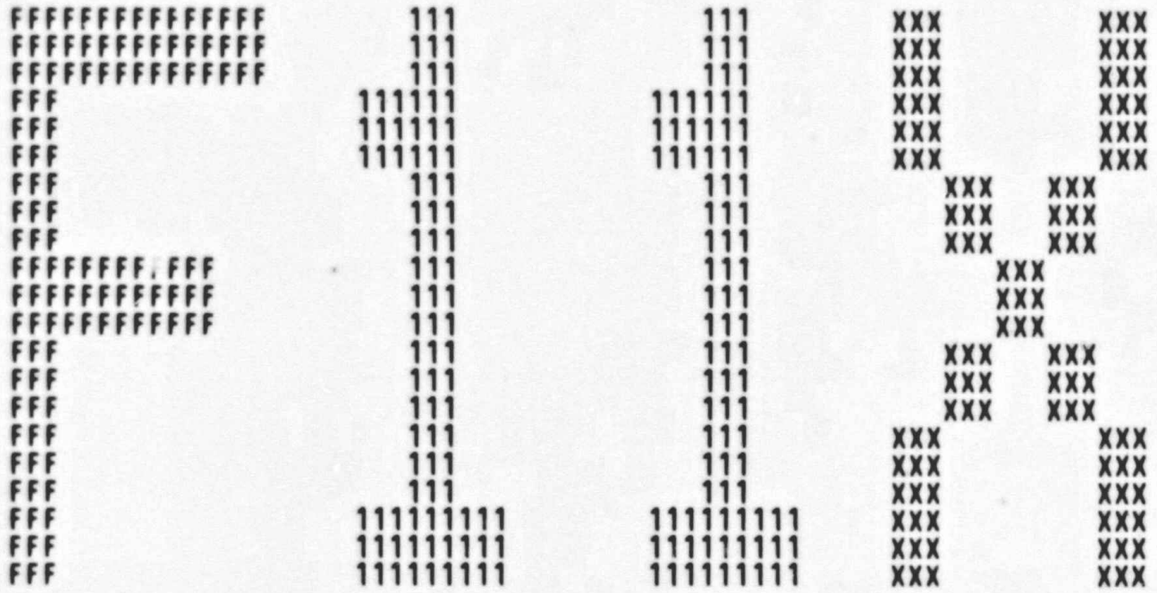

_$2

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

```
  SSSSSSSS  WW      WW    IIIIII    TTTTTTTTTT  VV      VV  LL
  SSSSSSSS  WW      WW    IIIIII    TTTTTTTTTT  VV      VV  LL
SS          WW      WW      II          TT      VV      VV  LL
SS          WW      WW      II          TT      VV      VV  LL
SS          WW      WW      II          TT      VV      VV  LL
SS          WW      WW      II          TT      VV      VV  LL
  SSSSSS    WW      WW      II          TT      VV      VV  LL
  SSSSSS    WW      WW      II          TT      VV      VV  LL
        SS  WW  WW  WW      II          TT      VV      VV  LL
        SS  WW  WW  WW      II          TT      VV      VV  LL
        SS  WWWW    WWWW    II          TT        VV  VV    LL                ....
        SS  WWWW    WWWW    II          TT        VV  VV    LL                ....
SSSSSSSS    WW      WW    IIIIII        TT          VV      LLLLLLLLLL        ....
SSSSSSSS    WW      WW    IIIIII        TT          VV      LLLLLLLLLL        ....

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 SWITVL (
;     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 module contains routines that switch file ACP context from
;    38          0038  1 !       one volume to another.
;    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:  8-Nov-1978  13:35
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-004 CDS0004         Christian D. Saether    30-Dec-1983
;    53          0053  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    54          0054  1 !
;    55          0055  1 !       V03-003 CDS0003         Christian D. Saether    15-Oct-1983
;    56          0056  1 !               Remove call to flush_lock_basis.  This is called
;    57          0057  1 !               from allocation_unlock now.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-002 CDS0002         Christian D. Saether    14-Sep-1983
;    60          0060  1 !               Release and reacquire correct volume lock, if held.
;    61          0061  1 !
;    62          0062  1 !       V03-001 CDS0001         Christian D. Saether    27-Aug-1983
;    63          0063  1 !               Remove reference to GET_CCB and use IO_CCB cell instead.
;    64          0064  1 !
;    65          0065  1 !       B0103   ACG0082         Andrew C. Goldstein,    13-Nov-1979  23:56
;    66          0066  1 !               Remove FLUSH_BUFFERS call due to write-back cacheing
;    67          0067  1 !
;    68          0068  1 !       B0102   ACG0082         Andrew C. Goldstein,    5-Nov-1979  14:52
;    69          0069  1 !               Changes for write-back cacheing
;    70          0070  1 !
;    71          0071  1 !       B0101   ACG0037         Andrew C. Goldstein,  7-May-1979  15:15
;    72          0072  1 !       Range check RVN before indexing into RVT;
