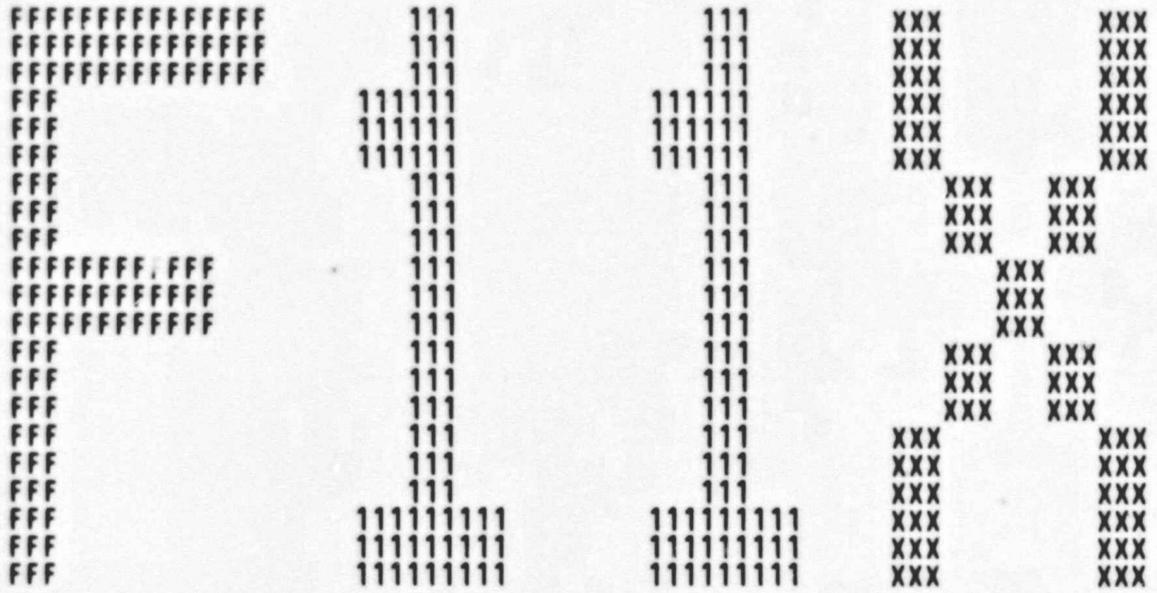

_$2

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**SNDSMB

```
  SSSSSSSS  NN      NN  DDDDDDDD     SSSSSSSS  MM      MM  BBBBBBBB
  SSSSSSSS  NN      NN  DDDDDDDD     SSSSSSSS  MM      MM  BBBBBBBB
SS          NN      NN  DD      DD  SS        MMMM  MMMM  BB      BB
SS          NN      NN  DD      DD  SS        MMMM  MMMM  BB      BB
SS          NNNN    NN  DD      DD  SS        MM  MM  MM  BB      BB
SS          NNNN    NN  DD      DD  SS        MM  MM  MM  BB      BB
  SSSSSS    NN  NN  NN  DD      DD    SSSSSS  MM      MM  BBBBBBBB
  SSSSSS    NN  NN  NN  DD      DD    SSSSSS  MM      MM  BBBBBBBB
        SS  NN    NNNN  DD      DD        SS  MM      MM  BB      BB
        SS  NN    NNNN  DD      DD        SS  MM      MM  BB      BB
        SS  NN      NN  DD      DD        SS  MM      MM  BB      BB
        SS  NN      NN  DD      DD        SS  MM      MM  BB      BB
SSSSSSSS    NN      NN  DDDDDDDD    SSSSSSSS  MM      MM  BBBBBBBB
SSSSSSSS    NN      NN  DDDDDDDD    SSSSSSSS  MM      MM  BBBBBBBB

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 SNDSMB (
;     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:  F11XQP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine constructs and sends a message to the symbiont manager
;    38          0038  1 !       to cause a file to be spooled and deleted.
