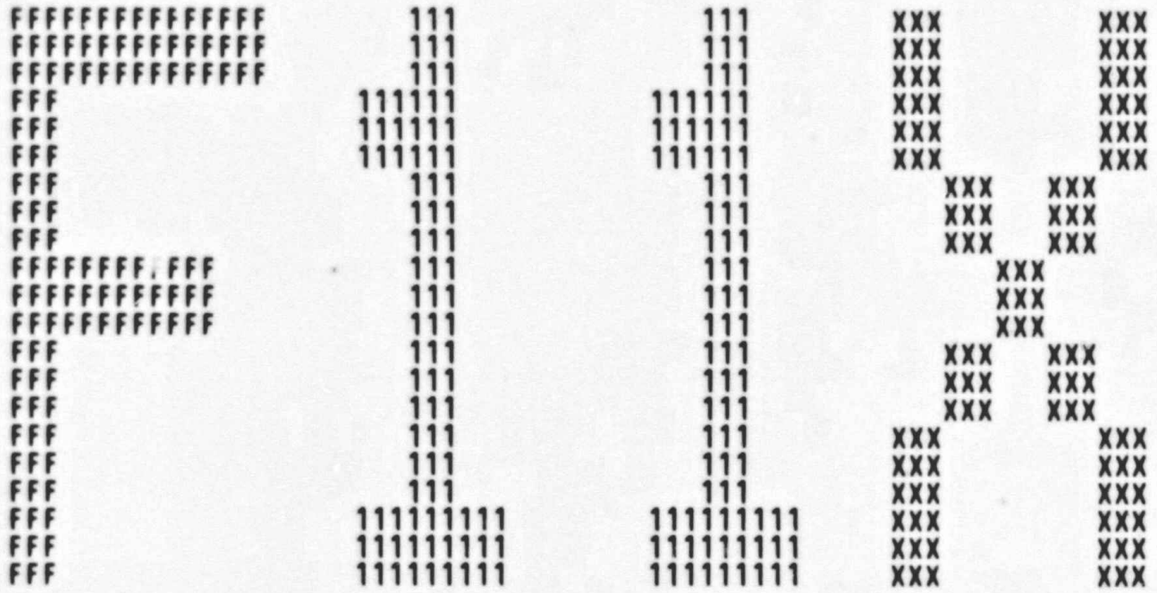

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
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**DELBAD

```
DDDDDDDD    EEEEEEEEEE  LL          BBBBBBBB      AAAAAA    DDDDDDDD
DDDDDDDD    EEEEEEEEEE  LL          BBBBBBBB      AAAAAA    DDDDDDDD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DD      DD  EEEEEEEE    LL          BBBBBBBB     AA    AA  DD      DD
DD      DD  EEEEEEEE    LL          BBBBBBBB     AA    AA  DD      DD
DD      DD  EE          LL          BB      BB   AAAAAAAAAA  DD      DD
DD      DD  EE          LL          BB      BB   AAAAAAAAAA  DD      DD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DD      DD  EE          LL          BB      BB   AA    AA  DD      DD
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  BBBBBBBB     AA    AA  DDDDDDDD
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  BBBBBBBB     AA    AA  DDDDDDDD

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 DELBAD (
;  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 removes the indicated blocks from the given file header
; 38     0038  1 !       and appends them to the bad block file.
; 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:  29-May-1978  22:43
; 49     0049  1 !
; 50     0050  1 ! MODIFIED BY:
; 51     0051  1 !
; 52     0052  1 !       V03-006 CDS0004         Christian D. Saether    14-Aug-1984
; 53     0053  1 !               Remove obsolete reference to update_filesize routine.
; 54     0054  1 !
; 55     0055  1 !       V03-005 CDS0003         Christian D. Saether    31-July-1984
; 56     0056  1 !               Remove local definition of get_map_pointer linkage.
; 57     0057  1 !
```

```
;    58          0058  1 !       V03-004 CDS0002         Christian D. Saether     2-May-1984
;    59          0059  1 !               Perform deallocation to bad block file in secondary
;    60          0060  1 !               context.  Add appropriate serialization.
;    61          0061  1 !
;    62          0062  1 !       V03-003 CDS0001         Christian D. Saether    29-Dec-1983
;    63          0063  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    64          0064  1 !
;    65          0065  1 !       V03-002 ACG0367         Andrew C. Goldstein,    26-Oct-1983  19:49
;    66          0066  1 !               Update BADBLK.SYS file highwater mark
;    67          0067  1 !
;    68          0068  1 !       V03-001 LMP0037         L. Mark Pilant,         28-Jun-1982  15:10
;    69          0069  1 !               Remove the addressing mode module switch.
;    70          0070  1 !
;    71          0071  1 !       V02-003 ACG0230         Andrew C. Goldstein,    24-Dec-1981  0:16
;    72          0072  1 !               Go to longword external addressing
;    73          0073  1 !
;    74          0074  1 !       V02-002 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
;    75          0075  1 !               Previous revision history moved to F11B.REV
;    76          0076  1 !**
;    77          0077  1
;    78          0078  1
;    79          0079  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    80          0080  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    82          1071  1 GLOBAL ROUTINE DEALLOCATE_BAD (FIB, FILE_HDR, POINTER, LAST_COUNT) : L_NORM NOVALUE =
;    83          1072  1
;    84          1073  1 !++
;    85          1074  1 !
;    86          1075  1 ! FUNCTIONAL DESCRIPTION:
;    87          1076  1 !
;    88          1077  1 !       This routine removes the indicated blocks from the given file header
;    89          1078  1 !       and appends them to the bad block file.
;    90          1079  1 !
;    91          1080  1 !
;    92          1081  1 ! CALLING SEQUENCE:
;    93          1082  1 !       DEALLOCATE_BAD (ARG1, ARG2, ARG3, ARG4)
;    94          1083  1 !
;    95          1084  1 ! INPUT PARAMETERS:
;    96          1085  1 !       ARG1: address of user FIB
;    97          1086  1 !       ARG2: address of file header
;    98          1087  1 !       ARG3: address of map pointer at which to start
;    99          1088  1 !       ARG4: new value for last pointer block count
;   100          1089  1 !
;   101          1090  1 ! IMPLICIT INPUTS:
;   102          1091  1 !       NONE
;   103          1092  1 !
;   104          1093  1 ! OUTPUT PARAMETERS:
;   105          1094  1 !       NONE
;   106          1095  1 !
;   107          1096  1 ! IMPLICIT OUTPUTS:
;   108          1097  1 !       NONE
;   109          1098  1 !
;   110          1099  1 ! ROUTINE VALUE:
;   111          1100  1 !       NONE
;   112          1101  1 !
;   113          1102  1 ! SIDE EFFECTS:
;   114          1103  1 !       file header updated, bad block log file updated, bad block file extended
;   115          1104  1 !
;   116          1105  1 !--
;   117          1106  1
;   118          1107  2 BEGIN
;   119          1108  2
;   120          1109  2 MAP
;   121          1110  2         FIB             : REF BBLOCK,   ! user FIB argument
;   122          1111  2         FILE_HDR        : REF BBLOCK;   ! address of file header
;   123          1112  2
;   124          1113  2 LINKAGE
;   125          1114  2         L_MAKE_POINTER  = CALL :
;   126          1115  2                         GLOBAL (BUILD_POINTER = 9);
;   127          1116  2
;   128          1117  2 GLOBAL REGISTER
;   129          1118  2         COUNT           = 6,            ! count of blocks returned
;   130          1119  2         LBN             = 7,            ! LBN of map entry
;   131          1120  2         MAP_POINTER     = 8 : REF BBLOCK, ! pointer to scan map
;   132          1121  2         BUILD_POINTER   = 9 : REF BBLOCK; ! pointer to build new map entry
;   133          1122  2
;   134          1123  2 LOCAL
;   135          1124  2         HEADER          : REF BBLOCK,   ! local address of file header
;   136          1125  2         EXT_HEADER      : REF BBLOCK;   ! address of extension header
;   137          1126  2
;   138          1127  2 BIND
```

```
;  139          1128  2          BB_FID = UPLIT WORD (BADBLK_FID, BADBLK_FID, 0);
;  140          1129  2
;  141          1130  2  BIND_COMMON;
;  142          1131  2
;  143          1132  2  EXTERNAL ROUTINE
;  144          1133  2          SAVE_CONTEXT     : L_NORM,          ! save primary context
;  145          1134  2          RESTORE_CONTEXT  : L_NORM,          ! restore primary context
;  146          1135  2          SERIAL_FILE      : L_NORM,          ! file serialization lock.
;  147          1136  2          RELEASE_SERIAL_LOCK : L_NORM,       ! relinquish file serialization
;  148          1137  2          WRITE_DIRTY      : L_NORM,          ! write modified buffers
;  149          1138  2          GET_MAP_POINTER  : L_MAP_POINTER,   ! get value of next map entry
;  150          1139  2          MAKE_POINTER     : L_MAKE_POINTER,  ! build new map entry
;  151          1140  2          NEXT_HEADER      : L_NORM,          ! read next extension header
;  152          1141  2          MARK_DIRTY       : L_NORM,          ! mark buffer for rewrite
;  153          1142  2          ZERO_WINDOWS     : L_NORM,          ! invalidate windows of file
;  154          1143  2          CHECKSUM         : L_NORM,          ! compute file header checksum
;  155          1144  2          READ_HEADER      : L_NORM,          ! read file header
;  156          1145  2          EXTEND_HEADER    : L_NORM,          ! create extention header
;  157          1146  2          SCAN_BADLOG      : L_NORM;          ! scan pending bad block log file
;  158          1147  2
;  159          1148  2
;  160          1149  2  ! Get into secondary context.
;  161          1150  2  !
;  162          1151  2