;    73          0073  1 !       allow RVN 1 on single volumes
;    74          0074  1 !
;    75          0075  1 !**
;    76          0076  1
;    77          0077  1
;    78          0078  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    79          0079  1 REQUIRE 'SRC$:FCPDEF.B32';
;    80          1070  1
;    81          1071  1
;    82          1072  1 FORWARD ROUTINE
;    83          1073  1         SWITCH_VOLUME   : L_NORM NOVALUE, ! switch context to specified RVN
;    84          1074  1         SWITCH_CHANNEL  : L_NORM NOVALUE; ! switch channel assignments
```

```
;    86          1075  1 GLOBAL ROUTINE SWITCH_VOLUME (NEW_RVN) : L_NORM NOVALUE =
;    87          1076  1
;    88          1077  1 !++
;    89          1078  1 !
;    90          1079  1 ! FUNCTIONAL DESCRIPTION:
;    91          1080  1 !
;    92          1081  1 !       This routine switches the ACP context to the specified RVN. It
;    93          1082  1 !       assigns the current channel to the new unit.
;    94          1083  1 !
;    95          1084  1 !
;    96          1085  1 ! CALLING SEQUENCE:
;    97          1086  1 !       SWITCH_VOLUME (ARG1)
;    98          1087  1 !
;    99          1088  1 ! INPUT PARAMETERS:
;   100          1089  1 !       ARG1: relative volume number to switch to
;   101          1090  1 !
;   102          1091  1 ! IMPLICIT INPUTS:
;   103          1092  1 !       CURRENT_UCB: UCB address of current volume
;   104          1093  1 !       CURRENT_VCB: VCB address of current volume
;   105          1094  1 !
;   106          1095  1 ! OUTPUT PARAMETERS:
;   107          1096  1 !       NONE
;   108          1097  1 !
;   109          1098  1 ! IMPLICIT OUTPUTS:
;   110          1099  1 !       NONE
;   111          1100  1 !
;   112          1101  1 ! ROUTINE VALUE:
;   113          1102  1 !       NONE
;   114          1103  1 !
;   115          1104  1 ! SIDE EFFECTS:
;   116          1105  1 !       context switched to new volume
;   117          1106  1 !
;   118          1107  1 !--
;   119          1108  1
;   120          1109  2 BEGIN
;   121          1110  2
;   122          1111  2 BIND_COMMON;
;   123          1112  2
;   124          1113  2 EXTERNAL ROUTINE
;   125          1114  2         ALLOCATION_LOCK : L_NORM NOVALUE, ! acquire volume lock for current volume
;   126          1115  2         ALLOCATION_UNLOCK : L_NORM;       ! release current volume lock.
;   127          1116  2
;   128          1117  2 LOCAL
;   129          1118  2         VOLOCK,                           ! remember whether volume lock held.
;   130          1119  2         RVN,                              ! filtered RVN desired
;   131          1120  2         RVT             : REF BBLOCK,     ! address of relative volume table
;   132          1121  2         UCB             : REF BBLOCK;     ! address of new UCB
;   133          1122  2
;   134          1123  2