;    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:  5-Jun-1978  11:23
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-007 ACG0408         Andrew C. Goldstein,    23-Mar-1984  14:48
;    53          0053  1 !               Add AST parameter so that impure storage is fully based
;    54          0054  1 !
;    55          0055  1 !       V03-006 CDS0005         Christian D. Saether    30-Dec-1983
;    56          0056  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-005 CDS0004         Christian D. Saether    26-Jul-1983
;    59          0059  1 !               Use new send job controller service.
;    60          0060  1 !
;    61          0061  1 !       V03-004 CDS0003         Christian D. Saether    13-May-1983
;    62          0062  1 !               Reflect change to IOC$CVT_DEVNAM interface.
;    63          0063  1 !
;    64          0064  1 !       V03-003 CWH1002         CW Hobbs                1-Mar-1983
;    65          0065  1 !               Use extended pid and owner in symbiont message
;    66          0066  1 !
;    67          0067  1 !       V03-002 CDS0002         Christian D. Saether    16-Dec-1982
;    68          0068  1 !               Make item list generation pic.
;    69          0069  1 !
;    70          0070  1 !       V03-001 CDS0001         C Saether               30-Jul-1982
;    71          0071  1 !               Changes for ACP to XQP.
;    72          0072  1 !               No timer on waiting for job controller reply.
;    73          0073  1 !
;    74          0074  1 !       V02-004 ACG0245         Andrew C. Goldstein,    23-Dec-1981  21:21
;    75          0075  1 !               Check error return from queue manager
;    76          0076  1 !
;    77          0077  1 !       V02-003 SPF0025         Steve Forgey            08-Sep-1981
;    78          0078  1 !               Add new header fields to symbiont manager message.
;    79          0079  1 !
;    80          0080  1 !       V02-001 GWF0043         Gary W. Fowler          12-May-1981  15:20
;    81          0081  1 !               Add file size option and file size to message sent to job
;    82          0082  1 !               controller.
;    83          0083  1 !
;    84          0084  1 !       V02-000 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
;    85          0085  1 !               Previous revision history moved to [F11B.SRC]F11B.REV
;    86          0086  1 !**
;    87          0087  1
;    88          0088  1
;    89          0089  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    90          0090  1 REQUIRE 'SRC$:FCPDEF.B32';
;    91          1081  1
;    92          1082  1
;    93          1083  1 MACRO
;    94          1084  1
;    95          1085  1 ! Layout of item list for job controller.
;    96          1086  1 ! The first item is the queue name.
;    97          1087  1 ! The second item is the file identification.
;    98          1088  1 ! The third item is the delete file item code.
;    99          1089  1 ! Only fields that are filled in with non-zero values are defined.
;   100          1090  1 !
;   101          1091  1
;   102          1092  1         ITM_QNAMSIZ     = 0, 0, 16, 0 %,        ! queue name size
;   103          1093  1         ITM_QNAMCODE    = 2, 0, 16, 0 %,        ! queue name item code
;   104          1094  1         ITM_QNAMADDR    = 4, 0, 32, 0 %,        ! queue name address
;   105          1095  1
;   106          1096  1         ITM_FILEINFOSIZ = 12, 0, 16, 0 %,       ! file identification size
;   107          1097  1         ITM_FILEINFOCODE = 14, 0, 16, 0 %,      ! file id item code
;   108          1098  1         ITM_FILEINFOADDR = 16, 0, 32, 0 %,      ! file id address
;   109          1099  1
;   110          1100  1         ITM_DELFILECODE = 26, 0, 16, 0 % ;      ! delete file item code.
;   111          1101  1
;   112          1102  1
;   113          1103  1 LITERAL
;   114          1104  1         ITM_LENGTH      = 3*12 + 4,             ! 3 item codes + stopper.
