```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
```

FILEID**LNKSYMOUT

```
LL          NN      NN  KK      KK    SSSSSSSS  YY      YY  MM      MM    000000    UU      UU  TTTTTTTTTT
LL          NN      NN  KK      KK    SSSSSSSS  YY      YY  MM      MM    000000    UU      UU  TTTTTTTTTT
LL          NN      NN  KK      KK  SS          YY      YY  MMMM  MMMM  00      00  UU      UU      TT
LL          NN      NN  KK      KK  SS          YY      YY  MMMM  MMMM  00      00  UU      UU      TT
LL          NNNN    NN  KK    KK    SS            YY  YY    MM  MM  MM  00      00  UU      UU      TT
LL          NNNN    NN  KK    KK    SS            YY  YY    MM  MM  MM  00      00  UU      UU      TT
LL          NN  NN  NN  KKKKK        SSSSSS          YY      MM      MM  00      00  UU      UU      TT
LL          NN  NN  NN  KKKKK        SSSSSS          YY      MM      MM  00      00  UU      UU      TT
LL          NN    NNNN  KK    KK            SS      YY      MM      MM  00      00  UU      UU      TT
LL          NN    NNNN  KK    KK            SS      YY      MM      MM  00      00  UU      UU      TT
LL          NN      NN  KK      KK          SS      YY      MM      MM  00      00  UU      UU      TT
LL          NN      NN  KK      KK          SS      YY      MM      MM  00      00  UU      UU      TT
LLLLLLLLLL  NN      NN  KK      KK  SSSSSSSS        YY      MM      MM    000000    UUUUUUUUUU      TT
LLLLLLLLLL  NN      NN  KK      KK  SSSSSSSS        YY      MM      MM    000000    UUUUUUUUUU      TT


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 lnk_symtblout                        ! LINKER GLOBAL SYMBOL OUTPUT ROUTINES
;     2          0002  0                 (ident = 'V04-000'
;     3          0003  0                 ,addressing_mode
;     4          0004  0                         (external    = general
;     5          0005  0                         ,nonexternal = long_relative
;     6          0006  0                         )
;     7          0007  0                 ) =
;     8          0008  1 begin
;     9          0009  1
;    10          0010  1 !
;    11          0011  1 !*****************************************************************************
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    14          0014  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    15          0015  1 !*  ALL RIGHTS RESERVED.                                                      *
;    16          0016  1 !*                                                                           *
;    17          0017  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    18          0018  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    19          0019  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    20          0020  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    21          0021  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    22          0022  1 !*  TRANSFERRED.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    25          0025  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    26          0026  1 !*  CORPORATION.                                                              *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    29          0029  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    30          0030  1 !*                                                                           *
;    31          0031  1 !*                                                                           *
;    32          0032  1 !*****************************************************************************
;    33          0033  1 !
;    34          0034  1
;    35          0035  1 !++
;    36          0036  1 ! FACILITY:     LINKER
;    37          0037  1 !
;    38          0038  1 ! ABSTRACT:     THIS MODULE CONTAINS ALL LOGIC TO OUTPUT THE GLOBAL
;    39          0039  1 !               SYMBOLS OF THE LINK TO SYMBOL TABLE FILE AND/OR IMAGE FILE
;    40          0040  1 !
;    41          0041  1 !
;    42          0042  1 ! ENVIRONMENT:  VMS NATIVE MODE
;    43          0043  1 !
;    44          0044  1 ! AUTHOR:       T.J. PORTER, CREATION DATE: 14-JUL-77
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       V03-023 ADE0005         Alan D. Eldridge        23-Aug-1984
;    49          0049  1 !               Prevent symbols from being written twice to the symbol table
;    50          0050  1 !               by flushing the symbols before writing the PSECT record.
;    51          0051  1 !
;    52          0052  1 !       V03-022 ADE0004         Alan D. Eldridge        10-Jul-1984
;    53          0053  1 !               Fix module name selection when the image name is null but there
;    54          0054  1 !               is no .STB requested.
;    55          0055  1 !
;    56          0056  1 !       V03-021 ADE0003         Alan D. Eldridge        22-Jun-1984
;    57          0057  1 !               Adhere to Grammer Rules for output file spec's as defined
```

```
;    58          0058  1 !                in the Command Language User's Guide.
;    59          0059  1 !
;    60          0060  1 !        V03-020 ADE0002         Alan D. Eldridge         1-May-1984
;    61          0061  1 !                Fix bug which resulted in zero-lenghted module name in .STB
;    62          0062  1 !                by using the .STB name if the image name is null.
;    63          0063  1 !
;    64          0064  1 !        V03-019 ADE0001         Alan D. Eldridge        12-Apr-1984
;    65          0065  1 !                Use 'output file parsing' only if /SYM was not an
;    66          0066  1 !                input file qualifier.
;    67          0067  1 !
;    68          0068  1 !        V03-018 JWT0134         Jim Teague              29-Aug-1983
;    69          0069  1 !                Undo JWT0129.  NOSHR psects in shr img sym tbls are
;    70          0070  1 !                good for forcing a CRF section.
;    71          0071  1 !
;    72          0072  1 !        V03-017 JWT0129         Jim Teague              28-Jul-1983
;    73          0073  1 !                Psect selection for inclusion in shareable image
;    74          0074  1 !                symbol tables was neglecting to check the SHR bit.
;    75          0075  1 !
;    76          0076  1 !        V03-016 JWT0113         Jim Teague              20-Apr-1983
;    77          0077  1 !                Call $getjpi to get number of open files left.
;    78          0078  1 !
;    79          0079  1 !        V03-015 JWT0053         Jim Teague              15-Sep-1982
;    80          0080  1 !                Fix bug which caused linker to skip writing some
;    81          0081  1 !                symbols to shr imgs.
;    82          0082  1 !
;    83          0083  1 !        V03-014 JWT0044         Jim Teague              30-Jul-1982
;    84          0084  1 !                Open file performance boost.
;    85          0085  1 !
;    86          0086  1 !        V03-013 JWT0038         Jim Teague              23-Jun-1982
;    87          0087  1 !                Clean up INFO#212 errors.
;    88          0088  1 !
;    89          0089  1 !--
```

```
;    91          0090  1 !
;    92          0091  1 ! TABLE OF CONTENTS:
;    93          0092  1 !
;    94          0093  1
;    95          0094  1 forward routine
;    96          0095  1         eomrecout,                              ! OUTPUT END OF MODULE
;    97          0096  1         hdrecsout,                              ! OUTPUT HEADER RECORDS
;    98          0097  1         outputpsects,                           ! TRAVERSE PSECT LIST TO OUTPUT PSECTS
;    99          0098  1         psectrecout,                            ! OUTPUT P-SECT RECORDS
;   100          0099  1         symrecout,                              ! OUTPUT SYMBOL RECORDS
;   101          0100  1         lnk$closymout : novalue,                ! CLOSE SYMBOL TABLES
;   102          0101  1         stbrecout,                              ! WRITE RECORD TO STB FILE
;   103          0102  1         imgrecout,                              ! WRITE RECORD TO IMAGE FILE
;   104          0103  1         outputrec;                              ! WRITE THE RECORDS
;   105          0104  1
;   106          0105  1 !
;   107          0106  1 ! INCLUDE FILES:
;   108          0107  1 !
;   109          0108  1 library
;   110          0109  1         'LIBL32';                       ! SYSTEM STRUCTURES
;   111          0110  1 require
;   112          0111  1         'PREFIX';                       ! USEFUL MACROS ETC.
;   113          0226  1 library
;   114          0227  1         'DATBAS';                       ! DATA BASE DEFINITIONS
;   115          0228  1 !
;   116          0229  1 ! MACROS:
;   117          0230  1 !
;   118          0231  1
;   119          0232  1 !
;   120          0233  1 ! EQUATED SYMBOLS:
;   121          0234  1 !
;   122          0235  1
;   123          0236  1 literal
;   124          0237  1         maxsymbolrec = 512;                     ! MAX LENGTH OF SYMBOL RECORD
;   125          0238  1 !
;   126          0239  1 ! EXTERNAL REFERENCES:
;   127          0240  1 !
;   128          0241  1 external literal
;   129          0242  1         lin$_closeout,                          ! CLOSE FAILURE
;   130          0243  1         lin$_faofail,                           ! FAO FAILURE
;   131          0244  1         lin$_openout,                           ! ERROR OPENING OUTPUT FILE
;   132          0245  1         lin$_writeerr,                          ! WRITE ERROR
;   133          0246  1         lnk$c_objmbc            : short;        ! MULTI-BLOCK COUNT
;   134          0247  1 !
;   135          0248  1 external
;   136          0249  1         lnk$gt_jpilst,
;   137          0250  1         lnk$gl_filesleft,
;   138          0251  1         lnk$gt_imgid            : vector[,byte],        ! IMAGE IDENT
;   139          0252  1         lnk$gl_pshrnum,                         ! NUMBER OF HIGHEST PSECT CREATED
;   140          0253  1         lnk$gl_clulst           : vector[2],    ! CLUSTER DESCRIPTOR LISTHEAD
;   141          0254  1         lnk$gl_inrelnam,                        ! POINTER TO FIRST INPUT FILE NAM BLOCK
;   142          0255  1         lnk$gl_relnam_sym,                      ! POINTER TO /SYM RELATED NAME BLOCK
;   143          0256  1         lnk$gb_locnov_sym       : byte,         ! SET IF /SYM WAS 'LOCAL' WITHOUT A VALUE
;   144          0257  1         lnk$al_imgrab           : block[,byte], ! OPEN IMAGE FILE RAB
;   145          0258  1         lnk$al_rab              : block[,byte], ! OBJECT RAB
;   146          0259  1         lnk$gb_maxercod         : byte,         ! MAXIMUM ERROR CODE
;   147          0260  1         lnk$gb_pass             : byte,         ! PASS NUMBER
```

```
;  148     0261  1          lnk$gl_ctlmsk          : block[,byte],          ! CONTROL MASK
;  149     0262  1          lnk$gl_imgfil          : ref block[,byte],      ! IMAGE FILE D.B.
;  150     0263  1          lnk$gl_symfil          : ref block[,byte],      ! SYMBOL TABLE FILE
;  151     0264  1          lnk$gw_imgifi          : word,                  ! IMAGE FILE IFI
;  152     0265  1          lnk$gl_maplst,                                  ! LISTHEAD FOR USEFUL P-SECTIONS
;  153     0266  1          lnk$gl_minva,                                   ! LOWEST VIRTUAL ADDRESS ALLOCATED
;  154     0267  1          lnk$gw_nsymbols        : word,                  ! NUMBER OF GLOBAL SYMBOLS
;  155     0268  1          lnk$gq_startim,                                 ! START TIME/DATE
;  156     0269  1          lnk$aw_version         : block[,byte];          ! LINKER VERSION
;  157     0270  1
;  158     0271  1  external routine
;  159     0272  1          lnk$closefile          : novalue,               ! ROUTINE TO CLOSE A FILE
;  160     0273  1          lib$traverse_tree,                              ! TRAVERSE A BINARY TREE
;  161     0274  1          lnk$filnamdsc,                                  ! GET FILE NAME FROM FAB
;  162     0275  1          lnk$closimgfil;                                 ! CLOSES IMAGE FILE
;  163     0276  1  !
;  164     0277  1  ! MODULE OWN STORAGE:
;  165     0278  1  !
;  166     0279  1  global
;  167     0280  1          lnk$gw_gstrecs         : word,                  ! COUNT OF RECORDS WRITTEN TO IMAGE GST
;  168     0281  1          lnk$gw_symrecs         : word;                  ! COUNT OF RECORDS WRITTEN STB FILE
;  169     0282  1  own
;  170     0283  1          eomcodes : vector [4, byte]                     ! TRANSLATE EXIT CODES
;  171     0284  1                     initial (byte (eom$c_warning         !  INTO EOM STATUS CODES
;  172     0285  1                                   ,eom$c_success
;  173     0286  1                                   ,eom$c_error
;  174     0287  1                                   ,eom$c_abort
;  175     0288  1                                  ) ),
;  176     0289  1          stbauxfnb       : ref block [,byte],            ! POINTER TO AUX. FNB. OF SYMBOL TABLE FILE
;  177     0290  1          stbrab          : $rab (rac=seq, mbc=lnk$c_objmbc), ! RECORD ACCESS BLOCK OF SYMBOL TABLE FILE
;  178     0291  1          symask          : word initial (sym$m_supres),
;  179     0292  1          symatch,
;  180     0293  1          stbfileifi,                                     ! INTERNAL FILE ID OF SYMBOL TABLE FILE
;  181     0294  1          imgauxfnb       : ref block[,byte],             ! POINTER TO AUX, FNB, OF OPEN IMAGE FILE
;  182     0295  1          gsdreclng       : word,                         ! LENGTH OF CURRENT GSD RECORD
;  183     0296  1          curpsectnum     : byte,                         ! NUMBER OF CURRENT P-SECTION
;  184     0297  1          objrecord       : ref block [,byte];            ! POINTER TO OBJECT RECORD
;  185     0298  1
;  186     0299  1  bind
;  187     0300  1          objrecvec = objrecord : ref vector [,byte];     ! POINT TO OBJECT RECORD AS BYTE VECTOR
;  188     0301  1
;  189     0302  1  psect   own = $plit$;                                   ! DEFINE READ ONLY STORAGE
;  190     0303  1  own
;  191     0304  1          abspsect : block[psc$c_size+9,byte]             ! FOR THE GENERATED ABSOLUTE P-SECTION
;  192     0305  1                     initial (long(0,0),word(0),
;  193     0306  1                              word ( gps$m_pic or        ! IT IS POSITION INDEPENDENT
;  194     0307  1                                     gps$m_rd o          ! READABLE
;  195     0308  1                                     gps$m_lib),         ! AND A "LIBRARY" P-SECTION
;  196     0309  1                              long (0,0,0,0,0,0,0,0),
;  197     0310  1                              long (0),
;  198     0311  1                              byte (0),
;  199     0312  1                              countedstring ('.$$ABS$$.')); ! NAMED ".$$ABS$$."
;  200     0313  1  !
;  201     0314  1  psect   own = $own$;
```

```
;  203          0315  1 global routine lnk$symtblout : novalue =
;  204          0316  1 !++
;  205          0317  1 ! FUNCTIONAL DESCRIPTION:
;  206          0318  1 !
;  207          0319  1 !       THIS ROUTINE OUTPUTS THE GLOBAL SYMBOLS OF THE LINK.
;  208          0320  1 !       THERE ARE THREE REASONS FOR GLOBAL SYMBOL OUTPUT:
;  209          0321  1 !
;  210          0322  1 !       1.  THE DEBUGGER HAS BEEN LINKED INTO AN EXECUTABLE
;  211          0323  1 !           IMAGE.
;  212          0324  1 !
;  213          0325  1 !       2.  THE IMAGE IS A SHAREABLE IMAGE.
;  214          0326  1 !
;  215          0327  1 !       3.  A SEPARATE OUTPUT FILE OF GLOBAL SYMBOLS WAS
;  216          0328  1 !           REQUESTED BY THE LINK COMMAND.
;  217          0329  1 !