;   135          1124  2 ! First check if a volume switch is necessary. Extract the true RVN part
;   136          1125  2 ! (removing extended file ID if present), check for zero and compare it
;   137          1126  2 ! against the current RVN.
;   138          1127  2 !
;   139          1128  2
;   140          1129  2 RVN = .NEW_RVN<0,16>;
;   141          1130  2 IF .CURRENT_VCB[VCB$V_EXTFID]
;   142          1131  2 THEN RVN = .NEW_RVN<0,8>;
```

```
;  143          1132  2
;  144          1133  2 IF .RVN EQL 0 OR .RVN EQL .CURRENT_RVN
;  145          1134  2 THEN RETURN;
;  146          1135  2
;  147          1136  2 ! Get the RVT and from it the UCB address we are switching to. Nonexistence
;  148          1137  2 ! of either is an error.
;  149          1138  2 !
;  150          1139  2
;  151          1140  2 RVT = .CURRENT_VCB[VCB$L_RVT];
;  152          1141  2 IF .RVT EQL .CURRENT_UCB
;  153          1142  2 THEN
;  154          1143  3     BEGIN
;  155          1144  3     IF .RVN EQL 1
;  156          1145  3     THEN RETURN
;  157          1146  3     ELSE ERR_EXIT (SS$_NOTVOLSET);
;  158          1147  2     END;
;  159          1148  2
;  160          1149  2 IF .RVN GTRU .RVT[RVT$B_NVOLS]
;  161          1150  2 THEN ERR_EXIT (SS$_DEVNOTMOUNT);
;  162          1151  2
;  163          1152  2 UCB = .VECTOR [RVT[RVT$L_UCBLST], .RVN-1];
;  164          1153  3 IF (
;  165          1154  3     IF .UCB EQL 0
;  166          1155  3     THEN 1
;  167          1156  3     ELSE NOT .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_MNT]
;  168          1157  3     )
;  169          1158  2 THEN ERR_EXIT (SS$_DEVNOTMOUNT);
;  170          1159  2
;  171          1160  2 IF .UCB[UCB$B_TYPE] NEQ DYN$C_UCB
;  172          1161  2 THEN BUG_CHECK (NOTUCBRVT, FATAL, 'Not UCB pointer in RVT');
;  173          1162  2
;  174          1163  2 ! Unlock current volume lock, if held, and remember whether there was one.
;  175          1164  2 !
;  176          1165  2
;  177          1166  2 VOLOCK = 0;
;  178          1167  2
;  179          1168  2 IF .LB_LOCKID [0] NEQ 0
;  180          1169  2 THEN
;  181          1170  3     BEGIN
;  182          1171  3     ALLOCATION_UNLOCK ();
;  183          1172  3     VOLOCK = 1;
;  184          1173  2     END;
;  185          1174  2
;  186          1175  2 ! Finally shuffle the channels and pointers about.
;  187          1176  2 !
;  188          1177  2
;  189          1178  2 SWITCH_CHANNEL (.UCB);
;  190          1179  2
;  191          1180  2 ! If we had a volume lock before, reacquire it for the volume we
;  192          1181  2 ! just switched to.
;  193          1182  2 !
;  194          1183  2
;  195          1184  2 IF .VOLOCK
;  196          1185  2 THEN
;  197          1186  2     ALLOCATION_LOCK ();
;  198          1187  2
;  199          1188  1 END;                                    ! end of routine SWITCH_VOLUME
```

