```
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
```

FILEID**BADSCN

```
BBBBBBBB       AAAAAA    DDDDDDDD      SSSSSSSS    CCCCCCCC  NN       NN
BBBBBBBB       AAAAAA    DDDDDDDD      SSSSSSSS    CCCCCCCC  NN       NN
BB       BB   AA      AA  DD      DD  SS          CC          NN       NN
BB       BB   AA      AA  DD      DD  SS          CC          NN       NN
BB       BB   AA      AA  DD      DD  SS          CC          NNNN     NN
BB       BB   AA      AA  DD      DD  SS          CC          NNNN     NN
BBBBBBBB      AA      AA  DD      DD    SSSSSS    CC          NN  NN   NN
BBBBBBBB      AA      AA  DD      DD    SSSSSS    CC          NN  NN   NN
BB       BB   AAAAAAAAAA  DD      DD          SS  CC          NN    NNNN
BB       BB   AAAAAAAAAA  DD      DD          SS  CC          NN    NNNN
BB       BB   AA      AA  DD      DD          SS  CC          NN      NN      ....
BB       BB   AA      AA  DD      DD          SS  CC          NN      NN      ....
BBBBBBBB      AA      AA  DDDDDDDD    SSSSSSSS      CCCCCCCC  NN      NN      ....
BBBBBBBB      AA      AA  DDDDDDDD    SSSSSSSS      CCCCCCCC  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 BADSCN (
;     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 2
;    34        0034  1 !
;    35        0035  1 ! ABSTRACT:
;    36        0036  1 !
;    37        0037  1 !       This routine scans the pending bad block list and enters or removes
;    38        0038  1 !       entries.
;    39        0039  1 !
;    40        0040  1 ! ENVIRONMENT:
;    41        0041  1 !
;    42        0042  1 !       STARLET operating system, including privileged system services
;    43        0043  1 !       and internal exec routines.
;    44        0044  1 !
;    45        0045  1 !--
;    46        0046  1 !
;    47        0047  1 !
;    48        0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  22-May-1978  10:33
;    49        0049  1 !
;    50        0050  1 ! MODIFIED BY:
;    51        0051  1 !
;    52        0052  1 !       V03-003 CDS0001         Christian D. Saether    29-Dec-1983
;    53        0053  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    54        0054  1 !
;    55        0055  1 !       V03-002 ACG0367         Andrew C. Goldstein,    26-Oct-1983  19:48
;    56        0056  1 !               Update highwater mark in BADLOG.SYS when extending
;    57        0057  1 !
```

```
;    58          0058  1 !       V3-001  STJ0312         Steven T. Jeffreys,      2-Jun-1982
;    59          0059  1 !                               Fix addressing mode to fix out-or-range reference.
;    60          0060  1 !
;    61          0061  1 !       B0101   ACG0121         Andrew C. Goldstein,    16-Jan-1980  21:36
;    62          0062  1 !                               Make context save and restore into subroutines
;    63          0063  1 !
;    64          0064  1 !       B0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  19:59
;    65          0065  1 !       Previous revision history moved to [F11B.SRC]F11B.REV
;    66          0066  1 !**
;    67          0067  1
;    68          0068  1
;    69          0069  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    70          0070  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    72          1061  1 GLOBAL ROUTINE SCAN_BADLOG (FID, BASE_VBN, BASE_LBN, MODE, BLOCK_COUNT) : L_NORM NOVALUE =
;    73          1062  1
;    74          1063  1 !++
;    75          1064  1 !
;    76          1065  1 ! FUNCTIONAL DESCRIPTION:
;    77          1066  1 !
;    78          1067  1 !       This routine scans the volume bad block log for the specified block(s)
;    79          1068  1 !       ande enters or removes them, depending on the mode.
;    80          1069  1 !
;    81          1070  1 !
;    82          1071  1 ! CALLING SEQUENCE:
;    83          1072  1 !       SCAN_BADLOG (ARG1, ARG2, ARG3, ARG4, ARG5)
;    84          1073  1 !
;    85          1074  1 ! INPUT PARAMETERS:
;    86          1075  1 !       ARG1: file ID of file containing bad block
;    87          1076  1 !       ARG2: VBN in file of bad block
;    88          1077  1 !       ARG3: LBN of bad block
;    89          1078  1 !       ARG4: mode of operation
;    90          1079  1 !               REMOVE_BADBLOCK = 0
;    91          1080  1 !               ENTER_READERR   = 1
;    92          1081  1 !               ENTER_WRITERR   = 2
;    93          1082  1 !       ARG5: count of blocks to process (remove only)
;    94          1083  1 !
;    95          1084  1 ! IMPLICIT INPUTS:
;    96          1085  1 !       NONE
;    97          1086  1 !