;  218          0330  1 !       1 AND 2 ARE MUTUALLY EXCLUSIVE, WHEREAS THE THIRD
;  219          0331  1 !       MAY ACCOMPANY EITHER. IN CASES 1 AND 2 THE GLOBAL SYMBOLS
;  220          0332  1 !       ARE OUTPUT TO THE END OF THE IMAGE FILE. IN ALL CASES,
;  221          0333  1 !       THE SYMBOL TABLE OUTPUT CONFORMS TO THE OBJECT LANGUAGE
;  222          0334  1 !       FORMAT. I.E. VARIABLE LENGTH RECORDS.
;  223          0335  1 !       THERE IS SOME FILTERING OF SYMBOLS AND P-SECTIONS
;  224          0336  1 !       ARE OUTPUT:
;  225          0337  1 !
;  226          0338  1 !       1.  NO WEAKLY DEFINED SYMBOLS
;  227          0339  1 !
;  228          0340  1 !       2.  SYMBOLS FROM THE DEBUGGER ITSELF AND FROM SYSTEM
;  229          0341  1 !           LIBRARIES ARE SUPPRESSED IN ACCORDANCE WITH
;  230          0342  1 !           THE LINK COMMAND GIVEN.
;  231          0343  1 !
;  232          0344  1 ! FORMAL PARAMETERS:
;  233          0345  1 !
;  234          0346  1 !       NONE
;  235          0347  1 !
;  236          0348  1 ! IMPLICIT INPUTS:
;  237          0349  1 !
;  238          0350  1 !       THE IMAGE FILE IS OPEN AND DESCRIPTORS OF IMAGE FILE
;  239          0351  1 !       AND SYMBOL TABLE FILE ARE IN DYNAMIC MEMORY.
;  240          0352  1 !
;  241          0353  1 ! IMPLICIT OUTPUTS:
;  242          0354  1 !
;  243          0355  1 !       SYMBOLS AND P-SECTIONS (AS REQUIRED) ARE WRITTEN TO
;  244          0356  1 !       THE (APPROPRIATE) FILE(S) AND IF TO AN IMAGE,
;  245          0357  1 !       THE IMAGE HEADER IS UPDATED WITH A POINTER TO
;  246          0358  1 !       THE SYMBOL TABLE PATITION OF THE FILE.
;  247          0359  1 !
;  248          0360  1 ! ROUTINE VALUE:
;  249          0361  1 !
;  250          0362  1 ! COMPLETION CODES:
;  251          0363  1 !
;  252          0364  1 !       NONE
;  253          0365  1 !
;  254          0366  1 ! SIDE EFFECTS:
;  255          0367  1 !
;  256          0368  1 !       NONE
;  257          0369  1 !
;  258          0370  1 !--
;  259          0371  2 begin
```

```
;  260          0372  2 local
;  261          0373  2         rmserror,                                          ! RMS ERROR CODE RETURNED
;  262          0374  2         stvcode,                                           ! RMS STV CODE RETURNED
;  263          0375  2         fablock   : block [fab$c_bln,byte],                ! FILE ACCESS BLOCK
;  264          0376  2         psectdesc : ref block [,byte];                     ! POINTER TO P-SECT DESCRIPTOR
;  265          0377  2 !
;  266          0378  4 if (.lnk$gl_ctlmsk and (lnk$m_shr or lnk$m_dbg or          ! IF A SHAREABLE IMAGE
;  267          0379  2                         lnk$m_symtbl)) eql 0                ! OR DEBUGGER WITH EXECUTABLE IMAGE
;  268          0380  2 then return;                                               ! OR A SYMBOL TABLE FILE WAS REQUESTED
;  269          0381  2
;  270          0382  2 objrecord = .lnk$al_rab [rab$l_ubf];                       ! INITIALIZE OUTPUT BUFFER TO BE THE
;  271          0383  2                                                            ! ONE USED FOR INPUT RECORDS CROSSING BLOCKS
;  272        P 0384  2 $fab_init (fab = fablock                                   ! INITIALIZE THE FAB
;  273        P 0385  2           ,fop = put
;  274        P 0386  2           ,rfm = var
;  275        P 0387  2           ,mrs = maxsymbolrec
;  276          0388  2           );
;  277          0389  2
;  278          0390  2 if .lnk$gl_ctlmsk [lnk$v_symtbl]                                  !
;  279          0391  3 then    begin                                                    ! IF A SYMBOL TABLE, BUILD
;  280          0392  3         stbauxfnb           =  lnk$gl_symfil [fdb$t_auxfnb];     ! A FILE ACCESS BLOCK TO
;  281          0393  3         fablock [fab$l_fna] = .lnk$gl_symfil [fdb$l_usrnamadr];  ! WITH USER SPECIFIED OR
;  282          0394  3         fablock [fab$b_fns] = .lnk$gl_symfil [fdb$w_usrnamlen];  ! COMMAND LANGUAGE DEFAULTED
;  283          0395  4         fablock [fab$b_dns] =  (if .lnk$gb_locnov_sym
;  284          0396  4                                 then %charcount ('.STB')
;  285          0397  4                                 else %charcount ('SYS$DISK:[].STB')
;  286          0398  3                                ) ;
;  287          0399  4         fablock [fab$l_dna] =  (if .lnk$gb_locnov_sym
;  288          0400  4                                 then uplit (byte ('.STB'))
;  289          0401  4                                 else uplit (byte ('SYS$DISK:[].STB'))
;  290          0402  3                                ) ;
;  291          0403  3         fablock [fab$l_nam] = .stbauxfnb;                        !
;  292          0404  3         fablock [fab$l_alq] = .lnk$gw_nsymbols/20;               ! SET INITIAL ALLOCATION
;  293          0405  3         stbrab  [rab$l_fab] =  fablock;                          !
;  294          0406  3
;  295          0407  3         if  .lnk$gb_locnov_sym                                   ! DON'T USE 'OUTPUT FILE PARSING'
;  296          0408  3         then fablock [fab$v_ofp] = false                         ! IF /SYM WAS A LOCAL QUALIFIER
;  297          0409  3         else fablock [fab$v_ofp] = true ;                        ! WITHOUT  A SPECIFED VALUE
;  298          0410  3
;  299          0411  3         stbauxfnb [nam$l_rlf] = .lnk$gl_relnam_sym ;             ! SET RELATED NAM BLOCK ADDRESS
;  300          0412  3
;  301          0413  4         if not  ($getjpi (itmlst = lnk$gt_jpilst);
;  302          0414  4                  if .lnk$gl_filesleft leq 3
;  303          0415  4                  then
;  304          0416  4                     lnk$closefile ();                         !   THEN CLOSE A FILE
;  305          0417  4                  rmserror = $create (fab=fablock);            !   AND TRY AGAIN
;  306          0418  4                  stvcode  = .fablock [fab$l_stv];
;  307          0419  4                  ch$move (dsc$c_s_bln, lnk$filnamdsc (fablock)  ! SET RESULTANT NAME DESCRIPTOR
;  308          0420  4                           ,lnk$gl_symfil [fdb$q_filename]
;  309          0421  4                          );
;  310          0422  4                  .rmserror
;  311          0423  4                 )
;  312          0424  4         or not  (rmserror = $connect (rab=stbrab);           ! RECORD STREAM AND
;  313          0425  4                  stvcode = .stbrab [rab$l_stv];
;  314          0426  4                  .rmserror
;  315          0427  4                 )
;  316          0428  4         then begin                                           ! IF ANY FAILURE REPORT
```

```
;  317     0429  4              signal (lin$_openout,1, lnk$gl_symfil [fdb$q_filename]     ! IT
;  318     0430  4                     ,.rmserror,.stvcode
;  319     0431  4                     );
;  320     0432  6              if (.lnk$gl_ctlmsk and (lnk$m_shr or lnk$m_dbg or          ! THEN IF THERE IS
;  321     0433  4                                      lnk$m_image)) eql 0               ! NOTHING ELSE TO DO
;  322     0434  4              then return;                                               ! EXIT NOW.
;  323     0435  4              end
;  324     0436  4          else begin
;  325     0437  4              stbfileifi          = .fablock [fab$w_ifi];                ! SAVE IFI IF CREATED OK
;  326     0438  4              stbrab [rab$l_rbf] = .objrecord;                           ! SET RECORD BUFFER ADDRESS
;  327     0439  3              end;
;  328     0440  2      end;
;  329     0441  2 !
;  330     0442  2 ! IF A SHAREABLE IMAGE OR A DEBUGGER HAS BEEN LINKED IN, AND THE
;  331     0443  2 ! IMAGE FILE EXISTS (I.E. IT IS STILL OPEN), CHANGE ITS ATTRIBUTES
;  332     0444  2 ! SO THAT VARIABLE LENGTH RECORDS MAY BE WRITTEN TO THE END OF
;  333     0445  2 ! IT.
;  334     0446  2 !
;  335     0447  2 if (.lnk$gl_ctlmsk and (lnk$m_shr or lnk$m_dbg)) neq 0                 ! SHAREABLE OR DEBUGGABLE
;  336     0448  2 and .lnk$gl_ctlmsk [lnk$v_image] neq 0                                 ! IMAGE WHICH HAS BEEN
;  337     0449  3 then begin                                                             ! CREATED SUCCESSFULLY
;  338     0450  3          imgauxfnb                    = lnk$gl_imgfil [fdb$t_auxfnb];  ! (AND IS STILL OPEN). JAM
;  339     0451  3          fablock [fab$w_ifi]          = .lnk$gw_imgifi;                ! IFI, SET FOR BOTH BLOCK
;  340     0452  3          fablock [fab$v_bro]          = true;                          ! AND RECORD I/O
;  341     0453  3          fablock [fab$v_esc]          = true;                          ! AND FOR VARIABLE
;  342     0454  3          fablock [fab$l_ctx]          = rme$c_setrfm;                  ! LENGTH RECORDS
;  343     0455  3          lnk$al_imgrab [rab$l_fab] = fablock;                          ! SET FAB POINTER IN RAB
;  344     0456  3          lnk$al_imgrab [rab$v_eof] = true;                             ! AND END OF FILE OPTION
;  345     0457  3
;  346     0458  4          if not  (rmserror = $modify (fab = fablock);                  ! AND TELL RMS ABOUT IT
;  347     0459  4                   stvcode  = .fablock[fab$l_stv];
;  348     0460  4                   .rmserror
;  349     0461  4                  )
;  350     0462  4          or not  (rmserror = $connect (rab=lnk$al_imgrab);
;  351     0463  4                   stvcode = .lnk$al_imgrab [rab$l_stv];
;  352     0464  4                   .rmserror
;  353     0465  4                  )
;  354     0466  4          then begin
;  355     0467  4               signal (lin$_openout,1,lnk$gl_imgfil [fdb$q_filename]
;  356     0468  4                     ,.rmserror, .stvcode
;  357     0469  4                     );
;  358     0470  4               lnk$closymout (.imgauxfnb);                              ! THEN CLOSE THE FILE
;  359     0471  4               if .stbfileifi eql 0                                     ! IF NO OTHER SYMBOL
;  360     0472  4               then return;                                             ! TABLE FILE, EXIT
;  361     0473  4               end                                                      ! HERE NOW
;  362     0474  4          else begin
;  363     0475  4               lnk$al_imgrab [rab$b_mbc] = lnk$c_objmbc;          ! SET MULTI-BLOCK COUNT
;  364     0476  4               lnk$al_imgrab [rab$l_rbf] = .objrecord;                  ! SET RECORD BUFFER ADDRESS
;  365     0477  3               end;
;  366     0478  3      end
;  367     0479  2 else if .stbfileifi eql 0 then return;                                 ! GIVE UP IF NOTHING TO DO
;  368     0480  2 !                                                                      ! ADDRESS (USING OBJ INPUT BUFFER)
;  369     0481  2 if not hdrecsout ()                                                    ! OUTPUT HEADER RECORDS
;  370     0482  2 then return;                                                           ! AND GIVE UP ON FAILURE
;  371     0483  2 !
;  372     0484  2 if not psectrecout (abspsect)                                          ! OUTPUT THE ABSOLUTE P-SECTION
;  373     0485  2 then return;                                                           ! GIVING UP ON FAILURE
```

```
;  374          0486  2 !
;  375          0487  2 ! OUTPUT THE PSECTS
;  376          0488  2 !
;  377          0489  2 outputpsects ();
;  378          0490  2 !
;  379          0491  2 ! ALL SYMBOLS AND P-SECTIONS ARE PROCESSED. WRITE AN
;  380          0492  2 ! END OF MODULE RECORD THEN CLOSE THE FILE(S).