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

                                                        .EXTRN  ALLOCATION_LOCK
                                                        .EXTRN  ALLOCATION_UNLOCK
                                                        .EXTRN  BUG$_NOTUCBRVT

                                                        .PSECT  $CODE$,NOWRT,2

                                                000C 00000      .ENTRY  SWITCH_VOLUME, Save R2,R3                       ; 1075
                                          51    04   AC  3C 00002      MOVZWL  NEW_RVN, RVN                            ; 1129
                                          50    98   AA  D0 00006      MOVL    -104(BASE), R0                          ; 1130
                                    04          0B   A0  05 E1 0000A      BBC     #5, 11(R0), 1$                      ;
                                          51    04   AC  9A 0000F      MOVZBL  NEW_RVN, RVN                            ; 1131
                                          51           D5 00013 1$:   TSTL    RVN                                     ; 1133
                                                5F     13 00015       BEQL    7$                                      ;
                                          A0    AA     51 D1 00017      CMPL    RVN, -96(BASE)                          ;
                                                59     13 0001B       BEQL    7$                                      ;
                                          50    98   AA  D0 0001D      MOVL    -104(BASE), R0                          ; 1140
                                          50    20   A0  D0 00021      MOVL    32(R0), RVT                             ;
                                          94    AA       50 D1 00025      CMPL    RVT, -108(BASE)                         ; 1141
                                                0A     12 00029       BNEQ    2$                                      ;
                                                01       51 D1 0002B      CMPL    RVN, #1                               ; 1144
                                                46     13 0002E       BEQL    7$                                      ;
                                                0998   8F   BF 00030       CHMU    #2456                               ; 1146
                                                       04 00034       RET                                             ;
                                    51    0B    A0         08           00 ED 00035 2$:   CMPZV   #0, #8, 11(RVT), RVN   ; 1149
                                                0C     1F 0003B       BLSSU   3$                                      ;
                                          52          40 A041  D0 0003D      MOVL    64(RVT)[RVN], UCB               ; 1152
                                                05     13 00042       BEQL    3$                                      ; 1154
                                    05          3A   A2         03 E0 00044      BBS     #3, 58(UCB), 4$             ; 1156
                                                007C   8F   BF 00049 3$:   CHMU    #124                                ; 1158
                                                       04 0004D       RET                                             ;
                                                10         0A   A2  91 0004E 4$:   CMPB    10(UCB), #16                ; 1160
                                                           04   13  00052       BEQL    5$                              ;
                                                              FEFF  00054       BUGW                                    ; 1161
                                                             0000*  00056       .WORD   <BUG$_NOTUCBRVT!4>              ;
                                                           53   D4  00058 5$:   CLRL    VOLOCK                          ; 1166
                                                           6C   AA  D5 0005A       TSTL    108(BASE)                   ; 1168
                                                           08   13  0005D       BEQL    6$                              ;
                                          0000G    CF              00   FB 0005F       CALLS   #0, ALLOCATION_UNLOCK   ; 1171
                                                   53              01   D0 00064       MOVL    #1, VOLOCK              ; 1172
                                                                   52   DD 00067 6$:   PUSHL   UCB                     ; 1178
                                          0000V    CF              01   FB 00069       CALLS   #1, SWITCH_CHANNEL      ;
                                                   05              53   E9 0006E       BLBC    VOLOCK, 7$              ; 1184
                                          0000G    CF              00   FB 00071       CALLS   #0, ALLOCATION_LOCK     ; 1186
                                                                   04   00076 7$:   RET                               ; 1188

; Routine Size:  119 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  201          1189  1 GLOBAL ROUTINE SWITCH_CHANNEL (UCB) : L_NORM NOVALUE =
;  202          1190  1
;  203          1191  1 !++
;  204          1192  1 !
;  205          1193  1 ! FUNCTIONAL DESCRIPTION:
;  206          1194  1 !
;  207          1195  1 !       This routine reassigns the ACP's channels to the specified UCB
;  208          1196  1 !       and fixes up the associated pointers. It must be called in
;  209          1197  1 !       kernel mode.
;  210          1198  1 !
;  211          1199  1 !
;  212          1200  1 ! CALLING SEQUENCE:
;  213          1201  1 !       SWITCH_CHANNEL (ARG1)
;  214          1202  1 !
;  215          1203  1 ! INPUT PARAMETERS:
;  216          1204  1 !       ARG1: UCB address of new device
;  217          1205  1 !
;  218          1206  1 ! IMPLICIT INPUTS:
;  219          1207  1 !
;  220          1208  1 !       IO_CHANNEL: channel number of primary channel
;  221          1209  1 !       IO_CCB: CCB of IO_CHANNEL
;  222          1210  1 !       CURRENT_UCB: address of current UCB
;  223          1211  1 !
;  224          1212  1 ! OUTPUT PARAMETERS:
;  225          1213  1 !       NONE
;  226          1214  1 !
;  227          1215  1 ! IMPLICIT OUTPUTS:
;  228          1216  1 !       CURRENT_UCB: contains address of new UCB
;  229          1217  1 !       CURRENT_VCB: address of new VCB
;  230          1218  1 !       CURRENT_RVN: RVN of new volume
;  231          1219  1 !
;  232          1220  1 ! ROUTINE VALUE:
;  233          1221  1 !       1
;  234          1222  1 !
;  235          1223  1 ! SIDE EFFECTS:
;  236          1224  1 !       channels reassigned
;  237          1225  1 !
;  238          1226  1 !--
;  239          1227  1
;  240          1228  2 BEGIN
;  241          1229  2
;  242          1230  2 MAP
;  243          1231  2         UCB             : REF BBLOCK;   ! UCB address arg
;  244          1232  2
;  245          1233  2 BIND_COMMON;
;  246          1234  2
;  247          1235  2
;  248          1236  2 ! Stuff the desired UCB address into IO_CHANNEL's CCB.
;  249          1237  2 ! Fix up other global pointers.
;  250          1238  2 !
;  251          1239  2
;  252          1240  2 IO_CCB [CCB$L_UCB] = .UCB;
;  253          1241  2
;  254          1242  2 CURRENT_UCB = .UCB;
;  255          1243  2 CURRENT_VCB = .UCB[UCB$L_VCB];
;  256          1244  2
;  257          1245  2 IF .CURRENT_VCB EQL 0
```