```

```
;  115          1105  1          FS_LENGTH       = 16 + 2*FID$C_LENGTH + 4 + 4 + 2
;  116          1106  1                          + 16 + 2 + FI2$S_FILENAME + FI2$S_FILENAMEXT;
```

```
;  118          1107  1 GLOBAL ROUTINE SEND_SYMBIONT (HEADER, FCB) : L_NORM NOVALUE =
;  119          1108  1
;  120          1109  1 !++
;  121          1110  1 !
;  122          1111  1 ! FUNCTIONAL DESCRIPTION:
;  123          1112  1 !
;  124          1113  1 !       This routine constructs and sends a message to the symbiont manager
;  125          1114  1 !       to cause a file to be spooled and deleted.
;  126          1115  1 !
;  127          1116  1 !
;  128          1117  1 ! CALLING SEQUENCE:
;  129          1118  1 !       SEND_SYMBIONT (ARG1, ARG2)
;  130          1119  1 !
;  131          1120  1 ! INPUT PARAMETERS:
;  132          1121  1 !       ARG1: address of file header
;  133          1122  1 !       ARG2: address of file control block
;  134          1123  1 !
;  135          1124  1 ! IMPLICIT INPUTS:
;  136          1125  1 !       IO_PACKET: address of I/O packet of this request
;  137          1126  1 !
;  138          1127  1 ! OUTPUT PARAMETERS:
;  139          1128  1 !       NONE
;  140          1129  1 !
;  141          1130  1 ! IMPLICIT OUTPUTS:
;  142          1131  1 !       NONE
;  143          1132  1 !
;  144          1133  1 ! ROUTINE VALUE:
;  145          1134  1 !       NONE
;  146          1135  1 !
;  147          1136  1 ! SIDE EFFECTS:
;  148          1137  1 !       message sent to symbiont manager
;  149          1138  1 !
;  150          1139  1 !--
;  151          1140  1
;  152          1141  2 BEGIN
;  153          1142  2
;  154          1143  2 MAP
;  155          1144  2         HEADER          : REF BBLOCK,   ! file header arg
;  156          1145  2         FCB             : REF BBLOCK;   ! file control block arg
;  157          1146  2
;  158          1147  2 BUILTIN
;  159          1148  2         LOCC;
;  160          1149  2
;  161          1150  2 LINKAGE
;  162          1151  2         L_IOC_CVT       = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 5,
;  163          1152  2                                REGISTER = 4, REGISTER = 1) :
;  164          1153  2                           NOTUSED (6, 7, 8, 9, 10, 11);
;  165          1154  2
;  166          1155  2 LOCAL
;  167          1156  2         ITEMLIST        : BBLOCK [ITM_LENGTH], ! item list for jbc.
;  168          1157  2         FILEINFO_STR    : VECTOR [FS_LENGTH, BYTE], ! everything the jbc wants
;  169          1158  2                                                     ! to know about the file.
;  170          1159  2         P,                                          ! string scan pointer
;  171          1160  2         JBCSTS          : VECTOR [2],   ! io status block for SNDJBC
;  172          1161  2         LENGTH,                         ! length of converted device name
;  173          1162  2         IDENT_AREA      : REF BBLOCK,   ! address of file header ident area
;  174          1163  2         UCB             : REF BBLOCK,   ! UCB of spooled device
```

```
;  175          1164  2          VCB             : REF BBLOCK;   ! VCB of spooled device
;  176          1165  2
;  177          1166  2  BIND_COMMON;
;  178          1167  2
;  179          1168  2  EXTERNAL ROUTINE
;  180          1169  2          WAIT_FOR_AST    : L_NORM NOVALUE, ! wait for completion AST
;  181          1170  2          CONTINUE_THREAD : L_NORM NOVALUE, ! completion AST to resume thread
;  182          1171  2          IOC$CVT_DEVNAM  : L_IOC_CVT ADDRESSING_MODE (GENERAL);
;  183          1172  2                                            ! get device name of UCB
;  184          1173  2
;  185          1174  2  ! Initialize item list to zeroes.
;  186          1175  2  !
;  187          1176  2
;  188          1177  2  CH$FILL (0, ITM_LENGTH, ITEMLIST);
;  189          1178  2
;  190          1179  2  ! Get UCB and VCB addresses for the spooled device.