;  381          0493  2 !                                                                                              ;
;  382          0494  2 if not eomrecout ()                                             ! GIVE UP ON EOM RECORD
;  383          0495  2 then return;                                                    ! OUTPUT ERROR
;  384          0496  2 lnk$closymout (0);                                              ! AND CLOSE FILE(S)
;  385          0497  2 return;
;  386          0498  1 end;                                                           ! AND ALL DONE


                                                        .TITLE  LNK_SYMTBLOUT
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    00000000  00000000  00000 ABSPSECT:
                                                        .LONG   0, 0                                                ;
                                        0000  00008     .WORD   0                                                   ;
                                        0083  0000A     .WORD   131                                                 ;
00000000  00000000  00000000  00000000  00000000  00000000  0000C     .LONG   0, 0, 0, 0, 0, 0, 0                   ;
                                    00000000  00024                                                                 ;
                                    00000000  00028     .LONG   0                                                   ;
                                          00  0002C     .BYTE   0                                                   ;
                                          09  0002D     .BYTE   9                                                   ;
                  2E  24  24  53  42  41  24  24  2E  0002E     .ASCII  \.$$ABS$$.\                                 ;
                                              00037     .BLKB   1                                                   ;
                                  42  54  53  2E  00038 P.AAA:  .ASCII  \.STB\                                      ;
42  54  53  2E  5D  5B  3A  4B  53  49  44  24  53  59  53  0003C P.AAB:  .ASCII  \SYS$DISK:[].STB\                   ;

                                                        .PSECT  $OWN$,NOEXE,2

                                03  02  00  01  00000 EOMCODES:
                                                        .BYTE   1, 0, 2, 3                                          ;
                                              00004 STBAUXFNB:
                                                        .BLKB   4
                                          01  00008 STBRAB:   .BYTE   1                                             ;
                                          44  00009           .BYTE   68                                            ;
                                        0000  0000A           .WORD   0                                             ;
                                    00000000  0000C           .LONG   0                                             ;
                                    00000000  00010           .LONG   0                                             ;
                                    00000000  00014           .LONG   0                                             ;
                                       0000# 00018           .WORD   0[3]                                          ;
                                        0000  0001E           .WORD   0                                             ;
                                    00000000  00020           .LONG   0                                             ;
                                        0000  00024           .WORD   0                                             ;
                                          00  00026           .BYTE   0                                             ;
                                          00  00027           .BYTE   0                                             ;
                                        0000  00028           .WORD   0                                             ;
                                        0000  0002A           .WORD   0                                             ;
                                    00000000  0002C           .LONG   0                                             ;
                                    00000000  00030           .LONG   0                                             ;
                                    00000000  00034           .LONG   0                                             ;
```

```
00000000  00038          .LONG   0
       00  0003C          .BYTE   0
       00  0003D          .BYTE   0
       00  0003E          .BYTE   0
      00G  0003F          .BYTE   LNK$C_OBJMBC
00000000  00040          .LONG   0
00000000  00044          .LONG   0
00000000  00048          .LONG   0
     2000  0004C SYMASK:  .WORD   8192
           0004E          .BLKB   2
           00050 SYMATCH: .BLKB   4
           00054 STBFILEIFI:
                          .BLKB   4
           00058 IMGAUXFNB:
                          .BLKB   4
           0005C GSDRECLNG:
                          .BLKB   2
           0005E CURPSECTNUM:
                          .BLKB   1
           0005F          .BLKB   1
           00060 OBJRECORD:
                          .BLKB   4

                          .PSECT  $GLOBAL$,NOEXE,2

           00000 LNK$GW_GSTRECS::
                          .BLKB   2
           00002 LNK$GW_SYMRECS::
                          .BLKB   2

                    OBJRECVEC=          OBJRECORD
                          .EXTRN  LIN$_CLOSEOUT, LIN$_FAOFAIL
                          .EXTRN  LIN$_OPENOUT, LIN$_WRITEERR
                          .EXTRN  LNK$C_OBJMBC, LNK$GT_JPILST
                          .EXTRN  LNK$GL_FILESLEFT
                          .EXTRN  LNK$GT_IMGID, LNK$GL_PSHRNUM
                          .EXTRN  LNK$GL_CLULST, LNK$GL_INRELNAM
                          .EXTRN  LNK$GL_RELNAM_SYM
                          .EXTRN  LNK$GB_LOCNOV_SYM
                          .EXTRN  LNK$AL_IMGRAB, LNK$AL_RAB
                          .EXTRN  LNK$GB_MAXERCOD
                          .EXTRN  LNK$GB_PASS, LNK$GL_CTLMSK
                          .EXTRN  LNK$GL_IMGFIL, LNK$GL_SYMFIL
                          .EXTRN  LNK$GW_IMGIFI, LNK$GL_MAPLST
                          .EXTRN  LNK$GL_MINVA, LNK$GW_NSYMBOLS
                          .EXTRN  LNK$GQ_STARTIM, LNK$AW_VERSION
                          .EXTRN  LNK$CLOSEFILE, LIB$TRAVERSE_TREE
                          .EXTRN  LNK$FILNAMDSC, LNK$CLOSIMGFIL
                          .EXTRN  SYS$GETJPI, SYS$CREATE
                          .EXTRN  SYS$CONNECT, SYS$MODIFY

                          .PSECT  $CODE$,NOWRT,2

                    0FFC 00000    .ENTRY  LNK$SYMTBLOUT, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0315
                                          R9,R10,R11
5B 00000000G  00  9E 00002    MOVAB   LNK$GL_SYMFIL, R11
5A 00000000G  00  9E 00009    MOVAB   LNK$GL_CTLMSK, R10
```

```
                                 59 00000000G   00   9E 00010          MOVAB   LNK$AL_IMGRAB+60, R9                                ;
                                 58 00000000'   EF   9E 00017          MOVAB   OBJRECORD, R8                                       ;
                                 5E       B0    AE   9E 0001E          MOVAB   -80(SP), SP                                         ;
                           2044  8F             6A   B3 00022          BITW    LNK$GL_CTLMSK, #8260                                ; 0379
                                                01   12 00027          BNEQ    1$                                                  ;
                                                04   00029          RET                                                         ;
                                 68 00000000G   00   D0 0002A 1$:      MOVL    LNK$AL_RAB+36, OBJRECORD                            ; 0382
    0050   8F             00                   6E   00 2C 00031          MOVC5   #0, (SP), #0, #80, $RMS_PTR                         ; 0388
                                                6E      00038
                                 6E       5003  8F   B0 00039          MOVW    #20483, $RMS_PTR                                    ;
                             04  AE             01   D0 0003E          MOVL    #1, $RMS_PTR+4                                      ;
                             16  AE             02   90 00042          MOVB    #2, $RMS_PTR+22                                     ;
                             1F  AE             02   90 00046          MOVB    #2, $RMS_PTR+31                                     ;
                             36  AE       0200  8F   B0 0004A          MOVW    #512, $RMS_PTR+54                                   ;
                       03    01  AA             05   E0 00050          BBS     #5, LNK$GL_CTLMSK+1, 2$                             ; 0390
                                             00EF   31 00055          BRW     12$                                                 ;
                                 50             6B   D0 00058 2$:      MOVL    LNK$GL_SYMFIL, R0                                   ; 0392
                             A4  A8        26   A0   9E 0005B          MOVAB   38(R0), STBAUXFNB                                   ;
                             2C  AE        10   A0   D0 00060          MOVL    16(R0), FABLOCK+44                                  ; 0393
                             34  AE        0C   A0   90 00065          MOVB    12(R0), FABLOCK+52                                  ; 0394
                                 51 00000000G   00   9A 0006A          MOVZBL  LNK$GB_LOCNOV_SYM, R1                               ; 0395
                                 05             51   E9 00071          BLBC    R1, 3$                                              ;
                                 50             04   D0 00074          MOVL    #4, R0                                              ;
                                                03   11 00077          BRB     4$                                                  ;
                                 50             0F   D0 00079 3$:      MOVL    #15, R0                                             ;
                             35  AE             50   90 0007C 4$:      MOVB    R0, FABLOCK+53                                      ;
                                 09             51   E9 00080          BLBC    R1, 5$                                              ; 0399
                                 50 00000000'   EF   9E 00083          MOVAB   P.AAA, R0                                           ; 0400
                                                07   11 0008A          BRB     6$                                                  ;
                                 50 00000000'   EF   9E 0008C 5$:      MOVAB   P.AAB, R0                                           ; 0401
                             30  AE             50   D0 00093 6$:      MOVL    R0, FABLOCK+48                                      ; 0399
                                 50        A4   A8   D0 00097          MOVL    STBAUXFNB, R0                                       ; 0403
                             28  AE             50   D0 0009B          MOVL    R0, FABLOCK+40                                      ;
                                 52 00000000G   00   3C 0009F          MOVZWL  LNK$GW_NSYMBOLS, R2                                 ; 0404
                       10    AE             52   14   C7 000A6          DIVL3   #20, R2, FABLOCK+16                                 ;
                             E4  A8             6E   9E 000AB          MOVAB   FABLOCK, STBRAB+60                                  ; 0405
                                 06             51   E9 000AF          BLBC    R1, 7$                                              ; 0407
                             07  AE             20   8A 000B2          BICB2   #32, FABLOCK+7                                      ; 0408
                                                04   11 000B6          BRB     8$                                                  ;
                             07  AE             20   88 000B8 7$:      BISB2   #32, FABLOCK+7                                      ; 0409
                             10  A0 00000000G    00   D0 000BC 8$:      MOVL    LNK$GL_RELNAM_SYM, 16(R0)                           ; 0411
                                                7E   7C 000C4          CLRQ    -(SP)                                               ; 0413
                                                7E   D4 000C6          CLRL    -(SP)                                               ;
                                    00000000G   00   9F 000C8          PUSHAB  LNK$GT_JPILST                                       ;
                                                7E   7C 000CE          CLRQ    -(SP)                                               ;
                                                7E   D4 000D0          CLPL    -(SP)                                               ;
                       00000000G   00                   07   FB 000D2          CALLS   #7, SYS$GETJPI                              ;
                                 03 00000000G   00   D1 000D9          CMPL    LNK$GL_FILESLEFT, #3                                ; 0414
                                                07   14 000E0          BGTR    9$                                                  ;
                       00000000G   00                   00   FB 000E2          CALLS   #0, LNK$CLOSEFILE                           ; 0416
                                                5E   DD 000E9 9$:      PUSHL   SP                                                  ; 0417
                       00000000G   00                   01   FB 000EB          CALLS   #1, SYS$CREATE                              ;
                                 56             50   D0 000F2          MOVL    R0, RMSERROR                                        ;
                                 57        0C   AE   D0 000F5          MOVL    FABLOCK+12, STVCODE                                 ; 0418
                                                5E   DD 000F9          PUSHL   SP                                                  ; 0419
                       00000000G   00                   01   FB 000FB          CALLS   #1, LNK$FILNAMDSC                           ;
                                 51             6B   D0 00102          MOVL    LNK$GL_SYMFIL, R1                                   ; 0420
```

```
                14   A1              60                  08  28 00105          MOVC3   #8, (R0), 20(R1)                        ;
                                     14                  56  E9 0010A          BLBC    RMSERROR, 10$                           ; 0422
                                                   A8    A8  9F 0010D          PUSHAB  STBRAB                                  ; 0424
                                 00000000G           00  01  FB 00110          CALLS   #1, SYS$CONNECT                         ;
                                                     56  50  D0 00117          MOVL    R0, RMSERROR                            ;
                                                     57  B4  A8  D0 0011A      MOVL    STBRAB+12, STVCODE                      ; 0425
                                                     1D  56  E8 0011E          BLBS    RMSERROR, 11$                           ; 0426
                                                     7E  56  7D 00121 10$:     MOVQ    RMSERROR, -(SP)                         ; 0430
                                 7E                  6B  14  C1 00124          ADDL3   #20, LNK$GL_SYMFIL, -(SP)               ; 0429
                                                         01  DD 00128          PUSHL   #1                                      ;
                                                00000000G  8F  DD 0012A        PUSHL   #LIN$_OPENOUT                           ;
                                 00000000G           00  05  FB 00130          CALLS   #5, LIB$SIGNAL                          ;
                                       45    8F          6A  93 00137          BITB    LNK$GL_CTLMSK, #69                      ; 0433
                                                         0A  12 0013B          BNEQ    12$                                     ;
                                                             04 0013D          RET                                             ; 0434
                                       F4    A8    02    AE  3C 0013E 11$:     MOVZWL  FABLOCK+2, STBFILEIFI                   ; 0437
                                       D0    A8          68  D0 00143          MOVL    OBJRECORD, STBRAB+40                    ; 0438
                                       44    8F          6A  93 00147 12$:     BITB    LNK$GL_CTLMSK, #68                      ; 0447
                                                         7F  13 0014B          BEQL    15$                                     ;
                                                     7C      6A  E9 0014D      BLBC    LNK$GL_CTLMSK, 15$                      ; 0448
                                F8   A8 00000000G           00  26  C1 00150  ADDL3   #38, LNK$GL_IMGFIL, IMGAUXFNB          ; 0450
                                             02    AE 00000000G  00  B0 00159  MOVW    LNK$GW_IMGIFI, FABLOCK+2                ; 0451
                                             16    AE      40    8F  88 00161  BISB2   #64, FABLOCK+22                        ; 0452
                                             07    AE          08  88 00166  BISB2   #8, FABLOCK+7                          ; 0453
                                             18    AE          01  D0 0016A  MOVL    #1, FABLOCK+24                          ; 0454
                                                   69          6E  9E 0016E  MOVAB   FABLOCK, LNK$AL_IMGRAB+60               ; 0455
                                             C9    A9          01  88 00171  BISB2   #1, LNK$AL_IMGRAB+5                     ; 0456
                                                               5E  DD 00175  PUSHL   SP                                      ; 0458
                                 00000000G           00  01  FB 00177          CALLS   #1, SYS$MODIFY                          ;
                                                     56  50  D0 0017E          MOVL    R0, RMSERROR                            ;
                                                     57  0C  AE  D0 00181      MOVL    FABLOCK+12, STVCODE                     ; 0459
                                                     14  56  E9 00185          BLBC    RMSERROR, 13$                           ; 0460
                                                   C4    A9  9F 00188          PUSHAB  LNK$AL_IMGRAB                           ; 0462
                                 00000000G           00  01  FB 0018B          CALLS   #1, SYS$CONNECT                         ;
                                                     56  50  D0 00192          MOVL    R0, RMSERROR                            ;
                                                     57  D0  A9  D0 00195      MOVL    LNK$AL_IMGRAB+12, STVCODE               ; 0463
                                                     26  56  E8 00199          BLBS    RMSERROR, 14$                           ; 0464
                                                     7E  56  7D 0019C 13$:     MOVQ    RMSERROR, -(SP)                         ; 0468
                             7E  00000000G           00  14  C1 0019F          ADDL3   #20, LNK$GL_IMGFIL, -(SP)               ; 0467
                                                         01  DD 001A7          PUSHL   #1                                      ;
                                                00000000G  8F  DD 001A9        PUSHL   #LIN$_OPENOUT                           ;
                                 00000000G           00  05  FB 001AF          CALLS   #5, LIB$SIGNAL                          ;
                                                   F8    A8  DD 001B6          PUSHL   IMGAUXFNB                               ; 0470
                                 00000000V           EF  01  FB 001B9          CALLS   #1, LNK$CLOSYMOUT                       ;
                                                         0A  11 001C0          BRB     15$                                     ; 0471
                                       FB    A9          00G 90 001C2 14$:     MOVB    S^LNK$C_OBJMBC, LNK$AL_IMGRAB+55        ; 0472
                                       EC    A9          68  D0 001C6          MOVL    OBJRECORD, LNK$AL_IMGRAB+40             ; 0478
                                                         05  11 001CA          BRB     16$                                     ; 0447
                                                   F4    A8  D5 001CC 15$:     TSTL    STBFILEIFI                              ; 0479
                                                         34  13 001CF          BEQL    17$                                     ;
                                 00000000V           EF  00  FB 001D1 16$:     CALLS   #0, HDRECSOUT                           ; 0481
                                                     2A      50  E9 001D8      BLBC    R0, 17$                                 ;
                                                00000000'  EF  9F 001DB        PUSHAB  ABSPSECT                                ; 0484
                                 00000000V           EF  01  FB 001E1          CALLS   #1, PSECTRECOUT                         ;
                                                     1A      50  E9 001E8      BLBC    R0, 17$                                 ;
                                 00000000V           EF  00  FB 001EB          CALLS   #0, OUTPUTPSECTS                        ; 0489
                                 00000000V           EF  00  FB 001F2          CALLS   #0, EOMRECOUT                           ; 0494
```

```
                         09        50  E9 001F9          BLBC    R0, 17$                                 ;
                                   7E  D4 001FC          CLRL    -(SP)                                   ; 0496
                00000000V  EF      01  FB 001FE          CALLS   #1, LNK$CLOSYMOUT                       ;
                                       04 00205 17$:     RET                                             ; 0498

; Routine Size:  518 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  388          0499  1 routine hdrecsout =
;  389          0500  2 begin
;  390          0501  2 !
;  391          0502  2 ! THIS ROUTINE OUTPUTS MODULE HEADER RECORDS TO THE
;  392          0503  2 ! SYMBOL TABLE FILE.
;  393          0504  2 !
;  394          0505  2 bind     mhdrec          = .objrecord : block [,byte];
;  395          0506  2
;  396          0507  2 own      datecntrl       : descriptor('!17%D!17%D'),
;  397          0508  2          linknamever     : descriptor ('VAX-11 Linker V!AD-!AD');
;  398          0509  2
;  399          0510  2 literal  filenamelen     =  9,
;  400          0511  2          datefieldlen    = 17,
;  401          0512  2          maj_ident_lng   =  2,
;  402          0513  2          min_ident_lng   =  2;
;  403          0514  2
;  404          0515  2 local    filename        : ref block[,byte],
;  405          0516  2          modheadfield    : ref vector[,byte],
;  406          0517  2          datefield       : vector [2],
;  407          0518  2          reclng          : word;
;  408          0519  2
;  409          0520  2 bind     bufferdesc      = datefield : vector;
;  410          0521  2 !
;  411          0522  2 if  (filename = .imgauxfnb) neq 0                                ! SETUP DEFAULT MODULE FNB
;  412          0523  3 then begin                                                      !