```
;  258          1246  2 THEN BUG_CHECK (NOTUCBRVT, FATAL, 'Bad UCB pointer in RVT');
;  259          1247  2 IF .CURRENT_VCB[VCB$B_TYPE] NEQ DYN$C_VCB
;  260          1248  2 THEN BUG_CHECK (NOTVCBUCB, FATAL, 'Bad VCB pointer in UCB');
;  261          1249  2
;  262          1250  2 CURRENT_RVN = .CURRENT_VCB[VCB$W_RVN];
;  263          1251  2
;  264          1252  1 END;                                    ! end of routine SWITCH_CHANNEL


                                                        .EXTRN  BUG$_NOTVCBUCB

                                    0000 00000          .ENTRY  SWITCH_CHANNEL, Save nothing                ; 1189
                          FF74  DA  04  AC  D0 00002    MOVL    UCB, @-140(BASE)                            ; 1240
                            94  AA  04  AC  D0 00008    MOVL    UCB, -108(BASE)                             ; 1242
                                50  04  AC  D0 0000D    MOVL    UCB, R0                                     ; 1243
                            98  AA  34  A0  D0 00011    MOVL    52(R0), -104(BASE)                          ;
                                        04  12 00016    BNEQ    1$                                          ; 1245
                                          FEFF 00018    BUGW                                                ; 1246
                                        0000* 0001A     .WORD   <BUG$_NOTUCBRVT!4>                          ;
                                50  98  AA  D0 0001C 1$: MOVL    -104(BASE), R0                              ; 1247
                                11  0A  A0  91 00020    CMPB    10(R0), #17                                 ;
                                        04  13 00024    BEQL    2$                                          ;
                                          FEFF 00026    BUGW                                                ; 1248
                                        0000* 00028     .WORD   <BUG$_NOTVCBUCB!4>                          ;
                                50  98  AA  D0 0002A 2$: MOVL    -104(BASE), R0                              ; 1250
                              A0  AA  0E  A0  3C 0002E  MOVZWL  14(R0), -96(BASE)                           ;
                                            04 00033    RET                                                 ; 1252

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 0077

;  265          1253  1
;  266          1254  1 END
;  267          1255  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                           171  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        32         0        1000        00:02.0
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         171 code + 0 data bytes
; Run Time:       00:24.6
; Elapsed Time:   00:56.4
; Lines/CPU Min:   3065
; Lexemes/CPU-Min: 60012
; Memory Used:  206 pages
; Compilation Complete
```

SCHFCB
LIS

SNDSMB
LIS

SHFDIR
LIS

SNDERR
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MDL

SMALOC
LIS

SNDBAD
LIS

SWITVL
LIS

WITURN
LIS