;  191          1180  2  !
;  192          1181  2
;  193          1182  2  UCB = .IO_PACKET[IRP$L_MEDIA];
;  194          1183  2  VCB = .UCB[UCB$L_VCB];
;  195          1184  2
;  196          1185  2  ! Point the first item at the queue name in the spooled device's VCB.
;  197          1186  2  ! This might be a little racy in that we could become unscheduled and
;  198          1187  2  ! the device set non-spooled or changed before the sndjbc service gets
;  199          1188  2  ! a chance to look at it, but that should be no more harmful than if
;  200          1189  2  ! we copied it off to local storage and it changed before the service
;  201          1190  2  ! executed anyway.  At worst the VCB gets deallocated and some garbage
;  202          1191  2  ! is picked up for the name.  BFD.
;  203          1192  2  !
;  204          1193  2
;  205          1194  2  ITEMLIST [ITM_QNAMSIZ] = .VCB [VCB$B_QNAMECNT];
;  206          1195  2  ITEMLIST [ITM_QNAMCODE] = SJC$_QUEUE;
;  207          1196  2  ITEMLIST [ITM_QNAMADDR] = VCB [VCB$B_QNAMECNT] + 1;
;  208          1197  2
;  209          1198  2  ! Fill in the file identification string.
;  210          1199  2  ! "Normal" callers of this service supply only the counted device
;  211          1200  2  ! string, the file ID and the directory ID.  The service then performs
;  212          1201  2  ! an access function on that file to recover other information it needs
;  213          1202  2  ! in a trustworthy manner.
;  214          1203  2  ! However, we ARE the file system, and therefore can be trusted.  Also,
;  215          1204  2  ! an attempt to call us back recursively with an access function just
;  216          1205  2  ! flat out will not work, because we'll be sitting here waiting for the
;  217          1206  2  ! SNDJBC service to finish.
;  218          1207  2  ! Therefore, we will also put the end of file block, access mask,
;  219          1208  2  ! and expanded file spec in this string also.  SNDJBC will look at it
;  220          1209  2  ! because we are calling from exec mode or better and use this info
;  221          1210  2  ! rather than doing an access function on the file.
;  222          1211  2  !
;  223          1212  2
;  224          1213  2  ! First get the counted device string.   Leave the trailing ":" on
;  225          1214  2  ! the string for now, because we'll want it a little later when
;  226          1215  2  ! the full file name string is built.
;  227          1216  2  !
;  228          1217  2
;  229          1218  2  IOC$CVT_DEVNAM (15, FILEINFO_STR [1],
;  230          1219  2                  .CURRENT_UCB, 0; LENGTH);
;  231          1220  2  FILEINFO_STR [0] = .LENGTH;
```

```
;  232          1221  2
;  233          1222  2 ! Pick up the file ID from the FCB instead of the file header
;  234          1223  2 ! because the RVN has already been normalized in the FCB, i.e.,
;  235          1224  2 ! APPLY_RVN has already been called.
;  236          1225  2 ! Also zero out the DID while we're at it.
;  237          1226  2 !
;  238          1227  2
;  239          1228  2 CH$COPY (FID$C_LENGTH, FCB[FCB$W_FID], 0, 2*FID$C_LENGTH, FILEINFO_STR [16]);
;  240          1229  2
;  241          1230  2 ! Next is the end of file block for the file.
;  242          1231  2 !
;  243          1232  2
;  244          1233  2 (FILEINFO_STR [28])<0,32> = ROT (.BBLOCK[HEADER[FH2$W_RECATTR],FAT$L_EFBLK], 16);
;  245          1234  2 IF .(FILEINFO_STR [28])<0,32> NEQ 0
;  246          1235  2 AND .BBLOCK[HEADER[FH2$W_RECATTR],FAT$W_FFBYTE] EQL 0
;  247          1236  2 THEN
;  248          1237  2     (FILEINFO_STR [28])<0,32> = .(FILEINFO_STR [28])<0,32> - 1;
;  249          1238  2
;  250          1239  2 ! File access mask (everything allowed).