;    98          1087  1 ! OUTPUT PARAMETERS:
;    99          1088  1 !       NONE
;   100          1089  1 !
;   101          1090  1 ! IMPLICIT OUTPUTS:
;   102          1091  1 !       NONE
;   103          1092  1 !
;   104          1093  1 ! ROUTINE VALUE:
;   105          1094  1 !       NONE
;   106          1095  1 !
;   107          1096  1 ! SIDE EFFECTS:
;   108          1097  1 !       volume bad block list file altered
;   109          1098  1 !
;   110          1099  1 !--
;   111          1100  1
;   112          1101  2 BEGIN
;   113          1102  2
;   114          1103  2 MAP
;   115          1104  2         FID             : REF BBLOCK;   ! file ID argument
;   116          1105  2
;   117          1106  2 LABEL
;   118          1107  2         SEARCH_LOOP;                    ! bad block list search
;   119          1108  2
;   120          1109  2 LOCAL
;   121          1110  2         VBN,                            ! VBN of current block of file
;   122          1111  2         LBN,                            ! LBN of current block
;   123          1112  2         FIRST_FREE,                     ! VBN containing first free slot
;   124          1113  2         FREE_OFFSET,                    ! byte offset in block of free slot
;   125          1114  2         P               : REF BBLOCK,   ! record pointer
;   126          1115  2         FIB             : REF BBLOCK,   ! local pointer to secondary FIB
;   127          1116  2         WINDOW          : REF BBLOCK,   ! address of bad block file window
;   128          1117  2         FCB             : REF BBLOCK,   ! address of bad block file FCB
```

```
;  129          1113  2          HEADER          : REF BBLOCK;   . address of bad block file header
;  130          1119  2
;  131          1120  2 BIND_COMMON;
;  132          1121  2
;  133          1122  2 EXTERNAL ROUTINE
;  134          1123  2          SAVE_CONTEXT    : L_NORM,       ! save reentrant context area
;  135          1124  2          RESTORE_CONTEXT : L_NORM,       ! restore reentrant context area
;  136          1125  2          OPEN_FILE       : L_NORM,       ! open a data file
;  137          1126  2          READ_DATA       : L_NORM,       ! read a file block
;  138          1127  2          CLOSE_FILE      : L_NORM,       ! close the data file
;  139          1128  2          CREATE_BLOCK    : L_NORM,       ! fabricate a block buffer
;  140          1129  2          MARK_DIRTY      : L_NORM,       ! mark buffer dirty
;  141          1130  2          READ_HEADER     : L_NORM,       ! read file header
;  142          1131  2          EXTEND          : L_NORM,       ! extend a file
;  143          1132  2          CHECKSUM        : L_NORM,       ! checksum file header
;  144          1133  2          WRITE_HEADER    : L_NORM,       ! write a file header
;  145          1134  2          MAP_VBN         : L_NORM;       ! map virtual to logical
;  146          1135  2
;  147          1136  2
;  148          1137  2 ! Check the reserved file count to see if there is a bad block list file.
;  149          1138  2 ! If not, forget the whole thing. If there is, set up secondary context and
;  150          1139  2 ! open the file.
;  151          1140  2 !
;  152          1141  2
;  153          1142  2 IF .CURRENT_VCB[VCB$B_RESFILES] LSSU BADLOG_FID THEN RETURN;
;  154          1143  2
;  155          1144  2 SAVE_CONTEXT ();
;  156          1145  2 FIB = SECOND_FIB;
;  157          1146  2 FIB[FIB$W_FID_NUM] = BADLOG_FID;
;  158          1147  2 FIB[FIB$W_FID_SEQ] = BADLOG_FID;
;  159          1148  2
;  160          1149  2 WINDOW = OPEN_FILE (FIB[FIB$W_FID], 1);
;  161          1150  2
;  162          1151  2 ! Scan the pending bad block file for a match on the given LBN (in remove mode,
;  163          1152  2 ! scan for the given range). Also look for the first available free space.
;  164          1153  2 !
;  165          1154  2
;  166          1155  2 FIRST_FREE = 0;
;  167          1156  2 VBN = 0;
;  168          1157  2
;  169          1158  3 SEARCH_LOOP: BEGIN
;  170          1159  3 WHILE 1 DO
;  171          1160  4     BEGIN
;  172          1161  4     VBN = .VBN + 1;
;  173          1162  4     P = READ_DATA (.WINDOW, .VBN, 1);
;  174          1163  4     IF .P EQL 0 THEN EXITLOOP;
;  175          1164  4
;  176          1165  4     INCR J FROM 0 TO 512/PBB$C_LENGTH - 1
;  177          1166  4     DO
;  178          1167  5         BEGIN
;  179          1168  5         IF .MODE EQL REMOVE_BADBLOCK
;  180          1169  5         THEN
;  181          1170  6             BEGIN
;  182          1171  6             IF  .P[PBB$L_LBN] GEQU .BASE_LBN
;  183          1172  6             AND .P[PBB$L_LBN] LSSU .BASE_LBN + .BLOCK_COUNT
;  184          1173  6             THEN
;  185          1174  7                 BEGIN
```

```
;  186          1175  7                  CH$FILL (0, PBB$C_LENGTH, .P);
;  187          1176  7                  MARK_DIRTY (.P);
;  188          1177  6                  END;
;  189          1178  6              END
;  190          1179  5          ELSE
;  191          1180  6              BEGIN
;  192          1181  6              IF .P[PBB$B_FLAGS] EQL 0
;  193          1182  6              AND .FIRST_FREE EQL 0
;  194          1183  6              THEN
;  195          1184  7                  BEGIN
;  196          1185  7                  FIRST_FREE = .VBN;
;  197          1186  7                  FREE_OFFSET = .J * PBB$C_LENGTH;
;  198          1187  6                  END;
;  199          1188  6              IF .P[PBB$L_LBN] EQL .BASE_LBN
;  200          1189  6              THEN
;  201          1190  7                  BEGIN
;  202          1191  7                  IF .P[PBB$B_COUNT] LSSU 255
;  203          1192  7                  THEN P[PBB$B_COUNT] = .P[PBB$B_COUNT] + 1;
;  204          1193  7                  IF .MODE EQL ENTER_READERR
;  205          1194  7                  THEN P[PBB$V_READERR] = 1
;  206          1195  7                  ELSE P[PBB$V_WRITERR] = 1;
;  207          1196  7                  MARK_DIRTY (.P);
;  208          1197  7                  LEAVE SEARCH_LOOP;
;  209          1198  6                  END;
;  210          1199  5              END;
;  211          1200  5
;  212          1201  5          P = .P + PBB$C_LENGTH;
;  213          1202  4          END;                                ! end of loop within block
;  214          1203  3      END;                                    ! end of loop scanning blocks
;  215          1204  3
;  216          1205  3 ! We get here if we fail to match on the block (or were scanning to remove).
;  217          1206  3 ! On a remove, we are now done. For an enter, take the first free slot found.
;  218          1207  3 ! If there was none, we have to extend the file.
;  219          1208  3 !
;  220          1209  3
;  221          1210  3 IF .MODE EQL REMOVE_BADBLOCK THEN LEAVE SEARCH_LOOP;
;  222          1211  3
;  223          1212  3 IF .FIRST_FREE EQL 0
;  224          1213  3 THEN
;  225          1214  4     BEGIN
;  226          1215  4     FCB = .WINDOW[WCB$L_FCB];
;  227          1216  4     HEADER = READ_HEADER (0, .FCB);
;  228          1217  4
;  229          1218  4     IF .FCB[FCB$L_EFBLK] GEQU .FCB[FCB$L_FILESIZE]
;  230          1219  4     THEN
;  231          1220  5         BEGIN
;  232          1221  5         FIB[FIB$L_EXSZ] = 1;
;  233          1222  5         FIB[FIB$V_NOHDREXT] = 1;
;  234          1223  5         EXTEND (.FIB, .HEADER);
;  235          1224  4         END;
;  236          1225  4
;  237          1226  4     BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_EFBLKL] =
;  238          1227  4                 .BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_EFBLKL] + 1;
;  239          1228  4
;  240          1229  4 ! If this file header supports it, stuff the high water field to
;  241          1230  4 ! be the allocated size.
;  242          1231  4 !
```

```
;  243          1232  4
;  244          1233  4      IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
;  245          1234  4      THEN
;  246          1235  4          HEADER[FH2$L_HIGHWATER] = .BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_EFBLKL];
;  247          1236  4
;  248          1237  4      CHECKSUM (.HEADER);
;  249          1238  4      WRITE_HEADER ();
;  250          1239  4
;  251          1240  4      LBN = MAP_VBN (.VBN, .WINDOW);
;  252          1241  4      IF .LBN EQL -1
;  253          1242  4      THEN BUG_CHECK (HDRNOTMAP, FATAL, 'Block just created not mapped');
;  254          1243  4      P = CREATE_BLOCK (.LBN, 1, DATA_TYPE);
;  255          1244  4      FREE_OFFSET = 0;
;  256          1245  4      END
;  257          1246  4
;  258          1247  3  ELSE
;  259          1248  3      P = READ_DATA (.WINDOW, .FIRST_FREE, 1);
;  260          1249  3
;  261          1250  3  ! Now build the new bad block list entry.
;  262          1251  3  !
;  263          1252  3
;  264          1253  3  P = .P + .FREE_OFFSET;
;  265          1254  3
;  266          1255  3  CH$COPY (FID$C_LENGTH, .FID, 0, PBB$C_LENGTH, P[PBB$W_FID]);
;  267          1256  3  P[PBB$L_VBN] = .BASE_VBN;
;  268          1257  3  P[PBB$L_LBN] = .BASE_LBN;
;  269          1258  3  P[PBB$B_COUNT] = .P[PBB$B_COUNT] + 1;
;  270          1259  3  IF .MODE EQL ENTER_READERR
;  271          1260  3  THEN P[PBB$V_READERR] = 1
;  272          1261  3  ELSE P[PBB$V_WRITERR] = 1;
;  273          1262  3
;  274          1263  3  MARK_DIRTY (.P);
;  275          1264  3
;  276          1265  2  END;                                   ! end of block SEARCH_LOOP
;  277          1266  2
;  278          1267  2  CLOSE_FILE (.WINDOW);                  ! close the bad block list file
;  279          1268  2  RESTORE_CONTEXT ();
;  280          1269  2
;  281          1270  1  END;                                   ! end of routine SCAN_BADLOG

                                                .TITLE  BADSCN
                                                .IDENT  \V04-000\

                                                .EXTRN  SAVE_CONTEXT, RESTORE_CONTEXT
                                                .EXTRN  OPEN_FILE, READ_DATA
                                                .EXTRN  CLOSE_FILE, CREATE_BLOCK
                                                .EXTRN  MARK_DIRTY, READ_HEADER
                                                .EXTRN  EXTEND, CHECKSUM
                                                .EXTRN  WRITE_HEADER, MAP_VBN
                                                .EXTRN  BUG$_HDRNOTMAP

                                                .PSECT  $CODE$,NOWRT,2

                                  0BFC 00000    .ENTRY  SCAN_BADLOG, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1061
                                                        R11                                         ;
                              5E        08  C2 00002    SUBL2   #8, SP                              ;
```

```
                        50        98    AA  D0 00005          MOVL    -104(BASE), R0                                  ; 1142
                        09        4F    A0  91 00009          CMPB    79(R0), #9                                      ;
                                        01  1E 0000D          BGEQU   1$                                              ;
                                            04 0000F          RET                                                     ;
                  0000G CF                  00  FB 00010 1$:      CALLS   #0, SAVE_CONTEXT                            ; 1144
                        57        0244      CA  9E 00015          MOVAB   580(BASE), FIB                              ; 1145
                     04 A7 00090009         8F  D0 0001A          MOVL    #589833, 4(FIB)                             ; 1146
                                            01  DD 00022          PUSHL   #1                                          ; 1149
                              04            A7  9F 00024          PUSHAB  4(FIB)                                      ;
                  0000G CF                  02  FB 00027          CALLS   #2, OPEN_FILE                               ;
                        5B                  50  D0 0002C          MOVL    R0, WINDOW                                  ;
                                            59  D4 0002F          CLRL    FIRST_FREE                                  ; 1155
                                            6E  D4 00031          CLRL    VBN                                         ; 1156
                                            6E  D6 00033 2$:      INCL    VBN                                         ; 1161
                                            01  DD 00035          PUSHL   #1                                          ; 1162
                              04            AE  DD 00037          PUSHL   VBN                                         ;
                                            5B  DD 0003A          PUSHL   WINDOW                                      ;
                  0000G CF                  03  FB 0003C          CALLS   #3, READ_DATA                               ;
                        56                  50  D0 00041          MOVL    R0, P                                       ;
                                            57  13 00044          BEQL    8$                                          ; 1163
                                            58  D4 00046          CLRL    J                                           ; 1165
                              10            AC  D5 00048 3$:      TSTL    MODE                                        ; 1168
                                            22  12 0004B          BNEQ    4$                                          ;
                     0C       AC       0C    A6  D1 0004D          CMPL    12(P), BASE_LBN                             ; 1171
                                            40  1F 00052          BLSSU   7$                                          ;
                  50 0C       AC       14    AC  C1 00054          ADDL3   BLOCK_COUNT, BASE_LBN, R0                   ; 1172
                              50        0C    A6  D1 0005A          CMPL    12(P), R0                                   ;
                                            34  1E 0005E          BGEQU   7$                                          ;
                10 00                 6E        00  2C 00060          MOVC5   #0, (SP), #0, #16, (P)                      ; 1175
                                                66  00065                                                             ;
                                            56  DD 00066          PUSHL   P                                           ; 1176
                  0000G CF                  01  FB 00068          CALLS   #1, MARK_DIRTY                              ;
                                            25  11 0006D          BRB     7$                                          ; 1168
                                    06      A6  95 0006F 4$:      TSTB    6(P)                                        ; 1181
                                            0C  12 00072          BNEQ    5$                                          ;
                                            59  D5 00074          TSTL    FIRST_FREE                                  ; 1182
                                            08  12 00076          BNEQ    5$                                          ;
                              59            6E  D0 00078          MOVL    VBN, FIRST_FREE                             ; 1185
04 AE                         58            04  78 0007B          ASHL    #4, J, FREE_OFFSET                          ; 1186
                     0C       AC       0C    A6  D1 00080 5$:      CMPL    12(P), BASE_LBN                             ; 1188
                                            0D  12 00085          BNEQ    7$                                          ;
                     FF       8F       07    A6  91 00087          CMPB    7(P), #255                                  ; 1191
                                            03  1F 0008C          BLSSU   6$                                          ;
                                        00A4    31  0008E          BRW     16$                                         ;
                                        009E    31  00091 6$:      BRW     15$                                         ;
                              56            10  C0 00094 7$:      ADDL2   #16, P                                      ; 1201
                        AD    58            1F  F3 00097          AOBLEQ  #31, J, 3$                                  ; 1165
                                            96  11 0009B          BRB     2$                                          ; 1159
                                    10      AC  D5 0009D 8$:      TSTL    MODE                                        ; 1210
                                            03  12 000A0          BNEQ    9$                                          ;
                                        00A7    31  000A2          BRW     19$                                         ;
                                            59  D5 000A5 9$:      TSTL    FIRST_FREE                                  ; 1212
                                            6B  12 000A7          BNEQ    13$                                         ;
                              53        18    AB  D0 000A9          MOVL    24(WINDOW), FCB                             ; 1215
                                            53  DD 000AD          PUSHL   FCB                                         ; 1216
                                            7E  D4 000AF          CLRL    -(SP)                                       ;
                  0000G CF                  02  FB 000B1          CALLS   #2, READ_HEADER                             ;
```

```
                52              50  D0 000B6          MOVL    R0, HEADER
              38  A3        3C  A3  D1 000B9          CMPL    60(FCB), 56(FCB)                ; 1218
                                11  1F 000BE          BLSSU   10$
              18  A7            01  D0 000C0          MOVL    #1, 24(FIB)                     ; 1221
              17  A7            02  88 000C4          BISB2   #2, 23(FIB)                     ; 1222
                                52  DD 000C8          PUSHL   HEADER                          ; 1223
                                57  DD 000CA          PUSHL   FIB
          0000G   CF            02  FB 000CC          CALLS   #2, EXTEND
                            1E  A2  B6 000D1 10$:     INCW    30(HEADER)                      ; 1227
                    28          62  91 000D4          CMPB    (HEADER), #40                   ; 1233
                                05  1F 000D7          BLSSU   11$
              4C    A2        1E  A2  3C 000D9          MOVZWL  30(HEADER), 76(HEADER)          ; 1235
                                52  DD 000DE 11$:     PUSHL   HEADER                          ; 1237
          0000G   CF            01  FB 000E0          CALLS   #1, CHECKSUM
          0000G   CF            00  FB 000E5          CALLS   #0, WRITE_HEADER                ; 1238
                                5B  DD 000EA          PUSHL   WINDOW                          ; 1240
                            04  AE  DD 000EC          PUSHL   VBN
          0000G   CF            02  FB 000EF          CALLS   #2, MAP_VBN
      FFFFFFFF    8F            50  D1 000F4          CMPL    LBN, #-1                        ; 1241
                                04  12 000FB          BNEQ    12$
                                  FEFF 000FD          BUGW                                    ; 1242
                                 0000* 000FF          .WORD   <BUG$_HDRNOTMAP!4>
                                04  DD 00101 12$:     PUSHL   #4                              ; 1243
                                01  DD 00103          PUSHL   #1
                                50  DD 00105          PUSHL   LBN
          0000G   CF            03  FB 00107          CALLS   #3, CREATE_BLOCK
                    56          50  D0 0010C          MOVL    R0, P
                            04  AE  D4 0010F          CLRL    FREE_OFFSET                     ; 1244
                                0E  11 00112          BRB     14$                             ; 1212
                                01  DD 00114 13$:     PUSHL   #1                              ; 1248
                                59  DD 00116          PUSHL   FIRST_FREE
                                5B  DD 00118          PUSHL   WINDOW
                      0000G     CF  03  FB 0011A      CALLS   #3, READ_DATA
                                56  50  D0 0011F      MOVL    R0, P
                                56      04  AE  C0 00122 14$:  ADDL2   FREE_OFFSET, P          ; 1253
          10            00      04  BC      06  2C 00126      MOVC5   #6, @FID, #0, #16, (P)  ; 1255
                                                66     0012C
                                08  A6      08  AC 7D 00128  MOVQ    BASE_VBN, 8(P)          ; 1256
                                            07  A6 96 00132 15$:  INCB    7(P)                ; 1258
                                01          10  AC D1 00135 16$:  CMPL    MODE, #1            ; 1259
                                                06 12 00139      BNEQ    17$
                                06  A6          01 88 0013B      BISB2   #1, 6(P)            ; 1260
                                                04 11 0013F      BRB     18$
                                06  A6          02 88 00141 17$: BISB2   #2, 6(P)            ; 1261
                                                56 DD 00145 18$: PUSHL   P                   ; 1263
                      0000G     CF              01 FB 00147      CALLS   #1, MARK_DIRTY
                                                5B DD 0014C 19$: PUSHL   WINDOW              ; 1267
                      0000G     CF              01 FB 0014E      CALLS   #1, CLOSE_FILE
                      0000G     CF              00 FB 00153      CALLS   #0, RESTORE_CONTEXT     ; 1268
                                                   04 00158      RET                             ; 1270

; Routine Size:  345 bytes,    Routine Base:  $CODE$ + 0000

;  282          1271  1
;  283          1272  1 END
;  284          1273  0 ELUDOM
```

```
;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         345  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        39         0      1000          00:02.0


;                                   COMMAND QUALIFIERS

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

; Size:          345 code + 0 data bytes
; Run Time:          00:21.3
; Elapsed Time:      00:49.9
; Lines/CPU Min:      3589
; Lexemes/CPU-Min: 43192
; Memory Used:  278 pages
; Compilation Complete
```

CHKDMO
LIS

CPYNAM
LIS

CLENUP
LIS

CHKPRO
LIS

CHARGEQ
LIS

COMMON
LIS

CREATE
LIS

BADSCN
LIS

CHKHD2
LIS

CHKSUM
LIS

ALLOCB
LIS