;  413          0524  3      if  .imgauxfnb [nam$b_name] eql 0                          ! IF IMAGE NAME IS NULL
;  414          0525  3      and .stbauxfnb neq 0                                       ! AND .STB EXITS,
;  415          0526  3      then filename = .stbauxfnb;                                ! USE .STB NAME
;  416          0527  3      end
;  417          0528  2 else filename = .stbauxfnb;                                     ! USE .STB NAME IF NO IMAGE
;  418          0529  2
;  419          0530  2 objrecord [obj$b_rectyp] = obj$c_hdr;                           ! SET RECORD TYPE
;  420          0531  2 mhdrec [mhd$b_hdrtyp]    = obj$c_hdr_mhd;                      ! AND HEADER SUB-TYPE
;  421          0532  2 mhdrec [mhd$b_strlvl]    = obj$c_strlvl;                       ! SET STRUCTURE LEVEL
;  422          0533  2 mhdrec [mhd$w_recsiz]    = maxsymbolrec;                       ! SET MAX RECORD LENGTH
;  423          0534  2 mhdrec [mhd$b_namlng]    = .filename [nam$b_name];             ! SET MODULE NAME LENGTH
;  424          0535  2
;  425          0536  2 modheadfield = ch$move  (.mhdrec [mhd$b_namlng]
;  426          0537  2                         ,.filename [nam$l_name]                 ! AND COPY THE NAME, SETTING
;  427          0538  2                         , mhdrec [mhd$t_name]                   ! POINTER TO NEXT FIELD
;  428          0539  2                         ) ;
;  429          0540  2
;  430          0541  2 modheadfield [0] = .lnk$gt_imgid [0];                           ! SET LENGTH OF IDENT
;  431          0542  2 datefield [1]    = ch$move (.modheadfield [0],lnk$gt_imgid [1],modheadfield [1]); ! COPY IN THE IDENT
;  432          0543  2 datefield [0]    = 2 * datefieldlen;                           ! SET UP DESCRIPTOR FOR DATE
;  433          0544  2
;  434        P 0545  2 if not $fao (datecntrl, reclng, datefield                     ! FIELDS AND CALL FAO TO
;  435        P 0546  2             ,lnk$gq_startim, lnk$gq_startim                   ! CONVERT AND MOVE IN DATE AND TIME
;  436          0547  3             )
;  437          0548  3 then begin
;  438          0549  3      signal (lin$_faofail);                                     ! GIVE UP WITH MESSAGE IF AN ERROR
;  439          0550  3      return false;
;  440          0551  2      end;
;  441          0552  2
;  442          0553  2 reclng = .reclng + .modheadfield [0] + .mhdrec [mhd$b_namlng] + 2 +     ! COMPUTE TOTAL RECORD
;  443          0554  2           mhdrec [mhd$b_namlng] - objrecord [obj$b_rectyp];             ! LENGTH
;  444          0555  2
```

```
;  445          0556  2 if not outputrec (.reclng)                                  ! AND OUTPUT THE
;  446          0557  2 then return false;                                         ! RECORD
;  447          0558  2
;  448          0559  2 !
;  449          0560  2 ! NOW BUILD THE RECORD WITH LINKER'S NAME AND VERSION.
;  450          0561  2 !
;  451          0562  2
;  452          0563  2 objrecord [obj$b_subtyp] = obj$c_hdr_lnm;                   ! CREATOR ID HEADER
;  453          0564  2 bufferdesc [0]           = maxsymbolrec;                   ! SET LENGTH AND
;  454          0565  2 bufferdesc [1]           = objrecord [obj$b_subtyp]+1;     ! ADDRESS AND FAO
;  455        P 0566  2 if not $fao (linknamever, reclng, bufferdesc, maj_ident_lng ! FILLS IN THE RECORD.
;  456        P 0567  2                ,lnk$aw_version [lid$w_major], min_ident_lng ! WITH MAJOR AND MINOR
;  457        P 0568  2                ,lnk$aw_version [lid$w_minor]                ! LINKER IDENT
;  458          0569  3                )
;  459          0570  3 then begin                                                 ! REPORT FAO ERROR
;  460          0571  3      signal (lin$_faofail);
;  461          0572  3      return false;                                         ! AND GIVE UP
;  462          0573  2      end;
;  463          0574  2
;  464          0575  2 reclng = .reclng+.bufferdesc [1]-objrecord [obj$b_rectyp]; ! COMPUTE RECORD LENGTH
;  465          0576  2 return outputrec (.reclng)                                 ! OUTPUT THE RECORD AND RETURN STATU
;  466          0577  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 0004B  .BLKB   1
        00  00  44  25  37  31  21  44  25  37  31  21  0004C P.AAC:  .ASCII  \!17%D!17%D\<0><0>                    ;
56  20  72  65  6B  6E  69  4C  20  31  31  2D  58  41  56  00058 P.AAD:  .ASCII  \VAX-11 Linker V!AD-!AD\<0><0>      ;
                    00  00  44  41  21  2D  44  41  21  00067                                                      ;

                                                        .PSECT  $OWN$,NOEXE,2

                                 0000000A  00064 DATECNTRL:
                                                        .LONG   10                                                  ;
                                 00000000' 00068        .ADDRESS P.AAC                                              ;
                                 00000016  0006C LINKNAMEVER:
                                                        .LONG   22                                                  ;
                                 00000000' 00070        .ADDRESS P.AAD                                              ;

                                                        .EXTRN  SYS$FAO

                                                        .PSECT  $CODE$,NOWRT,2

                                      0FFC 00000 HDRECSOUT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                ; 0499
                5B 00000000V   EF  9E 00002             MOVAB   OUTPUTREC, R11                                      ;
                5A 00000000G   00  9E 00009             MOVAB   SYS$FAO, R10                                        ;
                59 00000000G   00  9E 00010             MOVAB   LNK$GQ_STARTIM, R9                                  ;
                58 00000000'   EF  9E 00017             MOVAB   OBJRECORD, R8                                       ;
                5E             0C  C2 0001E             SUBL2   #12, SP                                             ;
                52             68  D0 00021             MOVL    OBJRECORD, R2                                       ; 0505
                56             52  D0 00024             MOVL    R2, R6                                              ;
                50       F8    A8  D0 00027             MOVL    IMGAUXFNB, R0                                       ; 0522
                51             50  D0 0002B             MOVL    R0, FILENAME                                        ;
                               0A  13 0002E             BEQL    1$                                                  ;
```

```
                                                3B   A0  95 00030          TSTB    59(R0)                                  ; 0524
                                                     09  12 00033          BNEQ    2$                                      ;
                                                A4   A8  D5 00035          TSTL    STBAUXFNB                               ; 0525
                                                     04  13 00038          BEQL    2$                                      ;
                                          51    A4   A8  D0 0003A 1$:      MOVL    STBAUXFNB, FILENAME                     ; 0528
                                                     62  94 0003E 2$:      CLRB    (R2)                                    ; 0530
                                     01   A6 02000000   8F  D0 00040          MOVL    #33554432, 1(R6)                        ; 0531
                                     05   A6       3B   A1  90 00048          MOVB    59(FILENAME), 5(R6)                     ; 0534
                                          50       05   A6  9A 0004D          MOVZBL  5(R6), R0                               ; 0536
                                06   A6       4C   B1      50 28 00051       MOVC3   R0, @76(FILENAME), 6(R6)                ; 0538
                                          57            53  D0 00057          MOVL    R3, MODHEADFIELD                        ;
                                          67 00000000G  00  90 0005A          MOVB    LNK$GT_IMGID, (MODHEADFIELD)            ; 0541
                                          50            67  9A 00061          MOVZBL  (MODHEADFIELD), R0                      ; 0542
                                01   A7 00000000G  00       50 28 00064       MOVC3   R0, LNK$GT_IMGID+1, 1(MODHEADFIELD)     ;
                                               08   AE       53  D0 0006D      MOVL    R3, DATEFIELD+4                         ;
                                               04   AE       22  D0 00071      MOVL    #34, DATEFIELD                          ; 0543
                                                             59  DD 00075      PUSHL   R9                                      ; 0547
                                                             59  DD 00077      PUSHL   R9                                      ;
                                                        0C   AE  9F 00079      PUSHAB  DATEFIELD                               ;
                                                        0C   AE  9F 0007C      PUSHAB  RECLNG                                  ;
                                                        04   A8  9F 0007F      PUSHAB  DATECNTRL                               ;
                                          6A                 05  FB 00082      CALLS   #5, SYS$FAO                             ;
                                          54                 50  E9 00085      BLBC    R0, 3$                                  ;
                                          50                 6E  3C 00088      MOVZWL  RECLNG, R0                              ; 0553
                                          51                 67  9A 0008B      MOVZBL  (MODHEADFIELD), R1                      ;
                                          50                 51  C0 0008E      ADDL2   R1, R0                                  ;
                                          51            05   A6  9A 00091      MOVZBL  5(R6), R1                               ;
                                          50                 51  C0 00095      ADDL2   R1, R0                                  ;
                                          50                 56  C0 00098      ADDL2   R6, R0                                  ;
                                          50                 68  C2 0009B      SUBL2   OBJRECORD, R0                           ; 0554
                                     6E   50                 07  A1 0009E      ADDW3   #7, R0, RECLNG                          ;
                                          7E                 6E  3C 000A2      MOVZWL  RECLNG, -(SP)                           ; 0556
                                          6B                 01  FB 000A5      CALLS   #1, OUTPUTREC                           ;
                                          52                 50  E9 000A8      BLBC    R0, 5$                                  ;
                                          50                 68  D0 000AB      MOVL    OBJRECORD, R0                           ; 0563
                                     01   A0                 01  90 000AE      MOVB    #1, 1(R0)                               ;
                                     04   AE       0200      8F  3C 000B2      MOVZWL  #512, BUFFERDESC                        ; 0564
                                     08   AE         02      A0  9E 000B8      MOVAB   2(R0), BUFFERDESC+4                     ; 0565
                                          00000000G          00  9F 000BD      PUSHAB  LNK$AW_VERSION+2                        ; 0569
                                                             02  DD 000C3      PUSHL   #2                                      ;
                                          00000000G          00  9F 000C5      PUSHAB  LNK$AW_VERSION                          ;
                                                             02  DD 000CB      PUSHL   #2                                      ;
                                                   14   AE  9F 000CD          PUSHAB  BUFFERDESC                              ;
                                                   14   AE  9F 000D0          PUSHAB  RECLNG                                  ;
                                                   0C   A8  9F 000D3          PUSHAB  LINKNAMEVER                             ;
                                          6A                 07  FB 000D6      CALLS   #7, SYS$FAO                             ;
                                          0F                 50  E8 000D9      BLBS    R0, 4$                                  ;
                                          00000000G          8F  DD 000DC 3$:   PUSHL   #LIN$_FAOFAIL                          ; 0571
                                00000000G 00                 01  FB 000E2      CALLS   #1, LIB$SIGNAL                          ;
                                                             12  11 000E9      BRB     5$                                      ; 0572
                                          50                 6E  3C 000EB 4$:   MOVZWL  RECLNG, R0                              ; 0575
                                          50            08   AE  C0 000EE      ADDL2   BUFFERDESC+4, R0                        ;
                                     6E   50                 68  A3 000F2      SUBW3   OBJRECORD, R0, RECLNG                   ;
                                          7E                 6E  3C 000F6      MOVZWL  RECLNG, -(SP)                           ; 0576
                                          6B                 01  FB 000F9      CALLS   #1, OUTPUTREC                           ;
                                                             04  000FC        RET                                             ;
                                                             50  D4 000FD 5$:   CLRL    R0                                      ; 0577
```

```
                                        04 000FF        RET                                             ;

; Routine Size:  256 bytes,    Routine Base:  $CODE$ + 0206

;  467          0578  1
```

```
;  469          0579  1 routine eomrecout =
;  470          0580  2 begin
;  471          0581  2 !
;  472          0582  2 ! THIS ROUTINE BUILDS AND OUTPUTS AN END OF MODULE RECORD
;  473          0583  2 !
;  474          0584  2 objrecord [obj$b_rectyp] = obj$c_eom;                                          ! SET RECORD TYPE
;  475          0585  2 objrecord [eom$b_comcod] = .eomcodes [minu (eom$c_abort,.lnk$gb_maxercod)];    ! AND ERROR CODE
;  476          0586  2 return outputrec (eom$c_eommin);                                               ! AND OUTPUT IT
;  477          0587  1 end;


                                        0000 00000 EOMRECOUT:
                                                           .WORD   Save nothing                          ; 0579
                          51 00000000'  EF  D0 00002       MOVL    OBJRECORD, R1                         ; 0584
                          61            03  90 00009       MOVB    #3, (R1)                              ;
                          50 00000000G  00  9A 0000C       MOVZBL  LNK$GB_MAXERCOD, R0                   ; 0585
                          03            50  91 00013       CMPB    R0, #3                                ;
                                        03  1B 00016       BLEQU   1$                                    ;
                          50            03  D0 00018       MOVL    #3, R0                                ;
                    01    A1 00000000'EF40  90 0001B 1$:     MOVB    EOMCODES[R0], 1(R1)                   ;
                                        02  DD 00024       PUSHL   #2                                    ; 0586
              00000000V   EF            01  FB 00026       CALLS   #1, OUTPUTREC                         ;
                                            04 0002D       RET                                           ; 0587

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 0306
```

```
;  479     0588  1 routine outputpsects =
;  480     0589  2 begin
;  481     0590  2 !
;  482     0591  2 ! THIS ROUTINE OUTPUTS THE PSECTS TO THE SYMBOL TABLE
;  483     0592  2 !
;  484     0593  2 routine psect_out(node) =
;  485     0594  3 begin
;  486     0595  3 !
;  487     0596  3 ! THIS ROUTINE IS CALLED BY LIB$TRAVERSE_TREE FOR EACH PSECT IN THE
;  488     0597  3 ! MAPPING LIST
;  489     0598  3 !
;  490     0599  3 !
;  491     0600  3 ! THE SYMBOLS IN THE SYMBOL TABLE ARE ALL LINKED ON A (SINGLY THREADED) LIST FROM
;  492     0601  3 ! THE PROGRAM SECTIONS WITHIN WHICH THE SYMBOLS WERE DEFINED. THEREFORE TO FIND
;  493     0602  3 ! ALL SYMBOLS, WE SCAN DOWN THE LINKED LIST OF P-SECTION DESCRIPTORS, THEN DOWN
;  494     0603  3 ! THE LIST OF SYMBOLS STRUNG OFF EACH P-SECTION DESCRIPTOR.
;  495     0604  3 !