;  251          1240  2 !
;  252          1241  2
;  253          1242  2 (FILEINFO_STR [32])<0,32> = %X'FFFFFFFF';
;  254          1243  2
;  255          1244  2 ! Build an expanded file name string including the device, null directory
;  256          1245  2 ! spec, and the file name.
;  257          1246  2 !
;  258          1247  2
;  259          1248  2 P = CH$MOVE (.FILEINFO_STR [0], FILEINFO_STR [1], FILEINFO_STR [38]);
;  260          1249  2
;  261          1250  2 ! Now that the entire device spec has been copied, adjust the count
;  262          1251  2 ! on the first string to drop the trailing ":", just like RMS does
;  263          1252  2 ! for the DVI field in the NAM block.
;  264          1253  2 !
;  265          1254  2
;  266          1255  2 FILEINFO_STR [0] = .FILEINFO_STR [0] - 1;
;  267          1256  2
;  268          1257  2 ! Add in null directory spec.
;  269          1258  2 !
;  270          1259  2
;  271          1260  2 (.P)<0,16> = '[]';
;  272          1261  2 P = .P + 2;
;  273          1262  2
;  274          1263  2 ! Now pick up the file name from the ident area in the header.
;  275          1264  2 !
;  276          1265  2
;  277          1266  2 IDENT_AREA = .HEADER + .HEADER[FH2$B_IDOFFSET] * 2;
;  278          1267  2
;  279          1268  2 P = CH$MOVE (FI2$S_FILENAME, IDENT_AREA [FI2$T_FILENAME], .P);
;  280          1269  2 P = CH$MOVE (FI2$S_FILENAMEXT, IDENT_AREA [FI2$T_FILENAMEXT], .P);
;  281          1270  2
;  282          1271  2 ! Find the end of the filename.  It is delimited by the space character
;  283          1272  2 ! if less than the full filename size.
;  284          1273  2 ! Turns out that the maximum legal filename is 39 + 1 + 39 + 1 + 5, or
;  285          1274  2 ! 85, whereas the filename plus filenamext add up to 86.
;  286          1275  2 ! Scan for 85 characters from the beginning of the concatenated filename.
;  287          1276  2 ! Set P back to the beginning of the string we just copied and scan.
;  288          1277  2 ! Using P as the second output argument causes it to get the address
```

```
;  289          1278  2 ! of the located space character after the locc executes, which will
;  290          1279  2 ! either be at the space character, if found, or at the 86'th byte.
;  291          1280  2 !
;  292          1281  2
;  293          1282  2 LOCC (%REF(' '), %REF (85), (.P - FI2$S_FILENAME - FI2$S_FILENAMEXT) ; ,P);
;  294          1283  2
;  295          1284  2 ! We skipped filling in the size field at byte 36 in this string so
;  296          1285  2 ! get it now.  The filename string started at byte 38.
;  297          1286  2
;  298          1287  2 (FILEINFO_STR [36])<0,16> = .P - FILEINFO_STR [38];
;  299          1288  2
;  300          1289  2 ! Set up the item list entry for the file information string.
;  301          1290  2 !
;  302          1291  2
;  303          1292  2 ITEMLIST [ITM_FILEINFOSIZ] = .F - FILEINFO_STR [0];
;  304          1293  2 ITEMLIST [ITM_FILEINFOCODE] = SJC$_FILE_IDENTIFICATION;
;  305          1294  2 ITEMLIST [ITM_FILEINFOADDR] = FILEINFO_STR [0];
;  306          1295  2
;  307          1296  2 ! Finally the item code to delete the file after printing.
;  308          1297  2 !
