header
;  150          1139  2         TURN_WINDOW     : L_NORM ADDRESSING_MODE (GENERAL), ! turn window
;  151          1140  2         REMAP_FILE      : L_NORM;         ! remap the file into segmented windows
;  152          1141  2
;  153          1142  2
;  154          1143  2 ! Check the VBN for legality - i.e., non-zero
;  155          1144  2 !
;  156          1145  2
;  157          1146  2 FCB = .WINDOW[WCB$L_FCB];
;  158          1147  2
;  159          1148  2 IF .VBN EQL 0
;  160          1149  2 THEN
;  161          1150  2     RETURN -1;
;  162          1151  2
;  163          1152  2 IF .VBN GTRU .FCB [FCB$L_FILESIZE]
;  164          1153  2 THEN
;  165          1154  3     BEGIN
;  166          1155  3
;  167          1156  3     IF .FCB [FCB$B_ACCLKMODE] NEQ 0
;  168          1157  3     THEN
;  169          1158  4         BEGIN
;  170          1159  4         IF NOT .FCB [FCB$V_STALE]
;  171          1160  4         THEN
;  172          1161  4             RETURN -1;
;  173          1162  4         END
;  174          1163  3     ELSE
;  175          1164  3         IF NOT .BBLOCK [CURRENT_UCB [UCB$L_DEVCHAR2], DEV$V_CLU]
;  176          1165  3             OR .CLU$GL_CLUB EQL 0
;  177          1166  3         THEN
;  178          1167  3             RETURN -1;
;  179          1168  3
;  180          1169  3 ! Either the FCB has been marked stale, or this is a nolock access (which
;  181          1170  3 ! means the fcb is always suspect because it cannot be marked stale),
;  182          1171  3 ! so rebuild the fcb and extension fcb chain, if there is one.
;  183          1172  3 !
;  184          1173  3
;  185          1174  3     HEADER = READ_HEADER (0, .FCB);
;  186          1175  3
;  187          1176  3     REBLD_PRIM_FCB (.FCB, .HEADER);
;  188          1177  3
;  189          1178  3     IF .HEADER [FH2$W_EX_FIDNUM] NEQ 0
;  190          1179  3         OR .HEADER [FH2$B_EX_FIDNMX] NEQ 0
;  191          1180  3     THEN
;  192          1181  3         BUILD_EXT_FCBS (.HEADER);
;  193          1182  3
;  194          1183  2     END;
;  195          1184  2
;  196          1185  2 ! If an extension was done on a file which was completely mapped, and more
;  197          1186  2 ! than one user was accessing it, it is necessary to remap the file to get
;  198          1187  2 ! all the blocks correctly mapped.
;  199          1188  2 !
```

```
;  200          1189  2
;  201          1190  2 IF .WINDOW[WCB$V_CATHEDRAL] AND NOT .WINDOW[WCB$V_COMPLETE]
;  202          1191  2 THEN REMAP_FILE ();
;  203          1192  2
;  204          1193  2 ! Make the filesize test again, in case we did a reconstruction of the
;  205          1194  2 ! chain above.  This allows the window to be remapped in that case, if
;  206          1195  2 ! necessary.
;  207          1196  2 !
;  208          1197  2
;  209          1198  2 IF .VBN GTRU .FCB [FCB$L_FILESIZE]
;  210          1199  2 THEN
;  211          1200  2     RETURN -1;
;  212          1201  2
;  213          1202  2 ! If the file is multi-header, scan the extension FCB's for the one
;  214          1203  2 ! containing the desired VBN. The right FCB is identified by noting that
;  215          1204  2 ! there are no more, or that the start VBN of the next one is greater than
;  216          1205  2 ! the desired VBN.
;  217          1206  2 !
;  218          1207  2
;  219          1208  2 UNTIL
;  220          1209  3     (IF .FCB[FCB$L_EXFCB] EQL 0 THEN 1
;  221          1210  3      ELSE .BBLOCK [.FCB[FCB$L_EXFCB], FCB$L_STVBN] GTRU .VBN
;  222          1211  3     )
;  223          1212  2 DO FCB = .FCB[FCB$L_EXFCB];
;  224          1213  2
;  225          1214  2 ! If chasing extension FCB's took us to another volume, switch the context to
;  226          1215  2 ! that volume.
;  227          1216  2 !
;  228          1217  2
;  229          1218  2 SWITCH_VOLUME (.FCB[FCB$W_FID_RVN]);
;  230          1219  2
;  231          1220  2 ! Default the optional arguments.
;  232          1221  2 !
;  233          1222  2
;  234          1223  3 COUNT = (IF ACTUALCOUNT GEQ 3
;  235          1224  3         THEN .BLOCK_COUNT
;  236          1225  3         ELSE 1
;  237          1226  2         );
;  238          1227  3 UNMAPPED = (IF ACTUALCOUNT GEQ 4
;  239          1228  3         THEN .UNMAPPED_BLOCKS
;  240          1229  3         ELSE DUMMY
;  241          1230  2         );
;  242          1231  2
;  243          1232  2 ! Attempt to map the transfer with the existing window. If the map fails
;  244          1233  2 ! completely, turn the window and try once more. When any blocks map,
;  245          1234  2 ! return the relevant data.
;  246          1235  2 !
;  247          1236  2
;  248          1237  2 DECR I FROM 2 TO 1 DO
;  249          1238  3     BEGIN
;  250          1239  3
;  251          1240  3     LBN = KERNEL_CALL (MAP_WINDOW, .VBN, .WINDOW, .COUNT, .UNMAPPED, UCB);
;  252          1241  3     IF .LBN NEQ -1 THEN EXITLOOP;
;  253          1242  3
;  254          1243  3     PMS$GL_TURN = .PMS$GL_TURN + 1;      ! count window turn in PMS data base
;  255          1244  3     HEADER = READ_HEADER (0, .FCB);
;  256          1245  3     KERNEL_CALL (TURN_WINDOW, .WINDOW, .HEADER, .VBN, .FCB[FCB$L_STVBN]);
```