;  496     0605  3 map
;  497     0606  3     node        : ref block [,byte];
;  498     0607  3
;  499     0608  3 bind
;  500     0609  3     psectdesc   = node  [node$l_ptr]       : ref block [,byte],
;  501     0610  3     cludesc     = psectdesc [psc$l_cludsc] : ref block [,byte];
;  502     0611  3
;  503     0612  3 local
;  504     0613  3         symdesc : ref block [,byte],                    ! POINTER TO SYMBOL DESCRIPTOR
;  505     0614  3         pscoutflg,                                      ! FLAG IF PSECT WAS OUTPUT TO SYMBOL FILE
;  506     0615  3         savpscnum;                                      ! SAVED PSECT NUMBER
;  507     0616  3
;  508     0617  3 if  .lnk$gl_ctlmsk [lnk$v_shr]                          ! IF MAKING A SHAREABLE IMAGE
;  509     0618  3 and .cludesc [clu$v_shrimg]                             !  AND THIS CLUSTER IS ANOTHER SHAREABLE IMA
;  510     0619  3 then return true;                                       !  THEN SKIP THIS CLUSTER
;  511     0620  3
;  512     0621  3 if .lnk$gl_ctlmsk [lnk$v_shr]                          ! IF SHAREABLE IMAGE
;  513     0622  4 and (.psectdesc [psc$w_flags] and (gps$m_rel or gps$m_gbl or gps$m_ovr )) ! AND PSECT IS RELOCATABLE,
;  514     0623  4         eql                   (gps$m_rel or gps$m_gbl or gps$m_ovr )  !  GLOBAL, OVERLAYED
;  515     0624  4 then begin
;  516     0625  4      pscoutflg   = true;                                ! PSECT WAS OUTPUT
;  517     0626  4      curpsectnum = .curpsectnum + 1;                    ! INCREMENT P-SECTION NUMBER
;  518     0627  4      if not psectrecout(.psectdesc)                     ! OUTPUT THE P-SECTION
;  519     0628  4      then return true;                                  ! RETURNING ON ERROR
;  520     0629  4      end
;  521     0630  4 else begin
;  522     0631  4      pscoutflg  =  false;                               ! FLAG PSECT NOT OUTPUT
;  523     0632  4      savpscnum  = .curpsectnum;                         ! SAVE THE PSECT NUMBER
;  524     0633  4     curpsectnum =  0;                                   ! DEFINE THE SYMBOLS IN THE ABSOLUTE PSECT
;  525     0634  3     end;
;  526     0635  3 if (symdesc = .psectdesc [psc$l_symlst]) neq 0          ! IF THERE ARE SYMBOLS
;  527     0636  3 then do if (.symdesc [sym$w_flags] and .symask) eql .symatch    ! THAT QUALIFY FOR OUTPUT
;  528     0637  4         then begin
;  529     0638  4              if .symdesc [sym$v_redef]                  ! IF FLAGGED FOR RE-DEFINITION
;  530     0639  5              then begin
;  531     0640  5                   symdesc [sym$l_value] = .symdesc [sym$l_newval];       ! THEN RE-DEFINE VALUE
;  532     0641  5                   if  .lnk$gl_ctlmsk [lnk$v_picimg]                      ! IF IMAGE IS STILL PIC
;  533     0642  5                   and .symdesc [sym$v_rerel]                             !  AND THIS SYMBOL SHOULD BE
;  534     0643  5                   then symdesc [sym$v_rel] = true;                       !  RELOCATABLE THEN MAKE IT SO
;  535     0644  4                   end;
```

```
;  536          0645  4
;  537          0646  4              if  .lnk$gl_ctlmsk [lnk$v_picimg]                       ! IF A PIC IMAGE
;  538          0647  4              and .lnk$gl_ctlmsk [lnk$v_shr]                          ! AND A SHAREABLE IMAGE
;  539          0648  4              and .symdesc [sym$v_rel]                                ! AND SYMBOL IS RELOCATABLE
;  540          0649  4              then symdesc [sym$l_value] = .symdesc [sym$l_value] -   ! MAKE IT IMAGE RELATIVE
;  541          0650  4                                           .lnk$gl_minva
;  542          0651  4              else symdesc [sym$v_rel] = false;                       ! THEN SYMBOL IS ABSOLUTE
;  543          0652  4
;  544          0653  4             if .symdesc [sym$v_intsym]                               ! IF INTERNAL SYMBOL
;  545          0654  4             or .symdesc [sym$v_def]                                  ! OR DEFINED
;  546          0655  4             then if not symrecout(.symdesc)                          ! THEN OUTPUT THE SYMBOL
;  547          0656  4                  then return true;                                   ! GIVING UP ON AN ERROR
;  548          0657  4             end
;  549          0658  3         until (symdesc = .symdesc [sym$l_psclst]) eql 0;            ! ON FAILURE
;  550          0659  3
;  551          0660  3 if not .pscoutflg
;  552          0661  3 then curpsectnum = .savpscnum;                                      ! RESTORE PSECT NUMBER IF NECESSARY
;  553          0662  3
;  554          0663  3 return true
;  555          0664  2 end;


                                00FC 00000 PSECT_OUT:
                                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 0593
                    57 00000000G   00   9E 00002        MOVAB   LNK$GL_CTLMSK, R7               ;
                    56 00000000'   EF   9E 00009        MOVAB   CURPSECTNUM, R6                 ;
               50         04     AC       0A   C1 00010 ADDL3   #10, NODE, R0                   ; 0609
                                 52       60   D0 00015 MOVL    (R0), R2                        ; 0610
          51                     01       02   EF 00018 EXTZV   #2, #1, LNK$GL_CTLMSK, R1       ; 0617
                                 2F             51   E9 0001D BLBC    R1, 2$                  ;
                                 50         24  A2   D0 00020 MOVL    36(R2), R0              ; 0618
                           24    58       A0       02   E0 00024 BBS     #2, 88(R0), 1$          ;
                                 23             51   E9 00029 BLBC    R1, 2$                  ; 0621
                                 50        0A   A2   3C 0002C MOVZWL  10(R2), R0              ; 0622
                                 50 FFFFFFE3   8F   CA 00030 BICL2   #-29, R0                 ;
                                 1C            50   D1 00037 CMPL    R0, #28                 ; 0623
                                               13   12 0003A BNEQ    2$                      ;
                                 53            01   D0 0003C MOVL    #1, PSCOUTFLG           ; 0625
                                               66   96 0003F INCB    CURPSECTNUM             ; 0626
                                               52   DD 00041 PUSHL   R2                      ; 0627
                      00000000V  EF            01   FB 00043 CALLS   #1, PSECTRECOUT         ;
                                 09            50   E8 0004A BLBS    R0, 3$                  ;
                                               72   11 0004D 1$:  BRB     11$                     ; 0628
                                               53   D4 0004F 2$:  CLRL    PSCOUTFLG               ; 0631
                                 54            66   9A 00051 MOVZBL  CURPSECTNUM, SAVPSCNUM  ; 0632
                                               66   94 00054 CLRB    CURPSECTNUM             ; 0633
                                 52        14  A2   D0 00056 3$:  MOVL    20(R2), SYMDESC         ; 0635
                                               5F   13 0005A BEQL    10$                     ;
                                 50        0A  A2   9E 0005C 4$:  MOVAB   10(SYMDESC), R0        ; 0636
                                 51            60   3C 00060 MOVZWL  (R0), R1                ;
                                 55        EE  A6   3C 00063 MOVZWL  SYMASK, R5               ;
                                 55            55   D2 00067 MCOML   R5, R5                  ;
                                 51            55   CA 0006A BICL2   R5, R1                  ;
                           F2    A6            51   D1 0006D CMPL    R1, SYMATCH             ;
```

```
                                    42  12 00071          BNEQ    9$                                        ;
          11           60           0C  E1 00073          BBC     #12, (R0), 5$                             ; 0638
                       62      10   A2  D0 00077          MOVL    16(SYMDESC), (SYMDESC)                    ; 0640
          1E     02    A7           01  E1 0007B          BBC     #1, LNK$GL_CTLMSK+2, 6$                   ; 0641
          03     0C    A2           02  E1 00080          BBC     #2, 12(SYMDESC), 5$                       ; 0642
                       60           08  88 00085          BISB2   #8, (R0)                                  ; 0643
          11     02    A7           01  E1 00088 5$:      BBC     #1, LNK$GL_CTLMSK+2, 6$                   ; 0646
          0D           67           02  E1 0008D          BBC     #2, LNK$GL_CTLMSK, 6$                     ; 0647
          09           60           03  E1 00091          BBC     #3, (R0), 6$                              ; 0648
                       62 00000000G 00  C2 00095          SUBL2   LNK$GL_MINVA, (SYMDESC)                   ; 0650
                                    03  11 0009C          BRB     7$                                        ; 0649
                       60           08  8A 0009E 6$:      BICB2   #8, (R0)                                  ; 0651
          04           60           0A  E0 000A1 7$:      BBS     #10, (R0), 8$                             ; 0653
          0C           60           01  E1 000A5          BBC     #1, (R0), 9$                              ; 0654
                                    52  DD 000A9 8$:      PUSHL   SYMDESC                                   ; 0655
             00000000V EF           01  FB 000AB          CALLS   #1, SYMRECOUT                             ;
                       0C           50  E9 000B2          BLBC    R0, 11$                                   ;
                       52      04   A2  D0 000B5 9$:      MOVL    4(SYMDESC), SYMDESC                       ; 0658
                                    A1  12 000B9          BNEQ    4$                                        ;
                       03           53  E8 000BB 10$:     BLBS    PSCOUTFLG, 11$                            ; 0660
                       66           54  90 000BE          MOVB    SAVPSCNUM, CURPSECTNUM                    ; 0661
                       50           01  D0 000C1 11$:     MOVL    #1, R0                                    ; 0663
                                        04 000C4          RET                                               ; 0664

; Routine Size:  197 bytes,    Routine Base:  $CODE$ + 0334


;   556          0665  2 !
;   557          0666  2 ! MAIN BODY OF OUTPUTPSECTS
;   558          0667  2 !
;   559          0668  2 if not .lnk$gl_ctlmsk [lnk$v_shr]
;   560          0669  2 then symask = .symask or gsy$m_weak                     ! IF NOT SHAREABLE, EXCLUDE WEAK SYMBOLS
;   561          0670  3 else begin
;   562          0671  3      symatch = gsy$m_uni;                               ! IF SHAREABLE, SYMBOLS MUST BE UNIVERSAL
;   563          0672  3      symask = .symask or gsy$m_uni;
;   564          0673  2    end;
;   565          0674  2 !
;   566          0675  2 ! TRAVERSE THE TREE AND OUTPUT THE PSECTS
;   567          0676  2 !
;   568          0677  2 lib$traverse_tree (lnk$gl_maplst,psect_out);
;   569          0678  2
;   570          0679  2 return outputrec (.gsdreclng)                           ! RETURN, OUTPUTTING ANY PARTIAL RECORD
;   571          0680  1 end;                                                    !OF OUTPUTPSECTS


                                   0004 00000 OUTPUTPSECTS:
                                                          .WORD   Save R2                                   ; 0588
                       52 00000000'  EF  9E 00002          MOVAB   SYMASK, R2                                ;
          05 00000000G 00            02  E0 00009          BBS     #2, LNK$GL_CTLMSK, 1$                     ; 0668
                       62            01  88 00011          BISB2   #1, SYMASK                                ; 0669
                                     07  11 00014          BRB     2$                                        ;
                 04    A2            04  D0 00016 1$:      MOVL    #4, SYMATCH                               ; 0671
                       62            04  88 0001A          BISB2   #4, SYMASK                                ; 0672
                             FF1A    CF  9F 0001D 2$:      PUSHAB  PSECT_OUT                                 ; 0677
```

```
                          00000000G  00  9F 00021         PUSHAB  LNK$GL_MAPLST                          ;
                00000000G 00         02  FB 00027         CALLS   #2, LIB$TRAVERSE_TREE                 ;
                          7E     10  A2  3C 0002E         MOVZWL  GSDRECLNG, -(SP)                      ; 0679
                00000000V EF         01  FB 00032         CALLS   #1, OUTPUTREC                         ;
                                     04  00039         RET                                           ; 0680

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 03F9
```

```
;  573     0681  1 routine stbpscrecout(psectdesc) =
;  574     0682  2 begin
;  575     0683  2 !
;  576     0684  2 ! THIS ROUTINE OUTPUTS A PSECT DEFINITION RECORD TO THE STB FILE.
;  577     0685  2 !
;  578     0686  2 map
;  579     0687  2     psectdesc : ref block[,byte];
;  580     0688  2
;  581     0689  2 local
;  582     0690  2     psectdefrec : ref block[,byte];
;  583     0691  2
;  584     0692  2 if  .stbfileifi eql 0                                   ! IF NO STB FILE
;  585     0693  2 and .psectdesc [psc$v_rel]                              !  AND PSECT IS RELOCATABLE
;  586     0694  2 then return true;                                       !  THEN JUST SKIP IT
;  587     0695  2
;  588     0696  2 if .gsdreclng gtru 0                                    ! FLUSH BUFFER
;  589     0697  3 then begin
;  590     0698  3      if not outputrec (.gsdreclng)                      ! WRITE IT OUT
;  591     0699  3      then return false;                                 ! AND ZERO THE LENGTH
;  592     0700  3      gsdreclng = 0;
;  593     0701  2      end;
;  594     0702  2
;  595     0703  2 if .gsdreclng eql 0                                     ! IF BEGINNING A NEW
;  596     0704  3 then begin                                              ! GSD RECORD, SET
;  597     0705  3      objrecord [obj$b_rectyp] = obj$c_gsd;              ! RECORD TYPE AND INITIALIZE
;  598     0706  3      gsdreclng = 1;                                     ! THE LENGTH
;  599     0707  2      end;
;  600     0708  2
;  601     0709  2 psectdefrec = objrecvec [,gsdreclng];                   ! POINT TO P-SECTION PART OF RECORD
;  602     0710  2 psectdefrec [gps$b_gsdtyp] = gsd$c_psc;                 ! SET SUBRECORD TYPE
;  603     0711  2 psectdefrec [gps$b_align]  = .psectdesc [psc$b_align];  ! COPY ALIGNMENT
;  604     0712  2 psectdefrec [gps$w_flags]  = .psectdesc [psc$w_flags]   ! COPY FLAGS,
;  605     0713  3                              and not (psc$m_optpsc or psc$m_usrpsc or  ! AND CLEAR UNINTERESTING BITS
;  606     0714  3                                       psc$m_supres or psc$m_shrimg
;  607     0715  2                                      );
;  608     0716  2 psectdefrec [gps$l_alloc]  = .psectdesc [psc$l_base];   ! SET ALLOCATION AS PSECT BASE
;  609     0717  2 psectdefrec [gps$b_namlng] = .psectdesc [psc$b_namlng]; ! COPY LENGTH OF NAME
;  610     0718  2
;  611     0719  2 gsdreclng = .gsdreclng + ch$move (.psectdesc [psc$b_namlng]     ! AND THEN THE NAME AND UPDATE
;  612     0720  2                                  , psectdesc [psc$t_name]       ! LENGTH OF GSD RECORD
;  613     0721  2                                  , psectdefrec [gps$t_name]
;  614     0722  2                                  ) - .psectdefrec;
;  615     0723  2
;  616     0724  2 if  .imgauxfnb neq 0                                    ! IF ALSO WRITING TO IMAGE FILE
;  617     0725  2 and .psectdefrec [gps$v_rel]                            !  AND THIS IS A RELOCATABLE PSECT
;  618     0726  3 then begin
;  619     0727  3      stbrecout (.gsdreclng);                            ! THEN OUTPUT THE RECORD TO THE STB FILE
;  620     0728  3      gsdreclng = 0;
;  621     0729  2      end;
;  622     0730  2
;  623     0731  2 return true
;  624     0732  1 end;
```

```
                                        01FC 00000 STBPSCRECOUT:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8                     ; 0681
                   58 00000000'    EF   9E 00002          MOVAB   GSDRECLNG, R8                                 ;
                              F8   A8   D5 00009          TSTL    STBFILEIFI                                    ; 0692
                              09   12 0000C               BNEQ    1$                                            ;
                   50         04   AC   D0 0000E          MOVL    PSECTDESC, R0                                 ; 0693
        69         0A     A0   03   E0 00012              BBS     #3, 10(R0), 4$                                ;
                   50           68   3C 00017 1$:         MOVZWL  GSDRECLNG, R0                                 ; 0696
                                10   13 0001A             BEQL    2$                                            ;
                                50   DD 0001C             PUSHL   R0                                            ; 0698
           00000000V   EF       01   FB 0001E             CALLS   #1, OUTPUTREC                                 ;
                       5C       50   E9 00025             BLBC    R0, 5$                                        ;
                                68   B4 00028             CLRW    GSDRECLNG                                     ; 0700
                                07   12 0002A             BNEQ    3$                                            ; 0703
                04     B8       01   90 0002C 2$:         MOVB    #1, @OBJRECORD                                ; 0705
                       68       01   B0 00030             MOVW    #1, GSDRECLNG                                 ; 0706
                       57       68   3C 00033 3$:         MOVZWL  GSDRECLNG, R7                                 ; 0709
        56             57     04   A8   C1 00036          ADDL3   OBJRECVEC, R7, PSECTDEFREC                    ;
                       66       94 0003B                  CLRB    (PSECTDEFREC)                                 ; 0710
                       50     04   AC   D0 0003D          MOVL    PSECTDESC, R0                                 ; 0711
                01     A6     2C   A0   90 00041          MOVB    44(R0), 1(PSECTDEFREC)                        ;
  02    A6      0A     A0   3C00   8F   AB 00046          BICW3   #15360, 10(R0), 2(PSECTDEFREC)                ; 0713
                04     A6     18   A0   D0 0004E          MOVL    24(R0), 4(PSECTDEFREC)                        ; 0716
                08     A6     2D   A0   90 00053          MOVB    45(R0), 8(PSECTDEFREC)                        ; 0717
                       51     2D   A0   9A 00058          MOVZBL  45(R0), R1                                    ; 0719
  09    A6      2E     A0       51   28 0005C             MOVC3   R1, 46(R0), 9(PSECTDEFREC)                    ; 0721
        50             57       53   C1 00062             ADDL3   R3, R7, R0                                    ; 0719
        68             50       56   A3 00066             SUBW3   PSECTDEFREC, R0, GSDRECLNG                    ; 0722
                              FC   A8   D5 0006A          TSTL    IMGAUXFNB                                     ; 0724
                                11   13 0006D             BEQL    4$                                            ;
        0C      02     A6       03   E1 0006F             BBC     #3, 2(PSECTDEFREC), 4$                        ; 0725
                       7E       68   3C 00074             MOVZWL  GSDRECLNG, -(SP)                              ; 0727
           00000000V   EF       01   FB 00077             CALLS   #1, STBRECOUT                                 ;
                                68   B4 0007E             CLRW    GSDRECLNG                                     ; 0728
                       50       01   D0 00080 4$:         MOVL    #1, R0                                        ; 0731
                                     04 00083             RET                                                   ;
                                50   D4 00084 5$:         CLRL    R0                                            ; 0732
                                     04 00086             RET                                                   ;

; Routine Size:  135 bytes,    Routine Base:  $CODE$ + 0433
```

```
;  626          0733  1 routine imgpscrecout(psectdesc) =
;  627          0734  2 begin
;  628          0735  2 !