;  309          1298  2
;  310          1299  2 ITEMLIST [ITM_DELFILECODE] = SJC$_DELETE_FILE;
;  311          1300  2
;  312          1301  2 ! The status from the service is always written to the iosb and the
;  313          1302  2 ! completion ast is always delivered, regardless of status, so we
;  314          1303  2 ! do not need to separately check the status of the service call.
;  315          1304  2 !
;  316          1305  2
;  317        P 1306  2 $SNDJBC (EFN = EFN,
;  318        P 1307  2          FUNC = SJC$_ENTER_FILE,
;  319        P 1308  2          IOSB = JBCSTS,
;  320        P 1309  2          ASTADR = CONTINUE_THREAD,
;  321        P 1310  2          ASTPRM = .BASE,
;  322          1311  2          ITMLST = ITEMLIST);
;  323          1312  2
;  324          1313  2 ! The completion ast is always delivered, so we must always wait for it.
;  325          1314  2 !
;  326          1315  2
;  327          1316  2 WAIT_FOR_AST ();
;  328          1317  2
;  329          1318  2 ! A full longword of status is returned from sndjbc.
;  330          1319  2 !
;  331          1320  2
;  332          1321  2 IF NOT .JBCSTS [0]
;  333          1322  2 THEN
;  334          1323  3     BEGIN
;  335          1324  3     CLEANUP_FLAGS[CLF_DELFILE] = 1;
;  336          1325  3     USER_STATUS[1] = .JBCSTS [0];
;  337          1326  3     ERR_EXIT (SS$_NOTPRINTED);
;  338          1327  2     END;
;  339          1328  2
;  340          1329  1 END;                                    ! end of routine SEND_SYMBIONT


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

```
                                                        .EXTRN  WAIT_FOR_AST, CONTINUE_THREAD
                                                        .EXTRN  IOC$CVT_DEVNAM, SYS$SNDJBC

                                                        .PSECT  $CODE$,NOWRT,2

                                                00FC 00000      .ENTRY  SEND_SYMBIONT, Save R2,R3,R4,R5,R6,R7           ; 1107
                                             5E   FF40  CE  9E 00002    MOVAB   -192(SP), SP                           ;
                                             57     80  AA  9E 00007    MOVAB   -128(BASE), R7                         ; 1164
                                     28      00     6E      00  2C 0000B    MOVC5   #0, (SP), #0, #40, ITEMLIST            ; 1177
                                                       D8    AD      00010                                              ;
                                             50     90  AA  D0 00012    MOVL    -112(BASE), R0                         ; 1182
                                             50     38  A0  D0 00016    MOVL    56(R0), UCB                            ;
                                             50     34  A0  D0 0001A    MOVL    52(UCB), VCB                           ; 1183
                                         D8  AD     0B  A0  9B 0001E    MOVZBW  11(VCB), ITEMLIST                      ; 1194
                                         DA  AD     86  8F  9B 00023    MOVZBW  #134, ITEMLIST+2                       ; 1195
                                         DC  AD     0C  A0  9E 00028    MOVAB   12(R0), ITEMLIST+4                     ; 1196
                                             51     09  AE  9E 0002D    MOVAB   FILEINFO_STR+1, R1                     ; 1218
                                                        54  D4 00031    CLRL    R4                                     ;
                                             55     94  AA  D0 00033    MOVL    -108(BASE), R5                         ;
                                             50         0F  D0 00037    MOVL    #15, R0                                ;
                                                00000000G   00  16 0003A    JSB     IOC$CVT_DEVNAM                         ;
                                         08  AE             51  90 00040    MOVB    LENGTH, FILEINFO_STR                   ; 1220
                                             50     08  AC  D0 00044    MOVL    FCB, R0                                ; 1228
                                     0C      00     24  A0      06  2C 00048    MOVC5   #6, 36(R0), #0, #12, FILEINFO_STR+16   ;
                                                       18  AE      0004E                                              ;
                                             50     04  AC  D0 00050    MOVL    HEADER, R0                             ; 1233
                                     24  AE     1C  A0      10  9C 00054    ROTL    #16, 28(R0), FILEINFO_STR+28           ;
                                                        08  13 0005A    BEQL    1$                                     ; 1234