;  163          1152  2  SAVE_CONTEXT ();
;  164          1153  2
;  165          1154  2  ! Construct pointers into the file header and get the current contents of the
;  166          1155  2  ! last map pointer.
;  167          1156  2  !
;  168          1157  2
;  169          1158  2  HEADER = .FILE_HDR;
;  170          1159  2  MAP_POINTER = .POINTER;
;  171          1160  2
;  172          1161  2  GET_MAP_POINTER ();
;  173          1162  2
;  174          1163  2  ! Now append the blocks to the bad block file.
;  175          1164  2  !
;  176          1165  2
;  177          1166  2  LBN = .LBN + .LAST_COUNT;                  ! compute LBN of bad cluster
;  178          1167  2  COUNT = .COUNT - .LAST_COUNT;
;  179          1168  2
;  180          1169  2  ! Serialize on the bad block file.
;  181          1170  2  !
;  182          1171  2
;  183          1172  2  PRIM_LCKINDX = SERIAL_FILE (BB_FID);
;  184          1173  2
;  185          1174  2  HEADER = READ_HEADER (BB_FID, 0);
;  186          1175  2  WHILE 1 DO
;  187          1176  3      BEGIN
;  188          1177  3      EXT_HEADER = NEXT_HEADER (.HEADER, 0);
;  189          1178  3      IF .EXT_HEADER EQL 0 THEN EXITLOOP;
;  190          1179  3      HEADER = .EXT_HEADER;
;  191          1180  2      END;
;  192          1181  2  MARK_DIRTY (.HEADER);
;  193          1182  2  BUILD_POINTER = .HEADER + (.HEADER[FH2$B_MPOFFSET] + .HEADER[FH2$B_MAP_INUSE]) * 2;
;  194          1183  2
;  195          1184  2  IF NOT MAKE_POINTER (.COUNT, .LBN, .HEADER)
```

```
;  196          1185  2 THEN
;  197          1186  3     BEGIN
;  198          1187  3     HEADER = EXTEND_HEADER (UPLIT BYTE (REP FIB$C_LENGTH OF (0)), .HEADER, 0);
;  199          1188  3     BUILD_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET] * 2;
;  200          1189  3     MAKE_POINTER (.COUNT, .LBN, .HEADER);
;  201          1190  3     END;
;  202          1191  2
;  203          1192  2 BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK] =
;  204          1193  2         ROT (ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + .COUNT, 16);
;  205          1194  2
;  206          1195  2 ! If this file header supports it, stuff the high water field to
;  207          1196  2 ! be the allocated size.
;  208          1197  2 !
;  209          1198  2
;  210          1199  2 IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
;  211          1200  2 THEN
;  212          1201  2     HEADER [FH2$L_HIGHWATER] =  ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + 1;
;  213          1202  2
;  214          1203  2 CHECKSUM (.HEADER);
;  215          1204  2
;  216          1205  2 ! Write the modified header(s), release the serialization lock, and return to
;  217          1206  2 ! primary context.
;  218          1207  2 !
;  219          1208  2
;  220          1209  2 WRITE_DIRTY (.LB_BASIS [.PRIM_LCKINDX]);
;  221          1210  2
;  222          1211  2 RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
;  223          1212  2
;  224          1213  2 RESTORE_CONTEXT ();
;  225          1214  2
;  226          1215  2 ! Finally, remove the bad block cluster from the volume pending bad block log
;  227          1216  2 ! file, if it was there.
;  228          1217  2 !
;  229          1218  2
;  230          1219  2 SCAN_BADLOG (0, 0, .LBN, REMOVE_BADBLOCK, .COUNT);
;  231          1220  2
;  232          1221  1 END;                                    ! end of routine DEALLOCATE_BAD


                                                        .TITLE  DELBAD
                                                        .IDENT  \V04-000\

                                                        .PSECT  $CODE$,NOWRT,2

                                0000  0003  0003  00000 P.AAA:  .WORD   3, 3, 0                                 ;
                                                00# 00006 P.AAB:  .BYTE   0[64]                                 ;

                                                BB_FID=           P.AAA
                                                        .EXTRN  SAVE_CONTEXT, RESTORE_CONTEXT
                                                        .EXTRN  SERIAL_FILE, RELEASE_SERIAL_LOCK
                                                        .EXTRN  WRITE_DIRTY, GET_MAP_POINTER
                                                        .EXTRN  MAKE_POINTER, NEXT_HEADER
                                                        .EXTRN  MARK_DIRTY, ZERO_WINDOWS
                                                        .EXTRN  CHECKSUM, READ_HEADER
                                                        .EXTRN  EXTEND_HEADER, SCAN_BADLOG

                                             03C4 00000         .ENTRY  DEALLOCATE_BAD, Save R2,R6,R7,R8,R9         ; 1071
```

```
          0000G  CF          00  FB 00002        CALLS   #0, SAVE_CONTEXT                          ; 1152
                 52    08    AC  D0 00007        MOVL    FILE_HDR, HEADER                          ; 1158
                 58    0C    AC  D0 0000B        MOVL    POINTER, MAP_POINTER                      ; 1159
                           0000G 30 0000F        BSBW    GET_MAP_POINTER                           ; 1161
                 57    10    AC  C0 00012        ADDL2   LAST_COUNT, LBN                           ; 1166
                 56    10    AC  C2 00016        SUBL2   LAST_COUNT, COUNT                         ; 1167
                       9D    AF  9F 0001A        PUSHAB  BB_FID                                    ; 1172
          0000G  CF          01  FB 0001D        CALLS   #1, SERIAL_FILE                           ;
             18  AA          50  D0 00022        MOVL    R0, 24(BASE)                              ;
                             7E  D4 00026        CLRL    -(SP)                                     ; 1174
                       8F    AF  9F 00028        PUSHAB  BB_FID                                    ;
          0000G  CF          02  FB 0002B        CALLS   #2, READ_HEADER                           ;
                 52          50  D0 00030 1$:    MOVL    R0, HEADER                                ;
                             7E  D4 00033        CLRL    -(SP)                                     ; 1177
                             52  DD 00035        PUSHL   HEADER                                    ;
          0000G  CF          02  FB 00037        CALLS   #2, NEXT_HEADER                           ;
                             50  D5 0003C        TSTL    EXT_HEADER                                ; 1178
                             F0  12 0003E        BNEQ    1$                                        ;
                             52  DD 00040        PUSHL   HEADER                                    ; 1181
          0000G  CF          01  FB 00042        CALLS   #1, MARK_DIRTY                            ;
                 50    01    A2  9A 00047        MOVZBL  1(HEADER), R0                             ; 1182
                 51    3A    A2  9A 0004B        MOVZBL  58(HEADER), R1                            ;
                 50          51  C0 0004F        ADDL2   R1, R0                                    ;
                 59        6240  3E 00052        MOVAW   (HEADER)[R0], BUILD_POINTER               ;
                             52  DD 00056        PUSHL   HEADER                                    ; 1184
                 7E          56  7D 00058        MOVQ    COUNT, -(SP)                              ;
          0000G  CF          03  FB 0005B        CALLS   #3, MAKE_POINTER                          ;
                 22          50  E8 00060        BLBS    R0, 2$                                    ;
                             7E  D4 00063        CLRL    -(SP)                                     ; 1187
                             52  DD 00065        PUSHL   HEADER                                    ;
                     FF55    CF  9F 00067        PUSHAB  P.AAB                                     ;
          0000G  CF          03  FB 0006B        CALLS   #3, EXTEND_HEADER                         ;
                 52          50  D0 00070        MOVL    R0, HEADER                                ;
                 50    01    A2  9A 00073        MOVZBL  1(HEADER), R0                             ; 1188
                 59        6240  3E 00077        MOVAW   (HEADER)[R0], BUILD_POINTER               ;
                             52  DD 0007B        PUSHL   HEADER                                    ; 1189
                 7E          56  7D 0007D        MOVQ    COUNT, -(SP)                              ;
          0000G  CF          03  FB 00080        CALLS   #3, MAKE_POINTER                          ;
     50      18  A2          10  9C 00085 2$:    ROTL    #16, 24(HEADER), R0                       ; 1193
                 50          56  C0 0008A        ADDL2   COUNT, R0                                 ;
 18  A2          50          10  9C 0008D        ROTL    #16, R0, 24(HEADER)                       ;
                 28          62  91 00092        CMPB    (HEADER), #40                             ; 1199
                             0A  1F 00095        BLSSU   3$                                        ;
     50      18  A2          10  9C 00097        ROTL    #16, 24(HEADER), R0                       ; 1201
             4C  A2    01    A0  9E 0009C        MOVAB   1(R0), 76(HEADER)                         ;
                             52  DD 000A1 3$:    PUSHL   HEADER                                    ; 1203
          0000G  CF          01  FB 000A3        CALLS   #1, CHECKSUM                              ;
                 50    18    AA  D0 000A8        MOVL    24(BASE), R0                              ; 1209
                     0080  CA40  DD 000AC        PUSHL   128(BASE)[R0]                             ;
          0000G  CF          01  FB 000B1        CALLS   #1, WRITE_DIRTY                           ;
                       18    AA  DD 000B6        PUSHL   24(BASE)                                  ; 1211
          0000G  CF          01  FB 000B9        CALLS   #1, RELEASE_SERIAL_LOCK                   ;
          0000G  CF          00  FB 000BE        CALLS   #0, RESTORE_CONTEXT                       ; 1213
                             56  DD 000C3        PUSHL   COUNT                                     ; 1219
                             7E  D4 000C5        CLRL    -(SP)                                     ;
                             57  DD 000C7        PUSHL   LBN                                       ;
                             7E  7C 000C9        CLRQ    -(SP)                                     ;
```

```
                        0000G  CF          05  FB 000CB          CALLS   #5, SCAN_BADLOG                          ;
                                            04 000D0          RET                                             ; 1221

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 0046

;  233          1222  1
;  234          1223  1 END
;  235          1224  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                          279  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         24         0         1000        00:02.0



;                                       COMMAND QUALIFIERS

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

; Size:          209 code + 70 data bytes
; Run Time:          00:17.5
; Elapsed Time:      00:39.6
; Lines/CPU Min:      4206
; Lexemes/CPU-Min:  49494
; Memory Used:  222 pages
; Compilation Complete
```

DEACCS
LIS

DELETE
LIS

DIRSCN
LIS

DISPAT
LIS

CREHDR
LIS

DIRACC
LIS

DELBAD
LIS

CREFCB
LIS

CREWIN
LIS

DISPATCH
LIS

ENTER
LIS

DELFIL
LIS