;  629          0736  2 ! THIS ROUTINE OUTPUTS A PSECT DEFINITION RECORD TO THE IMAGE FILE
;  630          0737  2 !
;  631          0738  2 map
;  632          0739  2     psectdesc    : ref block [,byte];
;  633          0740  2
;  634          0741  2 local
;  635          0742  2     psectdefrec : ref block [,byte];
;  636          0743  2
;  637          0744  2 if not .psectdesc [psc$v_rel]                          ! IF PSECT IS ABSOLUTE
;  638          0745  3 then begin
;  639          0746  3      outputrec (.gsdreclng);                           ! OUTPUT THE RECORD, PSECT DEF ALREADY SET U
;  640          0747  3      gsdreclng = 0;
;  641          0748  3      return true;
;  642          0749  2      end;
;  643          0750  2
;  644          0751  2 if  .gsdreclng gtru 0                                  ! FLUSH BUFFER
;  645          0752  3 then begin
;  646          0753  3      if not outputrec (.gsdreclng)                     ! WRITE IT OUT
;  647          0754  3      then return false;                                ! AND ZERO THE LENGTH
;  648          0755  3      gsdreclng = 0;
;  649          0756  2      end;
;  650          0757  2
;  651          0758  2 if  .gsdreclng eql 0                                   ! IF BEGINNING A NEW
;  652          0759  3 then begin                                             ! GSD RECORD, SET
;  653          0760  3      objrecord [obj$b_rectyp] = obj$c_gsd;             ! RECORD TYPE AND INITIALIZE
;  654          0761  3      gsdreclng = 1;                                    ! THE LENGTH
;  655          0762  2      end;
;  656          0763  2
;  657          0764  2 psectdefrec = objrecvec [.gsdreclng];                  ! POINT TO P-SECTION PART OF RECORD
;  658          0765  2 psectdefrec [gps$b_gsdtyp] = gsd$c_spsc;                        ! SET SUBRECORD TYPE
;  659          0766  2 psectdefrec [gps$b_align]  = .psectdesc [psc$b_align];  ! COPY ALIGNMENT
;  660          0767  2 psectdefrec [gps$w_flags]  = .psectdesc [psc$w_flags]   ! COPY FLAGS,
;  661          0768  3                             and not (psc$m_optpsc or psc$m_usrpsc or   ! AND CLEAR UNINTERESTING BITS
;  662          0769  3                                      psc$m_supres or psc$m_shrimg
;  663          0770  2                                     );
;  664          0771  2 psectdefrec [sgps$l_alloc] = .psectdesc [psc$l_length]; ! SET PSECT ALLOCATION
;  665          0772  3 psectdefrec [sgps$l_base]  = (if .lnk$gl_ctlmsk [lnk$v_picimg]  ! IF A PIC IMAGE
;  666          0773  3                             then .psectdesc [psc$l_base] -    ! THEN RECORD BASE AS IMAGE RELATIVE
;  667          0774  3                                  .lnk$gl_minva
;  668          0775  3                             else .psectdesc [psc$l_base]      ! OTHERWISE ACTUAL ADDRESS
;  669          0776  2                            );
;  670          0777  2
;  671          0778  2 psectdefrec [sgps$b_namlng] = .psectdesc [psc$b_namlng];  ! SET LENGTH OF NAME
;  672          0779  2
;  673          0780  2 gsdreclng = .gsdreclng + ch$move (.psectdesc [psc$b_namlng]     ! COPY THE P-SECTION NAME
;  674          0781  2                                 , psectdesc [psc$t_name]       ! AND UPDATE RECORD LENGTH
;  675          0782  2                                 , psectdefrec [sgps$t_name]
;  676          0783  2                                 ) - .psectdefrec;
;  677          0784  2
;  678          0785  2 if .stbfileifi neq 0                                   ! IF ALSO WRITING STB FILE
;  679          0786  3 then begin
;  680          0787  3     imgrecout (.gsdreclng);                           ! THEN OUTPUT THE RECORD
;  681          0788  3     gsdreclng = 0;
;  682          0789  2     end;
```

```
;  683          0790  2
;  684          0791  2 return true
;  685          0792  1 end;


                                                03FC 00000 IMGPSCRECOUT:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 0733
                                            59 00000000V    EF  9E 00002        MOVAB   OUTPUTREC, R9                   ;
                                            58 00000000'    EF  9E 00009        MOVAB   GSDRECLNG, R8                   ;
                                            52      04      AC  D0 00010        MOVL    PSECTDESC, R2                   ; 0744
                                    08      0A      A2      03  E0 00014        BBS     #3, 10(R2), 1$                  ;
                                                    7E      68  3C 00019        MOVZWL  GSDRECLNG, -(SP)                ; 0746
                                                    69      01  FB 0001C        CALLS   #1, OUTPUTREC                   ;
                                                            76  11 0001F        BRB     6$                              ; 0747
                                                    50      68  3C 00021 1$:    MOVZWL  GSDRECLNG, R0                   ; 0751
                                                            0C  13 00024        BEQL    2$                              ;
                                                    50      DD  00026           PUSHL   R0                              ; 0753
                                                    69      01  FB 00028        CALLS   #1, OUTPUTREC                   ;
                                                    6F      50  E9 0002B        BLBC    R0, 8$                          ;
                                                            68  B4 0002E        CLRW    GSDRECLNG                       ; 0755
                                                            07  12 00030        BNEQ    3$                              ; 0758
                                            04      B8      01  90 00032 2$:    MOVB    #1, @OBJRECORD                  ; 0760
                                                    68      01  B0 00036        MOVW    #1, GSDRECLNG                   ; 0761
                                                    57      68  3C 00039 3$:    MOVZWL  GSDRECLNG, R7                   ; 0764
                                    56              57      04  A8 C1 0003C        ADDL3   OBJRECVEC, R7, PSECTDEFREC      ;
                                                    66      0C  90 00041        MOVB    #12, (PSECTDEFREC)              ; 0765
                                            01      A6      2C  A2 90 00044     MOVB    44(R2), 1(PSECTDEFREC)          ; 0766
                                02          A6      0A      A2  3C00 8F AB 00049    BICW3   #15360, 10(R2), 2(PSECTDEFREC)  ; 0768
                                            04      A6      1C  A2 D0 00051     MOVL    28(R2), 4(PSECTDEFREC)          ; 0771
                                    0B 00000000G    00          01 E1 00056     BBC     #1, LNK$GL_CTLMSK+2, 4$         ; 0772
                                    50          18  A2 00000000G    00 C3 0005E     SUBL3   LNK$GL_MINVA, 24(R2), R0        ; 0774
                                                            04  11 00067        BRB     5$                              ; 0773
                                                    50      18  A2 D0 00069 4$:    MOVL    24(R2), R0                  ; 0775
                                            08      A6          50 D0 0006D 5$:    MOVL    R0, 8(PSECTDEFREC)          ; 0772
                                            0C      A6      2D  A2 90 00071     MOVB    45(R2), 12(PSECTDEFREC)         ; 0778
                                                    50      2D  A2 9A 00076     MOVZBL  45(R2), R0                      ; 0780
                                D0      A6          2E      A2      50 28 0007A     MOVC3   R0, 46(R2), 13(PSECTDEFREC)     ; 0782
                                        50                  57      53 C1 00080     ADDL3   R3, R7, R0                      ; 0780
                                        68                  50      56 A3 00084     SUBW3   PSECTDEFREC, R0, GSDRECLNG      ; 0783
                                                                F8  A8 D5 00088     TSTL    STBFILEIFI                      ; 0785
                                                                    0C 13 0008B     BEQL    7$                              ;
                                                    7E              68 3C 0008D     MOVZWL  GSDRECLNG, -(SP)                ; 0787
                                        00000000V   EF              01 FB 00090     CALLS   #1, IMGRECOUT                   ;
                                                                    68 B4 00097 6$:     CLRW    GSDRECLNG               ; 0788
                                                    50              01 D0 00099 7$:     MOVL    #1, R0                  ; 0791
                                                                    04 0009C        RET                                 ;
                                                                    50 D4 0009D 8$:     CLRL    R0                      ; 0792
                                                                    04 0009F        RET                                 ;

; Routine Size:  160 bytes,    Routine Base:  $CODE$ + 04BA
```

```
;  687          0793  1 routine psectrecout(psectdesc) =
;  688          0794  2 begin
;  689          0795  2 !
;  690          0796  2 ! THIS ROUTINE OUTPUTS A P-SECTION DEFINITION RECORD. IT ASSUMES THAT GSD
;  691          0797  2 ! RECORDS ARE BEING WRITTEN AND BLOCKED UP. IF ANOTHER P-SECTION DEFINITION
;  692          0798  2 ! RECORD WILL NOT FIT IN THE CURRENT GSD RECORD, THE RECORD IS WRITTEN
;  693          0799  2 ! AND ANOTHER BEGUN.
;  694          0800  2 !
;  695          0801  2 map
;  696          0802  2     psectdesc : ref block[,byte];                          ! BLOCK POINTER
;  697          0803  2
;  698          0804  2 stbpscrecout(.psectdesc);                                  ! OUTPUT TO STB FILE
;  699          0805  2
;  700          0806  2 if  .imgauxfnb neq 0                                       ! IF WRITING TO IMAGE FILE
;  701          0807  2 then imgpscrecout (.psectdesc);                            !  THEN OUTPUT TO IMAGE FILE
;  702          0808  2
;  703          0809  2 return true                                                ! AND ALL DONE.
;  704          0810  1 end;

                                    0000 00000 PSECTRECOUT:
                                                       .WORD   Save nothing                ; 0793
                                04   AC  DD 00002        PUSHL   PSECTDESC                   ; 0804
                          FECF  CF       01  FB 00005        CALLS   #1, STBPSCRECOUT
                                00000000' EF  D5 0000A        TSTL    IMGAUXFNB           ; 0806
                                          08  13 00010        BEQL    1$
                                04   AC  DD 00012        PUSHL   PSECTDESC               ; 0807
                          FF46  CF       01  FB 00015        CALLS   #1, IMGPSCRECOUT
                                50       01  D0 0001A 1$:    MOVL    #1, R0              ; 0809
                                         04  0001D          RET                         ; 0810

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 055A
```

```
;  706          0811  1 routine symrecout (symdesc) =
;  707          0812  2 begin
;  708          0813  2 !
;  709          0814  2 ! THIS ROUTINE BLOCKS SYMBOL DEFINITION RECORDS INTO GSD RECORDS
;  710          0815  2 ! AND OUTPUTS THEM TO THE SYMBOL TABLE.
;  711          0816  2 !
;  712          0817  2 map symdesc : ref block[,byte];
;  713          0818  2 local   symdefrec : ref block[,byte],
;  714          0819  2         symbolstring : ref vector[,byte],
;  715          0820  2         valdatlng,                              ! LENGTH OF ARG VALIDATION DATA
;  716          0821  2         masklength;
;  717          0822  2 bind symdscnam = .symdesc - .symdesc[sym$b_namlng] - snb$c_fxdlen : block[,byte]; ! POINT TO NAME PART
;  718          0823  2 if (.symdesc[sym$w_flags] and sym$m_entmsk) neq 0                                 ! IF THERE IS AN ENTRY
;  719          0824  2 then masklength = 2                                                               ! MASK, SET THE EXTRA
;  720          0825  2 else masklength = 0;                                                             ! LENGTH
;  721          0826  2 if .symdesc[sym$l_valdata] neq 0                                                 ! IF THERE IS VALIDATION DATA
;  722          0827  3 then begin
;  723          0828  3     bind
;  724          0829  3         argvaldata = symdesc[sym$l_valdata] : ref vector[,byte];                 ! NAME IT
;  725          0830  3     valdatlng = (.argvaldata[0]-2)*2 + 2;                                       ! GET LENGTH OF VALIDATION INFORMATI
;  726          0831  3     end
;  727          0832  2 else valdatlng = 0;                                                             ! OTHERWISE THERE IS NONE
;  728          0833  3 if (.gsdreclng+.masklength+.symdesc[sym$b_namlng]+.valdatlng+                   ! IF THIS SYMBOL WOULD
;  729          0834  2                         sdf$c_name) gtru maxsymbolrec                          ! OVERFLOW THE CURRENT
;  730          0835  3 then begin                                                                      ! RECORD, THEN OUTPUT
;  731          0836  3         if not outputrec(.gsdreclng)                                            ! CURRENT RECORD AND
;  732          0837  3         then return false;                                                      ! EXIT ON ERROR
;  733          0838  3         gsdreclng = 0;                                                          ! RESET RECORD LENGTH
;  734          0839  2      end;
;  735          0840  2 if .gsdreclng eql 0                                                             ! SET NEW RECORD AS A
;  736          0841  3 then begin
;  737          0842  3         objrecord[obj$b_rectyp] = obj$c_gsd;                                    ! GSD RECORD
;  738          0843  3         gsdreclng = 1;
;  739          0844  2      end;
;  740          0845  2 symdefrec = objrecvec [.gsdreclng];                                            ! SET POINTER TO SYMBOL
;  741          0846  2 if .valdatlng neq 0                                                            ! IF THERE IS VALIDATION DATA
;  742          0847  3 then begin
;  743          0848  3     bind
;  744          0849  3         argvaldata = symdesc[sym$l_valdata] : ref vector[,byte],                ! POINT TO VALIDATION DATA
;  745          0850  3         formaldata = symdefrec[pro$t_name]+
;  746          0851  3                      .symdesc[sym$b_namlng] : block[,byte];                     ! POINTER TO THE FIXED PART OF FORMA
;  747          0852  3     symdefrec[pro$w_mask] = .symdesc[sym$w_entmsk];                             ! SET THE ENTRY MASK
;  748          0853  3     symbolstring = symdefrec[pro$b_namlng];                                     ! POINT TO THE NAME
;  749          0854  3     symdefrec[pro$b_gsdtyp] = obj$c_gsd_pro;                                    ! PROCEDURE DEFINITION
;  750          0855  3     formaldata[fml$b_minargs] = .argvaldata[1];                                 ! SET MINIMUM ARG COUNT
;  751          0856  3     formaldata[fml$b_maxargs] = .argvaldata[0] - 2;                             ! AND MAXIMUM
;  752          0857  3     incr i from 1 to .formaldata[fml$b_maxargs]                                 ! LOOP THROUGH THE ARGUMENTS
;  753          0858  4     do begin
;  754          0859  4         bind
;  755          0860  4             argdesc =
;  756          0861  4                 formaldata[fml$b_maxargs]+1+((.i-1)*arg$c_size) : block[,byte]; ! POINT TO CURRENT ARG DESCR
;  757          0862  4         argdesc[arg$b_valctl] = .(argvaldata[1] + .i);                          ! GET NEXT DESCRIPTOR
;  758          0863  4         argdesc[arg$b_bytecnt] = 0;                                             ! NO OTHER DESCRIPTOR BYTES
;  759          0864  3         end;
;  760          0865  3     end
;  761          0866  2 else if .masklength neq 0
;  762          0867  3 then begin                                                                     ! TO SYMBOL NAME
```

```
;  763          0868  3          symdefrec[epm$w_mask]=.symdesc[sym$w_entmsk];                    ! STRING AND IF AN
;  764          0869  3          symbolstring = symdefrec[epm$b_namlng];                          ! ENTRY POINT DEFINITION
;  765          0870  3          symdefrec[epm$b_gsdtyp] = obj$c_gsd_epm                          ! SET THE GSD TYPE
;  766          0871  3      end                                                                  ! ALSO COPY THE ENTRY
;  767          0872  3  else begin                                                               ! POINT MASK
;  768          0873  3          symbolstring = symdefrec[sdf$b_namlng];                          ! DO LIKEWISE FOR
;  769          0874  3          symdefrec[sdf$b_gsdtyp] = obj$c_gsd_sym;                         ! ORDINARY SYMBOL
;  770          0875  2      end;                                                                 ! DEFINITION
;  771          0876  2  symdefrec[sdf$b_datyp] = .symdesc[sym$b_datyp];                          ! COPY DATA TYPE
;  772          0877  3  symdefrec[sdf$w_flags] = .symdesc[sym$w_flags] and (gsy$m_rel or        ! AND FLAGS
;  773          0878  2                                    gsy$m_weak or gsy$m_uni or gsy$m_def);
;  774          0879  2  if not .symdesc[sym$v_rel]                                              ! IF ABSOLUTE P-SECTION
;  775          0880  2      then symdefrec[sdf$b_psindx] = 0                                     ! SET OWNING P-SECT NUMBER = 0
;  776          0881  2      else symdefrec[sdf$b_psindx] = .curpsectnum;                         ! SET OWNING P-SECT
;  777          0882  2  symdefrec[sdf$l_value] = .symdesc[sym$l_value];                         ! SYMBOL VALUE
;  778          0883  2  gsdreclng = .gsdreclng+ch$move(.symdscnam[snb$b_namlng]+1,              ! COPY THE SYMBOL
;  779          0884  2                               symdscnam[snb$b_namlng],symbolstring[0])-  ! NAME (COUNTED STRING)
;  780          0885  2                               .symdefrec+.valdatlng;                     ! AND UPDATE LENGTH
;  781          0886  2  return true;                                                             ! AND IT IS ALL
;  782          0887  1  end;                                                                    ! DONE.


                                    0FFC 00000 SYMRECOUT:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0811
                          5B 00000000'   EF  9E 00002  MOVAB   GSDRECLNG, R11                                 ;
                          54       04    AC  D0 00009  MOVL    SYMDESC, R4                                    ; 0822
                          52       0F    A4  9A 0000D  MOVZBL  15(R4), R2                                     ;
                50        54             52  C3 00011  SUBL3   R2, R4, R0                                    ;
                          59       FB    A0  9E 00015  MOVAB   -5(R0), R9                                     ;
                                   0A    A4  B5 00019  TSTW    10(R4)                                         ; 0823
                                         05  18 0001C  BGEQ    1$                                             ;
                          53             02  D0 0001E  MOVL    #2, MASKLENGTH                                 ; 0824
                                         02  11 00021  BRB     2$                                             ;
                                         53  D4 00023  1$:     CLRL    MASKLENGTH                             ; 0825
                                   18    A4  D5 00025  2$:     TSTL    24(R4)                                 ; 0826
                                         0C  13 00028  BEQL    3$                                             ;
                          57       18    B4  9A 0002A  MOVZBL  @24(R4), R7                                    ; 0830
                          57             02  C4 0002E  MULL2   #2, VALDATLNG                                  ;
                          57             02  C2 00031  SUBL2   #2, VALDATLNG                                  ;
                                         02  11 00034  BRB     4$                                             ; 0826
                                         57  D4 00036  3$:     CLRL    VALDATLNG                              ; 0832
                          51             6B  3C 00038  4$:     MOVZWL  GSDRECLNG, R1                          ; 0833
                50        51             53  C1 0003B  ADDL3   MASKLENGTH, R1, R0                            ;
                          50             52  C0 0003F  ADDL2   R2, R0                                        ;
                          50       0A A740   9E 00042  MOVAB   10(VALDATLNG)[R0], R0                         ;
                    00000200      8F       50  D1 00047  CMPL    R0, #512                                   ; 0834
                                           11  1B 0004E  BLEQU   6$                                         ;
                                           51  DD 00050  PUSHL   R1                                         ; 0836
                    00000000V     EF       01  FB 00052  CALLS   #1, OUTPUTREC                             ;
                                  03       50  E8 00059  BLBS    R0, 5$                                     ;
                                         00A6  31 0005C  BRW     15$                                        ;
                                           6B  B4 0005F  5$:     CLRW    GSDRECLNG                          ; 0838
                                           6B  B5 00061  6$:     TSTW    GSDRECLNG                          ; 0840
                                           07  12 00063  BNEQ    7$                                         ;
```

```
                04    BB       01  90 00065       MOVB    #1, @OBJRECORD                     ; 0842
                      6B       01  B0 00069       MOVW    #1, GSDRECLNG                      ; 0843
                      58       6B  3C 0006C 7$:   MOVZWL  GSDRECLNG, R8                      ; 0845
          56          58  04   AB  C1 0006F       ADDL3   OBJRECVEC, R8, SYMDEFREC           ;
                               57  D5 00074       TSTL    VALDATLNG                          ; 0846
                               38  13 00076       BEQL    10$                                ;
                      51  0C A246  9E 00078       MOVAB   12(R2)[SYMDEFREC], R1              ; 0850
                09    A6  08   A4  B0 0007D       MOVW    8(R4), 9(SYMDEFREC)                ; 0852
                      50  0B   A6  9E 00082       MOVAB   11(R6), SYMBOLSTRING               ; 0853
                      66       03  90 00086       MOVB    #3, (SYMDEFREC)                    ; 0854
                      53  18   A4  D0 00089       MOVL    24(R4), R3                         ; 0855
                      61  01   A3  90 0008D       MOVB    1(R3), (R1)                        ;
       01     A1      63       02  83 00091       SUBB3   #2, (R3), 1(R1)                    ; 0856
                      5A  01   A1  9A 00096       MOVZBL  1(R1), R10                         ; 0857
                               52  D4 0009A       CLRL    I                                  ;
                               0C  11 0009C       BRB     9$                                 ;
                      55     6142  3E 0009E 8$:   MOVAW   (R1)[I], R5                        ; 0861
                      65  01 A243  90 000A2       MOVB    1(I)[R3], (R5)                     ; 0862
                          01   A5  94 000A7       CLRB    1(R5)                              ; 0863
              F0      52       5A  F3 000AA 9$:   AOBLEQ  R10, I, 8$                         ; 0857
                               19  11 000AE       BRB     12$                                ; 0846
                               53  D5 000B0 10$:  TSTL    MASKLENGTH                         ; 0866
                               0E  13 000B2       BEQL    11$                                ;
                09    A6  08   A4  B0 000B4       MOVW    8(R4), 9(SYMDEFREC)                ; 0868
                      50  0B   A6  9E 000B9       MOVAB   11(R6), SYMBOLSTRING               ; 0869
                      66       02  90 000BD       MOVB    #2, (SYMDEFREC)                    ; 0870
                               07  11 000C0       BRB     12$                                ;
                      50  09   A6  9E 000C2 11$:  MOVAB   9(SYMDEFREC), SYMBOLSTRING         ; 0873
                      66       01  90 000C6       MOVB    #1, (SYMDEFREC)                    ; 0874
                01    A6  0E   A4  90 000C9 12$:  MOVB    14(R4), 1(SYMDEFREC)               ; 0876
    51     0A   A4    04       00  EF 000CE       EXTZV   #0, #4, 10(R4), R1                 ; 0877
                02    A6       51  B0 000D4       MOVW    R1, 2(SYMDEFREC)                   ;
              05  0A  A4       03  E0 000D8       BBS     #3, 10(R4), 13$                    ; 0879
                          04   A6  94 000DD       CLRB    4(SYMDEFREC)                       ; 0880
                               05  11 000E0       BRB     14$                                ;
                04    A6  02   AB  90 000E2 13$:  MOVB    CURPSECTNUM, 4(SYMDEFREC)          ; 0881
                05    A6       64  D0 000E7 14$:  MOVL    (R4), 5(SYMDEFREC)                 ; 0882
                      51  04   A9  9A 000EB       MOVZBL  4(R9), R1                          ; 0883
                               51  D6 000EF       INCL    R1                                 ;
              60  04  A9       51  28 000F1       MOVC3   R1, 4(R9), (SYMBOLSTRING)          ; 0884
              50      58       53  C1 000F6       ADDL3   R3, R8, R0                         ; 0883
                      50       56  C2 000FA       SUBL2   SYMDEFREC, R0                      ; 0885
              6B      50       57  A1 000FD       ADDW3   VALDATLNG, R0, GSDRECLNG           ;
                      50       01  D0 00101       MOVL    #1, R0                             ; 0886
                                   04 00104       RET                                        ;
                               50  D4 00105 15$:  CLRL    R0                                 ; 0887
                                   04 00107       RET                                        ;

; Routine Size:  264 bytes,    Routine Base:  $CODE$ + 0578
```

```
;  784          0888  1 routine stbrecout(reclng) =
;  785          0889  2 begin
;  786          0890  2 !
;  787          0891  2 !       THIS ROUTINE WRITES TO THE STB FILE IF ONE IS BEING CREATED
;  788          0892  2 !
;  789          0893  2 !       RECLNG          LENGTH OF RECORD TO WRITE
;  790          0894  2 !
;  791          0895  2 local
;  792          0896  2     rmserror;
;  793          0897  2
;  794          0898  2 if .reclng neq 0                                ! IF NON-ZERO LENGTH RECORD
;  795          0899  2     and .stbfileifi neq 0                       ! AND WE ARE WRITING TO THE STB FILE
;  796          0900  3 then begin
;  797          0901  3     stbrab[rab$w_rsz] = .reclng;                ! SET RECORD LENGTH
;  798          0902  4     if not (rmserror = $put(rab=stbrab))        ! WRITE THE RECORD
;  799          0903  4     then begin
;  800          0904  4         signal(lin$_writeerr,1,                 ! SIGNAL ANY ERRORS
;  801          0905  4                 lnk$gl_symfi[[fdb$q_filename],
;  802          0906  4                 .rmserror,.stbrab[rab$l_stv]);
;  803          0907  4         lnk$closymout(.stbauxfnb);              ! CLOSE THE FILE IF ERROR
;  804          0908  4         if .imgauxfnb eql 0                     ! IF NO IMAGE FILE BEING CREATED
;  805          0909  4             then return false;                  !  THEN ALL DONE NOW
;  806          0910  4         end
;  807          0911  3     else lnk$gw_symrecs = .lnk$gw_symrecs + 1;  ! COUNT GOOD RECORD WRITTEN TO THE FILE
;  808          0912  2     end;
;  809          0913  2
;  810          0914  2 return true
;  811          0915  1 end;


                                                .EXTRN  SYS$PUT

                                0004 00000 STBRECOUT:
                                                .WORD   Save R2                 ;  0888
                             52 00000000'   EF  9E 00002        MOVAB   STBFILEIFI, R2
                                     04     AC  D5 00009        TSTL    RECLNG                  ;  0898
                                            49  13 0000C        BEQL    2$
                                            62  D5 0000E        TSTL    STBFILEIFI              ;  0899
                                            45  13 00010        BEQL    2$
                                   D6   A2      04  AC  B0 00012        MOVW    RECLNG, STBRAB+34       ;  0901
                                            B4  A2  9F 00017        PUSHAB  STBRAB                  ;  0902
                             00000000G      00      01  FB 0001A        CALLS   #1, SYS$PUT
                                            2D      50  E8 00021        BLBS    RMSERROR, 1$
                                            C0  A2  DD 00024        PUSHL   STBRAB+12               ;  0906
                                            50  DD 00027        PUSHL   RMSERROR
                          7E 00000000G      00      14  C1 00029        ADDL3   #20, LNK$GL_SYMFIL, -(SP)    ;  0905
                                            01  DD 00031        PUSHL   #1
                                 00000000G      8F  DD 00033        PUSHL   #LIN$_WRITEERR
                             00000000G      00      05  FB 00039        CALLS   #5, LIB$SIGNAL
                                            B0  A2  DD 00040        PUSHL   STBAUXFNB               ;  0907
                             00000000V      EF      01  FB 00043        CALLS   #1, LNK$CLOSYMOUT
                                            04  A2  D5 0004A        TSTL    IMGAUXFNB               ;  0908
                                            08  12  0004D        BNEQ    2$
                                            0A  11  0004F        BRB     3$                      ;  0909
                                 00000000'      EF  B6 00051 1$:    INCW    LNK$GW_SYMRECS          ;  0911
                                50              01  D0 00057 2$:    MOVL    #1, R0                  ;  0914
```

```
                        04 0005A          RET
                50      D4 0005B 3$:      CLRL    R0                                      ;
                        04 0005D          RET                                             ; 0915
                                                                                          ;

; Routine Size:  94 bytes,    Routine Base:  $CODE$ + 0680
```

```
;  813          0916  1 routine imgrecout(reclng) =
;  814          0917  2 begin
;  815          0918  2 !
;  816          0919  2 !       THIS ROUTINE WRITES TO THE IMAGE FILE
;  817          0920  2 !
;  818          0921  2 !       RECLNG          LENGTH OF RECORD
;  819          0922  2 !
;  820          0923  2 local
;  821          0924  2     rmserror;
;  822          0925  2
;  823          0926  2 if .reclng neq 0                                ! IF NON-ZERO LENGTH
;  824          0927  2     and .imgauxfnb neq 0                        ! AND IMAGE FILE IS OPEN
;  825          0928  3 then begin
;  826          0929  3     lnk$al_imgrab[rab$w_rsz] = .reclng;         ! SET RECORD LENGTH
;  827          0930  4     if not (rmserror = $put(rab = lnk$al_imgrab)) ! WRITE THE RECORD
;  828          0931  4     then begin
;  829          0932  4         signal(lin$_writeerr,1,                 ! IF ERROR, REPORT AND CLOSE FILE
;  830          0933  4                 lnk$gl_imgfil[fdb$g_filename],
;  831          0934  4                 .rmserror,.lnk$al_imgrab[rab$l_stv]);
;  832          0935  4         lnk$closymout(.imgauxfnb);
;  833          0936  4         if .stbfileifi eql 0                    ! IF NO STB FILE BEING CREATED
;  834          0937  4             then return false;                  !  THEN ALL DONE NOW
;  835          0938  4         end
;  836          0939  3     else lnk$gw_gstrecs = .lnk$gw_gstrecs + 1;  ! COUNT GOOD RECORD WRITTEN
;  837          0940  2     end;
;  838          0941  2
;  839          0942  2 return true
;  840          0943  1 end;

                                000C 00000 IMGRECOUT:
                                           .WORD   Save R2,R3                                  ; 0916
                    53 00000000'   EF  9E 00002    MOVAB   IMGAUXFNB, R3                      ;
                    52 00000000G   00  9E 00009    MOVAB   LNK$AL_IMGRAB+34, R2                ;
                               04  AC  D5 00010    TSTL    RECLNG                              ; 0926
                                   47  13 00013    BEQL    2$                                  ;
                                   63  D5 00015    TSTL    IMGAUXFNB                           ; 0927
                                   43  13 00017    BEQL    2$                                  ;
                               62  04  AC  B0 00019    MOVW    RECLNG, LNK$AL_IMGRAB+34        ; 0929
                                   DE  A2  9F 0001D    PUSHAB  LNK$AL_IMGRAB                   ; 0930
                    00000000G  00      01  FB 00020    CALLS   #1, SYS$PUT                     ;
                               2C      50  E8 00027    BLBS    RMSERROR, 1$                    ;
                                   EA  A2  DD 0002A    PUSHL   LNK$AL_IMGRAB+12                ; 0934
                                       50  DD 0002D    PUSHL   RMSERROR                        ;
                7E 00000000G   00      14  C1 0002F    ADDL3   #20, LNK$GL_IMGFIL, -(SP)       ; 0933
                                       01  DD 00037    PUSHL   #1                              ;
                           00000000G   8F  DD 00039    PUSHL   #LIN$_WRITEERR                  ;
                    00000000G  00      05  FB 0003F    CALLS   #5, LIB$SIGNAL                  ;
                                       63  DD 00046    PUSHL   IMGAUXFNB                       ; 0935
                    00000000V  EF      01  FB 00048    CALLS   #1, LNK$CLOSYMOUT               ;
                               FC  A3  D5 0004F    TSTL    STBFILEIFI                          ; 0936
                                   08  12 00052    BNEQ    2$                                  ;
                                   0A  11 00054    BRB     3$                                  ; 0937
                           00000000'   EF  B6 00056 1$:    INCW    LNK$GW_GSTRECS              ; 0939
```

```
                          50        01  D0 0005C 2$:    MOVL    #1, R0                          ; 0942
                                        04 0005F        RET                                     ;
                                    50  D4 00060 3$:    CLRL    R0                              ; 0943
                                        04 00062        RET                                     ;

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 05DE
```

```
;  842          0944  1 routine outputrec(reclng) =
;  843          0945  2 begin
;  844          0946  2 !
;  845          0947  2 !       THIS ROUTINE HANDLES THE ACTUAL RECORD O  PUT TO THE
;  846          0948  2 !       FILE(S) RECEIVING SYMBOL TABLE RECORDS. L  OR DETECTION
;  847          0949  2 !       OCCURS HERE BUT THE HANDLING IS DONE IN TH  FILE CLOSING
;  848          0950  2 !       ROUTINE.
;  849          0951  2 !
;  850          0952  2 if not stbrecout(.reclng)
;  851          0953  2     then return false;
;  852          0954  2
;  853          0955  2 return imgrecout(.reclng)
;  854          0956  1 end;


                                0000 00000 OUTPUTREC:
                                                .WORD   Save nothing                    ; 0944
                            04  AC  DD 00002          PUSHL   RECLNG                  ; 0952
                    FF35    CF      01  FB 00005      CALLS   #1, STBRECOUT           ;
                            08      50  E9 0000A      BLBC    R0, 1$                  ;
                            04  AC  DD 0000D          PUSHL   RECLNG                  ; 0955
                      89    AF      01  FB 00010      CALLS   #1, IMGRECOUT           ;
                                    04 00014          RET                             ;
                                50  D4 00015 1$:     CLRL    R0                      ; 0956
                                    04 00017          RET                             ;

; Routine Size:  24 bytes,    Routine Base:  $CODE$ + 0741
```

```
;  856          0957  1 global routine lnk$closymout(auxfnb) : novalue =
;  857          0958  2 begin
;  858          0959  2 !
;  859          0960  2 !       THIS ROUTINE HANDLES ERRORS WRITING THE SYMBOL TABLE RECORDS
;  860          0961  2 !       AND/OR CLOSES THE DESIRED FILE(S).
;  861          0962  2 !
;  862          0963  2 !       IF "AUXFNB" IS ZERO - BOTH FILES (IF BOTH EXIST) ARE CLOSED
;  863          0964  2 !       OTHERWISE "AUXFNB" IS THE ADDRESS OF THE AUXILIARY FILENAME BLOCK
;  864          0965  2 !       OF THE FILE ON WHICH AN ERROR OCCURRED.  THE FILE IS CLOSED.
;  865          0966  2 !
;  866          0967  2 !       WHEN OUTPUTTING RECORDS TO THE GST OF AN IMAGE, THE IMAGE FILE
;  867          0968  2 !       IS NOT ACTUALLY CLOSED (EXCEPT ON ERRORS). ITS ATTRIBUTES ARE MERELY
;  868          0969  2 !       MODIFIED (BACK TO FIXED 512 BYTE RECORD) AND IT IS LEFT OPEN SINCE
;  869          0970  2 !       THE IMAGE HEADER NEEDS TO BE WRITTEN AFTER THE GST IS DONE.
;  870          0971  2 !
;  871          0972  2 map     auxfnb : ref block[,byte];
;  872          0973  2
;  873          0974  2 local   fablock : block[fab$c_bln,byte],                    ! FAB FOR CLOSE AND MODIFY OPERATIONS
;  874          0975  2         closerror;                                          ! ERROR CODE IF CLOSE FAILS
;  875          0976  2
;  876        P 0977  2 $fab_init(fab=fablock,                                      ! INITIALIZE THE FAB
;  877          0978  2         fop=tef);
;  878          0979  2
;  879          0980  2 if .auxfnb eql 0                                           ! IF WE ARE CLOSING BOTH FILES
;  880          0981  2 or .auxfnb eql .stbauxfnb                                  ! OR THE SYMBOL TABLE FILE ONLY
;  881          0982  2 then if (fablock[fab$w_ifi] = .stbfileifi) neq 0           ! IF IT IS STILL OPEN
;  882          0983  3 then begin
;  883          0984  4     if not (closerror = $close(fab=fablock))              ! ATTEMPT TO CLOSE IT
;  884          0985  4     then begin
;  885          0986  4         signal(lin$_closeout,1,                           ! AND OUTPUT AN ERROR IF THAT
;  886          0987  4                lnk$gl_symfi[fdb$q_filename],
;  887          0988  4                .closerror,.fablock[fab$l_stv]);
;  888          0989  3         end;
;  889          0990  3     stbfileifi = 0;                                       ! AND FORGET THE FILE IN ANY CASE
;  890          0991  3     if .auxfnb neq 0 then return;                         ! RETURN IF THAT WAS ALL
;  891          0992  2     end;
;  892          0993  2 if .imgauxfnb neq 0                                       ! IF SYMBOLS ARE ALSO GOING TO IMAGE
;  893          0994  3 then begin                                                ! THEN SET UP TO MODIFY THE
;  894          0995  3     fablock[fab$w_ifi] = .lnk$gw_imgifi;                  ! THE ATTRIBUTES OF THIS FILE
;  895          0996  3     fablock[fab$b_rfm] = fab$c_fix;                       ! BACK TO FIXED LENGTH
;  896          0997  3     fablock[fab$w_mrs] = 512;                             ! 512 BYTE RECORDS IN THE
;  897          0998  3     fablock[fab$v_esc] = true;                            ! RMS DATA BASE
;  898          0999  3     fablock[fab$l_ctx] = rme$c_setrfm;                    ! SO THAT IT WILL HAVE
;  899          1000  4     if not (closerror = $modify(fab = fablock))           ! THE ATTRIBUTES OF AN IMAGE
;  900          1001  4     then begin
;  901          1002  4         signal(lin$_closeout,1,                           ! ISSUE MESSAGE IF MODIFY FAILED
;  902          1003  4                lnk$gl_imgfi[fdb$q_filename],
;  903          1004  4                .closerror,.fablock[fab$l_stv]);
;  904          1005  3         end;
;  905          1006  3     imgauxfnb = 0;
;  906          1007  3     return;
;  907          1008  2     end;
;  908          1009  1 end;


                                .EXTRN  SYS$CLOSE
```

```
                                           01FC 00000         .ENTRY  LNK$CLOSYMOUT, Save R2,R3,R4,R5,R6,R7,R8       ; 0957
                  58 00000000G      00   9E 00002          MOVAB   LIB$SIGNAL, R8
                  57 00000000G      8F   D0 00009          MOVL    #LIN$_CLOSEOUT, R7
                  56 00000000'      EF   9E 00010          MOVAB   STBFILEIFI, R6
                  5E        B0      AE   9E 00017          MOVAB   -80(SP), SP
  0050   8F          00       6E      00   2C 0001B          MOVC5   #0, (SP), #0, #80, $RMS_PTR                   ; 0978
                                  6E      00022
                  6E      5003      8F   B0 00023          MOVW    #20483, $RMS_PTR
        04        AE  10000000      8F   D0 00028          MOVL    #268435456, $RMS_PTR+4
        16        AE                02   90 00030          MOVB    #2, $RMS_PTR+22
        1F        AE                02   90 00034          MOVB    #2, $RMS_PTR+31
                  52        04      AC   D0 00038          MOVL    AUXFNB, R2                                    ; 0980
                                    06   13 0003C          BEQL    1$
        B0        A6                52   D1 0003E          CMPL    R2, STBAUXFNB                                 ; 0981
                                    34   12 00042          BNEQ    3$
                  50                66   D0 00044 1$:     MOVL    STBFILEIFI, R0                                ; 0982
        02        AE                50   B0 00047          MOVW    R0, FABLOCK+2
                                    50   D5 0004B          TSTL    R0
                                    29   13 0004D          BEQL    3$
                                    5E   DD 0004F          PUSHL   SP                                            ; 0984
         00000000G  00              01   FB 00051          CALLS   #1, SYS$CLOSE
                    53              50   D0 00058          MOVL    R0, CLOSERROR
                    14              53   E8 0005B          BLBS    CLOSERROR, 2$
                            0C      AE   DD 0005E          PUSHL   FABLOCK+12                                    ; 0988
                                    53   DD 00061          PUSHL   CLOSERROR
      7E 00000000G  00              14   C1 00063          ADDL3   #20, LNK$GL_SYMFIL, -(SP)                     ; 0987
                                    01   DD 0006B          PUSHL   #1
                                    57   DD 0006D          PUSHL   R7
                    68              05   FB 0006F          CALLS   #5, LIB$SIGNAL
                                    66   D4 00072 2$:     CLRL    STBFILEIFI                                    ; 0990
                                    52   D5 00074          TSTL    R2                                            ; 0991
                                    45   12 00076          BNEQ    5$
                            04      A6   D5 00078 3$:     TSTL    IMGAUXFNB                                     ; 0993
                                    40   13 0007B          BEQL    5$
        02        AE 00000000G      00   B0 0007D          MOVW    LNK$GW_IMGIFI, FABLOCK+2                      ; 0995
        1F        AE                01   90 00085          MOVB    #1, FABLOCK+31                                ; 0996
        36        AE      0200      8F   B0 00089          MOVW    #512, FABLOCK+54                              ; 0997
        07        AE                08   88 0008F          BISB2   #8, FABLOCK+7                                 ; 0998
        18        AE                01   D0 00093          MOVL    #1, FABLOCK+24                                ; 0999
                                    5E   DD 00097          PUSHL   SP                                            ; 1000
         00000000G  00              01   FB 00099          CALLS   #1, SYS$MODIFY
                    53              50   D0 000A0          MOVL    R0, CLOSERROR
                    14              53   E8 000A3          BLBS    CLOSERROR, 4$
                            0C      AE   DD 000A6          PUSHL   FABLOCK+12                                    ; 1004
                                    53   DD 000A9          PUSHL   CLOSERROR
      7E 00000000G  00              14   C1 000AB          ADDL3   #20, LNK$GL_IMGFIL, -(SP)                     ; 1003
                                    01   DD 000B3          PUSHL   #1
                                    57   DD 000B5          PUSHL   R7
                    68              05   FB 000B7          CALLS   #5, LIB$SIGNAL
                            04      A6   D4 000BA 4$:     CLRL    IMGAUXFNB                                     ; 1006
                                         04 000BD 5$:     RET                                                   ; 1009

; Routine Size:  190 bytes,    Routine Base:  $CODE$ + 0759

;  909         1010  0 end eludom                                  ! END OF MODULE
```

```
                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;   $GLOBAL$                        4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                         116  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                        112  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       2071  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       145         0        1000        00:01.9
;   _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1   538        47         8          28        00:00.8


;                                       COMMAND QUALIFIERS

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

; Size:         2071 code + 232 data bytes
; Run Time:       00:39.7
; Elapsed Time:   01:30.5
; Lines/CPU Min:   1527
; Lexemes/CPU-Min: 22610
; Memory Used:  220 pages
; Compilation Complete
```

LNKPROLTB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKVMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSO
LIS