```
;  257          1246  3
;  258          1247  2      END;
;  259          1248  2
;  260          1249  2 IF .UCB NEQ .CURRENT_UCB
;  261          1250  2 THEN BUG_CHECK (BADRVNWCB, FATAL, 'Inconsistent RVN in window map pointer');
;  262          1251  2 RETURN .LBN;
;  263          1252  2
;  264          1253  1 END;                                    ! end of routine MAP_VBN
```

```
                                                        .TITLE  MAPVBN
                                                        .IDENT  \V04-000\

                                                        .EXTRN  CLU$GL_CLUB, PMS$GL_TURN
                                                        .EXTRN  REBLD_PRIM_FCB, BUILD_EXT_FCBS
                                                        .EXTRN  SWITCH_VOLUME, MAP_WINDOW
                                                        .EXTRN  READ_HEADER, TURN_WINDOW
                                                        .EXTRN  REMAP_FILE, BUG$_BADRVNWCB

                                                        .PSECT  $CODE$,NOWRT,2

                                    00FC 00000          .ENTRY  MAP_VBN, Save R2,R3,R4,R5,R6,R7                 ; 1075
                                5E          08  C2 00002  SUBL2   #8, SP                                      ;
                                50      08  AC  D0 00005  MOVL    WINDOW, R0                                  ; 1146
                                52      18  A0  D0 00009  MOVL    24(R0), FCB                                 ;
                                        04  AC  D5 0000D  TSTL    VBN                                         ; 1148
                                            60  13 00010  BEQL    6$                                          ;
                            38  A2      04  AC  D1 00012  CMPL    VBN, 56(FCB)                                ; 1152
                                            3F  1B 00017  BLEQU   4$                                          ;
                                        0B  A2  95 00019  TSTB    11(FCB)                                     ; 1156
                                            06  13 0001C  BEQL    1$                                          ;
                                    12      23  A2  E8 0001E  BLBS    35(FCB), 2$                             ; 1159
                                            4E  11 00022  BRB     6$                                          ; 1161
                                    50      94  AA  D0 00024  1$:   MOVL    -108(BASE), R0                      ; 1164
                                    46      3C  A0  E9 00028  BLBC    60(R0), 6$                              ;
                                00000000G       00  D5 0002C  TSTL    CLU$GL_CLUB                             ; 1165
                                            3E  13 00032  BEQL    6$                                          ;
                                            52  DD 00034  2$:   PUSHL   FCB                                   ; 1174
                                            7E  D4 00036  CLRL    -(SP)                                       ;
                            0000G   CF          02  FB 00038  CALLS   #2, READ_HEADER                         ;
                                    53          50  D0 0003D  MOVL    R0, HEADER                              ;
                                                0C  BB 00040  PUSHR   #^M<R2,R3>                              ; 1176
                            0000G   CF          02  FB 00042  CALLS   #2, REBLD_PRIM_FCB                      ;
                                        0E  A3  B5 00047  TSTW    14(HEADER)                                  ; 1178
                                            05  12 0004A  BNEQ    3$                                          ;
                                        13  A3  95 0004C  TSTB    19(HEADER)                                  ; 1179
                                            07  13 0004F  BEQL    4$                                          ;
                                            53  DD 00051  3$:   PUSHL   HEADER                               ; 1181
                            0000G   CF          01  FB 00053  CALLS   #1, BUILD_EXT_FCBS                      ;
                                    50      08  AC  D0 00058  4$:   MOVL    WINDOW, R0                          ; 1190
                0A      0B  A0              06  E1 0005C  BBC     #6, 11(R0), 5$                              ;
                05      0B  A0              05  E0 00061  BBS     #5, 11(R0), 5$                              ;
                        0000G   CF          00  FB 00066  CALLS   #0, REMAP_FILE                              ; 1191
                          38    A2      04  AC  D1 0006B  5$:   CMPL    VBN, 56(FCB)                        ; 1198
                                            04  1B 00070  BLEQU   7$                                          ;
                                    50          01  CE 00072  6$:   MNEGL   #1, R0                             ; 1200
                                                04 00075  RET                                                 ;
```