                                                    20  A0  B5 0005C    TSTW    32(R0)                                 ; 1235
                                                        03  12 0005F    BNEQ    1$                                     ;
                                                    24  AE  D7 00061    DECL    FILEINFO_STR+28                        ; 1237
                                             28  AE         01  CE 00064 1$:    MNEGL   #1, FILEINFO_STR+32            ; 1242
                                             50     08  AE  9A 00068    MOVZBL  FILEINFO_STR, R0                       ; 1248
                                 2E  AE      09  AE          50  28 0006C    MOVC3   R0, FILEINFO_STR+1, FILEINFO_STR+38    ;
                                                 08  AE      97 00072    DECB    FILEINFO_STR                           ; 1255
                                             83     5D5B  8F  B0 00075    MOVW    #23899, (P)+                           ; 1260
                                             50     04  BC  9A 0007A    MOVZBL  @HEADER, R0                            ; 1266
                                             56     04 BC40  3E 0007E    MOVAW   @HEADER[R0], IDENT_AREA                ;
                                         63          66      14  28 00083    MOVC3   #20, (IDENT_AREA), (P)             ; 1268
                                         63      36  A6  0042  8F  28 00087    MOVC3   #66, 54(IDENT_AREA), (P)           ; 1269
                                 AA  A3      0055    8F          20  3A 0008E    LOCC    #32, #85, -86(P)               ; 1282
                                             53         51  D0 00095    MOVL    R1, R3                                 ;
                                             50     2E  AE  9E 00098    MOVAB   FILEINFO_STR+38, R0                    ; 1287
                                 2C  AE          53         50  A3 0009C    SUBW3   R0, P, FILEINFO_STR+36                 ;
                                             50     08  AE  9E 000A1    MOVAB   FILEINFO_STR, R0                       ; 1292
                                 E4  AD          53         50  A3 000A5    SUBW3   R0, P, ITEMLIST+12                     ;
                                         E6  AD             27  B0 000AA    MOVW    #39, ITEMLIST+14                       ; 1293
                                         E8  AD     08  AE  9E 000AE    MOVAB   FILEINFO_STR, ITEMLIST+16              ; 1294
                                         F2  AD             18  B0 000B3    MOVW    #24, ITEMLIST+26                       ; 1299
                                                            5A  DD 000B7    PUSHL   BASE                                   ; 1311
                                                    0000G   CF  9F 000B9    PUSHAB  CONTINUE_THREAD                        ;
                                                        08  AE  9F 000BD    PUSHAB  JBCSTS                                 ;
                                                        D8  AD  9F 000C0    PUSHAB  ITEMLIST                               ;
                                                 7E             13  7D 000C3    MOVQ    #19, -(SP)                     ;
                                                                1E  DD 000C6    PUSHL   #30                            ;
                                         00000000G   00             07  FB 000C8    CALLS   #7, SYS$SNDJBC                 ;
                                               0000G CF             00  FB 000CF    CALLS   #0, WAIT_FOR_AST               ; 1316
```

```
                     0C              6E  E8 000D4          BLBS    JBCSTS, 2$                              ; 1321
                02   AA              20  88 000D7          BISB2   #32, 2(BASE)                          ; 1324
                04   A7              6E  D0 000DB          MOVL    JBCSTS, 4(R7)                           ; 1325
                                2184    8F  BF 000DF          CHMU    #8580                                   ; 1326
                                        04 000E3 2$:          RET                                             ; 1329

; Routine Size:  228 bytes,    Routine Base:  $CODE$ + 0000

;  341          1330  1
;  342          1331  1 END
;  343          1332  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                          228  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        37         0        1000        00:01.9



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

; Size:          228 code + 0 data bytes
; Run Time:          00:19.1
; Elapsed Time:      00:40.5
; Lines/CPU Min:      4175
; Lexemes/CPU-Min: 49081
; Memory Used:  231 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