```
                50        0C    A2  DO 00076 7$:     MOVL    12(FCB), R0                                 ; 1209
                                  0C  13 0007A        BEQL    8$
              04    AC    2C    A0  D1 0007C        CMPL    44(R0), VBN                                 ; 1210
                                  05  1A 00081        BGTRU   8$
                52                50  DO 00083        MOVL    R0, FCB                                     ; 1212
                                  EE  11 00086        BRB     7$
                7E        28    A2  3C 00088 8$:     MOVZWL  40(FCB), -(SP)                              ; 1218
         0000G  CF                01  FB 0008C        CALLS   #1, SWITCH_VOLUME
                03                6C  91 00091        CMPB    (AP), #3                                    ; 1223
                                  06  1F 00094        BLSSU   9$
                56        0C      AC  DO 00096        MOVL    BLOCK_COUNT, COUNT                          ; 1224
                                  03  11 0009A        BRB     10$
                56                01  DO 0009C 9$:     MOVL    #1, COUNT                                   ; 1223
                04                6C  91 0009F 10$:    CMPB    (AP), #4                                    ; 1227
                                  06  1F 000A2        BLSSU   11$
                55        10      AC  DO 000A4        MOVL    UNMAPPED_BLOCKS, UNMAPPED                   ; 1228
                                  03  11 000A8        BRB     12$
                55                6E  9E 000AA 11$:    MOVAB   DUMMY, UNMAPPED                             ; 1227
                54                02  DO 000AD 12$:    MOVL    #2, I                                       ; 1237
                          04      AE  9F 000B0 13$:    PUSHAB  UCB                                         ; 1240
                                  55  DD 000B3        PUSHL   UNMAPPED
                                  56  DD 000B5        PUSHL   COUNT
                7E        04      AC  7D 000B7        MOVQ    VBN, -(SP)
         0000G  CF                05  FB 000BB        CALLS   #5, MAP_WINDOW
                57                50  DO 000C0        MOVL    R0, LBN
        FFFFFFFF    8F            57  D1 000C3        CMPL    LBN, #-1                                    ; 1241
                                  27  12 000CA        BNEQ    14$
                     00000000G    9F  D6 000CC        INCL    @#PMS$GL_TURN                               ; 1243
                                  52  DD 000D2        PUSHL   FCB                                         ; 1244
                                  7E  D4 000D4        CLRL    -(SP)
                         0000G    CF  02  FB 000D6        CALLS   #2, READ_HEADER
                                  53  50  DO 000DB        MOVL    R0, HEADER
                                    2C    A2  DD 000DE        PUSHL   44(FCB)                             ; 1245
                                    04    AC  DD 000E1        PUSHL   VBN
                                  53  DD 000E4        PUSHL   HEADER
                                    08    AC  DD 000E6        PUSHL   WINDOW
                      00000000G   00  04  FB 000E9        CALLS   #4, TURN_WINDOW
                                  BD  54  F5 000F0        SOBGTR  I, 13$                                  ; 1237
                       94   AA        04  AE  D1 000F3 14$:    CMPL    UCB, -108(BASE)                     ; 1249
                                      04  13 000F8        BEQL    15$
                                        FEFF 000FA        BUGW                                            ; 1250
                                       0000* 000FC        .WORD   <BUG$_BADRVNWCB!4>
                                  50  57  DO 000FE 15$:    MOVL    LBN, R0                                 ; 1251
                                      04 00101        RET                                                 ; 1253

; Routine Size:  258 bytes,    Routine Base:  $CODE$ + 0000

;  265          1254  1
;  266          1255  1 END
;  267          1256  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                          258  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                 Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619        30         0      1000          00:01.9




;                                    COMMAND QUALIFIERS

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

; Size:         258 code + 0 data bytes
; Run Time:        00:18.7
; Elapsed Time:    00:51.7
; Lines/CPU Min:    4021
; Lexemes/CPU-Min: 47055
; Memory Used:  237 pages
; Compilation Complete
```

PMS
LIS

QUOTAUTIL
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

PARSNM
LIS

MAPVBN
LIS

IODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS