```
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**LNKOBJPS1

```
LL            NN      NN  KK      KK    000000    BBBBBBBB          JJ  PPPPPPPP     SSSSSSSS      11
LL            NN      NN  KK      KK    000000    BBBBBBBB          JJ  PPPPPPPP     SSSSSSSS      11
LL            NN      NN  KK      KK  00      00  BB      BB        JJ  PP      PP   SS           1111
LL            NN      NN  KK      KK  00      00  BB      BB        JJ  PP      PP   SS           1111
LL            NNNN    NN  KK    KK    00      00  BB      BB        JJ  PP      PP   SS             11
LL            NNNN    NN  KK    KK    00      00  BB      BB        JJ  PP      PP   SS             11
LL            NN  NN  NN  KKKKKK      00      00  BBBBBBBB          JJ  PPPPPPPP     SSSSSS         11
LL            NN  NN  NN  KKKKKK      00      00  BBBBBBBB          JJ  PPPPPPPP     SSSSSS         11
LL            NN    NNNN  KK    KK    00      00  BB      BB  JJ    JJ  PP                 SS       11
LL            NN    NNNN  KK    KK    00      00  BB      BB  JJ    JJ  PP                 SS       11
LL            NN      NN  KK      KK  00      00  BB      BB  JJ    JJ  PP                 SS       11          ....
LL            NN      NN  KK      KK  00      00  BB      BB  JJ    JJ  PP                 SS       11          ....
LLLLLLLLLL    NN      NN  KK      KK    000000    BBBBBBBB      JJJJJJ    PP           SSSSSSSS   111111        ....
LLLLLLLLLL    NN      NN  KK      KK    000000    BBBBBBBB      JJJJJJ    PP           SSSSSSSS   111111        ....

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_objpass1
;  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 !*   THER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 22          0022  1 !*  T  NSFERRED.                                                           *
; 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 ! MODULE:       LNK_OBJPASS1
; 37          0037  1 !
; 38          0038  1 ! FACILITY:     LINKER
; 39          0039  1 !
; 40          0040  1 ! ABSTRACT:     Pass one of all object modules of this link.
; 41          0041  1 !
; 42          0042  1 ! HISTORY:
; 43          0043  1 !
; 44          0044  1 !       VERSION: X01.00
; 45          0045  1 !
; 46          0046  1 !       AUTHOR: T.J. PORTER 30-DEC-76
; 47          0047  1 !
; 48          0048  1 ! MODIFIED BY:
; 49          0049  1 !
; 50          0050  1 !       V04-045 JWT0182         Jim Teague              14-May-1984
; 51          0051  1 !               Adjust logic in PROSYMBOL so that lnk$gw_nudfsyms
; 52          0052  1 !               doesn't get decremented too many times.
; 53          0053  1 !
; 54          0054  1 !       V04-044 ADE0003         Alan D. Eldridge         4-May-1984
; 55          0055  1 !               Fix access violation in routine PROSYMBOL.
; 56          0056  1 !
; 57          0057  1 !       V04-043 ADE0002         Alan D. Eldridge         1-May-1984
```

```
;    58          0058  1 !               Fix bug in reporting 'illegal module name length'.
;    59          0059  1 !
;    60          0060  1 !       V04-042 ADE0001         Alan D. Eldridge        3-Mar-1984
;    61          0061  1 !               If GSMATCH high bit is set, then ignore date comparison
;    62          0062  1 !               between shareable image library entry and shareable image
;    63          0063  1 !               itself.
;    64          0064  1 !
;    65          0065  1 !       V04-041 JWT0154         Jim Teague              9-Feb-1984
;    66          0066  1 !               MPC blocks are not always clean when memory is
;    67          0067  1 !               allocated for them.  This caused ridiculous values
;    68          0068  1 !               for psect length to show up in the data produced
;    69          0069  1 !               in the Debugger Module/Psect Table (DMT).
;    70          0070  1 !
;    71          0071  1 !       V03-040 JWT0132         Jim Teague              16-Aug-1983
;    72          0072  1 !               Adjust ordering of LNKOPT fdbs that are created.
;    73          0073  1 !               A module that specifies several files in LNKOPT
;    74          0074  1 !               records caused the linker to append the LNKOPT
;    75          0075  1 !               file fdbs in reverse order!
;    76          0076  1 !
;    77          0077  1 !       V03-039 JWT0130         Jim Teague              28-Jul-1983
;    78          0078  1 !               More scrutiny for Linker Options Records.
;    79          0079  1 !
;    80          0080  1 !       V03-038 JWT0128         Jim Teague              24-Jul-1983
;    81          0081  1 !               Fix environment ref/def problem.
;    82          0082  1 !
;    83          0083  1 !       V03-037 JWT0125         Jim Teague              23-Jun-1983
;    84          0084  1 !               Reincarnate univ=* when accompanied by /noexe.
;    85          0085  1 !
;    86          0086  1 !       V03-036 JWT0121         Jim Teague              20-May-1983
;    87          0087  1 !               Sound tne death knell for UNIVERSAL=*.
;    88          0088  1 !
;    89          0089  1 !       V03-035 JWT0113         Jim Teague              20-Apr-1983
;    90          0090  1 !               Filter universal bit out of incoming symbols;
;    91          0091  1 !               Call $getjpi to get number of open files left.
;    92          0092  1 !
;    93          0093  1 !       V03-034 JWT0109         Jim Teague              13-Apr-1983
;    94          0094  1 !               Adjust code for Linker Options Record.
;    95          0095  1 !
;    96          0096  1 !       V03-033 JWT0099         Jim Teague              14-Mar-1983
;    97          0097  1 !               New CLI interface.  Also implemented code for
;    98          0098  1 !               Linker Options Record.
;    99          0099  1 !
;   100          0100  1 !       V03-032 JWT0071         Jim Teague              02-Dec-1982
;   101          0101  1 !               Added NAME and IDENTIFICATION options.
;   102          0102  1 !
;   103          0103  1 !       V03-031 JWT0061         Jim Teague              22-Oct-1982
;   104          0104  1 !               Add debugger image section to images linked /debug.
;   105          0105  1 !
;   106          0106  1 !       V03-030 JWT0052         Jim Teague              15-Sep-1982
;   107          0107  1 !               Fix bug which caused option-defined symbols to be
;   108          0108  1 !               excluded from STB.
;   109          0109  1 !
;   110          0110  1 !       V03-029 JWT0047         Jim Teague              06-Aug-1982
;   111          0111  1 !               Fix the linker psect alignment bug, once and for all!
;   112          0112  1 !
;   113          0113  1 !       V03-028 JWT0045         Jim Teague              30-Jul-1982
;   114          0114  1 !               Fix bug in entity id mismatch error message.
```

```
;  115          0115  1 !
;  116          0116  1 !       V03-027 JWT0043         Jim Teague              15-Jul-1982
;  117          0117  1 !               Fix psect alignment bug which caused all contributions to
;  118          0118  1 !               a psect to have maximized alignment.
;  119          0119  1 !
;  120          0120  1 !       V03-026 JWT0039         Jim Teague              23-Jun-1982
;  121          0121  1 !               Add mask in order to retain SYM$V_LCLSYM attribute.
;  122          0122  1 !               This keeps module local symbols out of cross references.
;  123          0123  1 !
;  124          0124  1 !       V03-025 JWT0033         Jim Teague              25-May-1982
;  125          0125  1 !               Or OMD$V_NOBINs for each FDB and propagate to FDB$V_OMDNOBIN
;  126          0126  1 !               Use option-specified psect alignment if present in PROPSECTDEF
;  127          0127  1 !
;  128          0128  1 !       V03-024 BLS0161         Benn Schreiber          19-Mar-1982
;  129          0129  1 !               Correct V03-021 handling of selective search procedures
;  130          0130  1 !
;  131          0131  1 !
;  132          0132  1 !--
```

```
;  134          0133  1 !++
;  135          0134  1 !
;  136          0135  1 ! FUNCTIONAL DESCRIPTION:
;  137          0136  1 !
;  138          0137  1 ! THIS ROUTINE IS CALLED TO PROCESS ALL OBJECT MODULES
;  139          0138  1 ! DURING PASS ONE OF THE LINK. IT VALIDATES OBJECT MODULE
;  140          0139  1 ! FORMAT, READS THE MODULE HEADER, ALL GSD AND THE
;  141          0140  1 ! END OF MODULE RECORDS AND BUILDS SYMBOL AND P-SECTION
;  142          0141  1 ! TABLES. IT IS IN THIS ROUTINE ALONE THAT THE LINKER
;  143          0142  1 ! DETERMINES WHETHER THE MODULE IS CONCATENATED. IT IS
;  144          0143  1 ! ALSO HERE THAT ANY USER DEFINED TRANSFER ADDRESS IS
;  145          0144  1 ! EXTRACTED.
;  146          0145  1 !
;  147          0146  1 ! THIS ROUTINE IGNORES THE CONTENT OF TIR, DBG AND LNK RECORDS.
;  148          0147  1 !
;  149          0148  1 !--
;  150          0149  1 require 'PREFIX';
;  151          0264  1 forward routine
;  152          0265  1     prolnkrec,                          ! PROCESS LINKER OPTIONS RECORD
;  153          0266  1     checkpddentry,                      ! CHECK PSECT DEF. ENTRY AND DEFINE IT
;  154          0267  1     lnk$insudfsym : novalue,            ! INSERT SYMBOL IN UNDEFINED SYMBOL LIST
;  155          0268  1     seqchk,                             ! CHECK RECORD SEQUENCE
;  156          0269  1     prohdr,                             ! PROCESS MODULE HEADER RECORDS
;  157          0270  1     delete_psect,                       ! DELETE SHAREABLE IMAGE PSECT FROM OTHER CLUSTERS
;  158          0271  1     fndpscmapent,                       ! FIND PSECT MAPPING TABLE ENTRY
;  159          0272  1     alloc_omdnode,                      ! ALLOCATE OMD NODE BLOCK
;  160          0273  1     lnk$compare_omd,                    ! COMPARE OMD NODE BLOCKS
;  161          0274  1     propsectdef,                        ! PROCESS PSECT DEFINITIONS
;  162          0275  1     crefilter,                          ! CROSS REFERENCE FILTER FOR SYMBOLS
;  163          0276  1     prosymbol,                          ! PROCESS SYMBOLS
;  164          0277  1     symbols,                            ! FRONT END FOR SYMBOL DEFS/REFS
;  165          0278  1     entpnts,                            ! FRONT END FOR ENTRY POINTS
;  166          0279  1     procedef,                           ! FRONT END FOR PROCEDURE DEFINITIONS
;  167          0280  1     proeom,                             ! PROCESS END OF MODULE RECORDS
;  168          0281  1     countdbg,                           ! COUNT DEBUG AND TRACEBACK RECORDS
;  169          0282  1     compare_idc,                        ! COMPARE RANDOM IDENT BLOCKS
;  170          0283  1     alloc_idc,                          ! ALLOCATE RANDOM IDENT BLOCK
;  171          0284  1     randentity,                         ! PROCESS RANDOM IDENTS
;  172          0285  1     compare_env,                        ! COMPARE ENVIRONMENT BLOCKS
;  173          0286  1     alloc_env,                          ! ALLOCATE ENVIRONMENT BLOCK
;  174          0287  1     insudfenv           : novalue,      ! INSERT UNDEFINED ENVIRONMENT
;  175          0288  1     lnk$fndenvmap,                      ! FIND ENVIRONMENT MAPPING TABLE ENTRY
;  176          0289  1     proenv,                             ! PROCESS ENVIRONMENT
;  177          0290  1     localsymbols,                       ! FRONT END FOR LOCAL SYMBOLS
;  178          0291  1     localentpnt,                        ! FRONT END FOR LOCAL ENTRY POINTS
;  179          0292  1     localprocedef,                      ! FRONT END FOR LOCAL PROCEDURE DEFS
;  180          0293  1     progsd,                             ! PROCESS GSD RECORDS
;  181          0294  1     lnk$procsobj,                       ! PROCESS AN OBJECT MODULE
;  182          0295  1     lnk$objpass1 ;                      ! DO PASS 1
;  183          0296  1
;  184          0297  1 library 'LIBL32' ;                      ! SYSTEM DATA STRUCTURES
;  185          0298  1
;  186          0299  1 library 'DATBAS' ;                      ! INTERNAL DATA BASE DEFINITIONS
;  187          0300  1
;  188          0301  1 require 'ISGENC' ;                      ! IMAGE SECTION DEFINITIONS AND OBJECT LANG. DEFS.
;  189          0685  1 !
;  190          0686  1 !
```

```
;  191          0687  1 ! ENSURE THAT SYM$ SYMBOLS CORRESPOND TO GSY$ SYMBOLS
;  192          0688  1 !  AND THE PSC$ SYMBOLS CORRESPOND TO GPS$ SYMBOLS
;  193          0689  1 !
;  194          0690  1 %if (sym$m_weak and gsy$m_weak) eql 0   %then %warn ('SYM$M_WEAK NEQ GSY$M_WEAK') %fi;
;  195          0691  1 %if  (sym$m_def and gsy$m_def)  eql 0   %then %warn  ('SYM$M_DEF NEQ GSY$M_DEF') %fi;
;  196          0692  1 %if  (sym$m_uni and gsy$m_uni)  eql 0   %then %warn  ('SYM$M_UNI NEQ GSY$M_UNI') %fi;
;  197          0693  1 %if  (sym$m_rel and gsy$m_rel)  eql 0   %then %warn  ('SYM$M_REL NEQ GSY$M_REL') %fi;
;  198          0694  1 %if  (psc$m_pic and gps$m_pic)  eql 0   %then %warn  ('PSC$M_PIC NEQ GPS$M_PIC') %fi;
;  199          0695  1 %if  (psc$m_lib and gps$m_lib)  eql 0   %then %warn  ('PSC$M_LIB NEQ GPS$M_LIB') %fi;
;  200          0696  1 %if  (psc$m_ovr and gps$m_ovr)  eql 0   %then %warn  ('PSC$M_OVR NEQ GPS$M_OVR') %fi;
;  201          0697  1 %if  (psc$m_rel and gps$m_rel)  eql 0   %then %warn  ('PSC$M_REL NEQ GPS$M_REL') %fi;
;  202          0698  1 %if  (psc$m_gbl and gps$m_gbl)  eql 0   %then %warn  ('PSC$M_GBL NEQ GPS$M_GBL') %fi;
;  203          0699  1 %if  (psc$m_shr and gps$m_shr)  eql 0   %then %warn  ('PSC$M_SHR NEQ GPS$M_SHR') %fi;
;  204          0700  1 %if  (psc$m_exe and gps$m_exe)  eql 0   %then %warn  ('PSC$M_EXE NEQ GPS$M_EXE') %fi;
;  205          0701  1 %if  (psc$m_rd  and gps$m_rd)   eql 0   %then %warn   ('PSC$M_RD NEQ GPS$M_RD') %fi;
;  206          0702  1 %if  (psc$m_wrt and gps$m_wrt)  eql 0   %then %warn  ('PSC$M_WRT NEQ GPS$M_WRT') %fi;
;  207          0703  1 %if  (psc$m_vec and gps$m_vec)  eql 0   %then %warn  ('PSC$M_VEC NEQ GPS$M_VEC') %fi;
;  208          0704  1
;  209          0705  1 global literal
;  210          0706  1     psc$m_pscbits =    psc$m_pic or psc$m_lib or psc$m_ovr      ! PSECT ATTRIBUTES ALL TOGETHER
;  211          0707  1                     or psc$m_rel or psc$m_gbl or psc$m_shr
;  212          0708  1                     or psc$m_exe or psc$m_rd  or psc$m_wrt
;  213          0709  1                     or psc$m_vec,
;  214          0710  1
;  215          0711  1     psc$m_shrbits =    psc$m_gbl or psc$m_shr or psc$m_ovr ;    ! PSECTS CONSIDERED FOR SHAREABLE IMAGE
```

```
;  217          0712  1 external routine
;  218          0713  1      lnk$closefile,
;  219          0714  1      lnk$filnamdsc,
;  220          0715  1      lnk$openlib,
;  221          0716  1      lnk$allofdb,
;  222          0717  1      crf$insrtkey,                   ! INSERT KEY IN CREF TABLE
;  223          0718  1      crf$insrtref,                   ! INSERT REF TO KEY IN CREF TABLE
;  224          0719  1      lib$insert_tree,                ! INSERT INTO BINARY TREE
;  225          0720  1      lib$lookup_tree,                ! LOOKUP IN BINARY TREE
;  226          0721  1      lib$traverse_tree,              ! TRAVERSE BINARY TREE
;  227          0722  1      lnk$compare_pscnod,             ! COMPARE PSECT NAME WITH GPSLST NODE
;  228          0723  1      lnk$srcpscdef,                  ! FIND PSECT DEFINED IN OPTION FILE
;  229          0724  1      lnk$bintim,                     ! CONVERT DATE/TIME TO BINARY
;  230          0725  1      lnk$alloblk         : novalue,  ! DYNAMIC MEMORY ALLOCATION
;  231          0726  1      lnk$dealblk         : novalue,  ! AND DEALLOCATION
;  232          0727  1      lnk$exit            : novalue,  ! EXIT ROUTINE
;  233          0728  1      lnk$fndpscnam,                  ! SEARCH FOR A P-SECTION ENTRY BY NAME
;  234          0729  1      lnk$insert,                     ! SYMBOL TABLE INSERTION
;  235          0730  1      lnk$nxtfil,                     ! GET NEXT INPUT FILE
;  236          0731  1      lnk$nxtrec,                     ! OBTAIN NEXT RECORD OF OBJECT MODULE
;  237          0732  1      lnk$procshrim,                  ! PROCESS SHAREABLE IMAGES
;  238          0733  1      lnk$procslib,                   ! PROCESS AN OBJECT LIBRARY FILE
;  239          0734  1      lnk$search,                     ! SYMBOL TABLE SEARCH
;  240          0735  1      lnk$searchlocal,                ! MODULE-LOCAL SYMBOL TABLE SEARCH
;  241          0736  1      lnk$addimage;                   ! ADD SHAREABLE IMAGE CLUSTER
;  242          0737  1
;  243          0738  1 external literal
;  244          0739  1      lib$_keyalrins,                 ! KEY ALREADY INSERTED IN TREE
;  245          0740  1      lin$_openin,
;  246          0741  1      lin$_badccc,                    ! BAD COMPILER COMPLETION CODE
;  247          0742  1      lin$_badpsc,                    ! ILLEGAL P-SECTION REFERENCED
;  248          0743  1      lin$_datmismch,                 ! CREATION DATE/TIME MISMATCH
;  249          0744  1      lin$_eomftl,                    ! EOM RECORD SPECIFIES ABORT
;  250          0745  1      lin$_errors,                    ! COMPILATION HAD FATAL ERRORS
;  251          0746  1      lin$_excpsc,                    ! TOO MANY PSECTS DEFINED
;  252          0747  1      lin$_fatalerror,                ! FATAL ERROR MESSAGE ISSUED
;  253          0748  1      lin$_format,                    ! OBJECT MODULE FORMAT ERROR
;  254          0749  1      lin$_gsdtyp,                    ! ILLEGAL GSD TYPE
;  255          0750  1      lin$_entidmtch,                 ! ASCIC IDENT MISMATCH
;  256          0751  1      lin$_entidmtchb,                ! BINARY IDENT MISMATCH
;  257          0752  1      lin$_entidmtcho,                ! IDENT OBJECT TYPE MISMATCH
;  258          0753  1      lin$_entidmtcht,                ! IDENT TYPE MISMATCH
;  259          0754  1      lin$_illfmlcnt,                 ! ILLEGAL FORMAL ARGUMENT COUNTS
;  260          0755  1      lin$_illnamelen,                ! ILLEGAL NAME LENGTH
;  261          0756  1      lin$_illrectyp,                 ! ILLEGAL RECORD TYPE
;  262          0757  1      lin$_illreclen,                 ! ILLEGAL RECORD LENGTH
;  263          0758  1      lin$_modnam,                    ! ILLEGAL MODULE NAME
;  264          0759  1      lin$_muldef,                    ! MULTIPLE SYMBOL DEFINITION
;  265          0760  1      lin$_muldefpsc,                 ! MULT. DEF OF PSECT IN SAME MODULE
;  266          0761  1      lin$_mulpsc,                    ! MULTIPLE PSECT ATTRIBUTES DEF.
;  267          0762  1      lin$_multfr,                    ! MULTIPLE TRANSFER ADDRESSES
;  268          0763  1      lin$_mulshrpsc,                 ! MULTIPLY DEFINED SHAREABLE IMAGE PSECT
;  269          0764  1      lin$_noeom,                     ! MISSING END OF MODULE ERROR
;  270          0765  1      lin$_noimgfil,                  ! IMAGE FILE NOT CREATED
;  271          0766  1      lin$_nomods,                    ! NO MODULES FOUND (IN THE LIBRARIES)
;  272          0767  1      lin$_nopscts,                   ! NO PROGRAM SECTIONS FOUND
;  273          0768  1      lin$_nudfenvs,                  ! NUMBER OF UNDEFINED ENVIRONMENTS
```

```
;  274          0769  1      lin$_nudfsyms,                  ! NUMBER OF UNDEFINED SYMBOLS
;  275          0770  1      lin$_nudflsyms,                 ! NUMBER OF UNDEFINED LOCAL SYMBOLS
;  276          0771  1      lin$_ovrali,                    ! OVERLAYED P-SECTION WITH DIFFERERENT ALIGNMENT
;  277          0772  1      lin$_pscali,                    ! ILLEGAL P-SECTION ALIGNMENT
;  278          0773  1      lin$_pscnxr,                    ! P-SECTION CONTAINING TRANSFER ADDRESS IS NOT EXE/REL
;  279          0774  1      lin$_reclng,                    ! ILLEGAL RECORD LENGTH
;  280          0775  1      lin$_rectyp,                    ! ILLEGAL RECORD TYPE
;  281          0776  1      lin$_seqnce,                    ! RECORDS IN ILLEGAL SEQUENCE
;  282          0777  1      lin$_shrpsclng,                 ! PSECT IN OBJ MOD BIGGER THAN ONE IN SHR IMAGE
;  283          0778  1      lin$_strlvl,                    ! ILLEGAL STRUCTURE LEVEL IN MODULE
;  284          0779  1      lin$_udfenv,                    ! LIST AN UNDEFINED ENVIRONMENT
;  285          0780  1      lin$_udfsym,                    ! EACH UNDEFINED SYMBOL
;  286          0781  1      lin$_wrners,                    ! COMPILER ISSUED WARNINGS
;  287          0782  1      lnk$k_max_filename_length ;     ! MAXIMUM FILENAME LENGTH
;  288          0783  1
;  289          0784  1  external
;  290          0785  1      lnk$gt_jpilst,
;  291          0786  1      lnk$gl_filesleft,
;  292          0787  1      lnk$al_rab          : block [rab$c_bln, byte],      ! RAB USED TO OPEN THE FILE
;  293          0788  1      lnk$al_sytblfmt,                                    ! SYMBOL LISTING FORMAT DESCRIPTION TABLE ADDRESS
;  294          0789  1      lnk$al_valctltb,                                    ! CREF BY VALUE CONTROL TABLE ADDRESS
;  295          0790  1      lnk$aw_version      : block [lid$c_size, byte],     ! VERSION ARRAY
;  296          0791  1      lnk$gl_ctlmsk       : block [, byte],               ! LINK CONTROL MASK
;  297          0792  1      lnk$gl_pshrnum,                                     ! NUMBER OF PSECT TO BE OUTPUT TO SHAREABLE IMAGE
;  298          0793  1      lnk$gt_imgsta       : block [, byte],               ! NAME OF "SYS$IMGSTA"
;  299          0794  1      lnk$gl_clulst       : vector [2],                   ! CLUSTER DESCRIPTOR LISTHEAD
;  300          0795  1      lnk$gl_defclu       : block [, byte],
;  301          0796  1      lnk$gl_record,                                      ! RECORD NUMBER IN THE FILE
;  302          0797  1      lnk$gl_shrsyms,                                     ! NUMBER OF SHAREABLE IMAGE SYMBOLS REFERENCED
;  303          0798  1      lnk$gl_shrimgs,                                     ! NUMBER OF SHAREABLE IMAGES REFERENCED
;  304          0799  1      lnk$gl_pscdflst,                                    ! PSECTS DEFINED BY OPTION LIST
;  305          0800  1      lnk$gl_maxsymsz,                                    ! MAXIMUM SYMBOL LENGTH SEEN
;  306          0801  1      lnk$gl_maxmodsz,                                    ! MAXIMUM MODULE LENGTH SEEN
;  307          0802  1      lnk$gl_curclu       : ref block [, byte],           ! CURRENT CLUSTER POINTER
;  308          0803  1      lnk$gl_curfil       : ref block [, byte],           ! CURRENT FDB POINTER
;  309          0804  1      lnk$gl_curomd       : ref block [, byte],           ! CURRENT OMD POINTER
;  310          0805  1      lnk$gl_libsym       : ref block [, byte],           ! POINTS TO SYMBOL CAUSING LOAD OF LIBRARY MODULE
;  311          0806  1      lnk$gt_imgid        : vector [, byte],              ! STORAGE OF IMAGE IDENT
;  312          0807  1      shrfiltype,                                         ! .EXE string
;  313          0808  1      defiletype,                                         ! .OBJ string
;  314          0809  1      libfiltype;                                         ! .OLB string
;  315          0810  1
;  316          0811  1  global
;  317          0812  1      lnk$gl_cmddst       : initial (0),                  ! COUNT OF OBJMODS WHICH CONTRIBUTE TO DST
;  318          0813  1      lnk$gw_pscdst       : word initial (0),             ! COUNT OF PSECTS IN THE ABOVE OBJMODS
;  319          0814  1      lnk$gl_udflst       : vector [2] initial (          ! UNDEFINED SYMBOL LISTHEAD
;  320          0815  1                                 lnk$gl_udflst,       ! INITIALLY POINTS TO
;  321          0816  1                                 lnk$gl_udflst),      ! ITSELF, SINCE EMPTY
;  322          0817  1      lnk$gl_envtree,                                     ! TREE HEAD FOR ENVIRONMENTS
;  323          0818  1      lnk$gl_udfenv       : vector [2] initial (          ! UNDEFINED ENVIRONMENT LISTHEAD
;  324          0819  1                                 lnk$gl_udfenv,
;  325          0820  1                                 lnk$gl_udfenv),
;  326          0821  1      lnk$gl_udflsy       : vector [2] initial (          ! LIST HEAD FOR UNDEFINED LOCAL SYMBOLS
;  327          0822  1                                 lnk$gl_udflsy,
;  328          0823  1                                 lnk$gl_udflsy),
;  329          0824  1      lnk$gl_entitree,                                    ! TREE HEAD FOR ENTITIES IDENT CHECK
;  330          0825  1      lnk$gl_dbgtfr,                                      ! DEBUGGER TRANSFER ADDRESS
```

```
;  331          0826  1      lnk$gl_dbgtfps      : ref block [, byte],          ! POINTER TO PSECT CONTAINING IT
;  332          0827  1      lnk$gl_tfradr,                                     ! TRANSFER ADDRESS
;  333          0828  1      lnk$gl_tfrpsc       : ref block [, byte],          ! POINTER TO PSECT CONTAINING IT
;  334          0829  1      lnk$gl_dbgestim,                                   ! ESTIMATE OF NUMBER OF BYTES FOR DST
;  335          0830  1      lnk$gl_omdtree,                                    ! TREE HEAD FOR OBJ MODULE DESCRIPTORS
;  336          0831  1      lnk$gw_dbgrecs      : word,                        ! NUMBER OF DEBUG RECORDS
;  337          0832  1      lnk$gw_nfiles       : word,                        ! NUMBER OF FILES OPENED
;  338          0833  1      lnk$gw_nmodules     : word,                        ! NUMBER OF MODULES FOUND
;  339          0834  1      lnk$gw_npsects      : word,                        ! NUMBER OF PSECTS IN TABLE
;  340          0835  1      lnk$gw_nsymbols     : word,                        ! NUMBER OF SYMBOLS IN TABLE
;  341          0836  1      lnk$gw_lsymbols     : word,                        ! NUMBER OF SYMBOLS IN THIS IMAGE
;  342          0837  1      lnk$gw_ncrosrfs     : word,                        ! NUMBER OF CROSS REFERENCES ENTERED
;  343          0838  1      lnk$gw_nudfsyms     : word,                        ! NUMBER OF UNDEFINED SYMBOLS
;  344          0839  1      lnk$gw_nlsyms       : word,                        ! NUMBER OF MODULE-LOCAL SYMBOLS
;  345          0840  1      lnk$gw_nudflsyms    : word,                        ! NUMBER OF UNDEFINED MODULE-LOCAL SYMBOLS
;  346          0841  1      lnk$gw_nudfenvs     : word ;                       ! NUMBER OF UNDEFINED ENVIRONMENTS
;  347          0842  1
;  348          0843  1  own
;  349          0844  1      omd_has_dbg         : byte initial (0),            ! CURRENT OBJMOD HAS DBG RECORDS
;  350          0845  1      lastobjmod          : ref block [, byte],          ! POINTER TO LAST OBJ MODULE DESCRIPTOR
;  351          0846  1      weaktfradr          : byte,                        ! TRUE IF CURRENT XFR ADDR IS WEAK
;  352          0847  1      lclsymgsd           : byte,                        ! TRUE IF GSD SUBTYPE IS LOCAL SYMBOL
;  353          0848  1      wordpsectgsd        : byte,                        ! TRUE IF GSD SUBTYPE HAS WORD OF PSECT NUMBER
;  354          0849  1      mhdseen             : byte initial (0),            ! FLAG MHD SUB-HEADER AS YET NOT SEEN
;  355          0850  1      lnmseen             : byte initial (0),            ! FLAG THAT COMPILER NAME SUB-HEADER IS SEEN
;  356          0851  1      lastrectyp          : byte,                        ! TYPE OF THE PREVIOUS RECORD
;  357          0852  1      currectyp           : byte initial (obj$c_eom),    ! TYPE OF THE CURRENT RECORD
;  358          0853  1      gsdoffset           : word,                        ! OFFSET INTO CONCATENATED GSD RECORD
;  359          0854  1      entrymask           : word,                        ! ENTRY POINT MASK OF CURRENT SYMBOL
;  360          0855  1      maxreclng           : word initial (obj$c_maxrecsiz), ! MAXIMUM LENGTH PERMISSIBLE IN THIS MODULE
;  361          0856  1      objrecdesc          : block [dsc$c_s_bln, byte],   ! STRING DESCRIPTOR FOR OBJECT RECORD
;  362          0857  1      symbolstring        : ref vector [, byte],         ! POINTER TO CURRENT SYMBOL
;  363          0858  1      symtabent           : ref block [, byte],          ! POINTER TO SYMBOL TABLE ENTRY
;  364          0859  1      symentnam           : ref block [, byte],          ! POINTER TO SYMBOL NAME BLOCK
;  365          0860  1                                                      ! PART OF SYMBOL
;  366          0861  1      imageidstring       : vector [sym$c_maxlng+1, byte], ! DEFAULT IMAGE IDENT STORAGE
;  367          0862  1      obmodesc            : block [omd$c_size, byte],    ! STATIC COPY OF OBJECT MODULE DESCRIPTOR
;  368          0863  1      last_lnkopt_fdb     : ref block [,byte]     ;      ! FDB OF LAST LNKOPT FILE FOR THIS MODULE
;  369          0864  1
;  370          0865  1  bind
;  371          0866  1      reclng = objrecdesc [dsc$w_length]  : word,                 ! SIZE OF RECORD READ
;  372          0867  1      objrec = objrecdesc [dsc$a_pointer] : ref block [, byte],   ! NAME POINTER PART OF DESCRIPTOR
;  373          0868  1      objvec = objrecdesc [dsc$a_pointer] : ref vector [, byte],  ! MUST ALSO ACCESS RECORD AS BYTE VECTOR
;  374          0869  1      recdispatch = plit (                                        ! SET UP MAXIMUM ALLOWED RECORD TYPE (i.e.,
;  375          0870  1                          prohdr,                                 ! 0 - MODULE HEADER
;  376          0871  1                          progsd,                                 ! 1 - GSD RECORDS
;  377          0872  1                          seqchk,                                 ! 2 - TIR - JUST CHECK CORRECT SEQUENCE
;  378          0873  1                          proeom,                                 ! 3 - END OF MODULE
;  379          0874  1                          countdbg,                               ! 4 - DBG - CHECK SEQUENCE AND COUNT RECORDS
;  380          0875  1                          countdbg,                               ! 5 - TBT - COUNT AND CHECK SEQUENCE O.K.
;  381          0876  1                          prolnkrec) : vector ;                   ! 6 - OPTIONS RECORD
;  382          0877  1
;  383          0878  1  global bind
;  384          0879  1      lnk$gt_envstring = cstring ('Environment'),
;  385          0880  1      lnk$gt_modstring = cstring ('Module'),                      ! FOR THE ERROR MESSAGE
;  386          0881  1      lnk$gt_symstring = cstring ('Symbol'),
;  387          0882  1      lnk$gt_pscstring = cstring ('Psect'),
```

```
;  388          0883  1     lnk$gt_clustring = cstring ('Cluster'),
;  389          0884  1     lnk$gt_entity    = cstring ('Entity'),
;  390          0885  1     lnk$gt_ident     = cstring ('Ident'),
;  391          0886  1     lnk$gt_objnam    = cstring ('Entity type'),
;  392          0887  1     lnk$gt_sysver    = cstring ('SYS$K_VERSION') ;              ! SYSTEM VERSION SYMBOL
;  393          0888  1
;  394          0889  1 !
;  395          0890  1 routine dummy = return 1 ;                                    ! FORCE BLISS TO PRINT THE

                                                                .TITLE  LNK_OBJPASS1
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000007  00000                 .LONG   7
00000000V 00000000V 00000000V 00000000V 00000000V 00000000V 00004 P.AAA:        .ADDRESS PROHDR, PROGSD, SEQCHK, PROEOM, -
                                                00000000V 0001C                         COUNTDBG, COUNTDBG, PROLNKREC
                                                      0B  00020 P.AAB:          .BYTE   11
                  74  6E  65  6D  6E  6F  72  69  76  6E  45  00021            .ASCII  \Environment\
                                                      06  0002C P.AAC:          .BYTE   6
                              65  6C  75  64  6F  4D  0002D                      .ASCII  \Module\
                                                      06  00033 P.AAD:          .BYTE   6
                              6C  6F  62  6D  79  53  00034                      .ASCII  \Symbol\
                                                      05  0003A P.AAE:          .BYTE   5
                                  74  63  65  73  50  0003B                      .ASCII  \Psect\
                                                      07  00040 P.AAF:          .BYTE   7
                          72  65  74  73  75  6C  43  00041                      .ASCII  \Cluster\
                                                      06  00048 P.AAG:          .BYTE   6
                              79  74  69  74  6E  45  00049                      .ASCII  \Entity\
                                                      05  0004F P.AAH:          .BYTE   5
                                  74  6E  65  64  49  00050                      .ASCII  \Ident\
                                                      0B  00055 P.AAI:          .BYTE   11
              65  70  79  74  20  79  74  69  74  6E  45  00056                  .ASCII  \Entity type\
                                                      0D  00061 P.AAJ:          .BYTE   13
4E  4F  49  53  52  45  56  5F  4B  24  53  59  53  00062                      .ASCII  \SYS$K_VERSION\

                                                                .PSECT  $OWN$,NOEXE,2

                                                      00  00000 OMD_HAS_DBG:
                                                                .BYTE   0
                                                  00001         .BLKB   3
                                                  00004 LASTOBJMOD:
                                                                .BLKB   4
                                                  00008 WEAKTFRADR:
                                                                .BLKB   1
                                                  00009 LCLSYMGSD:
                                                                .BLKB   1
                                                  0000A WORDPSECTGSD:
                                                                .BLKB   1
                                                      00  0000B MHDSEEN:.BYTE   0
                                                      00  0000C LNMSEEN:.BYTE   0
                                                  0000D LASTRECTYP:
                                                                .BLKB   1
                                                      03  0000E CURRECTYP:
                                                                .BYTE   3
                                                  0000F         .BLKB   1
```

```
                        00010 GSDOFFSET:
                                .BLKB   2
                        00012 ENTRYMASK:
                                .BLKB   2
                    0800 00014 MAXRECLNG:
                                .WORD   2048                                    ;
                        00016           .BLKB   2
                        00018 OBJRECDESC:
                                .BLKB   8
                        00020 SYMBOLSTRING:
                                .BLKB   4
                        00024 SYMTABENT:
                                .BLKB   4
                        00028 SYMENTNAM:
                                .BLKB   4
                        0002C IMAGEIDSTRING:
                                .BLKB   32
                        0004C OBMODESC:
                                .BLKB   2120
                        00894 LAST_LNKOPT_FDB:
                                .BLKB   4

                                .PSECT  $GLOBAL$,NOEXE,2

                00000000 00000 LNK$GL_OMDDST::
                                .LONG   0                                       ;
                    0000 00004 LNK$GW_PSCDST::
                                .WORD   0                                       ;
                        00006           .BLKB   2
00000000' 00000000' 00008 LNK$GL_UDFLST::
                                .ADDRESS LNK$GL_UDFLST, LNK$GL_UDFLST           ;
                        00010 LNK$GL_ENVTREE::
                                .BLKB   4
00000000' 00000000' 00014 LNK$GL_UDFENV::
                                .ADDRESS LNK$GL_UDFENV, LNK$GL_UDFENV           ;
00000000' 00000000' 0001C LNK$GL_UDFLSY::
                                .ADDRESS LNK$GL_UDFLSY, LNK$GL_UDFLSY           ;
                        00024 LNK$GL_ENTITREE::
                                .BLKB   4
                        00028 LNK$GL_DBGTFR::
                                .BLKB   4
                        0002C LNK$GL_DBGTFPS::
                                .BLKB   4
                        00030 LNK$GL_TFRADR::
                                .BLKB   4
                        00034 LNK$GL_TFRPSC::
                                .BLKB   4
                        00038 LNK$GL_DBGESTIM::
                                .BLKB   4
                        0003C LNK$GL_OMDTREE::
                                .BLKB   4
                        00040 LNK$GW_DBGRECS::
                                .BLKB   2
                        00042 LNK$GW_NFILES::
                                .BLKB   2
                        00044 LNK$GW_NMODULES::
                                .BLKB   2
```

```
00046 LNK$GW_NPSECTS::
                .BLKB   2
00048 LNK$GW_NSYMBOLS::
                .BLKB   2
0004A LNK$GW_LSYMBOLS::
                .BLKB   2
0004C LNK$GW_NCROSRFS::
                .BLKB   2
0004E LNK$GW_NUDFSYMS::
                .BLKB   2
00050 LNK$GW_NLSYMS::
                .BLKB   2
00052 LNK$GW_NUDFLSYMS::
                .BLKB   2
00054 LNK$GW_NUDFENVS::
                .BLKB   2

        ISD$C_SIZE==            88
        HDR$K_FILLCHR==         255
        IHD$K_SHR==             2
        IHD$K_ACTIVOFF==        48
        IHD$K_SYMDBGOFF==       68
        IHD$K_IMGIDOFF==        88
        IHD$K_PATCHOFF==        168
        IHD$K_MAXLENGTH==       168
        HDR$K_MINFILL==         2
        PSC$M_PSCBITS==         1023
        PSC$M_SHRBITS==         52
        RECLNG=                 OBJRECDESC
        OBJREC=                 OBJRECDESC+4
        OBJVEC=                 OBJRECDESC+4
        RECDISPATCH=            P.AAA
        LNK$GT_ENVSTRING==      P.AAB
        LNK$GT_MODSTRING==      P.AAC
        LNK$GT_SYMSTRING==      P.AAD
        LNK$GT_PSCSTRING==      P.AAE
        LNK$GT_CLUSTRING==      P.AAF
        LNK$GT_ENTITY==         P.AAG
        LNK$GT_IDENT==          P.AAH
        LNK$GT_OBJNAM==         P.AAI
        LNK$GT_SYSVER==         P.AAJ
                .EXTRN  LNK$CLOSEFILE, LNK$FILNAMDSC
                .EXTRN  LNK$OPENLIB, LNK$ALLOFDB
                .EXTRN  CRF$INSRTKEY, CRF$INSRTREF
                .EXTRN  LIB$INSERT_TREE
                .EXTRN  LIB$LOOKUP_TREE
                .EXTRN  LIB$TRAVERSE_TREE
                .EXTRN  LNK$COMPARE_PSCNOD
                .EXTRN  LNK$SRCPSCDEF, LNK$BINTIM
                .EXTRN  LNK$ALLOBLK, LNK$DEALBLK
                .EXTRN  LNK$EXIT, LNK$FNDPSCNAM
                .EXTRN  LNK$INSERT, LNK$NXTFIL
                .EXTRN  LNK$NXTREC, LNK$PROCSHRIM
                .EXTRN  LNK$PROCSLIB, LNK$SEARCH
                .EXTRN  LNK$SEARCHLOCAL
                .EXTRN  LNK$ADDIMAGE, LIB$_KEYALRINS
                .EXTRN  LIN$_OPENIN, LIN$_BADCCC
```

```
                                              .EXTRN   LIN$_BADPSC, LIN$_DATMISMCH
                                              .EXTRN   LIN$_EOMFTL, LIN$_ERRORS
                                              .EXTRN   LIN$_EXCPSC, LIN$_FATALERROR
                                              .EXTRN   LIN$_FORMAT, LIN$_GSDTYP
                                              .EXTRN   LIN$_ENTIDMTCH, LIN$_ENTIDMTCHB
                                              .EXTRN   LIN$_ENTIDMTCHO
                                              .EXTRN   LIN$_ENTIDMTCHT
                                              .EXTRN   LIN$_ILLFMLCNT, LIN$_ILLNAMELEN
                                              .EXTRN   LIN$_ILLRECTYP, LIN$_ILLRECLEN
                                              .EXTRN   LIN$_MODNAM, LIN$_MULDEF
                                              .EXTRN   LIN$_MULDEFPSC, LIN$_MULPSC
                                              .EXTRN   LIN$_MULTFR, LIN$_MULSHRPSC
                                              .EXTRN   LIN$_NOEOM, LIN$_NOIMGFIL
                                              .EXTRN   LIN$_NOMODS, LIN$_NOPSCTS
                                              .EXTRN   LIN$_NUDFENVS, LIN$_NUDFSYMS
                                              .EXTRN   LIN$_NUDFLSYMS, LIN$_OVRALI
                                              .EXTRN   LIN$_PSCALI, LIN$_PSCNXR
                                              .EXTRN   LIN$_RECLNG, LIN$_RECTYP
                                              .EXTRN   LIN$_SEQNCE, LIN$_SHRPSCLNG
                                              .EXTRN   LIN$_STRLVL, LIN$_UDFENV
                                              .EXTRN   LIN$_UDFSYM, LIN$_WRNERS
                                              .EXTRN   LNK$K_MAX_FILENAME_LENGTH
                                              .EXTRN   LNK$GT_JPILST, LNK$GL_FILESLEFT
                                              .EXTRN   LNK$AL_RAB, LNK$AL_SYTBLFMT
                                              .EXTRN   LNK$AL_VALCTLTB
                                              .EXTRN   LNK$AW_VERSION, LNK$GL_CTLMSK
                                              .EXTRN   LNK$GL_PSHRNUM, LNK$GT_IMGSTA
                                              .EXTRN   LNK$GL_CLULST, LNK$GL_DEFCLU
                                              .EXTRN   LNK$GL_RECORD, LNK$GL_SHRSYMS
                                              .EXTRN   LNK$GL_SHRIMGS, LNK$GL_PSCDFLST
                                              .EXTRN   LNK$GL_MAXSYMSZ
                                              .EXTRN   LNK$GL_MAXMODSZ
                                              .EXTRN   LNK$GL_CURCLU, LNK$GL_CURFIL
                                              .EXTRN   LNK$GL_CUROMD, LNK$GL_LIBSYM
                                              .EXTRN   LNK$GT_IMGID, SHRFILTYPE
                                              .EXTRN   DEFILETYPE, LIBFILTYPE

                                              .PSECT   $CODE$,NOWRT,2

                         0000 00000 DUMMY:    .WORD    Save nothing                        ; 0890
           50         01   D0 00002           MOVL     #1, R0                              ;
                           04 00005           RET                                          ;

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 0000

;  396         0891  1                                   ! ABOVE DECLARATIONS
```

```
;  398          0892  1 routine prolnkrec =
;  399          0893  2 begin
;  400          0894  2
;  401          0895  3         routine setup_include_list ( omdlstadr ) : novalue =
;  402          0896  3         begin
;  403          0897  3         !
;  404          0898  3         !  This routine sets up an include list (in simple linked
;  405          0899  3         !    list form) for modules in a library file spec
;  406          0900  3         !
;  407          0901  3         bind    lnkoptrec = .objrec : block [, byte];
;  408          0902  3         local   recptr,
;  409          0903  3                 prevmodule,
;  410          0904  3                 currentmodule;
;  411          0905  3
;  412          0906  3         recptr = .objvec + $byteoffset (lnk$t_name)             ! Point to module list in the
;  413          0907  3                          + .lnkoptrec[lnk$w_namlng] ;           !  LNKOPT record
;  414          0908  3         prevmodule = 0;
;  415          0909  3         while .(.recptr)<0,8> neq 0                             ! Count field of 0 is end of
;  416          0910  4         do  begin                                               !   module list
;  417          0911  4             lnk$alloblk ((.(.recptr)<0,8>)+5, currentmodule);   ! Allocate memory for name
;  418          0912  4             if .prevmodule eql 0                                ! If first module
;  419          0913  4             then .omdlstadr  = .currentmodule                   !   then set up list header
;  420          0914  4             else .prevmodule = .currentmodule;                  !   otherwise, just place in
;  421          0915  4
;  422          0916  4             prevmodule = .currentmodule;                        !   linked list
;  423          0917  4             ch$move (.(.recptr)<0,8>, .recptr+1, .currentmodule+5);     ! Move in name,
;  424          0918  4             (.currentmodule+4)<0,8> = .(.recptr)<0,8>;          !   fill in length
;  425          0919  4             .currentmodule          =  0;
;  426          0920  4             recptr                  = .recptr + .(.recptr)<0,8> + 1;    : Move ptr to next module name
;  427          0921  3             end;
;  428          0922  3         return;
;  429          0923  2         end;


                                07FC 00000 SETUP_INCLUDE_LIST:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10        ; 0895
                5A 00000000'    EF  9E 00002        MOVAB   OBJREC, R10
                5E              04  C2 00009        SUBL2   #4, SP
                50              6A  D0 0000C        MOVL    OBJREC, R0                      ; 0901
                56        04    A0  3C 0000F        MOVZWL  4(R0), R6                       ; 0907
                56              6A  C0 00013        ADDL2   OBJREC, R6
                56              06  C0 00016        ADDL2   #6, RECPTR
                                59  D4 00019        CLRL    PREVMODULE                      ; 0908
                58              66  9A 0001B 1$:     MOVZBL  (RECPTR), R8                    ; 0909
                                32  13 0001E        BEQL    4$
                                5E  DD 00020        PUSHL   SP                              ; 0911
                          05    A8  9F 00022        PUSHAB  5(R8)
00000000G       00              02  FB 00025        CALLS   #2, LNK$ALLOBLK
                57              6E  D0 0002C        MOVL    CURRENTMODULE, R7               ; 0913
                                59  D5 0002F        TSTL    PREVMODULE                      ; 0912
                                06  12 00031        BNEQ    2$
              04    BC          57  D0 00033        MOVL    R7, @OMDLSTADR                  ; 0913
                                03  11 00037        BRB     3$
                69              57  D0 00039 2$:     MOVL    R7, (PREVMODULE)                ; 0914
```

```
                                59              57  D0 0003C 3$:    MOVL    R7, PREVMODULE                          ; 0916
                        05  A7      01  A6      58  28 00043        MOVC3   R8, 1(RECPTR), 5(R7)                    ; 0917
                                    04  A7      58  90 00045        MOVB    R8, 4(R7)                               ; 0918
                                                67  D4 00049        CLRL    (R7)                                    ; 0919
                                56          01 A846 9E 0004B        MOVAB   1(R8)[RECPTR], RECPTR                   ; 0920
                                                C9  11 00050        BRB     1$                                      ; 0909
                                                    04 00052 4$:    RET                                             ; 0923

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 0006
```

```
;  431          0924  2 local
;  432          0925  2         lastfdb         : ref block [,byte],
;  433          0926  2         auxfnb          : ref block [,byte],
;  434          0927  2         input_fab       : block [fab$c_bln, byte],
;  435          0928  2         errorcode,
;  436          0929  2         next_fdb,
;  437          0930  2         filnamadr,
;  438          0931  2         fildesblk       : ref block [,byte];
;  439          0932  2 bind
;  440          0933  2         lnkoptrec = .objrec : block [, byte];
;  441          0934  2 !
;  442          0935  2 ! PROCESS LINKER OPTIONS RECORD
;  443          0936  2 !
;  444          0937  2 if .lnkoptrec [lnk$b_lnktyp] gtr lnk$c_maxrectyp                 ! Valid subrecord type?
;  445          0938  2 or .lnkoptrec [lnk$w_namlng] gtr nam$c_maxrss                    ! Reasonable-looking name length?
;  446          0939  2 then return false;
;  447          0940  2
;  448          0941  2 lnk$alloblk (.lnkoptrec [lnk$w_namlng], filnamadr);              ! allocate storage for filename
;  449          0942  2 ch$move (.lnkoptrec [lnk$w_namlng] , lnkoptrec [lnk$t_name] , .filnamadr);
;  450          0943  2 lnk$allofdb (fildesblk);
;  451          0944  2 auxfnb = fildesblk [fdb$t_auxfnb];
;  452        P 0945  2 $fab_init (fab = input_fab                                      ! intialize the fab
;  453        P 0946  2           ,fac = get                                            ! for input
;  454        P 0947  2           ,fop = sqo
;  455        P 0948  2           ,shr = (upi,get,put)
;  456        P 0949  2           ,fns = .lnkoptrec [lnk$w_namlng]
;  457        P 0950  2           ,fna = .filnamadr
;  458        P 0951  2           ,nam = .auxfnb
;  459        P 0952  2           ,dns = 4
;  460        P 0953  2           ,dna = ch$ptr (libfiltype)
;  461          0954  2           );
;  462          0955  2 fildesblk [fdb$w_usrnamlen] = .lnkoptrec [lnk$w_namlng];        ! fill in file name
;  463          0956  2 fildesblk [fdb$l_usrnamadr] = .filnamadr;
;  464          0957  2
;  465          0958  2 select .lnkoptrec [lnk$b_lnktyp] of                             ! determine the file type
;  466          0959  2         set
;  467          0960  2         [lnk$c_olb,
;  468          0961  2          lnk$c_oli] : fildesblk [fdb$v_libr]    = true;         ! all of these are libraries
;  469          0962  2         [lnk$c_olb] : fildesblk [fdb$v_libsrch] = true;         ! olb to be searched
;  470          0963  2         [lnk$c_shr] : fildesblk [fdb$v_imglib]  = true;         ! shareable image LIBRARY
;  471          0964  3         [lnk$c_oli] : begin                                    ! object lib with include list
;  472          0965  3                       fildesblk [fdb$v_libextr] = true;
;  473          0966  3                       fildesblk [fdb$v_libsrch] = .lnkoptrec [lnk$v_libsrch];
;  474          0967  3                       setup_include_list (fildesblk [fdb$l_omdlst]) ;   ! put modules in linked list
;  475          0968  2                       end ;
;  476          0969  3         [lnk$c_obj] : begin                                    ! regular object module
;  477          0970  3                       fildesblk [fdb$v_selser]  = .lnkoptrec [lnk$v_selser];
;  478          0971  3                       input_fab [fab$l_dna]     = ch$ptr (defiletype) ;
;  479          0972  2                       end ;
;  480          0973  3         [lnk$c_sha] : begin                                    ! shareable image
;  481          0974  3                       fildesblk [fdb$v_shr]     = true;
;  482          0975  3                       input_fab [fab$l_dna]     = ch$ptr (shrfiltype) ;
;  483          0976  2                       end ;
;  484          0977  2         tes;
;  485          0978  2
;  486          0979  2
;  487          0980  2 if  .fildesblk[fdb$v_libr] or .fildesblk[fdb$v_imglib]          ! if a library file
```

```
;  488          0981  3 then begin
;  489          0982  3      $parse (fab=input_fab);                          !   then fill in name block enough
;  490          0983  4      if (errorcode = $search (fab=input_fab))
;  491          0984  4      then begin                                       !    to do a library open by name block
;  492          0985  4           local          savcurfil;
;  493          0986  4           savcurfil      = .lnk$gl_curfil;            ! set ptr to current fdb for the temporary b
;  494          0987  4           lnk$gl_curfil = .fildesblk;
;  495          0988  4           errorcode      =  lnk$openlib (.auxfnb);    !   of lnk$openlib
;  496          0989  4           lnk$gl_curfil = .savcurfil;                 ! must now restore it
;  497          0990  3           end;
;  498          0991  3      end
;  499          0992  3 else begin                                            ! not a library file
;  500          0993  3      $getjpi (itmlst=lnk$gt_jpilst);
;  501          0994  3      if .lnk$gl_filesleft leq 3 then lnk$closefile (); ! close a file if needed
;  502          0995  3      errorcode = $open (fab=input_fab);               !     and try again
;  503          0996  3      fildesblk [fdb$w_ifi] = .input_fab [fab$w_ifi];  ! save ifi
;  504          0997  2      end;
;  505          0998  2
;  506          0999  2
;  507          1000  2 if not .errorcode                                     !  if either of the opens was a failure,
;  508          1001  2 then signal_stop (lin$_openin, 1                      !    then signal and quit
;  509          1002  2                   ,lnk$filnamdsc (input_fab), .errorcode
;  510          1003  2                   ,.input_fab [fab$l_stv]
;  511          1004  2                   );
;  512          1005  2
;  513          1006  2 if .fildesblk [fdb$v_shr]                             ! Shareable image...?
;  514          1007  3 then begin
;  515          1008  3      local      moduledsc : vector [2, long];
;  516          1009  3
;  517          1010  3      moduledsc [0] = .auxfnb [nam$b_name];            ! Shareable image spec: set up name
;  518          1011  3      moduledsc [1] = .auxfnb [nam$l_name];            !   and create its very own cluster
;  519          1012  3      lnk$addimage (moduledsc);
;  520          1013  3      end
;  521          1014  3 else begin                                            ! Not a shareable image spec
;  522          1015  3      if  .last_lnkopt_fdb eql 0                       ! if first LNKOPT rec for this file
;  523          1016  3      then last_lnkopt_fdb = .lnk$gl_curfil;           ! set last_lnkopt_fdb to current file
;  524          1017  3
;  525          1018  3      if  .lnk$gl_curclu[clu$l_lstfdb] eql .last_lnkopt_fdb      ! If last fdb is last in current cluster
;  526          1019  3      then lnk$gl_curclu[clu$l_lstfdb] = .fildesblk;             ! then must update last cluster fdb
;  527          1020  3
;  528          1021  3      next_fdb                         = .last_lnkopt_fdb [fdb$l_nxtfdb];  ! Save ptr to next fdb
;  529          1022  3      last_lnkopt_fdb [fdb$l_nxtfdb] = .fildesblk;               ! Point current to this new fdb
;  530          1023  3      fildesblk [fdb$l_nxtfdb]       = .next_fdb;               ! Point new fdb to last
;  531          1024  3      last_lnkopt_fdb                = .fildesblk;              ! Update fdb of last LNKOPT file
;  532          1025  2      end;
;  533          1026  2 return true;
;  534          1027  1 end;


                                                .EXTRN  SYS$PARSE, SYS$SEARCH
                                                .EXTRN  SYS$GETJPI, SYS$OPEN

                                    OFFC 00000 PROLNKREC:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0892
                5B 00000000G   00   9E 00002    MOVAB   LNK$GL_CURFIL, R11                          ;
                5A 00000000'   EF   9E 00009    MOVAB   LAST_LNKOPT_FDB, R10                        ;
```

```
                                    5E      A0    AE   9E 00010          MOVAB   -96(SP), SP
                                    57    F788    CA   D0 00014          MOVL    OBJREC, R7                                  ;0933
                                    59      01    A7   9A 00019          MOVZBL  1(R7), R9                                   ;0937
                                    04            59   91 0001D          CMPB    R9, #4
                                                  06   1A 00020          BGTRU   1$
                                  00FF    8F      04   A7   B1 00022          CMPW    4(R7), #255                            ;0938
                                                  03   1B 00028 1$:       BLEQU   2$
                                                01A2   31 0002A          BRW     20$
                                                  5E   DD 0002D 2$:       PUSHL   SP                                          ;0941
                                                7E      04   A7   3C 0002F          MOVZWL  4(R7), -(SP)
                                00000000G       00           02   FB 00033          CALLS   #2, LNK$ALLOBLK
                                        00  BE      06   A7      04   A7   28 0003A          MOVC3   4(R7), 6(R7), @FILNAMADR           ;0942
                                                      04   AE   9F 00041          PUSHAB  FILDESBLK                           ;0943
                                00000000G       00           01   FB 00044          CALLS   #1, LNK$ALLOFDB
                                                58      04   AE   D0 0004B          MOVL    FILDESBLK, R8                       ;0944
                                                56      26   A8   9E 0004F          MOVAB   38(R8), AUXFNB
                0050    8F              00              6E           00   2C 00053          MOVC5   #0, (SP), #0, #80, $RMS_PTR         ;0954
                                                      10   AE           0005A
                                          10    AE  5003    8F   B0 0005C          MOVW    #20483, $RMS_PTR
                                          14    AE    40    8F   9A 00062          MOVZBL  #64, $RMS_PTR+4
                                          26    AE  4302    8F   B0 00067          MOVW    #17154, $RMS_PTR+22
                                          2F    AE              02   90 0006D          MOVB    #2, $RMS_PTR+31
                                          38    AE              56   D0 00071          MOVL    AUXFNB, $RMS_PTR+40
                                          3C    AE              6E   D0 00075          MOVL    FILNAMADR, $RMS_PTR+44
                                          40    AE  00000000G   00   9E 00079          MOVAB   LIBFILTYPE, $RMS_PTR+48
                                          44    AE      04    A7   90 00081          MOVB    4(R7), $RMS_PTR+52
                                          45    AE              04   90 00086          MOVB    #4, $RMS_PTR+53
                                          0C    A8      04    A7   B0 0008A          MOVW    4(R7), 12(R8)                       ;0955
                                          10    A8              6E   D0 0008F          MOVL    FILNAMADR, 16(R8)                   ;0956
                                                        59   D5 00093          TSTL    R9                                  ;0960
                                                        05   13 00095          BEQL    3$
                                                02            59   91 00097          CMPB    R9, #2
                                                              04   12 0009A          BNEQ    4$
                                          0A    A8              02   88 0009C 3$:       BISB2   #2, 10(R8)                      ;0961
                                                        59   D5 000A0 4$:       TSTL    R9                                  ;0962
                                                        05   12 000A2          BNEQ    5$
                                          0A    A8      80    8F   88 000A4          BISB2   #128, 10(R8)
                                                01            59   91 000A9 5$:       CMPB    R9, #1                              ;0963
                                                              04   12 000AC          BNEQ    6$
                                          0B    A8              01   88 000AE          BISB2   #1, 11(R8)
                                                02            59   91 000B2 6$:       CMPB    R9, #2                              ;0964
                                                              19   12 000B5          BNEQ    7$
                                          0A    A8      40    8F   88 000B7          BISB2   #64, 10(R8)                      ;0965
                50              02  A7              01           01   EF 000BC          EXTZV   #1, #1, 2(R7), R0                   ;0966
        0A      A8              01                  07           50   F0 000C2          INSV    R0, #7, #1, 10(R8)
                                                              04   A8   9F 000C8          PUSHAB  4(R8)                           ;0967
                                        FEDD    CF            01   FB 000CB          CALLS   #1, SETUP_INCLUDE_LIST
                                                03            59   91 000D0 7$:       CMPB    R9, #3                              ;0969
                                                              0F   12 000D3          BNEQ    8$
        0A      A8              01                  03      02   A7   F0 000D5          INSV    2(R7), #3, #1, 10(R8)               ;0970
                                          40    AE  00000000G   00   9E 000DC          MOVAB   DEFILETYPE, INPUT_FAB+48            ;0971
                                                04            59   91 000E4 8$:       CMPB    R9, #4                              ;0973
                                                              0C   12 000E7          BNEQ    9$
                                          0A    A8              04   88 000E9          BISB2   #4, 10(R8)                      ;0974
                                          40    AE  00000000G   00   9E 000ED          MOVAB   SHRFILTYPE, INPUT_FAB+48            ;0975
                                  04      0A    A8              01   E0 000F5 9$:       BBS     #1, 10(R8), 10$                     ;0980
                                                31      0B    A8   E9 000FA          BLBC    11(R8), 11$
```

```
                                10   AE  9F 000FE 10$:   PUSHAB  INPUT_FAB                          ; 0982
00000000G  00                       01  FB 00101        CALLS   #1, S^S$PARSE
                                10   AE  9F 00108        PUSHAB  INPUT_FAB                          ; 0983
00000000G  00                       01  FB 0010B        CALLS   #1, S^S$SEARCH
                                53      50  D0 00112    MOVL    R0, ERRORCODE
                                51      53  E9 00115    BLBC    ERRORCODE, 14$
                                52      6B  D0 00118    MOVL    LNK$GL_CURFIL, SAVCURFIL           ; 0986
                                6B      58  D0 0011B    MOVL    R8, LNK$GL_CURFIL                  ; 0987
                                        56  DD 0011E    PUSHL   AUXFNB                             ; 0988
00000000G  00                       01  FB 00120        CALLS   #1, LNK$OPENLIB
                                53      50  D0 00127    MOVL    R0, ERRORCODE
                                6B      52  D0 0012A    MOVL    SAVCURFIL, LNK$GL_CURFIL           ; 0989
                                        37  11 0012D    BRB     13$                                ; 0980
                                        7E  7C 0012F 11$:   CLRQ    -(SP)                          ; 0993
                                        7E  D4 00131        CLRL    -(SP)
                    00000000G   00      9F  00133        PUSHAB  LNK$GT_JPILST
                                        7E  7C 00139        CLRQ    -(SP)
                                        7E  D4 0013B        CLRL    -(SP)
00000000G  00                       07  FB 0013D        CALLS   #7, SYS$GETJPI
                03  00000000G  00      D1  00144        CMPL    LNK$GL_FILESLEFT, #3               ; 0994
                                07  14 0014B        BGTR    12$
00000000G  00                       00  FB 0014D        CALLS   #0, LNK$CLOSEFILE
                                10   AE  9F 00154 12$:   PUSHAB  INPUT_FAB                          ; 0995
00000000G  00                       01  FB 00157        CALLS   #1, S^S$OPEN
                                53      50  D0 0015E    MOVL    R0, ERRORCODE
          24    A8          12   AE  B0 00161        MOVW    INPUT_FAB+2, 36(R8)                ; 0996
                20                53  E8 00166 13$:   BLBS    ERRORCODE, 15$                     ; 1000
                                1C   AE  DD 00169 14$:   PUSHL   INPUT_FAB+12                      ; 1003
                                        53  DD 0016C    PUSHL   ERRORCODE                          ; 1002
                                18   AE  9F 0016E        PUSHAB  INPUT_FAB
00000000G  00                       01  FB 00171        CALLS   #1, LNK$FILNAMDSC                  :
                                        50  DD 00178    PUSHL   R0
                                        01  DD 0017A    PUSHL   #1                                 ; 1001
                    00000000G   8F      DD  0017C        PUSHL   #LIN$_OPENIN
00000000G  00                       05  FB 00182        CALLS   #5, LIB$STOP
          0A    A8                02  E1 00189 15$:   BBC     #2, 10(R8), 16$                    ; 1006
          08    AE          3B    A6  9A 0018E        MOVZBL  59(AUXFNB), MODULEDSC              ; 1010
          0C    AE          4C    A6  D0 00193        MOVL    76(AUXFNB), MODULEDSC+4            ; 1011
                            08    AE  9F 00198        PUSHAB  MODULEDSC                          ; 1012
00000000G  00                       01  FB 0019B        CALLS   #1, LNK$ADDIMAGE
                                        27  11 001A2    BRB     19$                                ; 1006
                                        6A  D5 001A4 16$:   TSTL    LAST_LNKOPT_FDB                ; 1015
                                        03  12 001A6        BNEQ    17$
                6A                      6B  D0 001A8        MOVL    LNK$GL_CURFIL, LAST_LNKOPT_FDB ; 1016
                50  00000000G       00  D0  001AB 17$:   MOVL    LNK$GL_CURCLU, R0                  ; 1018
                51                      6A  D0 001B2        MOVL    LAST_LNKOPT_FDB, R1
                51          0C      A0  D1  001B5        CMPL    12(R0), R1
                                        04  12 001B9        BNEQ    18$
          0C    A0                      58  D0 001BB        MOVL    R8, 12(R0)                     ; 1019
                50                      61  D0 001BF 18$:   MOVL    (R1), NEXT_FDB                 ; 1021
                61                      58  D0 001C2        MOVL    R8, (R1)                       ; 1022
                68                      50  D0 001C5        MOVL    NEXT_FDB, (R8)                 ; 1023
                6A                      58  D0 001C8        MOVL    R8, LAST_LNKOPT_FDB            ; 1024
                50                      01  D0 001CB 19$:   MOVL    #1, R0                         ; 1026
                                            04  001CE        RET
                                        50  D4 001CF 20$:   CLRL    R0                             ; 1027
                                            04  001D1        RET
```

; Routine Size: 466 bytes,   Routine Base: $CODE$ + 0059

```
;  536          1028  1 routine checkpddentry (pdd) =
;  537          1029  2     begin
;  538          1030  2 !
;  539          1031  2 ! ROUTINE CALLED BY LIB$TRAVERSE_TREE FOR EACH PSECT DEFINED BY OPTION
;  540          1032  2 !
;  541          1033  2 map
;  542          1034  2         pdd : ref block [, byte];
;  543          1035  2 local
;  544          1036  2         psctdesc : ref block [, byte];
;  545          1037  2
;  546          1038  2 if (.pdd [pdd$b_namlng] and %x'80') eql 0                        ! IF NOT DEFINED YET
;  547          1039  3 then begin
;  548          1040  3      lnk$fndpscnam (pdd [pdd$b_namlng], .pdd [pdd$w_flags]      ! THEN DO SO NOW
;  549          1041  4                     ,((.pdd [pdd$w_flags] and gps$m_gbl) neq 0)
;  550          1042  3                     ,psctdesc
;  551          1043  3                     );
;  552          1044  3       psctdesc [psc$w_flags] = .pdd [pdd$w_flags];              ! SET THE FLAGS
;  553          1045  3
;  554          1046  3      if  (psctdesc [psc$b_align] = .pdd [pdd$b_align]) eql %x'FF' ! AND THE ALIGNMENT
;  555          1047  3      then psctdesc [psc$b_align] = 0;
;  556          1048  2      end;
;  557          1049  2
;  558          1050  2 return true
;  559          1051  1 end;

                                0004 00000 CHECKPDDENTRY:
                                           .WORD   Save R2                          ; 1028
                          5E               04  C2 00002         SUBL2   #4, SP
                          52        04     AC  D0 00005         MOVL    PDD, R2     ; 1038
                                    0F     A2  95 00009         TSTB    15(R2)
                                           32  19 0000C         BLSS    2$
                                           5E  DD 0000E         PUSHL   SP          ; 1040
                                           7E  D4 00010         CLRL    -(SP)       ; 1041
                    02          0A     A2      04  E1 00012    BBC     #4, 10(R2), 1$
                                           6E  D6 00017         INCL    (SP)
                          7E        0A     A2  3C 00019 1$:     MOVZWL  10(R2), -(SP)   ; 1040
                                    0F     A2  9F 0001D         PUSHAB  15(R2)
                 00000000G  00             04  FB 00020         CALLS   #4, LNK$FNDPSCNAM
                          50               6E  D0 00027         MOVL    PSCTDESC, R0    ; 1044
                             0A     A0     0A  A2  B0 0002A     MOVW    10(R2), 10(R0)
                          52        0E     A2  9A 0002F         MOVZBL  14(R2), R2      ; 1046
                             2C     A0         52  90 00033     MOVB    R2, 44(R0)
                             FF     8F         52  91 00037     CMPB    R2, #255
                                               03  12 0003B     BNEQ    2$
                                    2C     A0  94 0003D         CLRB    44(R0)      ; 1047
                          50                   01  D0 00040 2$:  MOVL    #1, R0      ; 1050
                                               04  00043        RET                 ; 1051

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 022B
```

```
;  561          1052  1 global routine lnk$insudfsym (symboladdr) : novalue =
;  562          1053  2     begin
;  563          1054  2 !
;  564          1055  2 !       THIS ROUTINE INSERTS AN UNDEFINED SYMBOL INTO THE LINKED LIST
;  565          1056  2 !       OF UNDEFINED SYMBOLS.
;  566          1057  2 !
;  567          1058  2 map
;  568          1059  2         symboladdr : ref block [, byte];        ! REALLY A POINTER
;  569          1060  2 builtin
;  570          1061  2         insque;
;  571          1062  2 local
;  572          1063  2         ch_result,
;  573          1064  2         listhead        : ref vector [, long],
;  574          1065  2         nxtsyment       : ref block [, byte];           ! POINTER TO SYMBOL VALUE BLOCK
;  575          1066  2 bind
;  576          1067  2         symbolname = .symboladdr                        ! POINT TO NAME PART
;  577          1068  2                      - .symboladdr [sym$b_naming]
;  578          1069  2                      - snb$c_fxdlen                  : block [, byte];
;  579          1070  2
;  580          1071  2 !
;  581          1072  2 !       FIND THE SPOT TO INSERT THE NEW SYMBOL IN THE UNDEFINED LIST
;  582          1073  2 !
;  583          1074  2 listhead = lnk$gl_udflst;
;  584          1075  2 if .lclsymgsd then listhead = lnk$gl_udflsy;
;  585          1076  2
;  586          1077  2 nxtsyment = .listhead;                                 ! SCAN THE UNDEFINED LIST
;  587          1078  2
;  588          1079  2 while (nxtsyment = .nxtsyment [sym$l_udflink]) neq .listhead
;  589          1080  3 do  begin
;  590          1081  3     bind                                               ! POINT TO NAME PART
;  591          1082  3         nxtsymnam = .nxtsyment - .nxtsyment [sym$b_naming]
;  592          1083  3                               - snb$c_fxdlen           : block [, byte];
;  593          1084  3
;  594          1085  4     if (ch_result = ch$compare  (.nxtsymnam [snb$b_naming]      ! IF GREATER
;  595          1086  4                                 , nxtsymnam [snb$t_name]
;  596          1087  4                                 ,.symbolname [snb$b_naming]
;  597          1088  4                                 , symbolname [snb$t_name]
;  598          1089  4                                 )
;  599          1090  3         ) gtr 0
;  600          1091  3     then exitloop                                      ! THEN WE ARE ALL DONE
;  601          1092  3     else if .ch_result eql 0                           ! IF EQUAL
;  602          1093  3         then return;                                   ! THEN ALREADY IN LIST, SO RETURN
;  603          1094  2     end;
;  604          1095  2
;  605          1096  2 insque (symboladdr [sym$l_udflink], .nxtsyment [sym$l_udfblink]);       ! INSERT IN UNDEFINED LIST
;  606          1097  2
;  607          1098  2 if not .symboladdr [sym$v_weak]                         ! IF NOT A WEAK REFERENCE
;  608          1099  2 then if .lclsymgsd
;  609          1100  2      then lnk$gw_nudflsyms = .lnk$gw_nudflsyms + 1      ! THEN COUNT AS UNDEFINED
;  610          1101  2      else lnk$gw_nudfsyms  = .lnk$gw_nudfsyms + 1;
;  611          1102  2
;  612          1103  2 return;
;  613          1104  1 end;
```

```
                                   0FFC 00000         .ENTRY  LNK$INSUDFSYM, Save R2,R3,R4,R5,R6,R7,R8,-   ; 1052
                                                              R9,R10,R11                                 ;
                 5B 00000000'   EF  9E 00002          MOVAB   LCLSYMGSD, R11                             ;
                 5A 00000000'   EF  9E 00009          MOVAB   LNK$GL_UDFLST, R10                         ;
                 57        04   AC  D0 00010          MOVL    SYMBOLADDR, R7                             ; 1067
                 50        0F   A7  9A 00014          MOVZBL  15(R7), R0                                 ; 1068
        50       57             50  C3 00018          SUBL3   R0, R7, R0                                 ;
                 56        FB   A0  9E 0001C          MOVAB   -5(R0), R6                                 ; 1069
                 59             6A  9E 00020          MOVAB   LNK$GL_UDFLST, LISTHEAD                    ; 1074
                 04             6B  E9 00023          BLBC    LCLSYMGSD, 1$                              ; 1075
                 59        14   AA  9E 00026          MOVAB   LNK$GL_UDFLSY, LISTHEAD                    ;
                 54             59  D0 0002A 1$:      MOVL    LISTHEAD, NXTSYMENT                        ; 1077
                 54             64  D0 0002D 2$:      MOVL    (NXTSYMENT), NXTSYMENT                     ; 1079
                 59             54  D1 00030          CMPL    NXTSYMENT, LISTHEAD                        ;
                                2B  13 00033          BEQL    4$                                         ;
                 50        0F   A4  9A 00035          MOVZBL  15(NXTSYMENT), R0                          ; 1082
        50       54             50  C3 00039          SUBL3   R0, NXTSYMENT, R0                          ;
                 50             05  C2 0003D          SUBL2   #5, R0                                     ; 1083
                 52        04   A0  9A 00040          MOVZBL  4(R0), R2                                  ; 1085
                 51        04   A6  9A 00044          MOVZBL  4(R6), R1                                  ; 1087
                 55             01  D0 00048          MOVL    #1, R5                                     ; 1088
  51             00       05    A0             52  2D 0004B          CMPC5   R2, 5(R0), #0, R1, 5(R6)    ;
                           05   A6     00051                                                             ;
                                03  1A 00053          BGTRU   3$                                         ;
                 55             01  D9 00055          SBWC    #1, R5                                     ;
                 58             55  D0 00058 3$:      MOVL    R5, CH_RESULT                              ;
                                03  14 0005B          BGTR    4$                                         ; 1090
                                CE  12 0005D          BNEQ    2$                                         ; 1092
                                    04 0005F          RET                                                ; 1093
             04  B4             67  0E 00060 4$:      INSQUE  (R7), @4(NXTSYMENT)                        ; 1096
                 50        04   AC  D0 00064          MOVL    SYMBOLADDR, R0                             ; 1098
                 0A        0A   A0  E8 00068          BLBS    10(R0), 6$                                 ;
                 04             6B  E9 0006C          BLBC    LCLSYMGSD, 5$                              ; 1099
                           4A   AA  B6 0006F          INCW    LNK$GW_NUDFLSYMS                           ; 1100
                                    04 00072          RET                                                ;
                           46   AA  B6 00073 5$:      INCW    LNK$GW_NUDFSYMS                            ; 1101
                                    04 00076 6$:      RET                                                ; 1104

; Routine Size:  119 bytes,    Routine Base:  $CODE$ + 026F

;  614         1105  1
```

```
;  616          1106  1 routine seqchk =
;  617          1107  1 !
;  618          1108  1 !       ROUTINE WHICH VALIDATES THAT RECORDS ARE IN CORRECT SEQUENCE.
;  619          1109  1 !       RETURNS VALUE FALSE IF NOT, TRUE OTHERWISE. ALSO TURNS OFF THE NO
;  620          1110  1 !       BINARY FLAG WHEN A TIR RECORD IS SEEN.
;  621          1111  1 !
;  622          1112  2 begin
;  623          1113  2 bind
;  624          1114  2         hdrsubtyp = objrec [obj$b_subtyp] : byte;
;  625          1115  2
;  626          1116  2 if .currectyp eql obj$c_hdr                                  ! IF THIS RECORD IS A HEADER
;  627          1117  2 then if .hdrsubtyp eql obj$c_hdr_mhd                         ! AND IT IS THE MAIN MODULE HEADER
;  628          1118  2      then                                                    ! THEN WE HAVE VALID SEQUENCE
;  629          1119  3           if (.lastrectyp eql obj$c_eom)                     ! IF AND ONLY IF THE PREVIOUS WAS END OF MOD
;  630          1120  3           or (.lastrectyp eql obj$c_eomw)
;  631          1121  3           then  begin
;  632          1122  4                 mhdseen = true;
;  633          1123  4                 lnmseen = false;
;  634          1124  4                 return true                                ! IS THIS CASE SET MHD RECORD SEEN AND RETUR
;  635          1125  4                 end
;  636          1126  3           else  begin
;  637          1127  4                 signal  (lin$_seqnce, 2                    ! ELSE REPORT ERROR
;  638          1128  4                         ,obmodesc [omd$b_namlng]
;  639          1129  4                         ,lnk$gl_curfil [fdb$q_filename]
;  640          1130  4                         );
;  641          1131  4                 return false;
;  642          1132  4                 end
;  643          1133  2      else if .mhdseen                                        ! IF SOME OTHER KIND OF HEADER
;  644          1134  3           then  (if .hdrsubtyp eql obj$c_hdr_lnm            ! WE MUST HAVE SEEN A MAIN HEADER
;  645          1135  4                  then lnmseen = true;
;  646          1136  4                  return true
;  647          1137  3                 )
;  648          1138  3           else  begin
;  649          1139  4                 signal  (lin$_seqnce, 2                    ! ELSE REPORT ERROR
;  650          1140  4                         ,obmodesc [omd$b_namlng]
;  651          1141  4                         ,lnk$gl_curfil [fdb$q_filename]    ;
;  652          1142  4                         );
;  653          1143  4                 return false;
;  654          1144  4                 end                                        ;
;  655          1145  2 else if .mhdseen and .lnmseen
;  656          1146  3      then begin                                             ! IF WE HAVE SEEN A MAIN HEADER
;  657          1147  4           if (.currectyp eql obj$c_eom)                     ! THEN TURN OFF FLAG ON END OF MODULE.
;  658          1148  4           or (.currectyp eql obj$c_eomw)
;  659          1149  3           then  mhdseen = false                             ! SEQUENCE ERROR IF HAVE NOT SEEN
;  660          1150  3           else  if .currectyp eql obj$c_tir
;  661          1151  4                 then obmodesc [omd$v_nobin] = false;
;  662          1152  3
;  663          1153  3           return true;                                      ! MAIN HEADER AND THIS IS NOT ONE.
;  664          1154  3           end
;  665          1155  3      else begin
;  666          1156  3           signal (lin$_seqnce, 2                            ! ELSE REPORT ERROR
;  667          1157  3                  ,obmodesc [omd$b_namlng]
;  668          1158  3                  ,lnk$gl_curfil [fdb$q_filename]
;  669          1159  3                  );
;  670          1160  3           return false;
;  671          1161  2           end;
;  672          1162  1 end;
```

```
                                  0004 00000 SEQCHK: .WORD    Save R2                              ; 1106
           52 00000000'    EF   9E 00002          MOVAB    MHDSEEN, R2                          ;
 51        11  A2           01   C1 00009          ADDL3    #1, OBJREC, R1                       ; 1114
           50         03    A2   9A 0000E          MOVZBL   CURRECTYP, R0                        ; 1116
                            23   12 00012          BNEQ     3$                                   ;
                            61   95 00014          TSTB     (R1)                                 ; 1117
                            11   12 00016          BNEQ     2$                                   ;
           03         02    A2   91 00018          CMPB     LASTRECTYP, #3                       ; 1119
                            06   13 0001C          BEQL     1$                                   ;
           07         02    A2   91 0001E          CMPB     LASTRECTYP, #7                       ; 1120
                            35   12 00022          BNEQ     7$                                   ;
           62               01   B0 00024 1$:      MOVW     #1, MHDSEEN                          ; 1122
                            2C   11 00027          BRB      6$                                   ; 1124
           2D               62   E9 00029 2$:      BLBC     MHDSEEN, 7$                          ; 1133
           01               61   91 0002C          CMPB     (R1), #1                             ; 1134
                            24   12 0002F          BNEQ     6$                                   ;
       01  A2               01   90 00031          MOVB     #1, LNMSEEN                          ; 1135
                            1E   11 00035          BRB      6$                                   ; 1136
           1F               62   E9 00037 3$:      BLBC     MHDSEEN, 7$                          ; 1145
           1B         01    A2   E9 0003A          BLBC     LNMSEEN, 7$                          ;
           03               50   91 0003E          CMPB     R0, #3                               ; 1147
                            05   13 00041          BEQL     4$                                   ;
           07               50   91 00043          CMPB     R0, #7                               ; 1148
                            04   12 00046          BNEQ     5$                                   ;
                            62   94 00048 4$:      CLRB     MHDSEEN                              ; 1149
                            09   11 0004A          BRB      6$                                   ;
           02               50   91 0004C 5$:      CMPB     R0, #2                               ; 1150
                            04   12 0004F          BNEQ     6$                                   ;
       55  A2               02   8A 00051          BICB2    #2, OBMODESC+20                      ; 1151
           50               01   D0 00055 6$:      MOVL     #1, R0                               ; 1153
                                 04 00058          RET                                           ;
 7E 00000000G  00           14   C1 00059 7$:      ADDL3    #20, LNK$GL_CURFIL, -(SP)            ; 1158
                      69    A2   9F 00061          PUSHAB   OBMODESC+40                          ; 1157
                            02   DD 00064          PUSHL    #2                                   ; 1158
              00000000G     8F   DD 00066          PUSHL    #LIN$_SEQNCE                         ;
    00000000G  00           04   FB 0006C          CALLS    #4, LIB$SIGNAL                       ;
                            50   D4 00073          CLRL     R0                                   ; 1162
                                 04 00075          RET                                           ;

; Routine Size:  118 bytes,    Routine Base:  $CODE$ + 02E6

;  673          1163  1
```

```
;  675          1164  1 routine prohdr =
;  676          1165  1 !++
;  677          1166  1 !       PROCESS MODULE HEADER RECORDS AS FOLLOWS:
;  678          1167  1 !               (1) VALIDATE SEQUENCE
;  679          1168  1 !               (2) IGNORE ALL BUT MAIN MODULE HEADERS
;  680          1169  1 !               (3) VERIFY STRUCTURE LEVEL IS LESS THAN
;  681          1170  1 !                   OR EQUAL TO OBJ$C_STRLVL
;  682          1171  1 !               (4) VERIFY MAXIMUM RECORD LENGTH
;  683          1172  1 !                   PARAMETER IS LESS THAN OR EQUAL TO
;  684          1173  1 !                   OBJ$C_MAXRECSIZ
;  685          1174  1 !               (5) RECORD MAXIMUM RECORD LENGTH PARAMETER
;  686          1175  1 !                   FOR CHECKING SUBSEQUENT RECORDS
;  687          1176  1 !               (6) CHECK MODULE TITLE > 0 AND LESS THAN OR
;  688          1177  1 !                   EQUAL TO SYM$C_MAXLNG CHARACTERS
;  689          1178  1 !               (7) ALLOCATE AN OBJECT MODULE DESCRIPTOR
;  690          1179  1 !                   AND PUT ON END OF THE LIST (WHOSE
;  691          1180  1 !                   HEAD IS FILE DESCRIPTOR BLOCK).
;  692          1181  1 !               (8) COPY IN THE MODULE TITLE
;  693          1182  1 !               (9) INITIALIZE FLAGS AND P-SECTION COUNT IN
;  694          1183  1 !                       THE OBJECT MODULE DESCRIPTOR
;  695          1184  1 !--
;  696          1185  2 begin
;  697          1186  2 bind
;  698          1187  2         mhdrec = .objrec : block [, byte] ;
;  699          1188  2 local
;  700          1189  2         module_max_length ;
;  701          1190  2
;  702          1191  2 if not seqchk () then return false ;                      ! VALIDATE CORRECT SEQUENCE OF RECORD
;  703          1192  2
;  704          1193  2 if .objrec [obj$b_subtyp] neq obj$c_hdr_mhd then return true ;  ! IGNORE ALL HEADERS EXCEPT MAIN ONES
;  705          1194  2
;  706          1195  2 lnk$gw_nmodules = .lnk$gw_nmodules + 1 ;                  ! COUNT THIS MODULE
;  707          1196  2
;  708          1197  2 if  .mhdrec [mhd$b_strlvl] gtru obj$c_strlvl              ! COMPARE ITS OBJ FORMAT
;  709          1198  3 then begin                                               ! LEVEL AND IF BEYOND THIS AND GIVE UP
;  710          1199  3      signal (lin$_strlvl, 4
;  711          1200  3              ,.mhdrec [mhd$b_strlvl], obj$c_strlvl
;  712          1201  3              , mhdrec [mhd$b_namlng], lnk$gl_curfil [fdb$q_filename]
;  713          1202  3              ) ;
;  714          1203  3      return false ;
;  715          1204  2      end ;
;  716          1205  2
;  717          1206  2 if (maxreclng = .mhdrec [mhd$w_recsiz]) gtru obj$c_maxrecsiz  ! COMPARE MAX WITH MAX ALLOWED
;  718          1207  3 then begin                                               ! AND IF GREATER
;  719          1208  3      signal (lin$_illreclen, 3                           ! ISSUE ERROR MESSAGE
;  720          1209  3              ,.maxreclng, mhdrec [mhd$b_namlng]
;  721          1210  3              , lnk$gl_curfil [fdb$q_filename]
;  722          1211  3              ) ;
;  723          1212  3      return false ;
;  724          1213  2      end ;
;  725          1214  2
;  726          1215  2 module_max_length = lnk$k_max_filename_length ;          ! SETUP FOR ERROR REPORT
;  727          1216  2                                                          ! IN CASE mhd$b_namlng = 0
;  728          1217  2 if .mhdrec [mhd$b_namlng] eql 0                          ! CHECK MODULE NAME LENGTH
;  729          1218  3 or begin                                                 !  IS WITHIN LEGAL LIMITS...
;  730          1219  3    if   .lnk$gl_curclu [clu$v_shrimg]                    ! 1 to 39 for shareable images...
;  731          1220  3    then if .mhdrec [mhd$b_namlng] gtru lnk$k_max_filename_length
```

```
;  732          1221  4          then begin
;  733          1222  4               module_max_length = lnk$k_max_filenam _length ;
;  734          1223  4               true
;  735          1224  4               end
;  736          1225  3          else false
;  737          1226  3     else if .mhdrec [mhd$b_namlng] gtru sym$c_maxlng              ! 1 TO 31 FOR OBJECTS...
;  738          1227  4          then begin
;  739          1228  4               module_max_length = sym$c_maxlng ;
;  740          1229  4               true
;  741          1230  4               end
;  742          1231  3          else false
;  743          1232  3     end
;  744          1233  2 then
;  745          1234  3     begin                                                        ! AND IF NOT ISSUE ERROR
;  746          1235  3     signal ( lin$_illnamelen, 6, lnk$gt_modstring                ! MESSAGE AND QUIT
;  747          1236  3            , mhdrec [mhd$b_namlng], .mhdrec [mhd$b_namlng]
;  748          1237  3            ,.module_max_length, mhdrec [mhd$b_namlng]
;  749          1238  3            , lnk$gl_curfil [fdb$q_filename]
;  750          1239  3            ) ;
;  751          1240  3     return false ;
;  752          1241  2     end ;
;  753          1242  2
;  754          1243  2 obmodesc [omd$l_ownfdb]   = .lnk$gl_curfil ;                    ! SET POINTER TO OWNING FDB
;  755          1244  2 obmodesc [omd$l_modvbn]   = .lnk$al_rab [rab$l_rfa0] ;         ! THIS IS NOW THE LAST. CAPTURE THE
;  756          1245  2 obmodesc [omd$w_bytoff]   = .lnk$al_rab [rab$w_rfa4] ;         ! MODULE'S RFA FOR LATER
;  757          1246  2 obmodesc [omd$b_namlng]   = .mhdrec [mhd$b_namlng] ;           ! COPY THE STRING LENGTH
;  758          1247  2
;  759          1248  2 ch$copy (.mhdrec [mhd$b_namlng], mhdrec [mhd$t_name], 0       ! AND THE STRING ZERO FILLED
;  760          1249  2           ,sym$c_maxlng, obmodesc [omd$t_name]) ;               ! INTO THE DESCRIPTOR
;  761          1250  2
;  762          1251  2 obmodesc [omd$w_hipsct]   = -1 ;                              ! SET HIGHEST POSSIBLE P-SECTION NUMBER
;  763          1252  2 obmodesc [omd$w_h env]    = -1 ;                              ! SET HIGHEST POSSIBLE ENVIRONMENT NUMBER
;  764          1253  2 obmodesc [omd$b_flags]    = omd$m_nopsct or omd$m_nobin       ! FLAG NO P-SECTS AND NO BINARY YET
;  765          1254  2                                 or omd$m_noenv ;              !  AND NO ENVIRONMENTS DEFINED
;  766          1255  2 obmodesc [omd$v_shrimg]   = .lnk$gl_curfil [fdb$v_shr] ;     ! EXTRACT THE SHAREABLE IMAGE ATTRIB
;  767          1256  2                                                              !  FROM FILE DESCRIPTOR
;  768          1257  2 obmodesc [omd$v_selser]   = .lnk$gl_curfil [fdb$v_selser] ;  ! AS WELL AS THE SELECTIVE SEARCH ATTRIBUTE
;  769          1258  2 obmodesc [omd$v_debuger]  = .lnk$gl_curfil [fdb$v_debuger] ; ! COPY DEBUGGER MODULE FLAG
;  770          1259  2 obmodesc [omd$l_omdnum]   = .lnk$gw_nmodules ;                ! SET MODULE NUMBER INTO DESCRIPTOR
;  771          1260  2
;  772          1261  2 if not .lnk$gl_ctlmsk [lnk$v_intfil]                        ! IF THIS IS NOT STARLET
;  773          1262  2 then obmodesc [omd$v_mapmod] = true ;                       ! SET TO MAP THE MODULE ON PASS 2
;  774          1263  2
;  775          1264  2
;  776          1265  3     begin
;  777          1266  3     bind modidstring = mhdrec [mhd$t_name] +                   ! POINT TO THE MODULE ID IN HEADER
;  778          1267  3                        .mhdrec [mhd$b_namlng] : vector [, byte], ! RECORD (COUNTED STRING)
;  779          1268  3          modcredat     = modidstring [0] +                     !  POINT TO THE MODULE CREATION DATE/TIME
;  780          1269  3                        .modidstring [0] + 1   : vector [, byte] ;
;  781          1270  3
;  782          1271  3     local bincredat : vector [2, long] ;
;  783          1272  3
;  784          1273  3     if  .lnk$gl_curclu [clu$v_shrimg]                          ! IF THIS IS A SHAREABLE IMAGE THAT
;  785          1274  3           and                                                  ! WAS PICKED UP FROM A
;  786          1275  3         .(lnk$gl_curclu [clu$q_credat])<0, 32, 0> neq 0       ! SHAREABLE IMAGE LIBRARY
;  787          1276  4     then begin                                               !  (CREDAT WILL BE 0 IF NOT)
;  788          1277  4          lnk$bintim (modcredat, bincredat) ;                 ! CONVERT TIME TO BINARY
```

```
;  789          1278  4
;  790          1279  4          if  not .(lnk$gl_curclu [clu$l_gsmatch])<31, 1 ,0>      ! IF HIGH BIT CLEAR THEN DO DATE CHECK
;  791          1280  4          then if not ch$eql (8, bincredat, 8, lnk$gl_curclu [clu$q_credat])      ! IF THE DATES ARE DIFFERENT
;  792          1281  4               then signal (lin$_datmismch, 4, bincredat                          ! THEN WARN THE USER,
;  793          1282  4                           ,lnk$gl_curfil [fdb$q_filename]                       ! BUT CONTINUE
;  794          1283  4                           ,lnk$gl_curclu [clu$q_credat]
;  795          1284  4                           ,lnk$gl_curfil [fdb$q_libnamdsc]
;  796          1285  4                           ) ;
;  797          1286  4          ch$fill (0, 8, lnk$gl_curclu [clu$q_credat]) ;          ! ZERO QUADWORD DATE/TIME, SINCE IT
;  798          1287  3          end ;                                                   ! GETS REUSED
;  799          1288  3
;  800          1289  3     if not .obmodesc [omd$v_debuger]                             ! IF THIS IS NOT A DEBUGGER MODULE
;  801          1290  3         and
;  802          1291  4         (.lnk$gt_imgid [0] eql 0   or   .lnk$gl_tfrpsc eql 0)    ! IF THERE IS NO IMAGE IDENT SET UP
;  803          1292  4     then begin
;  804          1293  4          imageidstring [0] = minu (.modidstring [0]              ! EXTRACT THE LENGTH MINIMIZED WITH
;  805          1294  4                                  , sym$c_maxlng) ;               ! MAXIMUM ALLOWED STRING LENGTH
;  806          1295  4          if .imageidstring [0]  neq 0                           ! AND IF NOT NULL
;  807          1296  4          then  ch$move  (.imageidstring [0]                             ! SAVE IT TILL END OF MODULE SO THAT
;  808          1297  4                        , modidstring [1]                               ! MAY USE IT IF THIS HAS A TRANSFER
;  809          1298  4                        , imageidstring [1]                             ! ADDRESS
;  810          1299  4                        ) ;
;  811          1300  3          end ;
;  812          1301  2     end ;
;  813          1302  2 return true ;
;  814          1303  1 end ;                                                          ! OF MODULE HEADER PROCESSING
```

```
                                    0FFC 00000 PROHDR:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 1164
                5B 00000000G   00    9E 00002          MOVAB   LIB$SIGNAL, R11
                5A 00000000G   00    9E 00009          MOVAB   LNK$GL_CURFIL, R10
                59 00000000'   EF    9E 00010          MOVAB   OBMODESC+20, R9
                5E                   08 C2 00017       SUBL2   #8, SP
                56          BC       A9 D0 0001A       MOVL    OBJREC, R6                              ; 1187
          FF67  CF                    00 FB 0001E       CALLS   #0, SEQCHK                              ; 1191
                56                    50 E9 00023       BLBC    R0, 3$
                50          BC       A9 D0 00026       MOVL    OBJREC, R0                              ; 1193
                            01       A0 95 0002A       TSTB    1(R0)
                                     03 13 0002D       BEQL    1$
                                  0197 31 0002F       BRW     14$
                   00000000'     EF    B6 00032 1$:    INCW    LNK$GW_NMODULES                         ; 1195
                            02       A6 95 00038       TSTB    2(R6)                                   ; 1197
                                     1A 13 0003B       BEQL    2$
          7E        6A                 14 C1 0003D       ADDL3   #20, LNK$GL_CURFIL, -(SP)               ; 1201
                            05       A6 9F 00041       PUSHAB  5(R6)
                                     7E D4 00044       CLRL    -(SP)
                7E          02       A6 9A 00046       MOVZBL  2(R6), -(SP)
                                     04 DD 0004A       PUSHL   #4
                   00000000G     8F    DD 0004C       PUSHL   #LIN$_STRLVL
                6B                   06 FB 00052       CALLS   #6, LIB$SIGNAL
                                     77 11 00055       BRB     7$                                      ; 1203
                50          03       A6 3C 00057 2$:   MOVZWL  3(R6), R0                               ; 1206
             B4 A9                   50 B0 0005B       MOVW    R0, MAXRECLNG
           0800 8F                   50 B1 0005F       CMPW    R0, #2048
```

```
                                    18  1B 00064         BLEQU   4$                                     ;
           7E           6A          14  C1 00066         ADDL3   #20, LNK$GL_CURFIL, -(SP)             ; 1210
                             05     A6  9F 0006A         PUSHAB  5(R6)                                  ; 1209
                        7E   B4     A9  3C 0006D         MOVZWL  MAXRECLNG, -(SP)                       ; 1210
                                    03  DD 00071         PUSHL   #3                                     ;
                          00000000G 8F  DD 00073         PUSHL   #LIN$_ILLRECLEN                        ;
                        6B          05  FB 00079         CALLS   #5, LIB$SIGNAL                         ;
                                    50  11 0007C  3$:    BRB     7$                                     ; 1212
                        52 00000000G 8F D0 0007E  4$:    MOVL    #LNK$K_MAX_FILENAME_LENGTH, -          ; 1215
                                                                 MODULE_MAX_LENGTH                      ;
                        51   05     A6  9A 00085         MOVZBL  5(R6), R1                              ; 1217
                                    26  13 00089         BEQL    6$                                     ;
                        50 00000000G 00 D0 0008B         MOVL    LNK$GL_CURCLU, R0                      ; 1219
           12      58   A0          02  E1 00092         BBC     #2, 88(R0), 5$                         ;
              00000000G 8F          51  D1 00097         CMPL    R1, #LNK$K_MAX_FILENAME_LENGTH         ; 1220
                                    31  1B 0009E         BLEQU   8$                                     ;
                        52 00000000G 8F D0 000A0         MOVL    #LNK$K_MAX_FILENAME_LENGTH, -          ; 1222
                                                                 MODULE_MAX_LENGTH                      ;
                                    08  11 000A7         BRB     6$                                     ;
                        1F          51  91 000A9  5$:    CMPB    R1, #31                                ; 1226
                                    23  1B 000AC         BLEQU   8$                                     ;
                        52          1F  D0 000AE         MOVL    #31, MODULE_MAX_LENGTH                 ; 1228
           7E           6A          14  C1 000B1  6$:    ADDL3   #20, LNK$GL_CURFIL, -(SP)             ; 1238
                             05     A6  9F 000B5         PUSHAB  5(R6)                                  ; 1237
                                    06  BB 000B8         PUSHR   #^M<R1,R2>                             ; 1238
                             05     A6  9F 000BA         PUSHAB  5(R6)                                  ; 1236
                          00000000' EF  9F 000BD         PUSHAB  LNK$GT_MODSTRING                       ; 1235
                                    06  DD 000C3         PUSHL   #6                                     ; 1238
                          00000000G 8F  DD 000C5         PUSHL   #LIN$_ILLNAMELEN                       ;
                        6B          08  FB 000CB         CALLS   #8, LIB$SIGNAL                         ;
                                  00FC  31 000CE  7$:    BRW     15$                                    ; 1240
                        57          6A  D0 000D1  8$:    MOVL    LNK$GL_CURFIL, R7                      ; 1243
                   F0   A9          57  D0 000D4         MOVL    R7, OBMODESC+4                         ;
                   F8   A9 00000000G 00 D0 000D8         MOVL    LNK$AL_RAB+16, OBMODESC+12             ; 1244
                   FC   A9 00000000G 00 B0 000E0         MOVW    LNK$AL_RAB+20, OBMODESC+16             ; 1245
                   58        05     A6  9A 000E8         MOVZBL  5(R6), R8                              ; 1246
                   14   A9          58  90 000EC         MOVB    R8, OBMODESC+40                        ;
 1F        00      06   A6          58  2C 000F0         MOVC5   R8, 6(R6), #0, #31, OBMODESC+41        ; 1249
                             15     A9     000F6                                                        ;
                   FE   A9          01  AE 000F8         MNEGW   #1, OBMODESC+18                        ; 1251
                   02   A9          01  AE 000FC         MNEGW   #1, OBMODESC+22                        ; 1252
                   69        83     8F  90 00100         MOVB    #-125, OBMODESC+20                     ; 1254
 50     0A    A7        01          02  EF 00104         EXTZV   #2, #1, 10(R7), R0                     ; 1255
 69           01        02          50  F0 0010A         INSV    R0, #2, #1, OBMODESC+20                ;
 50     0A    A7        01          03  EF 0010F         EXTZV   #3, #1, 10(R7), R0                     ; 1257
 69           01        03          50  F0 00115         INSV    R0, #3, #1, OBMODESC+20                ;
 50     0A    A7        01          05  EF 0011A         EXTZV   #5, #1, 10(R7), R0                     ; 1258
 69           01        05          50  F0 00120         INSV    R0, #5, #1, OBMODESC+20                ;
                   08   A9 00000000' EF 3C 00125         MOVZWL  LNK$GW_NMODULES, OBMODESC+28           ; 1259
           03 00000000G 00          03  E0 0012D         BBS     #3, LNK$GL_CTLMSK+1, 9$                ; 1261
                        69          10  88 00135         BISB2   #16, OBMODESC+20                       ; 1262
                        56   06 A846    9E 00138  9$:    MOVAB   6(R8)[R6], R6                          ; 1266
                        57          66  9A 0013D         MOVZBL  (R6), R7                               ; 1269
                        50 00000000G 00 D0 00140         MOVL    LNK$GL_CURCLU, R0                      ; 1273
           4E      58   A0          02  E1 00147         BBC     #2, 88(R0), 11$                        ;
                             30     A0  D5 0014C         TSTL    48(R0)                                 ; 1275
                                    49  13 0014F         BEQL    11$                                    ;
```

```
                                                5E   DD 00151          PUSHL   SP                                          ; 1277
                                              01 A746  9F 00153          PUSHAB  1(R7)[R6]
                                 00000000G    00       02  FB 00157     CALLS   #2, LNK$BINTIM
                                              54 00000000G  00  D0 0015E     MOVL    LNK$GL_CURCLU, R4                      ; 1279
                                                   0087     C4  95 00165     TSTB    135(R4)
                                                            21  19 00169     BLSS    10$
                                        30    A4            6E      08  29 0016B     CMPC3   #8, BINCREDAT, 48(R4)          ; 1280
                                                            1A  13 00170     BEQL    10$
                                              50            6A  D0 00172     MOVL    LNK$GL_CURFIL, R0                      ; 1284
                                                      1C    A0  9F 00175     PUSHAB  28(R0)
                                                      30    A4  9F 00178     PUSHAB  48(R4)                                 ; 1283
                                                      14    A0  9F 0017B     PUSHAB  20(R0)                                 ; 1282
                                                      0C    AE  9F 0017E     PUSHAB  BINCREDAT                              ; 1281
                                                            04  DD 00181     PUSHL   #4                                     ; 1284
                                                 00000000G  8F  DD 00183     PUSHL   #LIN$_DATMISMCH
                                              6B            06  FB 00189     CALLS   #6, LIB$SIGNAL
                                              50 00000000G  00  D0 0018C 10$:  MOVL    LNK$GL_CURCLU, R0                      ; 1286
                                  08          00            6E  00  2C 00193     MOVC5   #0, (SP), #0, #8, 48(R0)
                                                      30    A0      00198
                                              2B            69  05  E0 0019A 11$:  BBS     #5, OBMODESC+20, 14$               ; 1289
                                                 00000000G  00  95  0019E     TSTB    LNK$GT_IMGID                           ; 1291
                                                            08  13  001A4     BEQL    12$
                                                 00000000'  EF  D5  001A6     TSTL    LNK$GL_TFRPSC
                                                            1B  12  001AC     BNEQ    14$
                                              50            57  D0  001AE 12$:  MOVL    R7, R0                                 ; 1293
                                              1F            50  91  001B1     CMPB    R0, #31
                                                            03  1B  001B4     BLEQU   13$
                                              50            1F  D0  001B6     MOVL    #31, R0
                                        CC    A9            50  90  001B9 13$:  MOVB    R0, IMAGEIDSTRING
                                              50      CC    A9  9A  001BD     MOVZBL  IMAGEIDSTRING, R0                      ; 1295
                                                            06  13  001C1     BEQL    14$
                              CD    A9        01    A6      50  28  001C3     MOVC3   R0, 1(R6), IMAGEIDSTRING+1             ; 1298
                                              50            01  D0  001C9 14$:  MOVL    #1, R0                                 ; 1302
                                                            04  001CC     RET
                                                            50  D4  001CD 15$:  CLRL    R0                                     ; 1303
                                                            04  001CF     RET

; Routine Size:  464 bytes,    Routine Base:  $CODE$ + 035C

;  815          1304  1
```

```
;  817          1305  1 routine delete_psect (psectname, retdescadr) =
;  818          1306  2     begin
;  819          1307  2 !
;  820          1308  2 ! THIS ROUTINE SEARCHES ALL CLUSTERS FOR ANOTHER DEFINITION OF THE
;  821          1309  2 ! NAMED GLOBAL PSECT.  IF ONE IS FOUND, IT IS DELETED (BY SETTING
;  822          1310  2 ! A FLAG IN THE DESCRIPTOR).  IF THE CLUSTER IT IS FOUND IN IS
;  823          1311  2 ! A SHAREABLE IMAGE, THEN AN ERROR MESSAGE IS ISSUED, SINCE A
;  824          1312  2 ! PSECT CANNOT BE CONTAINED IN TWO SHAREABLE IMAGES AT THE SAME
;  825          1313  2 ! TIME.
;  826          1314  2 !
;  827          1315  2     map
;  828          1316  2         psectname : ref vector [, byte],
;  829          1317  2         retdescadr : ref vector [, long];
;  830          1318  2     local
;  831          1319  2         cludesc : ref block [, byte],
;  832          1320  2         fdb : ref block [, byte],
;  833          1321  2         pdesc : ref block [, byte];
;  834          1322  2
;  835          1323  2     cludesc = lnk$gl_clulst;
;  836          1324  2 !
;  837          1325  2 ! LOOP OVER ALL CLUSTERS, SEARCHING THE GLOBAL PSECT LIST FOR
;  838          1326  2 ! A MATCH
;  839          1327  2 !
;  840          1328  2
;  841          1329  2     while (cludesc = .cludesc [clu$l_nxtclu]) neq 0 do
;  842          1330  2
;  843          1331  3         if lib$lookup_tree (cludesc [clu$l_gpslst], .psectname, lnk$compare_pscnod, pdesc) and not (pdesc =
;  844          1332  3             .pdesc [node$l_ptr];                ! GET PSECT DESCRIPTOR ADDRESS
;  845          1333  3             .pdesc [psc$v_deleted])             ! TEST IF PSECT HAS BEEN DELETED
;  846          1334  2         then
;  847          1335  3             begin
;  848          1336  3             pdesc [psc$v_deleted] = true;       ! FLAG PSECT DELETED FROM THIS CLUSTER
;  849          1337  3
;  850          1338  3             if .cludesc [clu$v_shrimg]          ! IF CLUSTER IS A SHAREABLE IMAGE
;  851          1339  3             then
;  852          1340  4                 begin
;  853          1341  4                 fdb = .cludesc [clu$l_fstfdb];  ! GET FDB ADDRESS FOR SHAREABLE IMAGE
;  854          1342  4                 signal (lin$_mulshrpsc, 3, .psectname,  ! ISSUE ERROR MESSAGE
;  855          1343  4                     fdb [fdb$q_filename], lnk$gl_curfil [fdb$q_filename], lin$_noimgfil);
;  856          1344  4                 lnk$gl_ctlmsk [lnk$v_image] = false;    ! IMAGE WOULD ONLY BE INVALID
;  857          1345  3                 end;
;  858          1346  3
;  859          1347  3             retdescadr [0] = .pdesc;            ! RETURN ADDRESS TO CALLER
;  860          1348  3             return true;
;  861          1349  2             end;
;  862          1350  2
;  863          1351  2     return false
;  864          1352  1     end;


                                001C 00000 DELETE_PSECT:
                                           .WORD   Save R2,R3,R4                                    ; 1305
                5E                 04  C2 00002             SUBL2   #4, SP
                52 00000000G       00  9E 00005             MOVAB   LNK$GL_CLULST, CLUDESC           ; 1323
```

```
                                52              62  D0 0000C 1$:    MOVL    (CLUDESC), CLUDESC                      ; 1329
                                                6A  13 0000F        BEQL    3$                                      ;
                                                5E  DD 00011        PUSHL   SP                                      ; 1331
                                    000000000G  00  9F 00013        PUSHAB  LNK$COMPARE_PSCNOD
                                            04  AC  DD 00019        PUSHL   PSECTNAME
                                            14  A2  9F 0001C        PUSHAB  20(CLUDESC)
                            00000000G   00      04  FB 0001F        CALLS   #4, LIB$LOOKUP_TREE
                                        E3      50  E9 00026        BLBC    R0, 1$
                                        50      6E  D0 00029        MOVL    PDESC, R0                               ; 1332
                                        6E  0A  A0  D0 0002C        MOVL    10(R0), PDESC
                                        50      6E  D0 00030        MOVL    PDESC, R0                               ; 1333
                        D4      0B      A0      06  E0 00033        BBS     #6, 11(R0), 1$
                                        53      6E  D0 00038        MOVL    PDESC, R3                               ; 1336
                                0B      A3  40  8F  88 0003B        BISB2   #64, 11(R3)
                        2E      58      A2      02  E1 00040        BBC     #2, 88(CLUDESC), 2$                     ; 1338
                                        54  08  A2  D0 00045        MOVL    8(CLUDESC), FDB                         ; 1341
                                    00000000G   8F  DD 00049        PUSHL   #LIN$_NOIMGFIL                          ; 1343
                    7E  00000000G   00          14  C1 0004F        ADDL3   #20, LNK$GL_CURFIL, -(SP)
                                            14  A4  9F 00057        PUSHAB  20(FDB)
                                            04  AC  DD 0005A        PUSHL   PSECTNAME
                                                03  DD 0005D        PUSHL   #3
                                    00000000G   8F  DD 0005F        PUSHL   #LIN$_MULSHRPSC
                            00000000G   00          06  FB 00065    CALLS   #6, LIB$SIGNAL
                            00000000G   00          01  8A 0006C    BICB2   #1, LNK$GL_CTLMSK                       ; 1344
                                    08  BC          53  D0 00073 2$: MOVL    R3, @RETDESCADR                         ; 1347
                                        50          01  D0 00077    MOVL    #1, R0                                  ; 1348
                                                        04 0007A    RET                                             ;
                                                    50  D4 0007B 3$: CLRL    R0                                      ; 1351
                                                        04 0007D    RET                                             ; 1352

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 052C
```

```
;  866          1353  1 routine fndpscmapent (psctnum, omdptr) =
;  867          1354  2     begin
;  868          1355  2 !
;  869          1356  2 !       THIS ROUTINE RETURNS THE ADDRESS OF THE MAPPING
;  870          1357  2 !       TABLE ENTRY FOR P-SECTION NUMBER "PSCTNUM" IN
;  871          1358  2 !       THE CURRENT MODULE.
;  872          1359  2 !       THE P-SECT MAPPING TABLE IS AN ARRAY APPENDED
;  873          1360  2 !       TO THE MODULE DESCRIPTOR BLOCK. IT IS INITIALLY
;  874          1361  2 !       ALLOCATED WITH SPACE SUFFICIENT FOR 256
;  875          1362  2 !       ENTRIES.  IF MORE THAN 256 PSECTS ARE ENCOUNTERED DURING
;  876          1363  2 !       THE PROCESSING OF THE OBJECT MODULE, THE INITIAL TABLE IS
;  877          1364  2 !       COPIED OUT TO ANOTHER BLOCK, AND THE MAP TABLE APPENDED TO
;  878          1365  2 !       THE OBJECT MODULE DESCRIPTOR BLOCK IS A TABLE OF TABLE ADDRESSES.
;  879          1366  2 !
;  880          1367  2 !       THE SECOND ARGUMENT IS OPTIONAL.  IF NOT SUPPLIED, THE OWN OBJ MODULE
;  881          1368  2 !       DESCRIPTOR (OBMODESC) WILL BE USED.
;  882          1369  2     literal
;  883          1370  2         tablesize = 256*pmt$c_size;             ! SIZE IN BYTES OF A BLOCK
;  884          1371  2     local
;  885          1372  2         blockoff,                               ! OFFSET OF ENTRY IN EXTENDED BLOCK
;  886          1373  2         first_entry : ref block[, byte],
;  887          1374  2         omdesc : ref block [, byte],            ! POINTER TO OBJ MODULE DESCRIPTOR
;  888          1375  2         mapent : ref block [, byte],            ! ADDRESS IN PSECT MAPPING TABLE
;  889          1376  2         first256 : ref block [,byte],           ! address of block of first 256
;  890          1377  2         mapoff : ref block [, byte];            ! ADDRESS OF EXTENDED MAPPING BLOCK
;  891          1378  2     builtin
;  892          1379  2         nullparameter;
;  893          1380  2
;  894          1381  2     if nullparameter (2)                        ! SETUP POINTER TO OBJ MODULE DESCRIPTOR
;  895          1382  2     then
;  896          1383  2         omdesc = obmodesc
;  897          1384  2     else
;  898          1385  2         omdesc = .omdptr;
;  899          1386  2
;  900          1387  2     mapent = omdesc [omd$t_pscmap] + .psctnum*pmt$c_size;       ! GET ENTRY ADDRESS IN BASE MAPPING TABLE
;  901          1388  2
;  902          1389  2     if not .omdesc [omd$v_p256]                 ! IF NOT INTO EXTENDED MAP TABLE
;  903          1390  2         and .psctnum lequ 255                   !  AND LEQU 255 PSECTS
;  904          1391  2     then
;  905          1392  2         return .mapent
;  906          1393  2 !
;  907          1394  2 ! EXTENDED PSECT ENTRY
;  908          1395  2 !
;  909          1396  2     else
;  910          1397  3         begin
;  911          1398  3         blockoff = (.psctnum mod 256)*pmt$c_size;       ! OFFSET OF ENTRY WITHIN EXTENDED TABLE
;  912          1399  3         mapent = omdesc [omd$t_pscmap] + (.psctnum/256)*pmt$c_size;
;  913          1400  3                                                 ! GET ADDRESS OF EXTENDED TABLE ADDRESS
;  914          1401  3
;  915          1402  3         if ..mapent neq 0                       ! IF EXTENDED TABLE PRESENT
;  916          1403  3             and .omdesc [omd$v_p256]            !  AND THERE REALLY IS AN EXTENDED TABLE
;  917          1404  3         then
;  918          1405  3             return ..mapent + .blockoff         !  THEN RETURN ENTRY ADDRESS
;  919          1406  3         else
;  920          1407  4             begin
;  921          1408  4             lnk$alloblk (tablesize, mapoff);    ! ALLOCATE EXTENDED TABLE BLOCK
;  922          1409  4
```

```
;  923          1410  4              if not .omdesc [omd$v_p256]         ! IF THIS IS THE FIRST ONE
;  924          1411  4              then
;  925          1412  5                  begin
;  926          1413  5                  first_entry = omdesc[omd$t_pscmap];
;  927          1414  5                  lnk$alloblk(tablesize,first256);                    ! first, allocate space for first 25
;  928          1415  5                  ch$move(tablesize,omdesc[omd$t_pscmap],.first256);  ! move data for first 256
;  929          1416  5                  ch$fill (0, tablesize, omdesc [omd$t_pscmap]);      ! ZERO BASE TABLE
;  930          1417  5                  first_entry[pmt$l_secpmt] = .first256;
;  931          1418  5                  omdesc [omd$v_p256] = true;                         ! FLAG INTO EXTENDED TABLE
;  932          1419  4                  end;
;  933          1420  4
;  934          1421  4              ch$fill (0, tablesize, .mapoff);    !  ZERO NEW BLOCK
;  935          1422  4
;  936          1423  4              !
;  937          1424  4              ! POINT ENTRY IN BASE TABLE TO NEW BLOCK
;  938          1425  4              !
;  939          1426  4              mapent [pmt$l_secpmt] = .mapoff;
;  940          1427  4              return .mapoff + .blockoff
;  941          1428  3              end;
;  942          1429  2          end;
;  943          1430  1      end;


                                07FC 00000 FNDPSCMAPENT:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 1353
5A 00000000G   00  9E 00002                MOVAB   LNK$ALLOBLK, R10                          ;
5E             08  C2 00009                SUBL2   #8, SP                                    ;
02             6C  91 0000C                CMPB    (AP), #2                                  ; 1381
               05  1F 0000F                BLSSU   1$                                        ;
         08    AC  D5 00011                TSTL    8(AP)                                     ;
               09  12 00014                BNEQ    2$                                        ;
57 00000000'   EF  9E 00016 1$:            MOVAB   OBMODESC, OMDESC                          ; 1383
               04  11 0001D                BRB     3$                                        ;
57       08    AC  D0 0001F 2$:           MOVL    OMDPTR, OMDESC                            ; 1385
50       04    AC  D0 00023 3$:           MOVL    PSCTNUM, R0                               ; 1387
56       48 A740  7E 00027                MOVAQ   72(OMDESC)[R0], MAPENT                    ;
   0D       14   A7  06 0002C                BBS     #6, 20(OMDESC), 4$                        ; 1389
      000000FF  8F  50 00031                CMPL    R0, #255                                  ; 1390
               04  1A 00038                BGTRU   4$                                        ;
50             56  D0 0003A                MOVL    MAPENT, R0                                ; 1397
               04  0003D                   RET                                               ;
7E 00          50  01 7A 0003E 4$:        EMUL    #1, R0, #0, -(SP)                         ; 1398
51 51          8E 00000100  8F 7B 00043   EDIV    #256, (SP)+, R1, R1                       ;
59             51  03 78 0004C             ASHL    #3, R1, BLOCKOFF                          ;
               50 00000100  8F C6 00050    DIVL2   #256, R0                                  ; 1399
               56       48 A740 7E 00057   MOVAQ   72(OMDESC)[R0], MAPENT                    ;
                           66 D5 0005C     TSTL    (MAPENT)                                  ; 1402
                           0A 13 0005E     BEQL    5$                                        ;
   05       14   A7  06 E1 00060          BBC     #6, 20(OMDESC), 5$                        ; 1403
50             66  59 C1 00065             ADDL3   BLOCKOFF, (MAPENT), R0                    ; 1405
                   04 00069                RET                                               ; 1407
                   5E DD 0006A 5$:         PUSHL   SP                                        ; 1408
7E       0800      8F 3C 0006C             MOVZWL  #2048, -(SP)                              ;
6A               02 FB 00071               CALLS   #2, LNK$ALLOBLK                           ;
```

```
                        29      14  A7              06  E0 00074          BBS     #6, 20(OMDESC), 6$                      ; 1410
                                    58      48      A7  9E 00079          MOVAB   72(OMDESC), FIRST_ENTRY                 ; 1413
                                            04      AE  9F 0007D          PUSHAB  FIRST256                                ; 1414
                                    7E      0800    8F  3C 00080          MOVZWL  #2048, -(SP)                            ;
                                    6A              02  FB 00085          CALLS   #2, LNK$ALLOBLK                         ;
                    04  BE      48  A7      0800    8F  28 00088          MOVC3   #2048, 72(OMDESC), @FIRST256            ; 1415
    0800    8F              00      6E              00  2C 00090          MOVC5   #0, (SP), #0, #2048, 72(OMDESC)         ; 1416
                                            48      A7      00097          ;                                               ;
                                    68      04      AE  D0 00099          MOVL    FIRST256, (FIRST_ENTRY)                 ; 1417
                                14  A7      40      8F  88 0009D          BISB2   #64, 20(OMDESC)                         ; 1418
    0800    8F              00      6E              00  2C 000A2  6$:      MOVC5   #0, (SP), #0, #2048, @MAPOFF            ; 1421
                                            00      BE      000A9          ;                                               ;
                                    66              6E  D0 000AB          MOVL    MAPOFF, (MAPENT)                        ; 1426
                        50              6E          59  C1 000AE          ADDL3   BLOCKOFF, MAPOFF, R0                    ; 1427
                                                        04 000B2          RET                                             ; 1430

; Routine Size:  179 bytes,    Routine Base:  $CODE$ + 05AA
```

```
;  945          1431  1 routine alloc_omdnode (omdnum, retnodeadr, omdesc) =
;  946          1432  1 !
;  947          1433  1 ! THIS ROUTINE IS CALLED BY LIB$INSERT_TREE TO ALLOCATE
;  948          1434  1 ! STORAGE FOR A NODE
;  949          1435  1 !
;  950          1436  2 begin
;  951          1437  2 map
;  952          1438  2         retnodeadr      : ref vector [, long],
;  953          1439  2         omdesc          : ref block [, byte];
;  954          1440  2 local
;  955          1441  2         node            : ref block [, byte];
;  956          1442  2
;  957          1443  2 lnk$alloblk (node$c_long, node);                    ! ALLOCATE A NODE
;  958          1444  2 node [node$l_ptr] = .omdesc;                        ! POINT TO OBJ MODULE DESCRIPTOR
;  959          1445  2 retnodeadr [0]    = .node;                          ! RETURN ADDRESS TO CALLER
;  960          1446  2 return true
;  961          1447  1 end;

                                    0000 00000 ALLOC_OMDNODE:
                                                       .WORD   Save nothing                    ; 1431
                          5E                 04  C2 00002     SUBL2   #4, SP                  ;
                                             5E  DD 00005     PUSHL   SP                      ; 1443
                                             0E  DD 00007     PUSHL   #14                     ;
                00000000G   00               02  FB 00009     CALLS   #2, LNK$ALLOBLK         ;
                          50                 6E  D0 00010     MOVL    NODE, R0                ; 1444
                       0A   A0          0C   AC  D0 00013     MOVL    OMDESC, 10(R0)          ;
                       08   BC               50  D0 00018     MOVL    R0, @RETNODEADR         ; 1445
                          50                 01  D0 0001C     MOVL    #1, R0                  ; 1446
                                                 04 0001F     RET                             ; 1447

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 065D

;  962          1448  1
```

```
;  964          1449  1 global routine lnk$compare_omd (omdnum, curnode, omdesc) =
;  965          1450  1 !
;  966          1451  1 ! THIS ROUTINE IS CALLED BY LIB$INSERT_TREE AND LIB$LOOKUP_TREE
;  967          1452  1 ! TO COMPARE CURRENT NODE WITH THE NUMBER WE ARE LOOKING FOR
;  968          1453  1 !
;  969          1454  2 begin
;  970          1455  2 map
;  971          1456  2         curnode : ref block [, byte];
;  972          1457  2 bind
;  973          1458  2         curomd = .curnode [node$l_ptr] : block [, byte];
;  974          1459  2
;  975          1460  2 if .omdnum lss .curomd [omd$l_omdnum]                  ! COMPARE THIS WITH CURRENT NODE
;  976          1461  2 then return -1                                       ! RETURN -1 IF LSS
;  977          1462  2 else if .omdnum eql .curomd [omd$l_omdnum]            ! IF EQUAL
;  978          1463  2      then return 0                                   !  THEN RETURN 0
;  979          1464  2      else return 1                                   !  ELSE IT'S GTR, RETURN 1
;  980          1465  1 end;


                                        0000 00000          .ENTRY  LNK$COMPARE_OMD, Save nothing                   ; 1449
                              50    08  AC  D0 00002          MOVL    CURNODE, R0                                     ; 1458
                              50    0A  A0  D0 00006          MOVL    10(R0), R0                                      ;
                        1C    A0    04  AC  D1 0000A          CMPL    OMDNUM, 28(R0)                                  ; 1460
                                        04  18 0000F          BGEQ    1$                                              ;
                              50        01  CE 00011          MNEGL   #1, R0                                          ; 1462
                                        04  00014          RET                                                     ;
                                        03  12 00015 1$:     BNEQ    2$                                              ;
                                        50  D4 00017          CLRL    R0                                              ; 1463
                                        04  00019          RET                                                     ;
                                  50        01  D0 0001A 2$:     MOVL    #1, R0                                      ; 1464
                                        04  0001D          RET                                                     ; 1465

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 067D

;  981          1466  1
```

```
;  983          1467  1 routine propsectdef =
;  984          1468  2     begin
;  985          1469  2 !++
;  986          1470  2 !       PROCESS P-SECTION DEFINITIONS AS FOLLOWS:
;  987          1471  2 !               (0) CHECK LEGAL P-SECTION NAME AND ALIGNMENT PARAMETER
;  988          1472  2 !               (1) SEARCH FOR P-SECTION DEFINED
;  989          1473  2 !               (2) IF DEFINED:
;  990          1474  2 !                       1. CHECK COMPATIBLE ATTRIBUTES
;  991          1475  2 !                       2. IF OVERLAYED, CHECK EQUAL ALIGNMENT
;  992          1476  2 !                       3. MAXIMIZE P-SECTIONS'S BASE ALIGNMENT
;  993          1477  2 !                       4. GO TO (3)
;  994          1478  2 !                   IF NOT DEFINED:
;  995          1479  2 !                       1. INSERT IN P-SECTION LIST
;  996          1480  2 !                       2. COPY FLAGS AND ALIGNMENT
;  997          1481  2 !               (3) ALLOCATE A MODULE CONTRIBUTION ENTRY AND INSERT
;  998          1482  2 !                       IT LEXICALLY IN THE LIST BY MODULE NAME
;  999          1483  2 !               (4) UPDATE CURRENT BASE BY THIS ALIGNMENT THEN COPY
; 1000          1484  2 !                       CURRENT BASE INTO MAPPING TABLE ENTRY
; 1001          1485  2 !               (5) ADD THAT ROUND UP TO PSECTIONS ACCUMULATED LENGTH
; 1002          1486  2 !               (6) ADD P-SECTIONS LENGTH CONTRIBUTION TO THE ACCUMULATION
; 1003          1487  2 !                       IF CONCATENATED. IF OVERLAYED, MAXIMIZE LENGTH.
; 1004          1488  2 !               (7) COPY CONTRIBUTION LENGTH INTO MAPPING TABLE ENTRY
; 1005          1489  2 !               (8) COPY ALIGNMENT INTO MAPPING TABLE
; 1006          1490  2 !               (9) COPY MODULE DESCRIPTOR ADDRESS INTO MAPPING TABLE
; 1007          1491  2 !               (10) COUNT THIS P-SECTION AND CLEAR NO P-SECTIONS FLAG
; 1008          1492  2 !               (11) IF NECESSARY EXTEND MODULE DESCRIPTOR FOR ANOTHER
; 1009          1493  2 !                       P-SECTION AND INSERT ADDRESS OF P-SECTION
; 1010          1494  2 !                       IN TABLE.
; 1011          1495  2 !               (12) IF ANY PREMATURE SYMBOL DEFINITIONS, REMOVE EACH FROM
; 1012          1496  2 !                       LIST, ADDING P-SECTION BASE TO RELOCATABLE VALUES
; 1013          1497  2 !                       AND LINKING THEM ON THE P-SECION'S LIST OF SYMBOLS
; 1014          1498  2 !--
; 1015          1499  2     local
; 1016          1500  2         psctdesc : ref block [, byte],          ! CURRENT P-SECT DESCRIPTOR
; 1017          1501  2         modpscontriblk : ref block [, byte],    ! MODULE CONTRIBUTION DATA BLOCK
; 1018          1502  2         lastmodpscontriblk : ref block [, byte],
; 1019          1503  2         psectname : ref vector [, byte],        ! POINTER TO PSECT NAME
; 1020          1504  2         newpsect,                               ! TRUE IF NEW PSECT
; 1021          1505  2         sgpstype,                               ! TRUE IF GSD$C_SGPS
; 1022          1506  2         pscope,                                 ! P-SECTION SEARCH SCOPE
; 1023          1507  2         psflags,
; 1024          1508  2         prevdfound,                             ! TRUE IF PREV. DEF. OF COMMON FROM SHR IMAGE
; 1025          1509  2         delpscdesc : ref block [, byte],        ! POINTER TO DELETED PSECT DESCRIPTOR
; 1026          1510  2         psctmapent : ref block [, byte],        ! P-SECTION MAPPING TABLE ENTRY
; 1027          1511  2         around,                                 ! ROUND UP FOR ALIGNMENT
; 1028          1512  2         pddptr : ref block [, byte],
; 1029          1513  2         localpdd : block [pdd$c_size, byte],
; 1030          1514  2         premsymlst,                             ! SAVE PREMATURE SYMBOL LIST
; 1031          1515  2         premsym : ref block [, byte];           ! PREMATURELY DEFINED SYMBOL
; 1032          1516  2
; 1033          1517  2     bind
; 1034          1518  2         length = around,
; 1035          1519  2         psctdef = objvec [.gsdoffset] : block [, byte];
; 1036          1520  2
; 1037          1521  2 !
; 1038          1522  2 ! DETERMINE WHICH PSECT SUBRECORD TYPE THIS IS
; 1039          1523  2 !
```

```
; 1040          1524  2     sgpstype = false;
; 1041          1525  2
; 1042          1526  2     if .psctdef [gps$b_gsdtyp] eql gsd$c_psc
; 1043          1527  2     then
; 1044          1528  2         psectname = psctdef [gps$b_namlng]
; 1045          1529  2     else
; 1046          1530  3         begin
; 1047          1531  3         psectname = psctdef [sgps$b_namlng];    ! SHAREABLE IMAGE PSECT
; 1048          1532  3         sgpstype = true;
; 1049          1533  3         end;
; 1050          1534  2
; 1051          1535  2 !
; 1052          1536  2 !       FIRST CHECK FOR LEGAL P-SECTION NAME
; 1053          1537  2 !
; 1054          1538  2
; 1055          1539  2     if .psectname [0] gtru sym$c_maxlng          ! CHECK NAME WITHIN THE LEGAL
; 1056          1540  2         or .psectname [0] eql 0                  ! RANGE FOR SYMBOL AND P-SECTION
; 1057          1541  2     then
; 1058          1542  3         begin
; 1059          1543  3         signal (lin$_illnamelen, 6,              ! NAMES AND IF NOT
; 1060          1544  3             lnk$gt_pscstring, .psectname, .psectname [0], sym$c_maxlng,          ! ISSUE AN ERROR MESSAGE
; 1061          1545  3             obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
; 1062          1546  3         return false;
; 1063          1547  3         end;
; 1064          1548  2
; 1065          1549  2 !
; 1066          1550  2 ! SEE IF PSECT DEFINED BY OPTION
; 1067          1551  2 !
; 1068          1552  2     ch$fill (0, pdd$c_size, localpdd);           ! ZERO ALL FIELDS IN LOCAL PDD
; 1069          1553  2     localpdd [pdd$b_align] = %x'FF';             ! SET ITS ALIGNMENT TO ALL ONES
; 1070          1554  2     pddptr = localpdd;                           ! PRESET IN CASE NOT FOUND
; 1071          1555  2     psflags = .psctdef [gps$w_flags];            ! ASSUME NOT DEFINED BY OPTION
; 1072          1556  2
; 1073          1557  2     if lnk$srcpscdef (.psectname, pddptr)        ! SEE IF DEFINED BY OPTION
; 1074          1558  2     then
; 1075          1559  3         begin
; 1076          1560  3         pddptr [pdd$b_namlng] = .pddptr [pdd$b_namlng] or %x'80';        ! AND IF SO, MARK AS DEFINED
; 1077          1561  3         psflags = .pddptr [pdd$w_flags] or       ! SET THE FLAGS
; 1078          1562  3         (.psflags and not .pddptr [pdd$w_flgmsk]);       ! QUALIFIED BY THE MASK
; 1079          1563  2         end;
; 1080          1564  2
; 1081          1565  2 !
; 1082          1566  2 ! IF THIS IS A SHAREABLE IMAGE CLUSTER AND THIS IS A GLOBAL PSECT
; 1083          1567  2 ! (COMMON), THEN SEE IF DEFINED SOMEWHERE ELSE
; 1084          1568  2 !
; 1085          1569  2
; 1086          1570  2     if .lnk$gl_curclu [clu$v_shrimg] and (.psctdef [gps$w_flags] and psc$m_shrbits) eql psc$m_shrbits
; 1087          1571  2     then
; 1088          1572  2         prevdfound = delete_psect (.psectname, delpscdesc)
; 1089          1573  2     else
; 1090          1574  2         prevdfound = false;
; 1091          1575  2
; 1092          1576  2 !
; 1093          1577  2 ! CHECK ALIGNMENT
; 1094          1578  2 !
; 1095          1579  2
; 1096          1580  2     if .psctdef [gps$b_align] gtru obj$c_pscalilim      ! IF THE ALIGNMENT IS GREATER
```

```
;  1097          1581  2      then
;  1098          1582  3          begin
;  1099          1583  3          signal (lin$_pscali, 4, .psectname, .psctdef [gps$b_align],     ! THAN THE MAXIMUM, ISSUE
;  1100          1584  3              obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);   ! ERROR AND QUIT
;  1101          1585  3          psctdef [gps$b_align] = obj$c_pscalilim;        ! USE OBJ LANG MAX
;  1102          1586  2          end;
;  1103          1587  2
;  1104          1588  2  !
;  1105          1589  2  !       SEARCH FOR P-SECTION DEFINED, CREATING DESCRIPTOR IF NOT.
;  1106          1590  2  !
;  1107          1591  2
;  1108          1592  2      if .obmodesc [omd$v_shrimg]                     ! IF THIS IS A SHAREABLE IMAGE
;  1109          1593  2      then
;  1110          1594  2          pscope = 0                                  ! THEN P-SECTION SEARCH IS ONLY IN THIS CLUSTER
;  1111          1595  2      else
;  1112          1596  2
;  1113          1597  2          if (pscope = .psctdef [gps$v_gbl]) eql 0        ! ELSE DEPENDS ON PSECT SCOPE
;  1114          1598  2          then
;  1115          1599  2              pscope = .pddptr [pdd$w_flags] and gps$m_gbl;       ! OR OPTION DEFINITION SCOPE
;  1116          1600  2
;  1117          1601  2      if lnk$fndpscnam (.psectname, .psflags, .pscope, psctdesc)  ! GET DESCRIPTOR ADDRESS
;  1118          1602  2          and .psctdesc [psc$v_usrpsc]                ! AND PSECT WAS SEEN IN SOURCE
;  1119          1603  2      then
;  1120          1604  3          begin                                       ! AND IF FOUND
;  1121          1605  3          newpsect = false;
;  1122          1606  3
;  1123          1607  3  ! ** THIS CHECK DISABLED UNTIL C ISSUE RESOLVED
;  1124          1608  3  !
;  1125          1609  3  !       if .psctdesc [psc$l_omdnum] eql .lnk$gw_nmodules        ! IF PSECT ALREADY DEFINED IN THIS MODULE
;  1126          1610  3  !       then signal (lin$_muldefpsc, 3                          !  THEN ISSUE A WARNING MESSAGE
;  1127          1611  3  !                   ,psctdesc [psc$b_namlng]
;  1128          1612  3  !                   ,obmodesc [omd$b_namlng]
;  1129          1613  3  !                   ,lnk$gl_curfil [fdb$q_filename]
;  1130          1614  3  !                   );
;  1131          1615  3  !
;  1132          1616  3  ! ** END OF DISABLED CODE
;  1133          1617  3
;  1134          1618  3          psctdesc [psc$l_omdnum] = .lnk$gw_nmodules;     ! SET DEFINED IN CURRENT MODULE
;  1135          1619  3
;  1136          1620  3          if not .psctdesc [psc$v_optpsc]                 ! IF PSECT NOT MODIFIED BY OPTION
;  1137          1621  3          then
;  1138          1622  4              begin
;  1139          1623  4
;  1140          1624  6              if .psctdef [gps$w_flags] neq ((.psctdesc [psc$w_flags]     ! CHECK COMPATIBLE
;  1141          1625  5                  and not (psc$m_supres or psc$m_usrpsc)) and psc$m_pscbits)
;  1142          1626  4              then
;  1143          1627  4                  signal (lin$_mulpsc, 3,                                 ! ATTRIBUTES, ISSUING ERROR
;  1144          1628  4                      .psectname, obmodesc [omd$b_namlng],                ! IF NOT
;  1145          1629  4                      lnk$gl_curfil [fdb$q_filename]);
;  1146          1630  4
;  1147          1631  4                  if .psctdesc [psc$v_ovr]                            ! IF OVERLAYED
;  1148          1632  4                  then
;  1149          1633  4
;  1150          1634  4                      if .psctdef [gps$b_align] neq .psctdesc [psc$b_align]   ! CHECK EQUAL
;  1151          1635  4                      then
;  1152          1636  4                          signal (lin$_ovrali, 3,     ! ALIGNMENT
;  1153          1637  4                              .psectname, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
```

```
;1154    1638  4
;1155    1639  4            end;                                    ! PSECT NOT MODIFIED BY OPTION
;1156    1640  4
;1157    1641  4        if .psctdef [gps$b_align] gtru .psctdesc [psc$b_align]  ! MAXIMIZE
;1158    1642  4            and .pddptr [pdd$b_align] eql %x'FF'
;1159    1643  4        then
;1160    1644  4            psctdesc [psc$b_align] = .psctdef [gps$b_align];    ! THE ALIGNMENT
;1161    1645  4
;1162    1646  4        end
;1163    1647  3    !
;1164    1648  3    !   P-SECTION IS NOT YET DEFINED - SO DO THAT
;1165    1649  3    !
;1166    1650  3    else
;1167    1651  3        begin                                       ! AND INSERT P-SECTION
;1168    1652  3        lnk$gw_npsects = .lnk$gw_npsects + 1;       ! AFTER COUNTING IT
;1169    1653  3        newpsect = true;
;1170    1654  3        psctdesc [psc$b_align] = .pddptr [pdd$b_align]; ! SET THE ALIGNMENT FROM PDD
;1171    1655  3
;1172    1656  3        if .psctdesc [psc$b_align] eql %x'FF'       ! BUT IF NONE SPECIFIED
;1173    1657  3        then
;1174    1658  3            psctdesc [psc$b_align] = .psctdef [gps$b_align];    ! THEN USE INPUT FILE
;1175    1659  3
;1176    1660  3        psctdesc [psc$w_flags] = .psflags;          ! SET THE FLAGS
;1177    1661  3        psctdesc [psc$v_usrpsc] = true;             ! FLAG DEFINED BY USER SOURCE
;1178    1662  3        psctdesc [psc$v_shrimg] = .lnk$gl_curclu [clu$v_shrimg];
;1179    1663  3                                                    ! PROPOGATE SHR IMG FLAG INTO PSECT DESCRIPTOR
;1180    1664  3        psctdesc [psc$v_newdef] = .sgpstype;        ! REMEMBER IF SGPS DEFINITION
;1181    1665  3
;1182    1666  3        if .lnk$gl_ctlmsk [lnk$v_shr]               ! IF CREATING A SHAREABLE IMAGE
;1183    1667  3            and not .obmodesc [omd$v_shrimg]        !  AND THIS IS NOT A SHAREABLE IMAGE INPUT MODULE
;1184    1668  4            and (.psctdesc [psc$w_flags] and psc$m_shrbits)
;1185    1669  4                                                    !  AND IF PSECT WILL BE OUTPUT TO SHAREABLE IMAGE
;1186    1670  3            eql psc$m_shrbits                       ! (MUST BE RELOCATABLE, GLOBAL AND OVERLAID
;1187    1671  3        then
;1188    1672  4            begin
;1189    1673  4            lnk$gl_pshrnum = .lnk$gl_pshrnum + 1;       ! THEN COUNT IT FOR LNKSYMOUT
;1190    1674  4
;1191    1675  4            if .lnk$gl_pshrnum eql lnk$c_maxpsects + 1  ! IF THERE ARE TOO MANY
;1192    1676  4            then
;1193    1677  4                signal_stop (lin$_excpsc, 2, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename],
;1194    1678  4                    lin$_format);
;1195    1679  4
;1196    1680  4            end;
;1197    1681  3
;1198    1682  3        if .lnk$gl_ctlmsk [lnk$v_intfil]            ! IF THIS IS THE INTERNAL (SYSLIB) FILE
;1199    1683  3            and .lnk$gl_ctlmsk [lnk$v_supsys]       ! AND SYSLIB SUPPRESSION IS ON
;1200    1684  3        then
;1201    1685  3            psctdesc [psc$v_supres] = true;         ! THEN SUPPRESS THIS P-SECTION
;1202    1686  3
;1203    1687  3        if .obmodesc [omd$v_shrimg]                 ! IF A SHAREABLE IMAGE P-SECTION
;1204    1688  3        then
;1205    1689  3
;1206    1690  4            if .sgpstype
;1207    1691  4            then
;1208    1692  4                psctdesc [psc$l_base] = .psctdef [sgps$l_base]
;1209    1693  4            else
;1210    1694  4                psctdesc [psc$l_base] = .psctdef [gps$l_alloc];
```

```
;  1211     1695  3
;  1212     1696  3                                          ! ITS BASE IS GIVEN BY SIZE FIELD IF NORMAL PSECT DEF
;  1213     1697  3          end;
;  1214     1698  3
;  1215     1699  2      !
;  1216     1700  2      ! IF THIS IS A SHAREABLE IMAGE PSECT THAT WAS FOUND ELSEWHERE,
;  1217     1701  2      ! WE HAVE TO RUN AROUND AND FIX UP THE POINTERS IN THE PSECT
;  1218     1702  2      ! MAPPING TABLES FOR THE PREVIOUS DEF. TO POINT TO THIS NEW
;  1219     1703  2      ! DEFINITION.  ALSO, PUT THE OLD PSECT MAPPING LIST ONTO THE
;  1220     1704  2      ! NEW DESCRIPTOR
;  1221     1705  2      !
;  1222     1706  2
;  1223     1707  2          if .prevdfound
;  1224     1708  2          then
;  1225     1709  3              begin
;  1226     1710  3              modpscontriblk = .delpscdesc [psc$l_mpclst];     ! GET MPC LIST POINTER
;  1227     1711  3              psctdesc [psc$l_mpclst] = .modpscontriblk;       ! SET INTO NEW DESCRIPTOR
;  1228     1712  3              psctdesc [psc$l_lstmpc] = .delpscdesc [psc$l_lstmpc];   ! COPY END OF LIST POINTER
;  1229     1713  3
;  1230     1714  3              while .modpscontriblk neq 0 do
;  1231     1715  4                  begin
;  1232     1716  4                  psctmapent = fndpscmapent (.modpscontriblk [mpc$w_pscnum],  ! FIND PSECT MAPPING ENTRY
;  1233     1717  4                      .modpscontriblk [mpc$l_ownomd]);
;  1234     1718  4                  psctmapent [pmt$l_pscdes] = .psctdesc;       ! SET POINTER TO NEW DESCRIPTOR
;  1235     1719  4                  modpscontriblk = .modpscontriblk [mpc$l_nxtmpc];
;  1236     1720  3                  end;
;  1237     1721  3
;  1238     1722  2              end;
;  1239     1723  2
;  1240     1724  2      !
;  1241     1725  2      !       NOW TO CREATE A BLOCK OF DATA DESCRIBING THIS
;  1242     1726  2      !       MODULE'S CONTRIBUTION TO THE P-SECTION, LINK THE
;  1243     1727  2      !       BLOCK TO END OF THE P-SECTION MODULAR CONTRIBUTION LIST
;  1244     1728  2      !       AND POINT MODULE DESCRIPTOR
;  1245     1729  2      !       MAPPING TABLE ENTRY AT THIS BLOCK.
;  1246     1730  2      !
;  1247     1731  2          lnk$alloblk (mpc$c_size, modpscontriblk);                    ! ALLOCATE THE BLOCK
;  1248     1732  2          ch$fill ( 0, mpc$c_size, .modpscontriblk );                  !  (make sure it's clean)
;  1249     1733  2          modpscontriblk [mpc$l_ownomd] = .obmodesc [omd$l_nxtomd];    ! LINK INTO TEMP LIST
;  1250     1734  2          obmodesc [omd$l_nxtomd] = .modpscontriblk;                   ! TO BE FIXED UP AT PROEOM TIME
;  1251     1735  2          lastmodpscontriblk = .psctdesc [psc$l_lstmpc];               ! GET POINTER TO LAST CONTRIB. BLOCK FOR PSE
;  1252     1736  2
;  1253     1737  2          if .lastmodpscontriblk neq 0                                 ! IF THERE IS A LAST
;  1254     1738  2          then
;  1255     1739  2              lastmodpscontriblk [mpc$l_nxtmpc] = .modpscontriblk;     ! THEN LINK NEW BLOCK INTO LIST
;  1256     1740  2
;  1257     1741  2          psctdesc [psc$l_lstmpc] = .modpscontriblk;                   ! AND MAKE IT THE NEW LAST.
;  1258     1742  2      !
;  1259     1743  2
;  1260     1744  2          if .obmodesc [omd$w_hipsct] eql lnk$c_maxpsects      ! IF TOO MANY PSECTS ALREADY
;  1261     1745  2              and not .obmodesc [omd$v_nopsct]
;  1262     1746  2          then                                             ! BEEN SEEN FOR THIS MODULE
;  1263     1747  2              signal_stop (lin$_excpsc, 2,                 ! ISSUE AN ERROR MESSAGE AND GIVE UP
;  1264     1748  2                  obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename], lin$_format);
;  1265     1749  2
;  1266     1750  2      !
;  1267     1751  2          obmodesc [omd$v_nopsct] = false;                 ! CLEAR NO PSECTS FLAG
```

```
;1268    1752  2     obmodesc [omd$w_hipsct] = .obmodesc [omd$w_hipsct] + 1;      ! COUNT NEW P-SECTION
;1269    1753  2     modpscontriblk [mpc$w_pscnum] = .obmodesc [omd$w_hipsct];    ! SAVE PSECT NUMBER IN MPC
;1270    1754  2     psctmapent = fndpscmapent (.obmodesc [omd$w_hipsct]);        ! GET MAP TABLE ENTRY
;1271    1755  2     premsymlst = .psctmapent [pmt$l_symlst];     ! SAVE PREMATURE SYMBOL LIST
;1272    1756  2     psctmapent [pmt$l_pscdes] = .psctdesc;       ! POINT IT TO PSECTION DESCRIPTOR
;1273    1757  2     psctmapent [pmt$l_modcon] = .modpscontriblk;         ! AND TO THE NEW CONTRIBUTION BLOCK.
;1274    1758  2     modpscontriblk [mpc$b_align] = .psctdef [gps$b_align];       ! COPY ITS ALIGNMENT
;1275    1759  2     modpscontriblk [mpc$l_offset] = 0;           ! ASSUME ABSOLUTE AND/OR OVERLAYED
;1276    1760  2                                                  ! AND THUS 0 OFFSET CONTRIBUTION
;1277    1761  2
;1278    1762  2     if .psctdesc [psc$v_rel]                     ! THAT IS ALL FOR ABSOLUTE
;1279    1763  2     then
;1280    1764  3         begin                                    ! P-SECTIONS
;1281    1765  3
;1282    1766  3         if not .psctdesc [psc$v_ovr]             ! IF A CONCATENATED PSECT NOT FROM SHR IMAGE
;1283    1767  3             and not .obmodesc [omd$v_shrimg]
;1284    1768  3         then
;1285    1769  4             begin
;1286    1770  4             around = (1^.psctdef [gps$b_align]) - 1;
;1287    1771  5             modpscontriblk [mpc$l_offset] = (.psctdesc [psc$l_base] + .around)
;1288    1772  5                                                  ! ADD IN THE ROUNDING AND TRUNCATE
;1289    1773  4             and not .around;
;1290    1774  4             around = .modpscontriblk [mpc$l_offset] - .psctdesc [psc$l_base];   ! THEN COMPUTE AMOUNT ADDED
;1291    1775  4             psctdesc [psc$l_base] = .psctdesc [psc$l_base] + .around +  ! UPDATE THE BASE FOR NEXT
;1292    1776  4             .psctdef [gps$l_alloc];              ! CONTRIBUTOR.
;1293    1777  4             psctdesc [psc$l_length] = .psctdesc [psc$l_base];   ! WHICH IS ALSO TOTAL LENGTH
;1294    1778  4             end
;1295    1779  3         else
;1296    1780  4             begin                                ! FOR OVERLAYED
;1297    1781  4
;1298    1782  4             local
;1299    1783  4                 omd : ref block [, byte],
;1300    1784  4                 fdb : ref block [, byte];
;1301    1785  4
;1302    1786  5             if (.sgpstype                        ! IF SHAREABLE IMAGE PSECT TYPE
;1303    1787  5                 and .psctdesc [psc$v_ovr] and .prevdfound        !  AND IT WAS DEFINED BEFORE IN OBJ MODULE
;1304    1788  5                 and (.delpscdesc [psc$l_length] gtru .psctdef [sgps$l_alloc]))
;1305    1789  5                                                  ! TO BE LARGER THAN ONE IN SHR IMG
;1306    1790  4             then
;1307    1791  5                 begin
;1308    1792  5                 lib$lookup_tree (lnk$gl_omdtree, .delpscdesc [psc$l_omdnum],    ! FIND DEFINING MODULE OMD
;1309    1793  5                     lnk$compare_omd, omd);
;1310    1794  5                 omd = .omd [node$l_ptr];         ! POINT TO OMD FROM NODE
;1311    1795  5                 fdb = .omd [omd$l_ownfdb];       ! GET POINTER TO FDB FOR MODULE
;1312    1796  5                 signal (lin$_shrpsclng, 6, .psectname, .delpscdesc [psc$l_length], omd [omd$b_namlng],
;1313    1797  5                     fdb [fdb$q_filename], .psctdef [sgps$l_alloc], lnk$gl_curfil [fdb$q_filename],
;1314    1798  5                     lin$_noimgfil);
;1315    1799  5                 lnk$gl_ctlmsk [lnk$v_image] = false;    ! DISABLE IMAGE PRODUCTION
;1316    1800  4                 end;
;1317    1801  4
;1318    1802  5             if (.psctdesc [psc$v_shrimg] and .psctdesc [psc$v_newdef]    ! IF PSECT DEFINED IN SHAREABLE IMAG
;1319    1803  5                 and .psctdesc [psc$v_ovr] and not .newpsect      ! AND THIS IS NOT FIRST DEFINITION
;1320    1804  5                 and (.psctdef [gps$l_alloc] gtru .psctdesc [psc$l_length]))
;1321    1805  5                                                  ! AND THIS DEFINITION IS BIGGER THAN SHR IMAGE ONE
;1322    1806  4             then
;1323    1807  5                 begin                            ! THAT IS AN ERROR
;1324    1808  5                 fdb = .psctdesc [psc$l_cludsc]; ! GET CLUSTER DESCRIPTOR POINTER
```

```
; 1325     1809  5                fdb = .fdb [clu$l_fstfdb];      ! GET POINTER TO FDB FOR SHAREABLE IMAGE
; 1326     1810  5                signal (lin$_shrpsclng, 6, .psectname, .psctdef [gps$l_alloc],  ! SIGNAL THE PROBLEM
; 1327     1811  5                    obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename], .psctdesc [psc$l_length],
; 1328     1812  5                    fdb [fdb$q_filename], lin$_noimgfil);
; 1329     1813  5                lnk$gl_ctlmsk [lnk$v_image] = false;    ! TURN OFF THE IMAGE
; 1330     1814  4                end;
; 1331     1815  4
; 1332     1816  4            if not .obmodesc [omd$v_shrimg]     ! IF NOT A SHAREABLE IMAGE
; 1333     1817  4            then
; 1334     1818  5                begin
; 1335     1819  5
; 1336     1820  5                if .psctdef [gps$l_alloc] gtru .psctdesc [psc$l_length] ! P-SECTION MAXIMIZE
; 1337     1821  5                then
; 1338     1822  5                    psctdesc [psc$l_length] = .psctdef [gps$l_alloc];   ! THE ALLOCATION
; 1339     1823  5
; 1340     1824  5                end
; 1341     1825  4            else
; 1342     1826  5                begin
; 1343     1827  5
; 1344     1828  5                if .sgpstype                    ! IF SPECIAL SHAREABLE IMAGE PSECT
; 1345     1829  5                then
; 1346     1830  5                    psctdesc [psc$l_length] = .psctdef [sgps$l_alloc];  !  THEN SET LENGTH OF PSECT
; 1347     1831  5
; 1348     1832  4                end;
; 1349     1833  4
; 1350     1834  3            end;
; 1351     1835  3
; 1352     1836  3        if not .obmodesc [omd$v_shrimg]
; 1353     1837  3        then
; 1354     1838  4            begin
; 1355     1839  4            modpscontriblk [mpc$l_length] = .psctdef [gps$l_alloc];     ! SET MODULE'S CONTRIBUTION
; 1356     1840  4            obmodesc [omd$l_alloc] = .obmodesc [omd$l_alloc] +  ! FINALLY ACCUMULATE THE CONTRIBUTION
; 1357     1841  4            .modpscontriblk [mpc$l_length];
; 1358     1842  3            end;
; 1359     1843  3
; 1360     1844  2        end;
; 1361     1845  2
; 1362     1846  2
; 1363     1847  2        NOW GO DOWN THE LIST OF PREMATURE SYMBOLS
; 1364     1848  2                (THOSE AWAITING DEFINITION OF THIS PSECTION)
; 1365     1849  2                AND RELOCATE THEM (IF THEY ARE RELOCATABLE)
; 1366     1850  2                AND THEN LINK THEM ON TO THE P-SECTON LIST
; 1367     1851  2
; 1368     1852  2
; 1369     1853  2    while (premsym = .premsymlst) neq 0 do       ! RELOCATABLE AND THERE
; 1370     1854  3        begin                                    ! ARE SOME PREMATURE
; 1371     1855  3
; 1372     1856  3        if .premsym [sym$v_rel]                  ! FOR EACH THAT IS RELOCATABLE
; 1373     1857  3        then
; 1374     1858  3            premsym [sym$l_value] = .premsym [sym$l_value] +    ! ADD IN THE BASE OFFSET
; 1375     1859  3            .modpscontriblk [mpc$l_offset];     ! OF THE MODULE'S
; 1376     1860  3
; 1377     1861  3        premsymlst = .premsym [sym$l_psclst];   ! TAKE IT OFF LIST
; 1378     1862  3        premsym [sym$l_psclst] = .psctdesc [psc$l_symlst];      ! AND PUT ON THE
; 1379     1863  3        psctdesc [psc$l_symlst] = .premsym;     ! P-SECTION SYMBOL LIST
; 1380     1864  2        end;
; 1381     1865  2
```

```
; 1382          1866  2 !
; 1383          1867  2 ! COMPUTE LENGTH OF THIS GSD SUBRECORD
; 1384          1868  2 !
; 1385          1869  2
; 1386          1870  2     if .sgpstype
; 1387          1871  2     then
; 1388          1872  2         length = psctdef [sgps$t_name] - psctdef [sgps$t_start] + .psectname [0]
; 1389          1873  2     else
; 1390          1874  2         length = psctdef [gps$t_name] - psctdef [gps$t_start] + .psectname [0];
; 1391          1875  2
; 1392          1876  2     gsdoffset = .gsdoffset + .length;          ! UPDATE GSD OFFSET INTO RECORD
; 1393          1877  2     return true;
; 1394          1878  1     end;                                       ! OF P-SECTION PROCESSOR


                                         0FFC 00000 PROPSECTDEF:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 1467
                          5E           2C  C2 00002       SUBL2   #44, SP                                   ;
                          58 00000000'  EF  3C 00005       MOVZWL  GSDOFFSET, R8                             ; 1519
                          58 00000000'  EF  C0 0000C       ADDL2   OBJVEC, R8                                ;
                                       7E  D4 00013       CLRL    SGPSTYPE                                  ; 1524
                                       68  95 00015       TSTB    (R8)                                      ; 1526
                                       06  12 00017       BNEQ    1$                                        ;
                          56       08  A8  9E 00019       MOVAB   8(R8), PSECTNAME                          ; 1528
                                       07  11 0001D       BRB     2$                                        ;
                          56       0C  A8  9E 0001F 1$:   MOVAB   12(R8), PSECTNAME                         ; 1531
                          6E           01  D0 00023       MOVL    #1, SGPSTYPE                              ; 1532
                          1F           66  91 00026 2$:   CMPB    (PSECTNAME), #31                          ; 1539
                                       04  1A 00029       BGTRU   3$                                        ;
                                       66  95 0002B       TSTB    (PSECTNAME)                               ; 1540
                                       2D  12 0002D       BNEQ    4$                                        ;
                7E 00000000G  00       14  C1 0002F 3$:   ADDL3   #20, LNK$GL_CURFIL, -(SP)                 ; 1545
                             00000000'  EF  9F 00037       PUSHAB  OBMODESC+40                               ;
                                       1F  DD 0003D       PUSHL   #31                                       ;
                          7E           66  9A 0003F       MOVZBL  (PSECTNAME), -(SP)                        ;
                                       56  DD 00042       PUSHL   PSECTNAME                                 ;
                             00000000'  EF  9F 00044       PUSHAB  LNK$GT_PSCSTRING                          ; 1543
                                       06  DD 0004A       PUSHL   #6                                        ; 1545
                             00000000G  8F  DD 0004C       PUSHL   #LIN$_ILLNAMELEN                          ;
                   00000000G  00       08  FB 00052       CALLS   #8, LIB$SIGNAL                            ;
                                     0474  31 00059       BRW     38$                                       ; 1546
     10         00            6E       00  2C 0005C 4$:   MOVC5   #0, (SP), #0, #16, LOCALPDD               ; 1552
                                   20  AE     00061                                                         ;
                     2E   AE           01  8E 00063       MNEGB   #1, LOCALPDD+14                           ; 1553
                     0C   AE       20  AE  9E 00067       MOVAB   LOCALPDD, PDDPTR                          ; 1554
                          52       02  A8  3C 0006C       MOVZWL  2(R8), PSFLAGS                            ; 1555
                                   0C  AE  9F 00070       PUSHAB  PDDPTR                                    ; 1557
                                       56  DD 00073       PUSHL   PSECTNAME                                 ;
                   00000000G  00       02  FB 00075       CALLS   #2, LNK$SRCPSCDEF                         ;
                          18           50  E9 0007C       BLBC    R0, 5$                                    ;
                          50       0C  AE  D0 0007F       MOVL    PDDPTR, R0                                ; 1560
                     0F   A0       80  8F  88 00083       BISB2   #128, 15(R0)                              ;
                          51       0C  A0  3C 00088       MOVZWL  12(R0), R1                                ; 1562
                51        52           51  CB 0008C       BICL3   R1, PSFLAGS, R1                           ;
```

```
                                52        0A    A0  3C 00090          MOVZWL  10(R0), PSFLAGS                          ;
                                52              C8  00094          BISL2   R1, PSFLAGS                             ;
                                50 00000000G    00  D0 00097 5$:    MOVL    LNK$GL_CURCLU, R0                       ; 1570
                          20        58    A0      02  E1 0009E          BBC     #2, 88(R0), 6$                          ;
                                50        02    A8  3C 000A3          MOVZWL  2(R8), R0                               ;
                                50 FFFFFFCB     8F  CA 000A7          BICL2   #-53, R0                                ;
                                34              50  D1 000AE          CMPL    R0, #52                                 ;
                                                10  12 000B1          BNEQ    6$                                      ;
                                        10      AE  9F 000B3          PUSHAB  DELPSCDESC                              ; 1572
                                                56  DD 000B6          PUSHL   PSECTNAME                               ;
                                      FDD4      CF      02  FB 000B8          CALLS   #2, DELETE_PSECT                        ;
                                        04      AE      50  D0 000BD          MOVL    R0, PREVDFOUND                  ;
                                                03  11 000C1          BRB     7$                                      ;
                                        04      AE  D4 000C3 6$:    CLRL    PREVDFOUND                              ; 1574
                                50        01    A8  9E 000C6 7$:    MOVAB   1(R8), R11                              ; 1580
                                09              6B  91 000CA          CMPB    (R11), #9                               ;
                                                25  1B 000CD          BLEQU   8$                                      ;
                          7E 00000000G    00      14  C1 000CF          ADDL3   #20, LNK$GL_CURFIL, -(SP)               ; 1584
                                      00000000'     EF  9F 000D7          PUSHAB  OBMODESC+40                             ;
                                7E              6B  9A 000DD          MOVZBL  (R11), -(SP)                            ;
                                                56  DD 000E0          PUSHL   PSECTNAME                               ;
                                                04  DD 000E2          PUSHL   #4                                      ;
                                      00000000G     8F  DD 000E4          PUSHL   #LIN$_PSCALI                            ;
                             00000000G    00      06  FB 000EA          CALLS   #6, LIB$SIGNAL                          ;
                                6B              09  90 000F1          MOVB    #9, (R11)                               ; 1585
                          04 00000000'      EF      02  E1 000F4 8$:    BBC     #2, OBMODESC+20, 9$                     ; 1592
                                                51  D4 000FC          CLRL    PSCOPE                                  ; 1594
                                                17  11 000FE          BRB     10$                                     ;
                    51        02    A8              01      04  EF 00100 9$:    EXTZV   #4, #1, 2(R8), PSCOPE           ; 1597
                                                0F  12 00106          BNEQ    10$                                     ;
                                50        0C    AE  D0 00108          MOVL    PDDPTR, R0                              ; 1599
                                51        0A    A0  3C 0010C          MOVZWL  10(R0), PSCOPE                          ;
                                51 FFFFFFEF     8F  CA 00110          BICL2   #-17, PSCOPE                            ;
                                        14      AE  9F 00117 10$:    PUSHAB  PSCTDESC                                ; 1601
                                                51  DD 0011A          PUSHL   PSCOPE                                  ;
                                                52  DD 0011C          PUSHL   PSFLAGS                                 ;
                                                56  DD 0011E          PUSHL   PSECTNAME                               ;
                             00000000G    00      04  FB 00120          CALLS   #4, LNK$FNDPSCNAM                       ;
                                03              50  E8 00127          BLBS    R0, 12$                                 ;
                                              008B  31 0012A 11$:    BRW     16$                                     ;
                                5C        14    AE  D0 0012D 12$:    MOVL    PSCTDESC, R0                            ; 1602
                          F4        0B    A0      03  E1 00131          BBC     #3, 11(R0), 11$                         ;
                                        08      AE  D4 00136          CLRL    NEWPSECT                                ; 1605
                                53        14    AE  D0 00139          MOVL    PSCTDESC, R3                            ; 1618
                                      28    A3 00000000'    EF  3C 0013D          MOVZWL  LNK$GW_NMODULES, 40(R3)         ;
                                59        0A    A3  9E 00145          MOVAB   10(R3), R9                              ; 1620
                          53            69      0A      E0 00149          BBS     #10, (R9), 14$                          ;
                                50        02    A8  3C 0014D          MOVZWL  2(R8), R0                               ; 1624
                    50            69      0A      00      ED 00151          CMPZV   #0, #10, (R9), R0                       ;
                                                1F  13 00156          BEQL    13$                                     ;
                          7E 00000000G    00      14  C1 00158          ADDL3   #20, LNK$GL_CURFIL, -(SP)               ; 1629
                                      00000000'     EF  9F 00160          PUSHAB  OBMODESC+40                             ; 1628
                                                56  DD 00166          PUSHL   PSECTNAME                               ; 1629
                                                03  DD 00168          PUSHL   #3                                      ;
                                      000U 00UG     8F  DD 0016A          PUSHL   #LIN$_MULPSC                            ;
                             00000000G    00      05  FB 00170          CALLS   #5, LIB$SIGNAL                          ;
                          25            69              02  E1 00177 13$:    BBC     #2, (R9), 14$                           ; 1631
```

```
                    2C   A3              6B  91 0017B          CMPB    (R11), 44(R3)                                   ; 1634
                                         1F  13 0017F          BEQL    14$
                7E 00000000G 00          14  C1 00181          ADDL3   #20, LNK$GL_CURFIL, -(SP)                       ; 1637
                             00000000'   EF  9F 00189          PUSHAB  OBMODESC+40
                                         56  DD 0018F          PUSHL   PSECTNAME
                                         03  DD 00191          PUSHL   #3
                             00000000G   8F  DD 00193          PUSHL   #LIN$_OVRALI
                    00000000G 00         05  FB 00199          CALLS   #5, LIB$SIGNAL
                    2C   A3              6B  91 001A0 14$:     CMPB    (R11), 44(R3)                                   ; 1641
                                         0F  1B 001A4          BLEQU   15$
                         50      0C      AE  D0 001A6          MOVL    PDDPTR, R0                                      ; 1642
                    FF   8F      0E      A0  91 001AA          CMPB    14(R0), #255
                                         04  12 001AF          BNEQ    15$
                    2C   A3              6B  90 001B1          MOVB    (R11), 44(R3)                                   ; 1644
                                   00C4  31  001B5 15$:        BRW     21$                                             ; 1601
                             00000000'   EF  B6 001B8 16$:     INCW    LNK$GW_NPSECTS                                  ; 1652
                    08   AE              01  D0 001BE          MOVL    #1, NEWPSECT                                    ; 1653
                         53      14      AE  D0 001C2          MOVL    PSCTDESC, R3                                    ; 1654
                         50      0C      AE  D0 001C6          MOVL    PDDPTR, R0
                    2C   A3      0E      A0  90 001CA          MOVB    14(R0), 44(R3)
                    FF   8F      2C      A3  91 001CF          CMPB    44(R3), #255                                    ; 1656
                                         04  12 001D4          BNEQ    17$
                    2C   A3              6B  90 001D6          MOVB    (R11), 44(R3)                                   ; 1658
                         59      0A      A3  9E 001DA 17$:     MOVAB   10(R3), R9                                      ; 1660
                         69              52  B0 001DE          MOVW    PSFLAGS, (R9)
                    01   A9              08  88 001E1          BISB2   #8, 1(R9)                                       ; 1661
                         50 00000000G    00  D0 001E5          MOVL    LNK$GL_CURCLU, R0                               ; 1662
          51    58   A0                  01  02 001EC          EXTZV   #2, #1, 88(R0), R1
          69         01                  0D  51 001F2          INSV    R1, #13, #1, (R9)
          69         01                  0F  6E 001F7          INSV    SGPSTYPE, #15, #1, (R9)                         ; 1664
          4D 00000000G 00                  02  E1 001FC          BBC     #2, LNK$GL_CTLMSK, 18$                          ; 1666
          45 00000000' EF                  02  E0 00204          BBS     #2, OBMODESC+20, 18$                            ; 1667
                         50                69  3C 0020C          MOVZWL  (R9), R0                                        ; 1668
                         50 FFFFFFCB       8F  CA 0020F          BICL2   #-53, R0
                         34                50  D1 00216          CMPL    R0, #52                                         ; 1670
                                           36  12 00219          BNEQ    18$
                             00000000G     00  D6 0021B          INCL    LNK$GL_PSHRNUM                                  ; 1673
                    00010000 8F 00000000G  00  D1 00221          CMPL    LNK$GL_PSHRNUM, #65536                          ; 1675
                                           23  12 0022C          BNEQ    18$
                             00000000G     8F  DD 0022E          PUSHL   #LIN$_FORMAT                                    ; 1677
                7E 00000000G 00          14  C1 00234          ADDL3   #20, LNK$GL_CURFIL, -(SP)
                             00000000'   EF  9F 0023C          PUSHAB  OBMODESC+40
                                         02  DD 00242          PUSHL   #2
                             00000000G   8F  DD 00244          PUSHL   #LIN$_EXCPSC
                    00000000G 00         05  FB 0024A          CALLS   #5, LIB$STOP
          0C 00000000G 00                  03  E1 00251 18$:     BBC     #3, LNK$GL_CTLMSK+1, 19$                    ; 1682
          04 00000000G 00                  06  E1 00259          BBC     #6, LNK$GL_CTLMSK+1, 19$                    ; 1683
                   01  A9                  10  88 00261          BISB2   #16, 1(R9)                                      ; 1685
          0F 00000000' EF                  02  E1 00265 19$:     BBC     #2, OBMODESC+20, 21$                        ; 1687
                         07                6E  E9 0026D          BLBC    SGPSTYPE, 20$                                   ; 1692
                    18   A3      08      A8  D0 00270          MOVL    8(R8), 24(R3)
                                         05  11 00275          BRB     21$
                    18   A3      04      A8  D0 00277 20$:     MOVL    4(R8), 24(R3)                                   ; 1694
                         35      04      AE  E9 0027C 21$:     BLBC    PREVDFOUND, 23$                                 ; 1707
                         50      10      AE  D0 00280          MOVL    DELPSCDESC, R0                                  ; 1710
                    18   AE      0C      A0  D0 00284          MOVL    12(R0), MODPSCONTRIBLK
                         53      14      AE  D0 00289          MOVL    PSCTDESC, R3                                    ; 1711
```

```
                    0C   A3        18   AE  D0 0028D          MOVL    MODPSCONTRIBLK, 12(R3)                          ;
                    10   A3        10   A0  D0 00292          MOVL    16(R0), 16(R3)                                  ; 1712
                         52        18   AE  D0 00297 22$:     MOVL    MODPSCONTRIBLK, R2                              ; 1714
                                   18   13  00299             BEQL    23$                                             ;
                                   04   A2  DD 0029D          PUSHL   4(R2)                                           ; 1717
                         7E        11   A2  3C 002A0          MOVZWL  17(R2), -(SP)                                   ; 1716
                       FC66        CF       02 FB 002A4      CALLS   #2, FNDPSCMAPENT                                ;
                         5A             50  D0 002A9          MOVL    R0, PSCTMAPENT                                  ;
                         6A             53  D0 002AC          MOVL    R3, (PSCTMAPENT)                                ; 1718
                         18   AE        62  D0 002AF          MOVL    (R2), MODPSCONTRIBLK                            ; 1719
                                        E2  11 002B3          BRB     22$                                             ; 1714
                                   18   AE  9F 002B5 23$:     PUSHAB  MODPSCONTRIBLK                                  ; 173*
                                        13  DD 002B8          PUSHL   #19                                             ;
                    00000000G        00       02 FB 002BA      CALLS   #2, LNK$ALLOBLK                                 ;
                         57        18   AE  D0 002C1          MOVL    MODPSCONTRIBLK, R7                              ; 1732
              13    00             6E       00 2C 002C5      MOVC5   #0, (SP), #0, #19, (R7)                         ;
                                             67 002CA          ;                                                       ;
                    04   A7 00000000'   EF  D0 002CB          MOVL    OBMODESC, 4(R7)                                 ; 1733
                00000000'   EF             57  D0 002D3      MOVL    R7, OBMODESC                                    ; 1734
                         53        14   AE  D0 002DA          MOVL    PSCTDESC, R3                                    ; 1735
                         50        10   A3  D0 002DE          MOVL    16(R3), LASTMODPSCONTRIBLK                      ;
                                        03  13 002E2          BEQL    24$                                             ; 1737
                         60             57  D0 002E4          MOVL    R7, (LASTMODPSCONTRIBLK)                        ; 1739
                    10   A3             57  D0 002E7 24$:     MOVL    R7, 16(R3)                                      ; 1741
                  FFFF   8F 00000000'   EF  B1 002EB          CMPW    OBMODESC+18, #65535                             ; 1744
                                        2A  12 002F4          BNEQ    25$                                             ;
                         23 00000000'   EF  E8 002F6          BLBS    OBMODESC+20, 25$                                ; 1745
                            00000000G   8F  DD 002FD          PUSHL   #LIN$_FORMAT                                    ; 1748
                7E 00000000G        00       14  C1 00303      ADDL3   #20, LNK$GL_CURFIL, -(SP)                       ;
                            00000000'   EF  9F 0030B          PUSHAB  OBMODESC+40                                     ;
                                        02  DD 00311          PUSHL   #2                                              ;
                            00000000G   8F  DD 00313          PUSHL   #LIN$_EXCPSC                                    ;
                    00000000G   00             05  FB 00319      CALLS   #5, LIB$STOP                                    ;
                    00000000'   EF             01  8A 00320 25$:     BICB2   #1, OBMODESC+20                         ; 1751
                                   00000000'   EF  B6 00327          INCW    OBMODESC+18                             ; 1752
                         11    A7 00000000'   EF  B0 0032D          MOVW    OBMODESC+18, 17(R7)                     ; 1753
                               7E 00000000'   EF  3C 00335          MOVZWL  OBMODESC+18, -(SP)                      ; 1754
                       FBCE    CF             01  FB 0033C      CALLS   #1, FNDPSCMAPENT                                ;
                               5A             50  D0 00341          MOVL    R0, PSCTMAPENT                          ;
                               54        04   AA  D0 00344          MOVL    4(PSCTMAPENT), PREMSYMLST                       ; 1755
                               6A             53  D0 00348          MOVL    R3, (PSCTMAPENT)                        ; 1756
                         04    AA             57  D0 0034B          MOVL    R7, 4(PSCTMAPENT)                       ; 1757
                         10    A7             6B  90 0034F          MOVB    (R11), 16(R7)                           ; 1758
                                        08   A7  D4 00353          CLRL    8(R7)                                   ; 1759
                    03                  69       03  E0 00356      BBS     #3, (R9), 26$                                   ; 1762
                                             012B  31 0035A          BRW     33$                                     ;
                    31                  69       02  E0 0035D 26$:     BBS     #2, (R9), 27$                           ; 1766
                    29 00000000'        EF       02  E0 00361      BBS     #2, OBMODESC+20, 27$                            ; 1767
                    52                  01       6B  78 00369      ASHL    (R11), #1, R2                           ; 1770
                                                 52  D7 0036D      DECL    AROUND                                  ;
                    50                  52   18  A3  C1 0036F      ADDL3   24(R3), AROUND, R0                      ; 1771
              08    A7                  50       52  CB 00374      BICL3   AROUND, R0, 8(R7)                       ; 1773
                    52             08   A7   18  A3  C3 00379      SUBL3   24(R3), 8(R7), AROUND                   ; 1774
                    50                  52   18  A3  C1 0037F      ADDL3   24(R3), AROUND, R0                      ; 1775
                              18    A3   04 B840  9E 00384      MOVAB   @4(R8)[R0], 24(R3)                      ; 1776
                              1C    A3   18  A3  D0 0038A      MOVL    24(R3), 28(R3)                          ; 1777
                                        00E1  31 0038F          BRW     32$                                     ; 1766
```

```
                        71            6E  E9 00392 27$:    BLBC    SGPSTYPE, 28$                       ; 1786
                    6D  69            02  E1 00395         BBC     #2, (R9), 28$                       ; 1787
                        69        04  AE  E9 00399         BLBC    PREVDFOUND, 28$
                        50        10  AE  D0 0039D         MOVL    DELPSCDESC, R0                      ; 1788
                    04  A8        1C  A0  D1 003A1         CMPL    28(R0), 4(R8)
                                      5E  1B 003A6         BLEQU   28$
                                  1C  AE  9F 003A8         PUSHAB  OMD                                 ; 1792
                                FC33  CF  9F 003AB         PUSHAB  LNK$COMPARE_OMD
                        55        18  AE  D0 003AF         MOVL    DELPSCDESC, R5
                                  28  A5  DD 003B3         PUSHL   40(R5)
                             00000000' EF  9F 003B6         PUSHAB  LNK$GL_OMDTREE
                 00000000G   00           04  FB 003BC     CALLS   #4, LIB$LOOKUP_TREE
                        50        1C  AE  D0 003C3         MOVL    OMD, R0                             ; 1794
                    1C  AE        0A  A0  D0 003C7         MOVL    10(R0), OMD
                        50        1C  AE  D0 003CC         MOVL    OMD, R0                             ; 1795
                        5A        04  A0  D0 003D0         MOVL    4(R0), FDB
                             00000000G  8F  DD 003D4         PUSHL   #LIN$_NOIMGFIL                      ; 1797
                7E 00000000G   00           14  C1 003DA     ADDL3   #20, LNK$GL_CURFIL, -(SP)
                                  04  A8  DD 003E2         PUSHL   4(R8)
                                  14  AA  9F 003E5         PUSHAB  20(FDB)
                                  28  A0  9F 003E8         PUSHAB  40(R0)                              ; 1796
                                  1C  A5  DD 003EB         PUSHL   28(R5)                              ; 1797
                                      56  DD 003EE         PUSHL   PSECTNAME
                                      06  DD 003F0         PUSHL   #6
                             00000000G  8F  DD 003F2         PUSHL   #LIN$_SHRPSCLNG
                 00000000G   00           09  FB 003F8     CALLS   #9, LIB$SIGNAL
                 00000000G   00           01  8A 003FF     BICB2   #1, LNK$GL_CTLMSK                   ; 1799
                    50      69            0D  E1 00406 28$:  BBC     #13, (R9), 29$                      ; 1802
                                          69  B5 0040A     TSTW    (R9)
                                          4C  18 0040C     BGEQ    29$
                48        69            02  E1 0040E     BBC     #2, (R9), 29$                       ; 1803
                        44        08  AE  E8 00412         BLBS    NEWPSECT, 29$
                    1C  A3        04  A8  D1 00416         CMPL    4(R8), 28(R3)                       ; 1804
                                      3D  1B 0041B         BLEQU   29$
                        5A        24  A3  D0 0041D         MOVL    36(R3), FDB                         ; 1808
                        5A        08  AA  D0 00421         MOVL    8(FDB), FDB                         ; 1809
                             00000000G  8F  DD 00425         PUSHL   #LIN$_NOIMGFIL                      ; 1812
                                  14  AA  9F 0042B         PUSHAB  20(FDB)
                                  1C  A3  DD 0042E         PUSHL   28(R3)
                7E 00000000G   00           14  C1 00431     ADDL3   #20, LNK$GL_CURFIL, -(SP)           ; 1811
                             00000000' EF  9F 00439         PUSHAB  OBMODESC+40
                                  04  A8  DD 0043F         PUSHL   4(R8)                               ; 1812
                                      56  DD 00442         PUSHL   PSECTNAME
                                      06  DD 00444         PUSHL   #6
                             00000000G  8F  DD 00446         PUSHL   #LIN$_SHRPSCLNG
                 00000000G   00           09  FB 0044C     CALLS   #9, LIB$SIGNAL
                 00000000G   00           01  8A 00453     BICB2   #1, LNK$GL_CTLMSK                   ; 1813
                09 00000000'   EF        02  E0 0045A 29$:  BBS     #2, OBMODESC+20, 30$                ; 1816
                        1C  A3        04  A8  D1 00462         CMPL    4(R8), 28(R3)                       ; 1820
                                          0A  1B 00467         BLEQU   32$
                                          03  11 00469         BRB     31$                                 ; 1822
                        05                6E  E9 0046B 30$:  BLBC    SGPSTYPE, 32$                       ; 1828
                        1C  A3        04  A8  D0 0046E 31$:  MOVL    4(R8), 28(R3)                       ; 1830
                0D 00000000'   EF        02  E0 00473 32$:  BBS     #2, OBMODESC+20, 33$                ; 1836
                            0C  A7    04  A8  D0 00478         MOVL    4(R8), 12(R7)                       ; 1839
                     00000000'  EF    0C  A7  C0 00480         ADDL2   12(R7), OBMODESC+8                  ; 1841
                        50                54  D0 00488 33$:  MOVL    PREMSYMLST, PREMSYM                 ; 1853
```

```
                                       18  13 0048B          BEQL    35$                          ;
          04       0A   A0             03  E1 0048D          BBC     #3, 10(PREMSYM), 34$         ; 1856
                        60       08    A7  C0 00492          ADDL2   8(R7), (PREMSYM)             ; 1859
                        54       04    A0  D0 00496 34$:     MOVL    4(PREMSYM), PREMSYMLST       ; 1861
                   04   A0       14    A3  D0 0049A          MOVL    20(R3), 4(PREMSYM)           ; 1862
                   14   A3             50  D0 0049F          MOVL    PREMSYM, 20(R3)              ; 1863
                                       E3  11 004A3          BRB     33$                          ; 1853
                        10             6E  E9 004A5 35$:     BLBC    SGPSTYPE, 36$                ; 1870
          50            58             58  C3 004A8          SUBL3   R8, R8, R0                   ; 1872
                        51             66  9A 004AC          MOVZBL  (PSECTNAME), R1              ;
                        50             51  C0 004AF          ADDL2   R1, R0                       ;
                        52       0D    A0  9E 004B2          MOVAB   13(R0), LENGTH               ;
                                       0D  11 004B6          BRB     37$                          ;
                        58             58  C2 004B8 36$:     SUBL2   R8, R8                       ; 1874
                        50             66  9A 004BB          MOVZBL  (PSECTNAME), R0              ;
                        58             50  C0 004BE          ADDL2   R0, R8                       ;
                        52       09    A8  9E 004C1          MOVAB   9(R8), LENGTH                ;
              00000000' EF             52  A0 004C5 37$:     ADDW2   LENGTH, GSDOFFSET            ; 1876
                        50             01  D0 004CC          MOVL    #1, R0                       ; 1877
                                           04 004CF          RET                                  ;
                                       50  D4 004D0 38$:     CLRL    R0                           ; 1878
                                           04 004D2          RET                                  ;

; Routine Size:  1235 bytes,    Routine Base:  $CODE$ + 069B
```

```
;  1396          1879  1 routine crefilter (definition, weakflag, altomd) =
;  1397          1880  2     begin
;  1398          1881  2 !
;  1399          1882  2 !       THIS ROUTINE FILTERS FROM THE CROSS REFERENCE THE SYMBOLS
;  1400          1883  2 !       OF THE DEBUGGER AND ALSO MASKS THEM FROM APPEARING IN ANY
;  1401          1884  2 !       OTHER SYMBOL TABLE OUTPUT. (STB FILE OR GST OF IMAGE)
;  1402          1885  2 !
;  1403          1886  2 !       IT ALSO DECIDES IF ANY CROSS REFERENCE IS TO BE DONE
;  1404          1887  2 !       AT ALL.
;  1405          1888  2 !
;  1406          1889  2 !       THE ARGUMENT "DEFINITION" HAS VALUE TRUE OR FALSE.
;  1407          1890  2 !
;  1408          1891  2     map
;  1409          1892  2         altomd : ref block [, byte];
;  1410          1893  2     builtin
;  1411          1894  2         nullparameter;
;  1412          1895  2     bind
;  1413          1896  3         objmodesc = (if nullparameter (3) then obmodesc ! IF NOT SPECIFIED, THEN USE THE LOCAL ONE
;  1414          1897  2             else .altomd) : block [, byte];     ! IF SPECIFIED, USE IT
;  1415          1898  2
;  1416          1899  2     if .symtabent [sym$v_lclsym]                ! NO CROSS REF OF MODULE LOCAL SYMBOLS
;  1417          1900  2     then
;  1418          1901  2         return true;
;  1419          1902  2
;  1420          1903  2     if .lnk$gl_curfil [fdb$v_debuger]           ! IF THIS FILE CONTAINS THE DEBUGGER
;  1421          1904  2     then
;  1422          1905  3         begin
;  1423          1906  3
;  1424          1907  3         if .definition                          ! AND IF THIS IS A DEFINITION
;  1425          1908  3         then
;  1426          1909  3             symtabent [sym$v_supres] = true;    ! SUPPRESS IT FROM STB OUTPUT
;  1427          1910  3
;  1428          1911  3         if .lnk$gl_ctlmsk [lnk$v_supdbg]        ! IF ALSO SUPRESSING
;  1429          1912  3         then
;  1430          1913  3             return true;                        ! DEBUGGER SYMBOLS FROM MAP
;  1431          1914  3
;  1432          1915  2         end;                                    ! THAT IS ALL
;  1433          1916  2
;  1434          1917  2     if .lnk$gl_ctlmsk [lnk$v_brief]             ! IF A BRIEF MAP THEN
;  1435          1918  2     then
;  1436          1919  2         return true;                            ! ALSO ALL DONE
;  1437          1920  2
;  1438          1921  2     if .lnk$gl_ctlmsk [lnk$v_intfil]            ! IF THIS IS THE INTERNAL (SYSLIB) FILE
;  1439          1922  2         and .lnk$gl_ctlmsk [lnk$v_supsys]       ! AND ITS CONTENTS ARE SUPPRESSED
;  1440          1923  2     then
;  1441          1924  2
;  1442          1925  2         if not .lnk$gl_ctlmsk [lnk$v_cros]      ! IF NO CROSS REFERENCE
;  1443          1926  2             or not .definition                  ! OR THIS IS A REFERENCE
;  1444          1927  2             or .symtabent [sym$v_supres]        ! OR THE SYMBOL HAS BEEN SUPPRESSED
;  1445          1928  2         then
;  1446          1929  2             return true                         ! THEN ALL DONE
;  1447          1930  2         else
;  1448          1931  2             objmodesc [omd$v_mapmod] = true;    ! BUT IF SYMBOL IS ENTERED MAKE SURE MODULE GETS MAPPED
;  1449          1932  2
;  1450          1933  3     if (.definition or .lnk$gl_ctlmsk [lnk$v_cros])     ! IF WE ARE GOING TO CREF THIS SYMBOL
;  1451          1934  2     then
;  1452          1935  3         begin
```

```
;  1453          1936  3
;  1454          1937  3          if .symentnam [snb$b_namlng] gtr .lnk$gl_maxsymsz
;  1455          1938  3                                                  ! IF THIS SYMBOL NAME IS BIGGER THAN BIGGEST SEEN
;  1456          1939  3          then
;  1457          1940  3              lnk$gl_maxsymsz = .symentnam [snb$b_namlng];        !  THEN MAKE THIS THE BIGGEST
;  1458          1941  3
;  1459          1942  3          if .objmodesc [omd$b_namlng] gtr .lnk$gl_maxmodsz       ! ALSO CHECK THE MODULE NAME LENGTH
;  1460          1943  3          then
;  1461          1944  3              lnk$gl_maxmodsz = .objmodesc [omd$b_namlng];
;  1462          1945  3
;  1463          1946  3          end;
;  1464          1947  3
;  1465          1948  2      if .definition                                  ! IF THIS IS A DEFINITION MUST
;  1466          1949  2      then
;  1467          1950  3          begin
;  1468          1951  3          crf$insrtkey (lnk$al_sytblfmt, symentnam [snb$b_namlng],        ! AT LEAST ENTER THE VALUE
;  1469          1952  3              symtabent [sym$l_value], .symtabent [sym$w_flags]);
;  1470          1953  3                                                  ! BY ADDRESS AND PASSING THE FLAGS AS VALUE FLAGS
;  1471          1954  3          symtabent [sym$v_crosref] = true;       ! FLAG SYMBOL HAS BEEN CROSS REFERENCED IF DEFINITION
;  1472          1955  3          end;
;  1473          1956  2
;  1474          1957  2      if .lnk$gl_ctlmsk [lnk$v_cros]              ! IF A CROSS REFERENCE IS BEING PRODUCED
;  1475          1958  2      then
;  1476          1959  3          begin
;  1477          1960  3          crf$insrtref (lnk$al_sytblfmt, symentnam [snb$b_namlng],        ! THEN INSERT THIS REFERENCE
;  1478          1961  3              objmodesc [omd$b_namlng], .weakflag, .definition);
;  1479          1962  3          lnk$gw_ncrosrfs = .lnk$gw_ncrosrfs + 1; ! AND COUNT IT
;  1480          1963  3          end;
;  1481          1964  2
;  1482          1965  2      return true
;  1483          1966  1      end;


                                        00FC 00000 CREFILTER:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1879
                                57 00000000G   00  9E 00002         MOVAB   LNK$AL_SYTBLFMT, R7     ;
                                56 00000000G   00  9E 00009         MOVAB   LNK$GL_MAXMODSZ, R6     ;
                                55 00000000G   00  9E 00010         MOVAB   LNK$GL_MAXSYMSZ, R5     ;
                                54 00000000'   EF  9E 00017         MOVAB   SYMENTNAM, R4           ;
                                53 00000000G   00  9E 0001E         MOVAB   LNK$GL_CTLMSK, R3       ;
                                03             6C  91 00025         CMPB    (AP), #3                ; 1896
                                               05  1F 00028         BLSSU   1$                      ;
                                         0C    AC  D5 0002A         TSTL    12(AP)                  ;
                                               09  12 0002D         BNEQ    2$                      ;
                                50       24    A4  9E 0002F 1$:     MOVAB   OBMODESC, R0            ;
                                52             50  D0 00033         MOVL    R0, R2                  ;
                                               04  11 00036         BRB     3$                      ;
                                52       0C    AC  D0 00038 2$:     MOVL    ALTOMD, R2              ; 1897
                                51       FC    A4  D0 0003C 3$:     MOVL    SYMTABENT, R1           ; 1899
                                03       0B    A1  E9 00040         BLBC    11(R1), 5$              ;
                                            009B  31 00044 4$:     BRW     13$                     ;
                                50 00000000G   00  D0 00047 5$:     MOVL    LNK$GL_CURFIL, R0       ; 1903
                          0D          0A    A0  05  E1 0004E         BBC     #5, 10(R0), 7$          ;
                                04       04    AC  E9 00053         BLBC    DEFINITION, 6$          ; 1907
```

```
                    0B    A1          20  88 00057       BISB2   #32, 11(R1)                              ; 1909
                              01      A3  95 0005B 6$:   TSTB    LNK$GL_CTLMSK+1                          ; 1911
                                      E4  19 0005E       BLSS    4$                                       ;
              7D    01    A3          01  E0 00060 7$:   BBS     #1, LNK$GL_CTLMSK+1, 13$                 ; 1917
              16    01    A3          03  E1 00065       BBC     #3, LNK$GL_CTLMSK+1, 8$                  ; 1921
              11    01    A3          06  E1 0006A       BBC     #6, LNK$GL_CTLMSK+1, 8$                  ; 1922
                                      63  95 0006F       TSTB    LNK$GL_CTLMSK                            ; 1925
                                      6F  18 00071       BGEQ    13$                                      ;
                          6B  04      AC  E9 00073       BLBC    DEFINITION, 13$                          ; 1926
              66    0B    A1          05  E0 00077       BBS     #5, 11(R1), 13$                          ; 1927
                    14    A2          10  88 0007C       BISB2   #16, 20(R2)                              ; 1931
                          04  04      AC  E8 00080 8$:   BLBS    DEFINITION, 9$                           ; 1933
                                      63  95 00084       TSTB    LNK$GL_CTLMSK                            ;
                                      1B  18 00086       BGEQ    11$                                      ;
                          50          64  D0 00088 9$:   MOVL    SYMENTNAM, R0                            ; 1937
  65      04      A0      08          00  ED 0008B       CMPZV   #0, #8, 4(R0), LNK$GL_MAXSYMSZ           ;
                                      04  15 00091       BLEQ    10$                                      ;
                          65  04      A0  9A 00093       MOVZBL  4(R0), LNK$GL_MAXSYMSZ                   ; 1940
  66      28      A2      08          00  ED 00097 10$:  CMPZV   #0, #8, 40(R2), LNK$GL_MAXMODSZ          ; 1942
                                      04  15 0009D       BLEQ    11$                                      ;
                          66  28      A2  9A 0009F       MOVZBL  40(R2), LNK$GL_MAXMODSZ                  ; 1944
                          1B  04      AC  E9 000A3 11$:  BLBC    DEFINITION, 12$                          ; 1948
                          7E  0A      A1  3C 000A7       MOVZWL  10(R1), -(SP)                            ; 1952
                                      51  DD 000AB       PUSHL   R1                                       ;
              7E          64          04  C1 000AD       ADDL3   #4, SYMENTNAM, -(SP)                     ; 1951
                                      57  DD 000B1       PUSHL   R7                                       ;
                   00000000G  00      04  FB 000B3       CALLS   #4, CRF$INSRTKEY                         ; 1952
                          50  FC      A4  D0 000BA       MOVL    SYMTABENT, R0                            ; 1954
                    0C    A0          02  88 000BE       BISB2   #2, 12(R0)                               ;
                                      63  95 000C2 12$:  TSTB    LNK$GL_CTLMSK                            ; 1957
                                      1C  18 000C4       BGEQ    13$                                      ;
                              04      AC  DD 000C6       PUSHL   DEFINITION                               ; 1961
                              08      AC  DD 000C9       PUSHL   WEAKFLAG                                 ;
                              28      A2  9F 000CC       PUSHAB  40(R2)                                   ;
              7E          64          04  C1 000CF       ADDL3   #4, SYMENTNAM, -(SP)                     ; 1960
                                      57  DD 000D3       PUSHL   R7                                       ;
                   00000000G  00      05  FB 000D5       CALLS   #5, CRF$INSRTREF                         ; 1961
                              00000000'   EF  B6 000DC   INCW    LNK$GW_NCROSRFS                          ; 1962
                          50          01  D0 000E2 13$:  MOVL    #1, R0                                   ; 1965
                                          04 000E5       RET                                              ; 1966

; Routine Size:  230 bytes,    Routine Base:  $CODE$ + 0B6E
```

```
;  1485          1967  1 routine prosymbol (entmskflg  =
;  1486          1968  1 !++
;  1487          1969  1 !   This routine does all the work of processing symbols on PASS 1, including
;  1488          1970  1 !   entry points, simple symbol definitions and references and procedure names.
;  1489          1971  1 !   the following operations are performed:
;  1490          1972  1 !
;  1491          1973  1 !     1. If the current object module is a selective search module, then only
;  1492          1974  1 !        definitions are considered. these are ignored unless there is a current
;  1493          1975  1 !        outstanding reference to that symbol.
;  1494          1976  1 !
;  1495          1977  1 !     2. If a definition already exists in the table and this incoming symbol is
;  1496          1978  1 !        a definition, produce an error unless they are equal, absolute symbols
;  1497          1979  1 !        or unless this is a selective search module.
;  1498          1980  1 !
;  1499          1981  1 !     3. If there is no symbol in the table, insert this one.
;  1500          1982  1 !
;  1501          1983  1 !     4. If the entry found is a reference and the in coming symbol is a
;  1502          1984  1 !        definition, the flags, data type and value are copied into the entry
;  1503          1985  1 !        after removing it from the undefined list.
;  1504          1986  1 !
;  1505          1987  1 !     5. If the p-section specification has been seen, the base of this module's
;  1506          1988  1 !        contribution to the p-section is added, and the symbol linked on to the
;  1507          1989  1 !        p-section symbol list.
;  1508          1990  1 !
;  1509          1991  1 !     6. If p-section has not been seen, find the mapping table entry and link
;  1510          1992  1 !        this symbol on to its list of prematurely defined symbols. The symbol
;  1511          1993  1 !        will be relocated when p-section is defined.
;  1512          1994  1 !
;  1513          1995  1 !     7. Undefined symbols are linked on to the undefined symbol list and a count
;  1514          1996  1 !        of them is kept if the references to them are not weak.
;  1515          1997  1 !
;  1516          1998  1 !
;  1517          1999  1 !   ENTMSKFLG =  1 if the symbol contains an entry mask,
;  1518          2000  1 !                0 otherwise.
;  1519          2001  1 !
;  1520          2002  1 !
;  1521          2003  1 !--
;  1522          2004  1 !
;  1523          2005  2 begin
;  1524          2006  2     routine shrimgsym =
;  1525          2007  3         begin
;  1526          2008  3         !
;  1527          2009  3         !   THIS ROUTINE LINKS A SYMBOL TABLE ENTRY INTO THE LIST OF REFERENCED SYMBOLS
;  1528          2010  3         !   IN A SHAREABLE IMAGE.  THE SYMBOL IS COUNTED ALSO.
;  1529          2011  3         !
;  1530          2012  3         bind    shrimgclu = .symtabent [sym$l_cludsc] : block [, byte];
;  1531          2013  3
;  1532          2014  3         if .shrimgclu [clu$v_based] or .lnk$gl_defclu [clu$v_based]
;  1533          2015  3         or  not .symtabent [sym$v_rel]
;  1534          2016  3         then return true;
;  1535          2017  3
;  1536          2018  3         if .shrimgclu [clu$l_shrsyms] eql 0                             ! IF NO SYMBOLS YET THIS CLUSTER
;  1537          2019  3         then lnk$gl_shrimgs = .lnk$gl_shrimgs + 1;                      ! THEN COUNT THIS SHAREABLE IMAGE
;  1538          2020  3
;  1539          2021  3         shrimgclu [clu$l_shrsyms] = .shrimgclu [clu$l_shrsyms] + 1;     ! COUNT THIS SYMBOL IN CLUSTER
;  1540          2022  3         lnk$gl_shrsyms            = .lnk$gl_shrsyms + 1;                ! COUNT IN TOTAL NUMBER OF SYMBOLS
;  1541          2023  3         symtabent [sym$l_shrlnk]  = .shrimgclu [clu$l_shrlst];          ! LINK SYMBOL INTO LIST OFF CLUSTER
```

```
;  1542          2024  3          symtabent [sym$v_gref]    =  true;                              ! FLAG SYMBOL HAS BEEN ENTERED INTO
;  1543          2025  3          shrimgclu [clu$l_shrlst]  = .symtabent;
;  1544          2026  3          return true
;  1545          2027  2          end;


                                                0000 00000 SHRIMGSYM:
                                                           .WORD    Save nothing                                    ; 2006
                          51 00000000'  EF   D0 00002      MOVL     SYMTABENT, R1                                   ; 2012
                          50       20   A1   D0 00009      MOVL     32(R1), R0                                      ;
                          2E       58   A0   E8 0000D      BLBS     88(R0), 2$                                      ; 2014
                          27 00000000G  00   E8 00011      BLBS     LNK$GL_DEFCLU+88, 2$                            ;
                    22        0A   A1        E3 00018      BBC      #3, 10(R1), 2$                                  ; 2015
                               2C   A0   D5 0001D          TSTL     44(R0)                                          ; 2018
                                    06   12 00020          BNEQ     1$                                              ;
                          00000000G  00   D6 00022          INCL     LNK$GL_SHRIMGS                                  ; 2019
                               2C   A0   D6 00028  1$:      INCL     44(R0)                                          ; 2021
                          00000000G  00   D6 0002B          INCL     LNK$GL_SHRSYMS                                  ; 2022
                    1C    A1        30   A0   D0 00031      MOVL     48(R0), 28(R1)                                  ; 2023
                    0B    A1        40   8F   88 00036      BISB2    #64, 11(R1)                                     ; 2024
                    30    A0             51   D0 0003B      MOVL     R1, 48(R0)                                      ; 2025
                          50             01   D0 0003F  2$:  MOVL     #1, R0                                          ; 2026
                                         04   00042          RET                                                      ; 2027

; Routine Size:  67 bytes,    Routine Base:  $CODE$ + 0C54

;  1546          2028  2 !
;  1547          2029  2 ! MAIN BODY OF PROSYMBOL
;  1548          2030  2 !
;  1549          2031  2 local
;  1550          2032  2          found,                                                  ! TRUE IF FOUND IN TABLE
;  1551          2033  2          symbolvalue,                                            ! VALUE OF SYMBOL FROM OBJECT FILE
;  1552          2034  2          symbolcluster    : ref block [, byte],                  ! POINTER TO CLUSTER DESCRIPTOR
;  1553          2035  2          nxtsyment        : ref block [, byte],                  ! USED AS POINTER IN SCAN OF UNDEFINED LIST
;  1554          2036  2          psctmapent       : ref block [, byte],                  ! POINTER TO MAPPING TABLE ENTRY
;  1555          2037  2          psctdesc         : ref block [, byte],                  ! POINTER TO P-SECT DESCRIPTION
;  1556          2038  2          modpscontriblk   : ref block [, byte];                  ! POINTER TO THIS MODULE'S
;  1557          2039  2                                                                  ! CONTRIBUTION BLOCK
;  1558          2040  2 bind
;  1559          2041  2          symbolrec = objvec [.gsdoffset] : block [, byte];
;  1560          2042  2 builtin
;  1561          2043  2          insque,
;  1562          2044  2          remque;
;  1563          2045  2
;  1564          2046  2 symtabent = 0;
;  1565          2047  2
;  1566          2048  2 !
;  1567          2049  2 !    FIRST CHECK VALID SYMBOL NAME
;  1568          2050  2 !
;  1569          2051  2
;  1570          2052  2 if .symbolstring [0] gtru sym$c_maxlng                            ! IF THE SYMBOL LENGTH IS OUTSIDE
;  1571          2053  2 or .symbolstring [0] eql 0                                       ! LEGAL RANGE
;  1572          2054  3 then begin
```

```
;1573    2055  3      signal (lin$_illnamelen, 0, lnk$gt_symstring                    ! ISSUE AN ERROR MESSAGE
;1574    2056  3             ,symbolstring [0], .symbolstring [0]
;1575    2057  3             ,sym$c_maxlng, obmodesc [omd$b_namlng]
;1576    2058  3             ,lnk$gl_curfil [fdb$q_filename]
;1577    2059  3             );
;1578    2060  3      return false;
;1579    2061  2      end;
;1580    2062  2
;1581    2063  3 symbolvalue =  (if .symbolrec [gsy$v_def]
;1582    2064  4                then (if not .lclsymgsd
;1583    2065  5                     then (if .wordpsectgsd                          ! EXTRACT THE SYMBOL VALUE FROM THE RECORD
;1584    2066  5                           then .symbolrec [sdfw$l_value]
;1585    2067  5                           else .symbolrec [sdf$l_value]
;1586    2068  5                          )
;1587    2069  4                     else .symbolrec [lsdf$l_value]
;1588    2070  4                    )
;1589    2071  3                else 0
;1590    2072  2                ) ;
;1591    2073  2
;1592    2074  4 if not (found = (if .lclsymgsd                                       ! Test for local symbol...
;1593    2075  4                 then lnk$searchlocal (symbolstring [0], .symbolrec [lsy$w_envindx], symtabent, symentnam)
;1594    2076  4                 else lnk$search (symbolstring [0], symtabent, symentnam)
;1595    2077  3          )      )
;1596    2078  2 then if  .obmodesc [omd$v_selser] and .symbolrec [gsy$v_def]     ! IF SELECTIVE SEARCH MODULE, AND THIS
;1597    2079  2      then return true                                            ! IS A DEFINITION, IGNORE IT
;1598    2080  3      else begin                                                  ! ELSE, NOT SEL SEARCH OR IT'S A REFERENCE
;1599    2081  3           lnk$insert (symbolstring [0], symtabent, symentnam);   ! INSERT IT
;1600    2082  3           if  .lclsymgsd                                         ! AND COUNT IT
;1601    2083  3           then lnk$gw_nlsyms   = .lnk$gw_nlsyms + 1
;1602    2084  3           else lnk$gw_nsymbols = .lnk$gw_nsymbols + 1;
;1603    2085  3
;1604    2086  3           symtabent [sym$v_weak]   = .symbolrec [gsy$v_weak];    ! COPY THE WEAK, RELOCATABLE
;1605    2087  3           symtabent [sym$v_rel]    = .symbolrec [gsy$v_rel];     ! (OMIT DEF FOR NOW)
;1606    2088  3           symtabent [sym$v_lclsym] = .lclsymgsd;                 ! SET FLAG IF LOCAL SYMBOL
;1607    2089  3
;1608    2090  3           if  .lnk$gl_ctlmsk [lnk$v_intfil]                      ! IF THIS IS THE INTERNAL (SYSTEM
;1609    2091  3                and
;1610    2092  3               .lnk$gl_ctlmsk [lnk$v_supsys]                    ! LIBARY FILE) AND THE DSUPPRESSION
;1611    2093  3           or  .symtabent [sym$v_lclsym]                        ! OR IT'S A MODULE-LOCAL SYMBOL
;1612    2094  3           then symtabent [sym$v_supres] = true;                ! THEN SUPRESS THIS SYMBOL
;1613    2095  3
;1614    2096  3           if not .symbolrec [gsy$v_def]                        ! IF IT WAS A REFERENCE
;1615    2097  4           then  begin                                          ! JUST INSERTED IN TABLE
;1616    2098  4                 lnk$insudfsym (.symtabent);                    ! INSERT IN UNDEFINED SYMBOL LIST
;1617    2099  4                 if  not .symtabent [sym$v_weak]                ! IF THE REFERENCE IS NOT WEAK REFERENCE
;1618    2100  4                 and not .symtabent [sym$v_lclsym]              !   AND NOT A LOCAL SYMBOL
;1619    2101  5                 then begin
;1620    2102  5                      if .lnk$gl_libsym neq 0
;1621    2103  6                      then begin
;1622    2104  6                           !
;1623    2105  6                           !   IF THE CURRENT FILE IS A SEARCH LIBRARY AND THE MODULE IS PULLED
;1624    2106  6                           !   AND THE LIST HAS EMPTIED DURING THE PROCESSING OF THE MODULE OR
;1625    2107  6                           !   FOR THE FILE SO THAT THE LIBRARY SEARCH WILL RETURN TO THE TOP OF
;1626    2108  6                           !   UNDEFINED LIST ONE MORE TIME.
;1627    2109  6                           !
;1628    2110  6                           bind   libsymnam = .lnk$gl_libsym - .lnk$gl_libsym [sym$b_namlng]
;1629    2111  6                                              - sn$$c_fxdlen          : block [, byte];
```

```
; 1630          2112  6
; 1631          2113  6                         if  .lnk$gl_curfil [fdb$v_libsrch]
; 1632          2114  6                             and
; 1633          2115  7                            (.lnk$gl_libsym eql lnk$gl_udflst
; 1634          2116  7                             or
; 1635          2117  7                             ch$lss (.symentnam [snb$b_namlng], symentnam [snb$t_name]
; 1636          2118  7                                    ,.libsymnam [snb$b_namlng], libsymnam [snb$t_name]
; 1637          2119  7                            )       )
; 1638          2120  6                         then lnk$gl_curfil [fdb$v_newudf] = true;
; 1639          2121  5                         end;
; 1640          2122  4                end;
; 1641          2123  4
; 1642          2124  4                crefilter (false, .symbolrec [gsy$v_weak]);
; 1643          2125  4                return true;                                    ! AND WE ARE DONE WITH THIS
; 1644          2126  3                end;                                            ! UNRESOLVED REFERENCE SO RETURN SUCCESS
; 1645          2127  2           end;
; 1646          2128  2
; 1647          2129  2 if .symtabent [sym$v_def]                                     ! IF THE ENTRY IN
; 1648          2130  3 then begin                                                    ! THE SYMBOL TABLE IS
; 1649          2131  3      if .obmodesc [omd$v_selser]                              ! A DEFINITION AND
; 1650          2132  3      then return true                                         ! THIS IS SELECTIVE
; 1651          2133  3      else if not .symbolrec [gsy$v_def]                       ! SEARCH, IGNORE SYMBOL.
; 1652          2134  4           then  begin                                  ! IGNORE REFERENCE TO DEFINED SYMBOL EXCEPT FOR CREF
; 1653          2135  4                 if      .symtabent [sym$v_shrimg]      ! IF SYMBOL IS FROM SHAREABLE IMAGE
; 1654          2136  4                 and not .symtabent [sym$v_crosref]     !  AND HAS NOT BEEN CROSS-REF'D YET
; 1655          2137  5                 then begin
; 1656          2138  5                      local      defomdptr : ref block [, byte];
; 1657          2139  5
; 1658          2140  5                      if not .symtabent [sym$v_gref]            ! ENTER SYMBOL INTO SHR LIST IF NOT DONE YET
; 1659          2141  5                      then shrimgsym ();
; 1660          2142  5
; 1661          2143  5                      lib$lookup_tree ( lnk$gl_omdtree          ! FIND OMD FOR MODULE THIS SYMBOL DEFINED IN
; 1662          2144  5                                      ,.symtabent [sym$l_omdnum]
; 1663          2145  5                                      , lnk$compare_omd
; 1664          2146  5                                      , defomdptr
; 1665          2147  5                                      );
; 1666          2148  5                      crefilter ( true, .symtabent [sym$v_weak]  ! CROSS REF THE DEFINITION
; 1667          2149  5                                ,.defomdptr [node$l_ptr]
; 1668          2150  5                                );
; 1669          2151  5
; 1670          2152  5                      if      .lnk$gl_ctlmsk [lnk$v_long]       ! IF A LONG FORM MAP IS REQUIRED
; 1671          2153  5                      and not .symtabent [sym$v_rel]            ! AND THIS SYMBOL IS BASED
; 1672          2154  5                      and not .symtabent [sym$v_supres]         !  AND NOT SUPRESSED
; 1673          2155  5                      then crf$insrtref  ( lnk$al_valctltb                ! INSERT THIS
; 1674          2156  5                                         , symtabent [sym$l_value]        ! SYMBOL AS A REFERENCE TO ITS
; 1675          2157  5                                         , symentnam [snb$b_namlng]       ! VALUE
; 1676          2158  5                                         ,.symtabent [sym$w_flags]
; 1677          2159  5                                         , 0
; 1678          2160  5                                         );
; 1679          2161  4                 end;
; 1680          2162  4
; 1681          2163  4           crefilter (false, .symbolrec [gsy$v_weak]);
; 1682          2164  4
; 1683          2165  4           if      .symtabent [sym$v_shrimg]                 ! IF SYMBOL IS FROM SHAREABLE IMAGE
; 1684          2166  4           and not .symtabent [sym$v_gref]                   !  AND HAS NOT BEEN LINKED INTO SHR LIST
; 1685          2167  4           then shrimgsym ();                                ! THEN DO SO NOW
; 1686          2168  4
```

```
;  1687          2169  4                 return true;                                          ! AND ALL DONE
;  1688          2170  3                 end;
;  1689          2171  3
;  1690          2172  3         !
;  1691          2173  3         !   HERE WE HAVE A MULTIPLE DEFINITION FROM A NON SELECTIVE SEARCH MODULE.
;  1692          2174  3         !   IF BOTH DEFINITIONS ARE ABSOLUTE AND THE VALUES ARE EQUAL ALL IS WELL.
;  1693          2175  3         !   OTHERWISE WE HAVE AN ERROR.
;  1694          2176  3         !
;  1695          2177  3
;  1696          2178  3         if      .symbolvalue eql .symtabent [sym$l_value]
;  1697          2179  3         and not .symbolrec [gsy$v_rel]
;  1698          2180  3         and not .symtabent [sym$v_rel]
;  1699          2181  3         then return true;
;  1700          2182  3
;  1701          2183  3         if      .symtabent [sym$v_optsym]                             ! IF DEFINED BY OPTION
;  1702          2184  3         and not .symbolrec [gsy$v_rel]                                ! AND 2ND DEF. IS ABSOLUTE THEN WE WON'T RED
;  1703          2185  4         then begin                                                    ! THE SYMBOL, BUT WE NEED TO INDICATE WHICH
;  1704          2186  5              psctmapent = fndpscmapent ((if not .lclsymgsd           ! OWNS IT
;  1705          2187  6                                         then (if .wordpsectgsd                    ! FIND PSECT MAPPING TABLE E
;  1706          2188  6                                               then .symbolrec [sdfw$w_psindx]
;  1707          2189  6                                               else .symbolrec [sdf$b_psindx]
;  1708          2190  6                                              )
;  1709          2191  5                                          else .symbolrec [lsdf$w_psindx]
;  1710          2192  4                                        ));
;  1711          2193  4              psctdesc = .psctmapent [pmt$l_pscdes];                  ! GET ADDRESS OF PSECT DESCRIPTOR
;  1712          2194  4
;  1713          2195  4              if .psctdesc neq 0                                      ! IF P-SECTION IS
;  1714          2196  5              then  begin                                             !  DEFINED, GET THIS
;  1715          2197  5                    symtabent [sym$l_psclst] = .psctdesc [psc$l_symlst];  !  P-SECT BASE
;  1716          2198  5                    psctdesc [psc$l_symlst]  = .symtabent;          !  LINK SYMBOL
;  1717          2199  5                    end                                             !  LIST
;  1718          2200  5              else  begin                                           ! P-SECTION NOT
;  1719          2201  5                    symtabent [sym$l_psclst]  = .psctmapent [pmt$l_symlst];  !  DEFINED YET, SO
;  1720          2202  5                    psctmapent [pmt$l_symlst] = .symtabent;         !  LINK SYMBOL
;  1721          2203  4                    end;                                            !  MAPPING TABLE
;  1722          2204  4
;  1723          2205  4              return true;                                          ! NOW WE CAN LEAVE QUIETLY...
;  1724          2206  3              end;
;  1725          2207  3
;  1726          2208  3         if  .symtabent [sym$v_shrimg]                              ! IF SYMBOL FROM SHAREABLE IMAGE
;  1727          2209  4         and (if (symbolcluster = .symtabent [sym$l_cludsc]) eql 0  !  AND THE IMAGE IS BASED
;  1728          2210  4              then  false
;  1729          2211  4              else .symbolcluster [clu$v_based]                     ! AND THIS IS A BASED
;  1730          2212  4             )
;  1731          2213  3         and .lnk$gl_curclu [clu$v_shrimg]                          ! SHARED IMAGE
;  1732          2214  3         and .lnk$gl_curclu [clu$v_based]                           !  WHICH IS BASED
;  1733          2215  4         and (.symtabent [sym$l_value] + .symbolcluster [clu$l_base]  !  AND THE VALUES ARE THE SAME
;  1734          2216  4                         eql
;  1735          2217  4               .symbolvalue + .lnk$gl_curclu [clu$l_base]
;  1736          2218  4             )
;  1737          2219  3         then  return true;                                         ! THEN IGNORE THIS DEFINITION COMPLE
;  1738          2220  3
;  1739          2221  3         signal (lin$_muldef, 3, symbolstring [0], obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  1740          2222  3
;  1741          2223  3         if   not .symbolrec [gsy$v_rel]                            ! IF BOTH ARE ABSOLUTE
;  1742          2224  3         and  not .symtabent [sym$v_rel]                            ! THEN WE SIMPLY RE-DEFINE
;  1743          2225  3         then symtabent [sym$l_value] = .symbolvalue;               ! THE VALUE
```

```
;  1744          2226  3
;  1745          2227  3      return true;
;  1746          2228  3      end
;  1747          2229  2  else
;  1748          2230  2      !
;  1749          2231  2      !   THE ENTRY IN THE SYMBOL TABLE IS EITHER A NEW REFERENCE
;  1750          2232  2      !   JUST INSERTED OR A REFERENCE AWAITING DEFINITION.
;  1751          2233  2      !
;  1752          2234  2      !   IF THIS IS AN INCOMING DEFINITION WE PROCEED TO DEFINE THE
;  1753          2235  2      !   SYMBOL. THAT IS:
;  1754          2236  2      !                       IF UDFLINK FIELD IS NON ZERO, THE TABLE ENTRY
;  1755          2237  2      !                       IS ON THE UNDEFINED SYMBOL LIST, SO REMOVE IT
;  1756          2238  2      !                       AND DECREMENT THE UNDEFINED COUNT (PROVIDED IT IS
;  1757          2239  2      !                       NOT A WEAK REFERENCE).
;  1758          2240  2      !
;  1759          2241  2      !                       THEN COPY INTO THE SYMBOL TABLE ENTRY THE FLAGS,
;  1760          2242  2      !                       VALUE, ENTRY MASK, DATA TYPE AND FINALLY
;  1761          2243  2      !                       LINK THE SYMBOL ON THE OWNING P-SECTION LIST
;  1762          2244  2      !
;  1763          2245  2      !   IF THIS IS (ANOTHER) INCOMING REFERENCE, AND IT IS
;  1764          2246  2      !   NOT A WEAK REFERENCE, ENSURE THAT THE SYMBOL TABLE ENTRY IS
;  1765          2247  2      !   ALSO NOT A WEAK REFERENCE AND THAT THIS STRONG REFERENCE
;  1766          2248  2      !   TO AN UNDEFINED SYMBOL IS COUNTED.
;  1767          2249  2      !
;  1768          2250  3      begin
;  1769          2251  3      if not .symbolrec [gsy$v_def]                          ! IF THIS IS
;  1770          2252  4      then begin                                             ! ANOTHER REFERENCE THEN
;  1771          2253  4           crefilter (false, .symbolrec [gsy$v_weak]);       ! CROSS REFERENCE IT
;  1772          2254  4
;  1773          2255  4           if .symbolrec [gsy$v_weak]                        ! IF WEAK, IGNORE IT SINCE THE ONE WE
;  1774          2256  4           then return true;                                 ! HAVE IS AT LEAST WEAK AND NOTHING TO
;  1775          2257  4                                                             ! BE DONE
;  1776          2258  4           if .symtabent [sym$v_weak]                        ! IF A STRONG REFERENCE AND SYMBOL TABLE
;  1777          2259  5           then   begin                                      ! ENTRY IS WEAK
;  1778          2260  5                  symtabent [sym$v_weak] = false;            ! MAKE SURE IT IS STRONG
;  1779          2261  5
;  1780          2262  5                  if .symtabent [sym$v_lclsym]
;  1781          2263  5                  then lnk$gw_nudflsyms = .lnk$gw_nudflsyms + 1   ! AND COUNT THE STRONG REFERENCE
;  1782          2264  5                  else lnk$gw_nudfsyms  = .lnk$gw_nudfsyms + 1;
;  1783          2265  4                  end;
;  1784          2266  4           return true;                                      ! AND THAT'S IT FOR REFERENCES.
;  1785          2267  3           end;
;  1786          2268  3
;  1787          2269  3      !
;  1788          2270  3      !   AND NOW FOR A DEFINITION
;  1789          2271  3      !
;  1790          2272  3      if .symtabent [sym$l_udflink] neq 0                    ! IF THIS SYMBOL TABLE ENTRY IS ON
;  1791          2273  4      then begin                                             ! UNDEFINED SYMBOL LIST,
;  1792          2274  4           remque (symtabent [sym$l_udflink], nxtsyment);    ! REMOVE FROM THE LIST
;  1793          2275  4
;  1794          2276  4           if .nxtsyment eql .lnk$gl_libsym                  ! IF THE ONE REMOVED IS THE ONE TO SEARCH
;  1795          2277  4           then lnk$gl_libsym = .nxtsyment [sym$l_udflink];  ! FOR NEXT, MOVE TO ONE BEYOND
;  1796          2278  4
;  1797          2279  4           if not .symtabent [sym$v_weak]                    ! IF IT IS NOT A WEAK REFERENCE
;  1798          2280  4           then
;  1799          2281  4                  if .symtabent [sym$v_lclsym]
;  1800          2282  4                  then lnk$gw_nudflsyms = .lnk$gw_nudflsyms - 1
```

```
;  1801          2283  4                 else lnk$gw_nudfsyms  = .lnk$gw_nudfsyms - 1;          ! DECREMENT NUMBER UNDEFINED
;  1802          2284  3           end;
;  1803          2285  3
;  1804          2286  3           !
;  1805          2287  3           !  Allow the symbol table entry to inherit the suppress and local symbol flags, but ignore
;  1806          2288  3           !  the universal flag.  If the symbol is to be universal, then it has either been
;  1807          2289  3           !  already declared that way (via the UNIVERSAL=name option), or will be set by a
;  1808          2290  3           !  redefinition later (i.e., a transfer vector).  It is not clear what an incoming
;  1809          2291  3           !  universal symbol (ref OR def) means.
;  1810          2292  3           !
;  1811          2293  3
;  1812          2294  4           symtabent [sym$w_flags] = (.symtabent [sym$w_flags] and (sym$m_lclsym or sym$m_supres or sym$m_uni))
;  1813          2295  3                                      or
;  1814          2296  3                                     (.symbolrec [gsy$w_flags] and (not sym$m_uni));
;  1815          2297  3
;  1816          2298  3           symtabent [sym$v_entmsk] = .entmskflg;                          ! COPY THE ENTRY MASK FLAG
;  1817          2299  3
;  1818          2300  3           if .lnk$gl_ctlmsk [lnk$v_alluniv]                               ! If all globals are to be promoted
;  1819          2301  3           then symtabent[sym$v_uni] = true;                               !   to universal, do this one now
;  1820          2302  3
;  1821          2303  3           symtabent [sym$b_datyp]   = .symbolrec [gsy$b_datyp];           ! COPY DATA TYPE
;  1822          2304  3           symtabent [sym$l_value]   = .symbolvalue;                       ! COPY THE VALUE
;  1823          2305  3           symtabent [sym$w_entmsk]  = .entrymask;                         ! COPY THE MASK
;  1824          2306  3           symtabent [sym$v_shrimg]  = .obmodesc [omd$v_shrimg],           ! SET/CLEAR SHAREABLE IMAGE FLAG
;  1825          2307  4           psctmapent = fndpscmapent ((if not .lclsymgsd
;  1826          2308  5                                       then (if .wordpsectgsd              ! FIND PSECT MAPPING TABLE ENTRY
;  1827          2309  5                                             then .symbolrec [sdfw$w_psindx]
;  1828          2310  5                                             else .symbolrec [sdf$b_psindx]
;  1829          2311  5                                            )
;  1830          2312  4                                       else .symbolrec [lsdt$w_psindx]
;  1831          2313  3                                     ));
;  1832          2314  3           psctdesc                = .psctmapent [pmt$l_pscdes];       ! GET ADDRESS OF PSECT DESCRIPTOR
;  1833          2315  3           symtabent [sym$l_cludsc] = .lnk$gl_curclu;                      ! SET POINTER TO CLUSTER DESCRIPTOR
;  1834          2316  3           symtabent [sym$l_omdnum] = .lnk$gw_nmodules;                    ! SET INDEX OF DEFINING MODULE
;  1835          2317  3
;  1836          2318  3           if .symtabent [sym$v_shrimg]                                    ! COPY SHR IMAGE FLAG
;  1837          2319  4           then begin
;  1838          2320  4                symtabent [sym$v_uni] = false;                             ! CLEARING UNI IF SYMBOL FROM ANOTHER IMAGE
;  1839          2321  4
;  1840          2322  4                if .found and not .symtabent [sym$v_gref]                  ! AND HAS NOT BEEN LINKED INTO THE LIST YET
;  1841          2323  4                then shrimgsym ();                                         ! THEN DO IT NOW
;  1842          2324  4
;  1843          2325  4                                                    ! DON'T CREF SHAREABLE IMAGE SYMBOLS ON DEF.  IT WILL
;  1844          2326  4                                                    ! BE CREF'ED WHEN A REFERENCE TO IT IS ENCOUNTERED
;  1845          2327  4                end
;  1846          2328  3           else                                                            ! NOT FROM A SHR IMAGE,
;  1847          2329  3                lnk$gw_lsymbols = .lnk$gw_lsymbols + 1;                    ! COUNT ANOTHER SYMBOL THIS IMAGE
;  1848          2330  3
;  1849          2331  3           if .found                                                       ! IF SYMBOL ALREADY REFERENCED
;  1850          2332  3           or not .symtabent [sym$v_shrimg]                                !  OR NOT FROM A SHAREABLE IMAGE
;  1851          2333  4           then begin
;  1852          2334  4                crefilter (true, .symbolrec [gsy$v_weak]);                 ! SO CROSS-REFERENCE IT NOW
;  1853          2335  4
;  1854          2336  4                if      .lnk$gl_ctlmsk [lnk$v_long]                       ! IF A LONG FORM MAP IS REQUIRED
;  1855          2337  4                and not .symtabent [sym$v_rel]                            !  AND SYMBOL IS ABSOLUTE
;  1856          2338  4                and not .symtabent [sym$v_supres]                         !  AND NOT SUPRESSED
;  1857          2339  4                then
```

```
;1858        2340  4                crf$insrtref ( lnk$al_valctltb                     ! INSERT THIS
;1859        2341  4                             , symtabent [sym$l_value]             ! SYMBOL AS A REFERENCE TO ITS
;1860        2342  4                             , symentnam [snb$b_namlng]            ! VALUE
;1861        2343  4                             ,.symtabent [sym$w_flags]
;1862        2344  4                             , 0
;1863        2345  4                             );
;1864        2346  3          end;
;1865        2347  3
;1866        2348  3     if .psctdesc neq 0                                              ! IF P-SECTION IS
;1867        2349  4     then begin                                                      ! DEFINED, GET THIS MODULE'S
;1868        2350  4          modpscontriblk           = .psctmapent [pmt$l_modcon];     ! CONTRIBUTION BLOCK AND
;1869        2351  4          symtabent [sym$l_value]  = .symtabent [sym$l_value] +      ! ADD THE OFFSET FROM
;1870        2352  4                                     .modpscontriblk [mpc$l_offset];
;1871        2353  4          symtabent [sym$l_psclst] = .psctdesc [psc$l_symlst];       ! P-SECT BASE
;1872        2354  4          psctdesc [psc$l_symlst]  = .symtabent;                     ! LINK SYMBOL ON TO P-SECT LIST
;1873        2355  4          end
;1874        2356  4     else begin                                                      ! P-SECTION NOT DEFINED YET
;1875        2357  4          symtabent [sym$l_psclst]  = .psctmapent [pmt$l_symlst];    ! LINK SYMBOL ON LIST OFF
;1876        2358  4          psctmapent [pmt$l_symlst] = .symtabent;                    ! MAPPING TABLE
;1877        2359  3          end;
;1878        2360  3
;1879        2361  3     return true;
;1880        2362  2     end;                                                            ! END OF DEFINITION
;1881        2363  2
;1882        2364  1 end;                                                                ! END OF SYMBOL ROUTINE


                                        0FFC 00000 PROSYMBOL:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ;1967
                    5B 00000000'    EF  9E 00002           MOVAB   LNK$GW_NUDFLSYMS, R11                 ;
                    5A       FECA   CF  9E 00009           MOVAB   CREFILTER, R10                        ;
                    59 00000000'    EF  9E 0000E           MOVAB   SYMTABENT, R9                         ;
                    5E              04  C2 00015           SUBL2   #4, SP                                ;
                    55         EC   A9  3C 00018           MOVZWL  GSDOFFSET, R5                         ;2041
                    55         F8   A9  C0 0001C           ADDL2   OBJVEC, R5                            ;
                                    69  D4 00020           CLRL    SYMTABENT                             ;2046
                    50         FC   B9  9A 00022           MOVZBL  @SYMBOLSTRING, R0                     ;2052
                    1F              50  91 00026           CMPB    R0, #31                               ;
                                    04  1A 00029           BGTRU   1$                                    ;
                                    50  D5 0002B           TSTL    R0                                    ;2053
                                    2A  12 0002D           BNEQ    2$                                    ;
7E 00000000G        00              14  C1 0002F 1$:       ADDL3   #20, LNK$GL_CURFIL, -(SP)             ;2058
                         50     A9  9F 00037               PUSHAB  OBMODESC+40                           ;2057
                                1F  DD 0003A               PUSHL   #31                                   ;2058
                                50  DD 0003C               PUSHL   R0                                    ;
                         FC     A9  DD 0003E               PUSHL   SYMBOLSTRING                          ;
                    00000000'   EF  9F 00041               PUSHAB  LNK$GT_SYMSTRING                      ;2055
                                06  DD 00047               PUSHL   #6                                    ;2058
                    00000000G   8F  DD 00049               PUSHL   #LIN$_ILLNAMELEN                      ;
   00000000G        00              08  FB 0004F           CALLS   #8, LIB$SIGNAL                        ;
                              03D3  31  00056           BRW     54$                                     ;2060
                    57         02   A5  9E 00059 2$:       MOVAB   2(R5), R7                             ;2063
1A                  67              01  E1 0005D           BBC     #1, (R7), 5$                          ;
                    10         E5   A9  E8 00061           BLBS    LCLSYMGSD, 4$                         ;2064
```

```
                                 06        E6   A9   E9 00065          BLBC    WORDPSECTGSD, 3$                          ; 2065
                                 54        06   A5   D0 00069          MOVL    6(R5), SYMBOLVALUE                        ; 2066
                                                0E   11 0006D          BRB     6$                                        ;
                                 54        05   A5   D0 0006F 3$:     MOVL    5(R5), SYMBOLVALUE                        ; 2067
                                                08   11 00073          BRB     6$                                        ; 2065
                                 54        08   A5   D0 00075 4$:     MOVL    8(R5), SYMBOLVALUE                        ; 2069
                                                02   11 00079          BRB     6$                                        ; 2064
                                                54   D4 0007B 5$:     CLRL    SYMBOLVALUE                               ; 2063
                                 15        E5   A9   E9 0007D 6$:     BLBC    LCLSYMGSD, 7$                             ; 2074
                                                04   A9   9F 00081          PUSHAB  SYMENTNAM                         ; 2075
                                                59   DD 00084          PUSHL   R9                                        ;
                                 7E        04   A5   3C 00086          MOVZWL  4(R5), -(SP)                              ;
                                           FC   A9   DD 0008A          PUSHL   SYMBOLSTRING                              ;
                        00000000G  00           04   FB 0008D          CALLS   #4, LNK$SEARCHLOCAL                       ;
                                                0F   11 00094          BRB     8$                                        ;
                                           04   A9   9F 00096 7$:     PUSHAB  SYMENTNAM                         ; 2076
                                                59   DD 00099          PUSHL   R9                                        ;
                                           FC   A9   DD 0009B          PUSHL   SYMBOLSTRING                              ;
                        00000000G  00           03   FB 0009E          CALLS   #3, LNK$SEARCH                            ;
                                 58             50   D0 000A5 8$:     MOVL    R0, FOUND                                 ;
                                 03             58   E9 000A8          BLBC    FOUND, 9$                         ; 2074
                                              00C2   31 000AB          BRW     19$                                       ;
                     07    3C    A9              03   E1 000AE 9$:     BBC     #3, OBMODESC+20, 10$              ; 2078
                     03          67              01   E1 000B3          BBC     #1, (R7), 10$                             ;
                                              036E   31 000B7          BRW     53$                                       ;
                                           04   A9   9F 000BA 10$:     PUSHAB  SYMENTNAM                         ; 2081
                                                59   DD 000BD          PUSHL   R9                                        ;
                                           FC   A9   DD 000BF          PUSHL   SYMBOLSTRING                              ;
                              00000000G    00        03   FB 000C2          CALLS   #3, LNK$INSERT                            ;
                                 05        E5   A9   E9 000C9          BLBC    LCLSYMGSD, 11$                    ; 2082
                                           FE   AB   B6 000CD          INCW    LNK$GW_NLSYMS                     ; 2083
                                                03   11 000D0          BRB     12$                                       ;
                                           F6   AB   B6 000D2 11$:     INCW    LNK$GW_NSYMBOLS                   ; 2084
                                 51             69   D0 000D5 12$:     MOVL    SYMTABENT, R1                     ; 2086
                                 50        0A   A1   9E 000D8          MOVAB   10(R1), R0                                ;
                     60          01        00        67   F0 000DC          INSV    (R7), #0, #1, (R0)                        ;
                     52          67        01        03   EF 000E1          EXTZV   #3, #1, (R7), R2                  ; 2087
                     60          01        03        52   F0 000E6          INSV    R2, #3, #1, (R0)                          ;
                01   A0          01        00        E5   A9   F0 000EB          INSV    LCLSYMGSD, #0, #1, 1(R0)          ; 2088
                                 08 00000000G  00        03   E1 000F2          BBC     #3, LNK$GL_CTLMSK+1, 13$          ; 2090
                                 04 00000000G  00        06   E0 000FA          BBS     #6, LNK$GL_CTLMSK+1, 14$          ; 209_
                                              04   01   A0   E9 00102 13$:     BLBC    1(R0), 15$                        ; 2093
                                       01     A0             20   88 00106 14$:     BISB2   #32, 1(R0)                ; 2094
                                 62          67             01   E0 0010A 15$:     BBS     #1, (R7), 19$             ; 2096
                                                            51   DD 0010E          PUSHL   R1                        ; 2098
                                       F701   CA             01   FB 00110          CALLS   #1, LNK$INSUDFSYM                 ;
                                                            50   D0 00115          MOVL    SYMTABENT, R0             ; 2099
                                                        47   0A   A0   E8 00118          BLBS    10(R0), 17$               ;
                                                        43   0B   A0   E8 0011C          BLBS    11(R0), 17$               ; 2100
                                                        50 00000000G  00   D0 00120          MOVL    LNK$GL_LIBSYM, R0         ; 2102
                                                                   3A   13 00127          BEQL    17$                       ;
                                                        51   0F   A0   9A 00129          MOVZBL  15(R0), R1                ; 2110
                                 51                     50             51   C3 0012D          SUBL3   R1, R0, R1                ;
                                                        51             05   C2 00131          SUBL2   #5, R1            ; 2111
                                                        56 00000000G  00   D0 00134          MOVL    LNK$GL_CURFIL, R6         ; 2113
                                                                   0A   A6   95 0013B          TSTB    10(R6)                    ;
                                                                        23   18 0013E          BGEQ    17$                       ;
```

```
                                52      B6   AB  9E 00140          MOVAB   LNK$GL_UDFLST, R2                          ; 2115
                                52           50  D1 00144          CMPL    R0, R2
                                             16  13 00147          BEQL    16$
                                50           04  A9  D0 00149     MOVL    SYMENTNAM, R0                             ; 2117
                                53           04  A0  9A 0014D     MOVZBL  4(R0), R3
                                52           04  A1  9A 00151     MOVZBL  4(R1), R2                                 ; 2118
                52      00      05      A0          53  2D 00155  CMPC5   R3, 5(R0), #0, R2, 5(R1)
                                             05  A1      0015B
                                             04  1E C015D      BGEQU   17$
                        0A      A6          01  88 0015F 16$:  BISB2   #1, 10(R6)                                 ; 2120
                7E      67              01      00  EF 00163 17$:  EXTZV   #0, #1, (R7), -(SP)                      ; 2124
                                             7E  D4 00168      CLRL    -(SP)
                                6A           02  FB 0016A      CALLS   #2, CREFILTER
                                         02B8    31 0016D 18$:  BRW     53$                                       ; 2125
                                50           69  D0 00170 19$:  MOVL    SYMTABENT, R0                             ; 2129
                        03      0A      A0          01  E0 00173      BBS     #1, 10(R0), 20$
                                         0157    31 00178      BRW     34$
                        ED      3C      A9          03  E0 0017B 20$:  BBS     #3, OBMODESC+20, 18$              ; 2131
                        03              67          01  E1 00180      BBC     #1, (R7), 21$                     ; 2133
                                         0091    31 00184      BRW     25$
                        6D      0B      A0          03  E1 00187 21$:  BBC     #3, 11(R0), 23$                   ; 2135
                        68      0C      A0          01  E0 0018C      BBS     #1, 12(R0), 23$                   ; 2136
                        05      0B      A0          06  E0 00191      BBS     #6, 11(R0), 22$                   ; 2140
                                00E6    CA          00  FB 00196      CALLS   #0, SHRIMGSYM                     ; 2141
                                             5E  DD 0019B 22$:  PUSHL   SP                                        ; 2143
                                    FB0F     CA  9F 0019D      PUSHAB  LNK$COMPARE_OMD
                                50           69  D0 001A1      MOVL    SYMTABENT, R0                             ; 2144
                                        14   A0  DD 001A4      PUSHL   20(R0)
                                        EA   AB  9F 001A7      PUSHAB  LNK$GL_OMDTREE                            ; 2143
                            00000000G   00       04  FB 001AA      CALLS   #4, LIB$LOOKUP_TREE
                                50           6E  D0 001B1      MOVL    DEFOMDPTR, R0                             ; 2149
                                        0A   A0  DD 001B4      PUSHL   10(R0)
                                50           69  D0 001B7      MOVL    SYMTABENT, R0                             ; 2148
        7E      0A      A0              01      00  EF 001BA      EXTZV   #0, #1, 10(R0), -(SP)
                                             01  DD 001C0      PUSHL   #1
                                6A           03  FB 001C2      CALLS   #3, CREFILTER
                                2D  00000000G   00  E9 001C5      BLBC    LNK$GL_CTLMSK+1, 23$                      ; 2152
                                50           69  D0 001CC      MOVL    SYMTABENT, R0                             ; 2153
                        25      0A      A0          03  E0 001CF      BBS     #3, 10(R0), 23$
                                50           69  D0 001D4      MOVL    SYMTABENT, R0                             ; 2154
                        1D      0B      A0          05  E0 001D7      BBS     #5, 11(R0), 23$
                                             7E  D4 001DC      CLRL    -(SP)                                     ; 2157
                                50           69  D0 001DE      MOVL    SYMTABENT, R0                             ; 2158
                                7E      0A   A0  3C 001E1      MOVZWL  10(R0), -(SP)
                        7E      04      A9          04  C1 001E5      ADDL3   #4, SYMENTNAM, -(SP)                      ; 2157
                                             50  DD 001EA      PUSHL   R0
                                00000000G    00  9F 001EC      PUSHAB  LNK$AL_VALCTLTB                           ; 2155
                            00000000G   00       05  FB 001F2      CALLS   #5, CRFSINSRTREF                          ; 2157
        7E              67              01      00  EF 001F9 23$:  EXTZV   #0, #1, (R7), -(SP)                      ; 2163
                                             7E  D4 001FE      CLRL    -(SP)
                                6A           02  FB 00200      CALLS   #2, CREFILTER
                                50           69  D0 00203      MOVL    SYMTABENT, R0                             ; 2165
                        0A      0B      A0          03  E1 00206      BBC     #3, 11(R0), 24$
                        05      0B      A0          06  E0 0020B      BBS     #6, 11(R0), 24$                   ; 2166
                                00E6    CA          00  FB 00210      CALLS   #0, SHRIMGSYM                     ; 2167
                                         0210    31 00215 24$:  BRW     53$                                       ; 2169
                                50           69  D0 00218 25$:  MOVL    SYMTABENT, R0                             ; 2178
```

```
                    60              54  D1 0021B          CMPL    SYMBOLVALUE, (R0)                                ;
                                    09  12 0021E          BNEQ    26$                                              ;
                05  67              03  E0 00220          BBS     #3, (R7), 26$                                    ; 2179
                EC  0A  A0          03  E1 00224          BBC     #3, 10(R0), 24$                                  ; 2180
                36  0B  A0          01  E1 00229 26$:      BBC     #1, 11(R0), 32$                                  ; 2183
                32  67              03  E0 0022E          BBS     #3, (R7), 32$                                    ; 2184
                    12      E5      A9  E8 00232          BLBS    LCLSYMGSD, 29$                                   ; 2186
                    06      E6      A9  E9 00236          BLBC    WORDPSECTGSD, 27$                                ; 2187
                    50      04      A5  3C 0023A          MOVZWL  4(R5), R0                                        ; 2188
                                    04  11 0023E          BRB     28$                                              ;
                    50      04      A5  9A 00240 27$:      MOVZBL  4(R5), R0                                        ; 2189
                                    50  DD 00244 28$:      PUSHL   R0                                               ; 2187
                                    04  11 00246          BRB     30$                                              ;
                    7E      06      A5  3C 00248 29$:      MOVZWL  6(R5), -(SP)                                     ; 2191
                FA3C    CA          01  FB 0024C 30$:      CALLS   #1, FNDPSCMAPENT                                 ; 2186
                    52              50  D0 00251          MOVL    R0, PSCTMAPENT                                   ;
                    53              62  D0 00254          MOVL    (PSCTMAPENT), PSCTDESC                           ; 2193
                    50              69  D0 00257          MOVL    SYMTABENT, R0                                    ; 2197
                                    53  D5 0025A          TSTL    PSCTDESC                                         ; 2195
                                    03  13 0025C          BEQL    31$                                              ;
                                01B3    31 0025E          BRW     51$                                              ;
                                01BB    31 00261 31$:      BRW     52$                                              ;
                    50              69  D0 00264 32$:      MOVL    SYMTABENT, R0                                    ; 2208
                37  0B  A0          03  E1 00267          BBC     #3, 11(R0), 33$                                  ;
                    51      20      A0  D0 0026C          MOVL    32(R0), SYMBOLCLUSTER                            ; 2209
                                    31  13 00270          BEQL    33$                                              ;
                    2D      58      A1  E9 00272          BLBC    88(SYMBOLCLUSTER), 33$                           ; 2211
                    56 00000000G    00  D0 00276          MOVL    LNK$GL_CURCLU, R6                                ; 2213
                21  58  A6          02  E1 0027D          BBC     #2, 88(R6), 33$                                  ;
                    56 00000000G    00  D0 00282          MOVL    LNK$GL_CURCLU, R6                                ; 2214
                    16      58      A6  E9 00289          BLBC    88(R6), 33$                                      ;
                50              60      4C  A1  C1 0028D      ADDL3   76(SYMBOLCLUSTER), (R0), R0                       ; 2215
                    51 00000000G    00  D0 00292          MOVL    LNK$GL_CURCLU, R1                                ; 2217
                51              54      4C  A1  C1 00299      ADDL3   76(R1), SYMBOLVALUE, R1                           ;
                                51      50  D1 0029E          CMPL    R0, R1                                           ;
                                        56  13 002A1          BEQL    36$                                              ;
                7E 00000000G    00              14  C1 002A3 33$:      ADDL3   #20, LNK$GL_CURFIL, -(SP)                ; 2221
                                        50  A9  9F 002AB      PUSHAB  OBMODESC+40                                      ;
                                        FC  A9  DD 002AE      PUSHL   SYMBOLSTRING                                     ;
                                            03  DD 002B1      PUSHL   #3                                               ;
                                00000000G   8F  DD 002B3      PUSHL   #LIN$_MULDEF                                     ;
                    00000000G   00          05  FB 002B9      CALLS   #5, LIB$SIGNAL                                   ;
                35              67          03  E0 002C0      BBS     #3, (R7), 36$                                    ; 2223
                                50          69  D0 002C4      MOVL    SYMTABENT, R0                                    ; 2224
                2D      0A      A0          03  E0 002C7      BBS     #3, 10(R0), 36$                                  ;
                        00      B9          54  D0 002CC      MOVL    SYMBOLVALUE, @SYMTABENT                          ; 2225
                                            27  11 002D0      BRB     36$                                              ; 2250
                26              67          01  E0 002D2 34$:      BBS     #1, (R7), 37$                                ; 2251
7E              67              01          00  EF 002D6      EXTZV   #0, #1, (R7), -(SP)                              ; 2253
                                            7E  D4 002DB      CLRL    -(SP)                                            ;
                                6A          02  FB 002DD      CALLS   #2, CREFILTER                                    ;
                                16          67  E8 002E0      BLBS    (R7), 36$                                        ; 2255
                                50          69  D0 002E3      MOVL    SYMTABENT, R0                                    ; 2258
                                0F      0A  A0  E9 002E6      BLBC    10(R0), 36$                                      ;
                        0A      A0          01  8A 002EA      BICB2   #1, 10(R0)                                       ; 2260
                                04      0B  A0  E9 002EE      BLBC    11(R0), 35$                                      ; 2262
                                            6B  B6 002F2      INCW    LNK$GW_NUDFLSYMS                                 ; 2263
```

```
                                                03  11 002F4          BRB     36$                                     ;
                                             FC AB  B6 002F6 35$:     INCW    LNK$GW_NUDFSYMS                         ; 2264
                                          012C  31  31 002F9 36$:     BRW     53$                                     ; 2266
                                        50      69  D0 002FC 37$:     MOVL    SYMTABENT, R0                           ; 2272
                                                60  D5 002FF          TSTL    (R0)                                    ;
                                                25  13 00301          BEQL    40$                                     ;
                                        51      60  0F 00303          REMQUE  (R0), NXTSYMENT                         ; 2274
                                00000000G 00    51  D1 00306          CMPL    NXTSYMENT, LNK$GL_LIBSYM                ; 2276
                                                07  12 0030D          BNEQ    38$                                     ;
                                00000000G 00    61  D0 0030F          MOVL    (NXTSYMENT), LNK$GL_LIBSYM              ; 2277
                                        50      69  D0 00316 38$:     MOVL    SYMTABENT, R0                           ; 2279
                                        0B   0A A0  E8 00319          BLBS    10(R0), 40$                             ;
                                        04   0B A0  E9 0031D          BLBC    11(R0), 39$                             ; 2281
                                                6B  B7 00321          DECW    LNK$GW_NUDFLSYMS                        ; 2282
                                                03  11 00323          BRB     40$                                     ;
                                             FC AB  B7 00325 39$:     DECW    LNK$GW_NUDFSYMS                         ; 2283
                                        50      69  D0 00328 40$:     MOVL    SYMTABENT, R0                           ; 2294
                                        51   0A A0  9E 0032B          MOVAB   10(R0), R1                              ;
                                        53      61  3C 0032F          MOVZWL  (R1), R3                                ;
                                        53 FFFFDEFB 8F  CA 00332      BICL2   #-8453, R3                              ;
                                        52      67  3C 00339          MOVZWL  (R7), R2                                ; 2296
                                        52      04  CA 0033C          BICL2   #4, R2                                  ;
                                61      52      53  A9 0033F          BISW3   R3, R2, (R1)                            ;
                            61  01      0F   04 AC  F0 00343          INSV    ENTMSKFLG, #15, #1, (R1)                ; 2298
                                03 00000000G 00  02 E1 00349          BBC     #2, LNK$GL_CTLMSK+3, 41$                ; 2300
                                        61      04  88 00351          BISB2   #4, (R1)                                ; 2301
                                     0E A0   01 A5  90 00354 41$:     MOVB    1(R5), 14(R0)                           ; 2303
                                        60      54  D0 00359          MOVL    SYMBOLVALUE, (R0)                       ; 2304
                                        08 A0   EE A9  B0 0035C      MOVW    ENTRYMASK, 8(R0)                        ; 2305
                            50      3C  A9          01  02 EF 00361  EXTZV   #2, #1, OBMODESC+20, R0                 ; 2306
                            61      01              0B  50 F0 00367  INSV    R0, #11, #1, (R1)                       ;
                                                    12  E5 A9 E8 0036C BLBS    LCLSYMGSD, 44$                          ; 2307
                                                    06  E6 A9 E9 00370 BLBC    WORDPSECTGSD, 42$                       ; 2308
                                                    50  04 A5 3C 00374 MOVZWL  4(R5), R0                              ; 2309
                                                    04  11 00378      BRB     43$                                     ;
                                                    50  04 A5 9A 0037A 42$:     MOVZBL  4(R5), R0                   ; 2310
                                                    50  DD 0037E 43$:     PUSHL   R0                              ; 2308
                                                    04  11 00380      BRB     45$                                     ;
                                                    7E  06 A5 3C 00382 44$:     MOVZWL  6(R5), -(SP)                ; 2312
                                                FA3C    CA  01 FB 00386 45$:     CALLS   #1, FNDPSCMAPENT            ; 2307
                                                    52  50 D0 0038B      MOVL    R0, PSCTMAPENT                          ;
                                                    53  62 D0 0038E      MOVL    (PSCTMAPENT), PSCTDESC                  ; 2314
                                                    50  69 D0 00391      MOVL    SYMTABENT, R0                           ; 2315
                                                20  A0 00000000G 00 D0 00394  MOVL    LNK$GL_CURCLU, 32(R0)                   ;
                                                14  A0   F2 AB  3C 0039C MOVZWL  LNK$GW_NMODULES, 20(R0)             ; 2316
                                            13  0B  A0      03  E1 003A1 BBC     #3, 11(R0), 46$                         ; 2318
                                                0A  A0      04  8A 003A6 BICB2   #4, 10(R0)                              ; 2320
                                                    12      58  E9 003AA BLBC    FOUND, 48$                              ; 2322
                                            0A  0B  A0      06  E0 003AD BBS     #6, 11(R0), 47$                         ;
                                                00E6 CA     00  FB 003B2 CALLS   #0, SHRIMGSYM                           ; 2323
                                                            03  11 003B7 BRB     47$                                     ; 2318
                                                         F8 AB  B6 003B9 46$:     INCW    LNK$GW_LSYMBOLS                 ; 2329
                                                    08      58  E8 003BC 47$:     BLBS    FOUND, 49$                      ; 2331
                                                    50      69  D0 003BF 48$:     MOVL    SYMTABENT, R0                   ; 2332
                                            3E  0B  A0      03  E0 003C2 BBS     #3, 11(R0), 50$                         ;
                                7E          67          01  00  EF 003C7 49$:     EXTZV   #0, #1, (R7), -(SP)         ; 2334
                                                            01  DD 003CC      PUSHL   #1                                  ;
```

```
                        6A              02  FB 003CE           CALLS   #2, CREFILTER                                    ;
                        2D 00000000G    00  E9 003D1           BLBC    LNK$GL_CTLMSK+1, 50$                            ; 2336
                        50              69  D0 003D8           MOVL    SYMTABENT, R0                                   ; 2337
                25      0A      A0      03  E0 003DB           BBS     #3, 10(R0), 50$                                 ;
                        50              69  D0 003E0           MOVL    SYMTABENT, R0                                   ; 2338
                1D      0B      A0      05  E0 003E3           BBS     #5, 11(R0), 50$                                 ;
                                        7E  D4 003E8           CLRL    -(SP)                                           ; 2342
                        50              69  D0 003EA           MOVL    SYMTABENT, R0                                   ; 2343
                        7E      0A      A0  3C 003ED           MOVZWL  10(R0), -(SP)                                   ;
                7E      04      A9          04  C1 003F1       ADDL3   #4, SYMENTNAM, -(SP)                            ; 2342
                                            50  DD 003F6       PUSHL   R0                                              ;
                            00000000G       00  9F 003F8       PUSHAB  LNK$AL_VALCTLTB                                 ; 2340
                    00000000G   00          05  FB 003FE       CALLS   #5, CRF$INSRTREF                                ; 2342
                        50              69  D0 00405 50$:   MOVL    SYMTABENT, R0                                   ; 2351
                                        53  D5 00408           TSTL    PSCTDESC                                        ; 2348
                                        13  13 0040A           BEQL    52$                                             ;
                        51      04      A2  D0 0040C           MOVL    4(PSCTMAPENT), MODPSCONTRIBLK                   ; 2350
                        60      08      A1  C0 00410           ADDL2   8(MODPSCONTRIBLK), (R0)                         ; 2352
                04      A0      14      A3  D0 00414 51$:      MOVL    20(PSCTDESC), 4(R0)                             ; 2353
                14      A3              50  D0 00419           MOVL    R0, 20(PSCTDESC)                                ; 2354
                                        09  11 0041D           BRB     53$                                             ; 2348
                04      A0      04      A2  D0 0041F 52$:      MOVL    4(PSCTMAPENT), 4(R0)                            ; 2357
                04      A2              50  D0 00424           MOVL    R0, 4(PSCTMAPENT)                               ; 2358
                        50              01  D0 00428 53$:      MOVL    #1, R0                                          ; 2361
                                            04 0042B           RET                                                     ; 2250
                                        50  D4 0042C 54$:      CLRL    R0                                              ; 2364
                                            04 0042E           RET                                                     ;

; Routine Size:  1071 bytes,    Routine Base:  $CODE$ + 0C97

; 1883          2365  1
```

```
;  1885          2366  1 routine symbols =
;  1886          2367  2 begin
;  1887          2368  2 local
;  1888          2369  2         length;
;  1889          2370  2 bind
;  1890          2371  2         symbolrec = objvec [.gsdoffset] : block [, byte];
;  1891          2372  2
;  1892          2373  2 entrymask = 0;                                    ! ZERO THE ENTRY MASK SINCE NONE EXISTS
;  1893          2374  2
;  1894          2375  2 if not .symbolrec [gsy$v_def]                    ! IF NOT DEFINED
;  1895          2376  3 then      begin
;  1896          2377  3           length       = symbolrec [srf$t_name] - symbolrec [srf$t_start] + .symbolrec [srf$b_namlng];
;  1897          2378  3           symbolstring = symbolrec [srf$b_namlng];
;  1898          2379  3           end
;  1899          2380  2 else if .wordpsectgsd                            ! IF A WORD OF PSECT NUMBER
;  1900          2381  3      then begin
;  1901          2382  3           length       = symbolrec [sdfw$t_name] - symbolrec [sdfw$t_start] + .symbolrec [sdfw$b_namlng];
;  1902          2383  3           symbolstring = symbolrec [sdfw$b_namlng];
;  1903          2384  3           end
;  1904          2385  3      else begin
;  1905          2386  3           length       = symbolrec [sdf$t_name] - symbolrec [sdf$t_start] + .symbolrec [sdf$b_namlng];
;  1906          2387  3           symbolstring = symbolrec [sdf$b_namlng];
;  1907          2388  2           end;
;  1908          2389  2
;  1909          2390  2 if not prosymbol (0)                             ! GO PROCESS THIS SYMBOL
;  1910          2391  2 then return false;                              ! AND EXIT IF FAILURE
;  1911          2392  2
;  1912          2393  2 gsdoffset = .gsdoffset + .length;               ! UPDATE THE GSD OFFSET FOR NEXT
;  1913          2394  2 return true;
;  1914          2395  1 end;


                                  001C 00000 SYMBOLS:.WORD   Save R2,R3,R4                    ; 2366
                          54 00000000'  EF  9E 00002        MOVAB   SYMBOLSTRING, R4          ;
                          50        F0  A4  3C 00009        MOVZWL  GSDOFFSET, R0             ; 2371
                          50        FC  A4  C0 0000D        ADDL2   OBJVEC, R0                ;
                                    F2  A4  B4 00011        CLRW    ENTRYMASK                 ; 2373
                    15        02  A0      01  E0 00014        BBS     #1, 2(R0), 1$             ; 2375
                    51            50      50  C3 00019        SUBL3   R0, R0, R1                ; 2377
                                  52        04  A0  9A 0001D        MOVZBL  4(R0), R2         ;
                                  51            52  C0 00021        ADDL2   R2, R1            ;
                                  52        05  A1  9E 00024        MOVAB   5(R1), LENGTH     ;
                                  64        04  A0  9E 00028        MOVAB   4(R0), SYMBOLSTRING ; 2378
                                              2C  11 0002C        BRB     3$                  ; 2375
                                  15        EA  A4  E9 0002E 1$:    BLBC    WORDPSECTGSD, 2$    ; 2380
                    51            50      50  C3 00032        SUBL3   R0, R0, R1              ; 2382
                                  53        0A  A0  9A 00036        MOVZBL  10(R0), R3        ;
                                  51            53  C0 0003A        ADDL2   R3, R1            ;
                                  52        0B  A1  9E 0003D        MOVAB   11(R1), LENGTH    ;
                                  64        0A  A0  9E 00041        MOVAB   10(R0), SYMBOLSTRING ; 2383
                                              13  11 00045        BRB     3$                  ; 2380
                    51            50      50  C3 00047 2$:    SUBL3   R0, R0, R1              ; 2386
                                  53        09  A0  9A 0004B        MOVZBL  9(R0), R3         ;
                                  51            53  C0 0004F        ADDL2   R3, R1            ;
```

```
                                52        0A    A1  9E 00052          MOVAB   10(R1), LENGTH                          ;
                                64        09    A0  9E 00056          MOVAB   9(R0), SYMBOLSTRING                    ; 2387
                                                  7E  D4 0005A 3$:      CLRL    -(SP)                                  ; 2390
                             FB70     CF          01  FB 0005C          CALLS   #1, PROSYMBOL                          ;
                                08                50  E9 00061          BLBC    R0, 4$                                 ;
                                F0    A4          52  A0 00064          ADDW2   LENGTH, GSDOFFSET                      ; 2393
                                50                01  D0 00068          MOVL    #1, R0                                 ; 2394
                                                      04 0006B          RET                                            ;
                                                  50  D4 0006C 4$:      CLRL    R0                                     ; 2395
                                                      04 0006E          RET                                            ;

; Routine Size:  111 bytes,    Routine Base:  $CODE$ + 10C6

; 1915          2396  1
```

```
;1917     2397  1 routine entpnts =
;1918     2398  2 begin
;1919     2399  2 local
;1920     2400  2         length;
;1921     2401  2 bind
;1922     2402  2         symbolrec = objvec [.gsdoffset] : block [, byte];
;1923     2403  2
;1924     2404  2 if .wordpsectgsd
;1925     2405  3 then begin
;1926     2406  3      entrymask    = .symbolrec [epmw$w_mask];          ! EXTRACT THE ENTRY POINT MASK
;1927     2407  3      symbolstring =  symbolrec [epmw$b_namlng];        ! POINT TO THE SYMBOL
;1928     2408  3      length       =  symbolrec [epmw$t_name] - symbolrec [epmw$t_start] + .symbolrec [epmw$b_namlng];
;1929     2409  3      end
;1930     2410  3 else begin
;1931     2411  3      entrymask    = .symbolrec [epm$w_mask];           ! EXTRACT THE ENTRY POINT MASK
;1932     2412  3      symbolstring =  symbolrec [epm$b_namlng];         ! POINT TO THE SYMBOL
;1933     2413  3      length       =  symbolrec [epm$t_name] - symbolrec [epm$t_start] + .symbolrec [epm$b_namlng];
;1934     2414  2      end;
;1935     2415  2
;1936     2416  2 if not prosymbol (1)                                   ! GO PROCESS THE SYMBOL
;1937     2417  2 then return false;                                     ! AND EXIT IF FAILURE
;1938     2418  2
;1939     2419  2 gsdoffset = .gsdoffset + .length;                      ! ELSE UPDATE THE OFFSET FOR NEXT
;1940     2420  2 return true;
;1941     2421  1 end;

                                  000C 00000 ENTPNTS:.WORD   Save R2,R3                                ;2397
                53 00000000'   EF   9E 00002          MOVAB   GSDOFFSET, R3                          ;
                52             63   3C 00009          MOVZWL  GSDOFFSET, R2                          ;2402
                52         0C  A3   C0 0000C          ADDL2   OBJVEC, R2                             ;
                1B         FA  A3   E9 00010          BLBC    WORDPSECTGSD, 1$                       ;2404
          02    A3         0A  A2   B0 00014          MOVW    10(R2), ENTRYMASK                      ;2406
          10    A3         0C  A2   9E 00019          MOVAB   12(R2), SYMBOLSTRING                   ;2407
     50         52             52   C3 0001E          SUBL3   R2, R2, R0                             ;2408
                51         0C  A2   9A 00022          MOVZBL  12(R2), R1                             ;
                50             51   C0 00026          ADDL2   R1, R0                                 ;
                52         0D  A0   9E 00029          MOVAB   13(R0), LENGTH                         ;
                               19   11 0002D          BRB     2$                                     ;2404
          02    A3         09  A2   B0 0002F 1$:      MOVW    9(R2), ENTRYMASK                       ;2411
          10    A3         0B  A2   9E 00034          MOVAB   11(R2), SYMBOLSTRING                   ;2412
     50         52             52   C3 00039          SUBL3   R2, R2, R0                             ;2413
                51         0B  A2   9A 0003D          MOVZBL  11(R2), R1                             ;
                50             51   C0 00041          ADDL2   R1, R0                                 ;
                52         0C  A0   9E 00044          MOVAB   12(R0), LENGTH                         ;
                               01   DD 00048 2$:       PUSHL   #1                                     ;2416
          FB13  CF             01   FB 0004A          CALLS   #1, PROSYMBOL                          ;
                07             50   E9 0004F          BLBC    R0, 3$                                 ;
                63             52   A0 00052          ADDW2   LENGTH, GSDOFFSET                      ;2419
                50             01   D0 00055          MOVL    #1, R0                                 ;2420
                               04   00058            RET                                            ;
                               50   D4 00059 3$:       CLRL    R0                                     ;2421
                               04   0005B            RET                                            ;
```

; Routine Size: 92 bytes,   Routine Base: $CODE$ + 1135

```
;  1943          2422  1 routine procedef =
;  1944          2423  2     begin
;  1945          2424  2 !
;  1946          2425  2 !       A PROCEDURE DEFINITION IS AN EXTENDED ENTRY POINT DEFINITION, CARRYING WITH
;  1947          2426  2 !       IT A DESCRIPTION OF THE PROCEDURE'S FORMAL ARGUMENTS. PROCESSING THESE CONSISTS
;  1948          2427  2 !       IN NORMAL SYMBOL DEFINITION PROCESSING FOLLOWED BY:-
;  1949          2428  2 !               (1) VALIDATION OF THE FORMAT OF FORMAL DESCRIPTION (I.E. JUST CHECK
;  1950          2429  2 !                       THAT MINIMUM NUMBER OF ARGUMENTS SPECIFIED IS LESS THAN
;  1951          2430  2 !                       OR EQUAL TO THE MAXIMUM.
;  1952          2431  2 !               (2) ALLOCATION OF AN ARGUMENT VALIDATION DATA ARRAY WHICH
;  1953          2432  2 !                       HAS THE FOLLOWING FORMAT:
;  1954          2433  2 !                                  ------------
;  1955          2434  2 !                                  !  N+2     ! LENGTH OF ARRAY WHERE N = MAXIMUM # ARGUMENTS
;  1956          2435  2 !                                  ------------
;  1957          2436  2 !                                  !  MIN     ! MINIMUM NUMBER OF ARGS
;  1958          2437  2 !                                  ------------
;  1959          2438  2 !                                  ! PASS M 1 ! PASSING MECHANISM OF ARG 1
;  1960          2439  2 !                                  ------------
;  1961          2440  2 !                                  ! PASS M 2 !    ''        ''        '' 2
;  1962          2441  2 !                                  ------------
;  1963          2442  2 !                                     ETC TO N ARGUMENTS
;  1964          2443  2 !
;  1965          2444  2 !       NOTE THAT ONLY THE VALIDATION CONTROL BYTE, ARGUMENT PASSING MECHANISM
;  1966          2445  2 !       IS EXTRACTED FROM THE FORMAL ARGUMENT DESCRIPTORS. THE REST OF SUCH
;  1967          2446  2 !       DESCRIPTORS IS IGNORED.
;  1968          2447  2 !
;  1969          2448  2 !       THE SYMBOL TABLE ENTRY POINTS TO THIS ARRAY. NOTE THAT MULTIPLE DEFINITION OF THE
;  1970          2449  2 !       SAME PROCEDURE WILL REPLACE THE VALIDATION DATA WITH NEW DATA.
;  1971          2450  2
;  1972          2451  2     local
;  1973          2452  2         entflag,
;  1974          2453  2         argvaldata : ref vector [, byte],
;  1975          2454  2         argcount;
;  1976          2455  2
;  1977          2456  2     if not entpnts () then return false;        ! PROCESS THE ENTRY POINT SYMBOL PART
;  1978          2457  2
;  1979          2458  2     entflag = .symtabent neq 0;                 ! SEE IF ENTERED
;  1980          2459  3     begin
;  1981          2460  3
;  1982          2461  3     bind
;  1983          2462  3         formals = objvec [.gsdoffset] : block [, byte]; ! THE FORMAL DESCRIPTION
;  1984          2463  3
;  1985          2464  3     if .formals [fml$b_minargs] gtru .formals [fml$b_maxargs]   ! IF THERE IS AN INCONSISTENT
;  1986          2465  3     then
;  1987          2466  4         begin                                   ! DESCRIPTION OF FORMALS
;  1988          2467  4         signal (lin$_illfmlcnt, 5,              ! OUTPUT ERROR MESSAGE
;  1989          2468  4             .formals [fml$b_minargs], .formals [fml$b_maxargs],
;  1990          2469  4             (if .wordpsectgsd then objrec [prow$b_nam[ng] else objrec [pro$b_namlng]),
;  1991          2470  4             obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  1992          2471  4         return false;
;  1993          2472  3         end;
;  1994          2473  3
;  1995          2474  3     gsdoffset = .gsdoffset + fml$c_size;        ! UPDATE RECORD POINTER
;  1996          2475  3
;  1997          2476  3     if .entflag and (argvaldata = .symtabent [sym$l_valdata]) neq 0     ! IF VALIDATION DATA ALREADY EXISTS
;  1998          2477  3     then
;  1999          2478  4         begin                                   ! MUST BE A NEW DEFINITION
```

```
;  2000          2479  4          lnk$dealblk (.argvaldata [0], argvaldata [0]);  ! SO DISPOSE OF OLD DATA
;  2001          2480  4          symtabent [sym$l_valdata] = 0;                  ! AND FORGET IT
;  2002          2481  3          end;
;  2003          2482  3
;  2004          2483  3      if (argcount = .formals [fml$b_maxargs]) eql 0      ! IF MAXIMUM NUMBER ARGUMENTS
;  2005          2484  3      then
;  2006          2485  3          return true;                                    ! IS ZERO, ALL DONE
;  2007          2486  3
;  2008          2487  3      if .entflag
;  2009          2488  3      then
;  2010          2489  4          begin
;  2011          2490  4          lnk$alloblk (.argcount + 2, symtabent [sym$l_valdata]); ! ALLOCATE A VALIDATION DATA BLOCK
;  2012          2491  4          argvaldata = .symtabent [sym$l_valdata];
;  2013          2492  4          argvaldata [0] = .argcount + 2;                 ! SET ITS SIZE
;  2014          2493  4          argvaldata [1] = .formals [fml$b_minargs];      ! SAVE MINIMUM ARGUMENT SPECIFICATION
;  2015          2494  3          end;
;  2016          2495  3
;  2017          2496  3      incr i from 1 to .argcount                          ! BEGIN A LOOP THROUGH
;  2018          2497  3          do
;  2019          2498  4          begin                                           ! ALL FORMAL ARGUMENT DESCRIPTORS
;  2020          2499  4
;  2021          2500  4          bind
;  2022          2501  4              argdesc = objvec [.gsdoffset] : block [, byte];     ! POINT TO NEXT DESCRIPTOR
;  2023          2502  4
;  2024          2503  4          if .entflag then argvaldata [.i + 1] = .argdesc [arg$v_passmech];
;  2025          2504  4
;  2026          2505  4                                                          ! EXTRACT THE PASSING MECHANISM
;  2027          2506  4          gsdoffset = .gsdoffset + .argdesc [arg$b_bytecnt] +     ! AND UPDATE RECORD POINTER
;  2028          2507  4          arg$c_size;
;  2029          2508  3          end;
;  2030          2509  3
;  2031          2510  3      return true;
;  2032          2511  2      end;
;  2033          2512  1      end;


                                        007C 00000 PROCEDEF:
                                                   .WORD   Save R2,R3,R4,R5,R6                          ; 2422
                    56 00000000'    EF  9E 00002           MOVAB   SYMTABENT, R6                        ;
                97  AF              00  FB 00009           CALLS   #0, ENTPNTS                          ; 2456
                    4C              50  E9 0000D           BLBC    R0, 4$                               ;
                                    50  D4 00010           CLRL    R0                                   ; 2458
                                    66  D5 00012           TSTL    SYMTABENT                            ;
                                    02  13 00014           BEQL    1$                                   ;
                                    50  D6 00016           INCL    R0                                   ;
                    55              50  D0 00018 1$:       MOVL    R0, ENTFLAG                          ;
                    50        F8    A6  D0 0001B           MOVL    OBJVEC, R0                           ; 2462
                    51        EC    A6  3C 0001F           MOVZWL  GSDOFFSET, R1                        ;
54                  50              51  C1 00023           ADDL3   R1, R0, R4                           ;
                01  A4              64  91 00027           CMPB    (R4), 1(R4)                          ; 2464
                                    31  1B 0002B           BLEQU   5$                                   ;
7E 00000000G    00                  14  C1 0002D           ADDL3   #20, LNK$GL_CURFIL, -(SP)            ; 2470
                              50    A6  9F 00035           PUSHAB  OBMODESC+40                          ;
                    05        E6    A6  E9 00038           BLBC    WORDPSECTGSD, 2$                     ; 2469
```

```
                    50              0C  C0 0003C          ADDL2   #12, R0                                 ;
                                    03  11 0003F          BRB     3$                                      ;
                    50              0B  C0 00041 2$:       ADDL2   #11, R0                             ;
                                    50  DD 00044 3$:       PUSHL   R0                                  ;
                    7E      01      A4  9A 00046          MOVZBL  1(R4), -(SP)                        ; 2470
                    7E              64  9A 0004A          MOVZBL  (R4), -(SP)                         ;
                                    05  DD 0004D          PUSHL   #5                                  ;
                        00000000G   8F  DD 0004F          PUSHL   #LIN$_ILLFMLCNT                     ;
            00000000G   00          07  FB 00055          CALLS   #7, LIB$SIGNAL                      ;
                                    7C  11 0005C 4$:       BRB     12$                                 ; 2471
                  EC    A6          02  A0 0005E 5$:       ADDW2   #2, GSDOFFSET                       ; 2474
                        1B          55  E9 00062          BLBC    ENTFLAG, 6$                         ; 2476
                        50          66  D0 00065          MOVL    SYMTABENT, R0                       ;
                        53      18  A0  D0 00068          MOVL    24(R0), ARGVALDATA                  ;
                                    12  13 0006C          BEQL    6$                                  ;
                                    53  DD 0006E          PUSHL   ARGVALDATA                          ; 2479
                        7E          63  9A 00070          MOVZBL  (ARGVALDATA), -(SP)                 ;
            00000000G   00          02  FB 00073          CALLS   #2, LNK$DEALBLK                     ;
                        50          66  D0 0007A          MOVL    SYMTABENT, R0                       ; 2480
                            18      A0  D4 0007D          CLRL    24(R0)                              ;
                        52      01  A4  9A 00080 6$:       MOVZBL  1(R4), ARGCOUNT                     ; 2483
                                    50  13 00084          BEQL    11$                                 ;
                        20          55  E9 00086          BLBC    ENTFLAG, 7$                         ; 2487
                7E      66          18  C1 00089          ADDL3   #24, SYMTABENT, -(SP)               ; 2490
                            02      A2  9F 0008D          PUSHAB  2(ARGCOUNT)                         ;
            00000000G   00          02  FB 00090          CALLS   #2, LNK$ALLOBLK                     ;
                        50          66  D0 00097          MOVL    SYMTABENT, R0                       ; 2491
                        53      18  AC  D0 0009A          MOVL    24(R0), ARGVALDATA                  ;
                        50      02  A2  9E 0009E          MOVAB   2(ARGCOUNT), R0                     ; 2492
                        63          50  90 000A2          MOVB    R0, (ARGVALDATA)                    ;
                    01  A3              64  90 000A5          MOVB    (R4), 1(ARGVALDATA)                 ; 2493
                                        51  D4 000A9 7$:       CLRL    I                               ; 2501
                                        25  11 000AB          BRB     10$                             ;
                        50      EC      A6  3C 000AD 8$:       MOVZWL  GSDOFFSET, R0                   ;
                        50      F8      A6  C0 000B1          ADDL2   OBJVEC, R0                      ;
                        0A              55  E9 000B5          BLBC    ENTFLAG, 9$                     ; 2503
              54            60          02  00  EF 000B8          EXTZV   #0, #2, (R0), R4                ;
                        01 A143             54  90 000BD          MOVB    R4, 1(I)[ARGVALDATA]            ;
                        54      EC      A6  3C 000C2 9$:       MOVZWL  GSDOFFSET, R4                   ; 2506
                        50      01      A0  9A 000C6          MOVZBL  1(R0), R0                       ;
                        50              54  C0 000CA          ADDL2   R4, R0                          ;
                    EC  A6              50  02  A1 000CD          ADDW3   #2, R0, GSDOFFSET               ;
                        D7              51  52  F3 000D2 10$:      AOBLEQ  ARGCOUNT, I, 8$                 ; 2496
                        50              01  D0 000D6 11$:       MOVL    #1, R0                          ; 2510
                                            04  000D9          RET                                     ;
                                        50  D4 000DA 12$:       CLRL    R0                              ; 2512
                                            04  000DC          RET                                     ;

; Routine Size:  221 bytes,    Routine Base:  $CODE$ + 1191
```

```
;  2035          2513  1 routine proeom (seqchkflg) =
;  2036          2514  2     begin
;  2037          2515  2 !
;  2038          2516  2 !       PROCESS END OF MODULE RECORDS:
;  2039          2517  2 !               (1) VALIDATE SEQUENCE
;  2040          2518  2 !               (2) INTERPRET COMPILER COMPLETION CODE,
;  2041          2519  2 !                   ISSUING APPROPRIATE ERROR OR WARNING
;  2042          2520  2 !                   MESSAGE
;  2043          2521  2 !               (3) VALIDATE THE TRANSFER ADDRESS
;  2044          2522  2 !                       (IF ANY) FOR:
;  2045          2523  2 !                       (A) P-SECTION DEFINED
;  2046          2524  2 !                       (B) P-SECTION EXECUTABLE AND RELOCATABLE
;  2047          2525  2 !                       (C) ADDRESS IS WITHIN THAT P-SECTION
;  2048          2526  2 !                       (D) NOT A MULTIPLE SPECIFICATION -( RETAIN THE FIRST)
;  2049          2527  2 !               (4) RECORD THE  TRANSFER ADDRESS, SEPARATING AN ADDRESS IN
;  2050          2528  2 !                   THE DEBUGGER FROM THAT OF THE USER.
;  2051          2529  2 !               (5) TRUNCATE THE MODULE DESCRIPTOR (MAPPING TABLE PORTION) TO ITS
;  2052          2530  2 !                   CORRECT SIZE FOR NUMBER OF P-SECTS CONTRIBUTED TO.
;  2053          2531  2 !
;  2054          2532  2     bind
;  2055          2533  2         eomrec = .objvec : block [, byte];
;  2056          2534  2     builtin
;  2057          2535  2         nullparameter;
;  2058          2536  2     local
;  2059          2537  2         comcode,
;  2060          2538  2         deferredsym : ref block [, byte],       ! POINTER TO DEFERRED SYMBOL
;  2061          2539  2         deferredsnb : ref block [, byte],       ! DEFERRED SYMBOL NAME BLOCK
;  2062          2540  2         modpscontriblk : ref block [, byte],    ! THE MODULE'S CONTRIBUTION TO P-SECT
;  2063          2541  2         wordpsecteom,                           ! TRUE IF EOMW RATHER THAN EOM
;  2064          2542  2         psctmap : ref block [, byte],           ! PSECT MAPPING TABLE POINTER
;  2065          2543  2         psctnum,                                ! P-SECTION NUMBER OF TRANSFER ADDRESS
;  2066          2544  2         tfrweak,                                ! TRUE IF TRANSFER ADDRESS THIS EOM IS WEAK
;  2067          2545  2         tfradr,                                 ! TEMP FOR TRANSFER ADDRESS
;  2068          2546  2         tfrpsc : ref block [, byte],            ! TEMP FOR P-SECT DESC. ADDR.
;  2069          2547  2         objdesc : ref block [, byte],           ! POINTER TO ALLOCATED OBJ MOD DESCRIPTOR
;  2070          2548  2         omdnode,
;  2071          2549  2         curcontriblk : ref block [, byte],
;  2072          2550  2         nextcontriblk : ref block [, byte],
;  2073          2551  2         objdescsize;                            ! SIZE OF OBJ MOD DESCRIPTOR TO ALLOCATE
;  2074          2552  2
;  2075          2553  2     if nullparameter (1)
;  2076          2554  2     then
;  2077          2555  2
;  2078          2556  2         if not seqchk () then return false;     ! VALIDATE THE SEQUENCE OF EOM
;  2079          2557  2
;  2080          2558  2     lnk$gl_curfil [fdb$v_omdnobin] =            ! IF ANY OBMODS FOR THIS FDB ARE
;  2081          2559  2     .lnk$gl_curfil [fdb$v_omdnobin] or .obmodesc [omd$v_nobin]; ! WITHOUT TIR RECS, FLAG IN FDB
;  2082          2560  2     wordpsecteom = (.eomrec [eom$b_rectyp] eql obj$c_eomw);     ! DECIDE IF EOM OR EOMW
;  2083          2561  2
;  2084          2562  3     if (if .wordpsecteom                        ! CHECK LENGTH OF RECORD
;  2085          2563  3         then (.reclng neq eomw$c_eommin and ((.reclng lss eomw$c_eommx1) or (.reclng gtr eomw$c_eommax))) el
;  2086          2564  4         (.reclng neq eom$c_eommin                ! AND ITS LENGTH
;  2087          2565  3             and ((.reclng lss eom$c_eommx1) or (.reclng gtr eom$c_eommax))))
;  2088          2566  2     then
;  2089          2567  3         begin
;  2090          2568  3         signal (lin$_illreclen, 3,              ! AND ISSUE ERROR
;  2091          2569  3             .reclng, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
```

```
;  2092          2570  3          return false;
;  2093          2571  2          end;
;  2094          2572  2
;  2095          2573  2      if (comcode = .eomrec [eom$b_comcod]) neq 0 ! IF NON ZERO COMPILATION CPLETE CODE
;  2096          2574  2      then
;  2097          2575  3          begin                                   ! CHECK
;  2098          2576  3
;  2099          2577  3          if .comcode gtru 3
;  2100          2578  3          then
;  2101          2579  4              begin
;  2102          2580  4              signal (lin$_badccc, 3,             ! IF AN ILLEGAL COMPLETION
;  2103          2581  4                  .comcode, obmodesc [omd$b_namlng],      !  THEN TELL THAT AND QUIT
;  2104          2582  4                  lnk$gl_curfil [fdb$q_filename]);
;  2105          2583  4              return false;
;  2106          2584  3              end;
;  2107          2585  3
;  2108          2586  3          case .comcode from eom$c_warning to eom$c_abort ! SIGNAL MESSAGE BASED ON ERROR CLASS
;  2109          2587  3          of
;  2110          2588  3              set
;  2111          2589  3
;  2112          2590  3              [eom$c_warning] :
;  2113          2591  3                  signal (lin$_wrners, 2, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2114          2592  3
;  2115          2593  3              [eom$c_error] :
;  2116          2594  4                  begin
;  2117          2595  4                  signal (lin$_errors, 2, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename],
;  2118          2596  4                      lin$_noimgfil);
;  2119          2597  4                  lnk$gl_ctlmsk [lnk$v_image] = false;    ! DISABLE IMAGE IF ERROR
;  2120          2598  3                  end;
;  2121          2599  3
;  2122          2600  3              [eom$c_abort] :
;  2123          2601  4                  begin
;  2124          2602  4                  signal (lin$_eomftl, 2, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2125          2603  4                  lnk$exit (lin$_fatalerror);     ! LINK ABORT
;  2126          2604  3                  end;
;  2127          2605  3              tes;
;  2128          2606  3
;  2129          2607  2          end;
;  2130          2608  2
;  2131          2609  2      if .obmodesc [omd$w_hipsct] eql lnk$c_maxpsects     ! IF HIGHEST NUMBERED P-SECT
;  2132          2610  2          and .obmodesc [omd$v_nopsct]                    ! IS LNK$C_MAXPSECTS AND STILL NO DEFINITIONS
;  2133          2611  2      then                                            ! BEEN SEEN
;  2134          2612  2          signal_stop (lin$_nopscts, 2,               ! THERE IS A FORMAT ERROR
;  2135          2613  2              obmodesc [omd$b_namlng],                ! ISSUE MESSAGE
;  2136          2614  2              lnk$gl_curfil [fdb$q_filename], lin$_format);
;  2137          2615  2
;  2138          2616  3      if (if .wordpsecteom then (.reclng geq eomw$c_eommx1) else (.reclng geq eom$c_eommx1))
;  2139          2617  3                                                      ! IF RECORD LENGTH IS (MX1)
;  2140          2618  3                                                      ! OR (MAX)
;  2141          2619  2      then
;  2142          2620  3          begin                                       ! A TRANSFER ADDRESS EXISTS, SO
;  2143          2621  4          psctnum = (if .wordpsecteom                 ! GET THE P-SECTION INDEX AND
;  2144          2622  3          then .eomrec [eomw$w_psindx] else .eomrec [eom$b_psindx]);
;  2145          2623  3
;  2146          2624  3          if .psctnum gtru .obmodesc [omd$w_hipsct] or .obmodesc [omd$v_nopsct]
;  2147          2625  3          then
;  2148          2626  4              begin
```

```
; 2149          2627  4            signal (lin$_badpsc, 3,                 ! IF P-SECTION NOT DEFINED
; 2150          2628  4                .psctnum, obmodesc [omd$b_namlng],      ! THEN OUTPUT ERROR AND QUIT
; 2151          2629  4                lnk$gl_curfil [fdb$q_filename]);
; 2152          2630  4            return false;
; 2153          2631  4            end
; 2154          2632  3        else
; 2155          2633  4            begin
; 2156          2634  4            psctmap = fndpscmapent (.psctnum);  . GET MAPPING TABLE ENTRY
; 2157          2635  4
; 2158          2636  4            if (tfrpsc = .psctmap [pmt$l_pscdes]) eql 0 ! SET THE POINTER TO THAT P-SECTION
; 2159          2637  4            then
; 2160          2638  5                begin
; 2161          2639  5                signal (lin$_badpsc, 3,             !  BUT ISSUE AN ERROR IF UNDEFINED
; 2162          2640  5                    .psctnum, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
; 2163          2641  5                return false;
; 2164          2642  4                end;
; 2165          2643  4
; 2166          2644  4            if not .tfrpsc [psc$v_exe]              ! IF THE P-SECTION CONTAINING THE
; 2167          2645  4                or not .tfrpsc [psc$v_rel]          ! TRANSFER ADDRESS IS NOT
; 2168          2646  4            then
; 2169          2647  4                signal (lin$_pscnxr, 2,             ! THEN ISSUE AN ERROR MESSAGE
; 2170          2648  4                    obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
; 2171          2649  4
; 2172          2650  4            modpscontriblk = .psctmap [pmt$l_modcon];
; 2173          2651  4            tfradr = (if .wordpsecteom then .eomrec [eomw$l_tfradr] else .eomrec [eom$l_tfradr]) +
; 2174          2652  4                                                    ! BUT IN ANY CASE COMPUTE THE
; 2175          2653  4            .modpscontriblk [mpc$l_offset];         ! ADDRESS RELATIVE TO THE WHOLE P-SECT
; 2176          2654  6            tfrweak = (if .wordpsecteom then ((.reclng eql eomw$c_eommax) and
; 2177          2655  6                                                    ! SET FLAG FOR WEAKNESS OF TRANSFER ADDRESS
; 2178          2656  4                .eomrec [eomw$v_wktfr]) else ((.reclng eql eom$c_eommax) and .eomrec [eom$v_wktfr]));
; 2179          2657  4
; 2180          2658  4            if not .lnk$gl_curfil [fdb$v_debuger]       ! IF THE CURRENT INPUT FILE
; 2181          2659  4                                                    ! DOES NOT CONTAIN THE DEBUGGER
; 2182          2660  5                and (.lnk$gl_tfrpsc eql 0           ! AND NO PREVIOUS TRANSFER ADDRESS WAS SPECIFIED
; 2183          2661  5                or (.weaktfradr and not .tfrweak))
; 2184          2662  5                                                    ! OR IT WAS A WEAK TRANSFER ADDRESS AND THIS ONE IS NOT
; 2185          2663  4            then
; 2186          2664  5                begin
; 2187          2665  5
; 2188          2666  5                if ( not .lnk$gl_ctlmsk [lnk$v_imgidopt]) and   ! IF IMAGE ID NOT MANUALLY SET
; 2189          2667  6                    (.lnk$gt_imgid [0] eql 0        ! IF STILL NO IMAGE IDENT
; 2190          2668  6                    or .lnk$gl_tfrpsc eql 0)        !  OR IDENT WAS FROM MODULE W/O TRANSFER ADDRESS
; 2191          2669  5                then
; 2192          2670  5                    ch$move (.imageidstring [0] + 1, imageidstring [0],         ! THEN MOVE IN THE ONE OF
; 2193          2671  5                        lnk$gt_imgid [0]);          ! THIS MODULE AS SAVED ON MHD RECORD
; 2194          2672  5
; 2195          2673  5                lnk$gl_tfrpsc = .tfrpsc;            ! THEN THIS IS THE USER'S
; 2196          2674  5                lnk$gl_tfradr = .tfradr;            ! TRANSFER ADDRESS
; 2197          2675  5                weaktfradr = .tfrweak;              ! REMEMBER IF THIS WAS WEAK OR STRONG TRANSFER ADDRESS
; 2198          2676  5                end
; 2199          2677  4            else
; 2200          2678  4
; 2201          2679  4                if .lnk$gl_curfil [fdb$v_debuger]
; 2202          2680  4                                                    ! OTHERWISE IF THIS IS THE DEBUGGER FILE AND THERE HAS NOT
; 2203          2681  4                    and .lnk$gl_dbgtfps eql 0       ! PREVIOUSLY BEEN A DEBUGGER TRANSFER
; 2204          2682  4                then
; 2205          2683  5                    begin                           ! ADDRESS, MAKE THIS THE
```

```
; 2206          2684  5                   lnk$gl_dbgtfps = .tfrpsc;    ! DEBUGGER TRANSFER ADDRESS
; 2207          2685  5                   lnk$gl_dbgtfr = .tfradr;     !
; 2208          2686  5                   end                          .
; 2209          2687  4               else
; 2210          2688  4
; 2211          2689  4                   if not .tfrweak
; 2212          2690  4 !
; 2213          2691  4 ! THERE WAS A MULTIPLE TRANSFER ADDRESS AND THE TRANSFER ADDRESS FLAG
; 2214          2692  4 ! BYTE WAS NOT PRESENT, OR WAS PRESENT AND THE WEAK TRANSFER ADDRESS
; 2215          2693  4 ! FLAG WAS NOT ON.
; 2216          2694  4 !
; 2217          2695  4                   then
; 2218          2696  4                       signal (lin$_multfr, 2, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
; 2219          2697  4
; 2220          2698  3               end;
; 2221          2699  3
; 2222          2700  2          end;                                     ! END OF TRANSFER ADDRESS PROCESSING
; 2223          2701  2
; 2224          2702  2      if .lnk$gt_imgid [0] eql 0                   ! IF NO IMAGE IDENT YET
; 2225          2703  2      then
; 2226          2704  2          ch$move (.imageidstring [0] + 1, imageidstring [0],
; 2227          2705  2                                   !  THEN SET THIS MODULE'S AS THE IDENT (WILL GET OVERWRITTEN IF XFR ADR
; 2228          2706  2              lnk$gt_imgid [0]);                   !    LATER)
; 2229          2707  2
; 2230          2708  2      objdescsize = obmodesc [omd$t_pscmap] - obmodesc +  ! COMPUTE SIZE OF OBJ MOD DESC TO ALLOCATE
; 2231          2709  4      pmt$c_size*((if not .obmodesc [omd$v_p256]
; 2232          2710  4                   then .obmodesc [omd$w_hipsct] + 1
; 2233          2711  2                   else (.obmodesc [omd$w_hipsct] + 256)/256));
; 2234          2712  2      lnk$alloblk (.objdescsize, objdesc);         ! ALLOCATE THE DESCRIPTOR
; 2235          2713  2      lastobjmod [omd$l_nxtomd] = .objdesc;        ! LINK INTO THE LIST
; 2236          2714  2      lastobjmod = .objdesc;                       ! AND NEW BECOMES THE LAST THEN
; 2237          2715  2      ch$move (.objdescsize, obmodesc, .objdesc); ! COPY THE DESCRIPTOR OUT
; 2238          2716  2      lib$insert_tree (lnk$gl_omdtree, .objdesc [omd$l_omdnum],   ! INSERT OMD INTO OMD TREE
; 2239          2717  2          %ref (0), lnk$compare_omd, alloc_omdnode, omdnode, .objdesc);
; 2240          2718  2      objdesc [omd$l_nxtomd] = 0;                  ! ZERO LINK IN THE DESCRIPTOR
; 2241          2719  2
; 2242          2720  2      if .obmodesc [omd$l_lsterr] eql obmodesc [omd$l_errtxt]     ! CHECK FOR NO ERRORS
; 2243          2721  2      then
; 2244          2722  2          objdesc [omd$l_lsterr] = objdesc [omd$l_errtxt];
; 2245          2723  2
; 2246          2724  2 !
; 2247          2725  2 ! IF DEBUG RECORDS IN THIS OBJ MOD, THEN ADD COUNT OF PSECTS IN DEBUG OBJ MODS.
; 2248          2726  2 !       THEN RESET PER-OBJ-MOD PSECT COUNT, AND DEBUG FLAG.
; 2249          2727  2 !
; 2250          2728  2
; 2251          2729  2      if .omd_has_dbg
; 2252          2730  2      then
; 2253          2731  3          begin
; 2254          2732  3          lnk$gw_pscdst = .lnk$gw_pscdst + .obmodesc [omd$w_hipsct] + 1;
; 2255          2733  3          omd_has_dbg = false;
; 2256          2734  2          end;
; 2257          2735  2
; 2258          2736  2 !
; 2259          2737  2 ! FILL IN THE FIELD MPC$L_OWNOMD FOR ALL PSECTS DEFINED IN THIS MODULE
; 2260          2738  2 !
; 2261          2739  2      curcontriblk = .obmodesc [omd$l_nxtomd];     ! GET LIST POINTER OF PSECT CONTRIBUTION BLOCKS
; 2262          2740  2
```

```
;  2263          2741  2     while .curcontriblk neq 0 do
;  2264          2742  3         begin
;  2265          2743  3         nextcontriblk = .curcontriblk [mpc$l_ownomd];   ! GET LINK TO NEXT
;  2266          2744  3         curcontriblk [mpc$l_ownomd] = .objdesc; ! SET OWNER POINTER
;  2267          2745  3         curcontriblk = .nextcontriblk;
;  2268          2746  2         end;
;  2269          2747  2
;  2270          2748  2     ch$fill (0, .objdescsize, obmodesc);          ! ZERO THE PROTOTYPE DESCRIPTOR
;  2271          2749  2     obmodesc [omd$l_lsterr] = obmodesc [omd$l_errtxt];
;  2272          2750  2     maxreclng = obj$c_maxrecsiz;                  ! RESET TO MAX ALLOWED BY LANGUAGE
;  2273          2751  2     return true;
;  2274          2752  1     end;                                          ! END OF EOM PROCESSING


                                        OFFC 00000 PROEOM:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 2513
                5B 00000000G    00  9E 00002            MOVAB   LNK$GL_CURFIL, R11                          ;
                5A 00000000'    EF  9E 00009            MOVAB   LNK$GL_TFRPSC, R10                          ;
                59 00000000'    EF  9E 00010            MOVAB   OBMODESC+40, R9                             ;
                5E              0C  C2 00017            SUBL2   #12, SP                                     ;
                52        A8    A9  D0 0001A            MOVL    OBJVEC, R2                                  ; 2533
                                6C  95 0001E            TSTB    (AP)                                        ; 2553
                                05  13 00020            BEQL    1$                                          ;
                          04    AC  D5 00022            TSTL    4(AP)                                       ;
                                0B  12 00025            BNEQ    2$                                          ;
                    F04C    CF      00  FB 00027 1$:    CALLS   #0, SEQCHK                                  ; 2556
                            03      50  E8 0002C        BLBS    R0, 2$                                      ;
                                  02F0  31 0002F        BRW     46$                                         ;
                            50      6B  D0 00032 2$:    MOVL    LNK$GL_CURFIL, R0                           ; 2558
                51    0B    A0      01  02 00035        EXTZV   #2, #1, 11(R0), R1                          ; 2559
                53    EC    A9      01  01 0003B        EXTZV   #1, #1, OBMODESC+20, R3                     ;
                            51      53  88 00041        BISB2   R3, R1                                      ;
          0B    A0              01      02  51  F0 00044        INSV    R1, #2, #1, 11(R0)                  ;
                                          51  D4 0004A        CLRL    R1                                  ; 2560
                            07          62  91 0004C        CMPB    (R2), #7                            ;
                                        02  12 0004F        BNEQ    3$                                  ;
                                        51  D6 00051        INCL    R1                                  ;
                            55          51  D0 00053 3$:    MOVL    R1, WORDPSECTEOM                    ;
                            51    A4    A9  3C 00056        MOVZWL  RECLNG, R1                          ; 2563
                            0F          55  E9 0005A        BLBC    WORDPSECTEOM, 4$                    ; 2562
                            02          51  D1 0005D        CMPL    R1, #2                              ; 2563
                                        2A  13 00060        BEQL    7$                                  ;
                            08          51  B1 00062        CMPW    R1, #8                              ;
                                        14  1F 00065        BLSSU   6$                                  ;
                            09          51  B1 00067        CMPW    R1, #9                              ;
                                        0D  11 0006A        BRB     5$                                  ;
                            02          51  B1 0006C 4$:    CMPW    R1, #2                              ; 2564
                                        1B  13 0006F        BEQL    7$                                  ;
                            07          51  B1 00071        CMPW    R1, #7                              ; 2565
                                        05  1F 00074        BLSSU   6$                                  ;
                            08          51  B1 00076        CMPW    R1, #8                              ;
                                        11  1B 00079 5$:    BLEQU   7$                                  ;
                                  14    A0  9F 0007B 6$:    PUSHAB  20(R0)                              ; 2569
                                0202    8F  BB 0007E        PUSHR   #^M<R1,R9>                          ;
                                        03  DD 00082        PUSHL   #3                                  ;
```

```
                        00000000G  8F   DD 00084          PUSHL   #LIN$_ILLRECLEN                                  ;
                                   1B   11 0008A          BRB     8$                                               ;
                     53       01   A2   9A 0008C 7$:     MOVZBL  1(R2), COMCODE                                   ; 2573
                                   7D   13 00090          BEQL    14$                                              ;
                     03            53   D1 00092          CMPL    COMCODE, #3                                      ; 2577
                                   13   1B 00095          BLEQU   9$                                               ;
                  7E 6B            14   C1 00097          ADDL3   #20, LNK$GL_CURFIL, -(SP)                        ; 2582
                           0208    8F   BB 0009B          PUSHR   #^M<R3,R9>                                       ;
                                   03   DD 0009F          PUSHL   #3                                               ;
                        00000000G  8F   DD 000A1          PUSHL   #LIN$_BADCCC                                     ;
                              00D6 31   000A7 8$:         BRW     22$                                              ;
                  50 6B            14   C1 000AA 9$:     ADDL3   #20, LNK$GL_CURFIL, R0                           ; 2591
                  02       01      53   CF 000AE          CASEL   COMCODE, #1, #2                                  ; 2586
                003D    001B       0006      000B2 10$:     .WORD   11$-10$,-                                        ;
                                                                    12$-10$,-                                       ;
                                                                    13$-10$                                         ;
                                   50   DD 000B8 11$:     PUSHL   R0                                               ; 2591
                                   59   DD 000BA          PUSHL   R9                                               ;
                                   02   DD 000BC          PUSHL   #2                                               ;
                        00000000G  8F   DD 000BE          PUSHL   #LIN$_WRNERS                                     ;
                  00000000G  00         04   FB 000C4          CALLS   #4, LIB$SIGNAL                             ;
                                   42   11 000CB          BRB     14$                                              ;
                        00000000G  8F   DD 000CD 12$:     PUSHL   #LIN$_NOIMGFIL                                   ; 2595
                                   50   DD 000D3          PUSHL   R0                                               ;
                                   59   DD 000D5          PUSHL   R9                                               ;
                                   02   DD 000D7          PUSHL   #2                                               ;
                        00000000G  8F   DD 000D9          PUSHL   #LIN$_ERRORS                                     ;
                  00000000G  00         05   FB 000DF          CALLS   #5, LIB$SIGNAL                             ;
                  00000000G  00         01   8A 000E6          BICB2   #1, LNK$GL_CTLMSK                          ; 2597
                                   20   11 000ED          BRB     14$                                              ; 2586
                                   50   DD 000EF 13$:     PUSHL   R0                                               ; 2602
                                   59   DD 000F1          PUSHL   R9                                               ;
                                   02   DD 000F3          PUSHL   #2                                               ;
                        00000000G  8F   DD 000F5          PUSHL   #LIN$_EOMFTL                                     ;
                  00000000G  00         04   FB 000FB          CALLS   #4, LIB$SIGNAL                             ;
                        00000000G  8F   DD 00102          PUSHL   #LIN$_FATALERROR                                 ; 2603
                  00000000G  00         01   FB 00108          CALLS   #1, LNK$EXIT                               ;
                     FFFF    8F   EA   A9   B1 0010F 14$:     CMPW    OBMODESC+18, #65535                      ; 2609
                                   1F   12 00115          BNEQ    15$                                              ;
                           1B      EC   A9   E9 00117          BLBC    OBMODESC+20, 15$                         ; 2610
                        00000000G  8F   DD 0011B          PUSHL   #LIN$_FORMAT                                     ; 2614
                  7E 6B            14   C1 00121          ADDL3   #20, LNK$GL_CURFIL, -(SP)                        ;
                                   59   DD 00125          PUSHL   R9                                               ; 2613
                                   02   DD 00127          PUSHL   #2                                               ; 2614
                        00000000G  8F   DD 00129          PUSHL   #LIN$_NOPSCTS                                    ;
                  00000000G  00         05   FB 0012F          CALLS   #5, LIB$STOP                               ;
                                   55   E9 00136 15$:     BLBC    WORDPSECTEOM, 16$                                ; 2616
                           08      A4   A9   B1 00139          CMPW    RECLNG, #8                               ;
                                   04   11 0013D          BRB     17$                                              ;
                           07      A4   A9   B1 0013F 16$:     CMPW    RECLNG, #7                               ;
                                   03   1E 00143 17$:     BGEQU   18$                                              ;
                              0111 31   00145          BRW     38$                                              ;
                                   55   E9 00148 18$:     BLBC    WORDPSECTEOM, 19$                                ; 2621
                           02      53   A2   3C 0014B          MOVZWL  2(R2), PSCTNUM                           ; 2622
                                   04   11 0014F          BRB     20$                                              ;
                           02      53   A2   9A 00151 19$:     MOVZBL  2(R2), PSCTNUM                           ;
  53    EA    A9                 10        00   ED 00155 20$:     CMPZV   #0, #16, OBMODESC+18, PSCTNUM            ; 2624
```

```
                                         13  1F 0015B          BLSSU   21$                                 ;
                        0F         EC    A9  E8 0015D          BLBS    OBMODESC+20, 21$                    ;
                                         53  DD 00161          PUSHL   PSCTNUM                             ; 2634
                  F1D4  CF               01  FB 00163          CALLS   #1, FNDPSCMAPENT                    ;
                        54               50  D0 00168          MOVL    R0, PSCTMAP                         ;
                        56               64  D0 0016B          MOVL    (PSCTMAP), TFRPSC                   ; 2636
                                         1A  12 0016E          BNEQ    23$                                 ;
               7E       6B               14  C1 00170 21$:     ADDL3   #20, LNK$GL_CURFIL, -(SP)           ; 2640
                                 0208    8F  BB 00174          PUSHR   #^M<R3,R9>                          ;
                                         03  DD 00178          PUSHL   #3                                  ;
                             00000000G   8F  DD 0017A          PUSHL   #LIN$_BADPSC                        ;
                  00000000G 00           05  FB 00180 22$:     CALLS   #5, LIB$SIGNAL                      ;
                                       0198  31 00187          BRW     46$                                 ; 2641
               05       0A  A6           06  E1 0018A 23$:     BBC     #6, 10(TFRPSC), 24$                 ; 2644
               15       0A  A6           03  E0 0018F          BBS     #3, 10(TFRPSC), 25$                 ; 2645
               7E           6B           14  C1 00194 24$:     ADDL3   #20, LNK$GL_CURFIL, -(SP)           ; 2648
                                         59  DD 00198          PUSHL   R9                                  ;
                                         02  DD 0019A          PUSHL   #2                                  ;
                             00000000G   8F  DD 0019C          PUSHL   #LIN$_PSCNXR                        ;
                  00000000G 00           04  FB 001A2          CALLS   #4, LIB$SIGNAL                      ;
                        50         04    A4  D0 001A9 25$:     MOVL    4(PSCTMAP), MODPSCONTRIBLK          ; 2650
                        06               55  E9 001AD          BLBC    WORDPSECTEOM, 26$                   ; 2651
                        51         04    A2  D0 001B0          MOVL    4(R2), R1                           ;
                                         04  11 001B4          BRB     27$                                 ;
                        51         03    A2  D0 001B6 26$:     MOVL    3(R2), R1                           ;
               58       51         08    A0  C1 001BA 27$:     ADDL3   8(MODPSCONTRIBLK), R1, TFRADR       ; 2653
                        12               55  E9 001BF          BLBC    WORDPSECTEOM, 29$                   ; 2654
                                         50  D4 001C2          CLRL    R0                                  ;
                        09         A4    A9  B1 001C4          CMPW    RECLNG, #9                          ;
                                         02  12 001C8          BNEQ    28$                                 ;
                                         50  D6 001CA          INCL    R0                                  ;
      57       08  A2           01       00  EF 001CC 28$:     EXTZV   #0, #1, 8(R2), TFRWEAK              ; 2656
                                         10  11 001D2          BRB     31$                                 ;
                                         50  D4 001D4 29$:     CLRL    R0                                  ;
                        08         A4    A9  B1 001D6          CMPW    RECLNG, #8                          ;
                                         02  12 001DA          BNEQ    30$                                 ;
                                         50  D6 001DC          INCL    R0                                  ;
      57       07  A2           01       00  EF 001DE 30$:     EXTZV   #0, #1, 7(R2), TFRWEAK              ;
                                57       57  D2 001E4 31$:     MCOML   TFRWEAK, TFRWEAK                    ;
               57               50       57  CB 001E7          BICL3   TFRWEAK, R0, TFRWEAK                ;
                                50       6B  D0 001EB          MOVL    LNK$GL_CURFIL, R0                   ; 2658
               40       0A      A0       05  E0 001EE          BBS     #5, 10(R0), 36$                     ;
                                         6A  D5 001F3          TSTL    LNK$GL_TFRPSC                       ; 2660
                                         07  13 001F5          BEQL    32$                                 ;
                        33         94    A9  E9 001F7          BLBC    WEAKTFRADR, 35$                     ; 2661
                        30               57  E8 001FB          BLBS    TFRWEAK, 35$                        ;
               1B 00000000G 00           06  E0 001FE 32$:     BBS     #6, LNK$GL_CTLMSK+3, 34$            ; 2666
                             00000000G   00  95 00206          TSTB    LNK$GT_IMGID                        ; 2667
                                         04  13 0020C          BEQL    33$                                 ;
                                         6A  D5 0020E          TSTL    LNK$GL_TFRPSC                       ; 2668
                                         0F  12 00210          BNEQ    34$                                 ;
                        50         B8    A9  9A 00212 33$:     MOVZBL  IMAGEIDSTRING, R0                   ; 2670
                                         50  D6 00216          INCL    R0                                  ;
   00000000G 00         B8  A9           50  28 00218          MOVC3   R0, IMAGEIDSTRING, LNK$GT_IMGID     ; 2671
                        6A               56  D0 00221 34$:     MOVL    TFRPSC, LNK$GL_TFRPSC               ; 2673
                    FC  AA               58  D0 00224          MOVL    TFRADR, LNK$GL_TFRADR               ; 2674
                    94  A9               57  90 00228          MOVB    TFRWEAK, WEAKTFRADR                 ; 2675
```

```
                                            2B  11 0022C        BRB     38$                                       ; 2658
                 0F        0A   A0          05  E1 0022E 35$:   BBC     #5, 10(R0), 37$                           ; 2679
                                       F8   AA  D5 00233 36$:   TSTL    LNK$GL_DBGTFPS                            ; 2681
                                            0A  12 00236        BNEQ    37$                                       ;
                           F8   AA          56  D0 00238        MOVL    TFRPSC, LNK$GL_DBGTFPS                    ; 2684
                           F4   AA          58  D0 0023C        MOVL    TFRADR, LNK$GL_DBGTFR                     ; 2685
                                            17  11 00240        BRB     38$                                       ; 2679
                                14          57  E8 00242 37$:   BLBS    TFRWEAK, 38$                              ; 2689
                                       14   A0  9F 00245        PUSHAB  20(R0)                                    ; 2696
                                            59  DD 00248        PUSHL   R9                                        ;
                                            02  DD 0024A        PUSHL   #2                                        ;
                                 00000000G  8F  DD 0024C        PUSHL   #LIN$_MULTFR                              ;
                   00000000G    00          04  FB 00252        CALLS   #4, LIB$SIGNAL                            ;
                                 00000000G  00  95 00259 38$:   TSTB    LNK$GT_IMGID                              ; 2702
                                            0F  12 0025F        BNEQ    39$                                       ;
                                50     B8   A9  9A 00261        MOVZBL  IMAGEIDSTRING, R0                         ; 2704
                                            50  D6 00265        INCL    R0                                        ;
 00000000G  00             B8   A9          50  28 00267        MOVC3   R0, IMAGEIDSTRING, LNK$GT_IMGID           ; 2706
            08             EC   A9          06  E0 00270 39$:   BBS     #6, OBMODESC+20, 40$                      ; 2709
                                56     EA   A9  3C 00275        MOVZWL  OBMODESC+18, R6                           ; 2710
                                            56  D6 00279        INCL    R6                                        ;
                                            10  11 0027B        BRB     41$                                       ;
                                56     EA   A9  3C 0027D 40$:   MOVZWL  OBMODESC+18, R6                           ; 2711
                                56   0100   C6  9E 00281        MOVAB   256(R6), R6                               ;
                                56 00000100 8F  C6 00286        DIVL2   #256, R6                                  ;
                                56          08  C4 0028D 41$:   MULL2   #8, OBJDESCSIZE                           ; 2708
                                56     48   A6  9E 00290        MOVAB   72(R6), OBJDESCSIZE                       ;
                                       04   AE  9F 00294        PUSHAB  OBJDESC                                   ; 2712
                                            56  DD 00297        PUSHL   OBJDESCSIZE                               ;
                   00000000G    00          02  FB 00299        CALLS   #2, LNK$ALLOBLK                           ;
                                57     04   AE  D0 002A0        MOVL    OBJDESC, R7                               ; 2713
                           90   B9          57  D0 002A4        MOVL    R7, @LASTOBJMOD                           ;
                           90   A9          57  D0 002A8        MOVL    R7, LASTOBJMOD                            ; 2714
                 67        D8   A9          56  28 002AC        MOVC3   OBJDESCSIZE, OBMODESC, (R7)               ; 2715
                                            57  DD 002B1        PUSHL   R7                                        ; 2717
                                       0C   AE  9F 002B3        PUSHAB  OMDNODE                                   ; 2716
                                     F135   CF  9F 002B6        PUSHAB  ALLOC_OMDNODE                             ;
                                     F151   CF  9F 002BA        PUSHAB  LNK$COMPARE_OMD                           ;
                                       10   AE  D4 002BE        CLRL    16(SP)                                    ; 2717
                                       10   AE  9F 002C1        PUSHAB  16(SP)                                    ;
                                       1C   A7  DD 002C4        PUSHL   28(R7)                                    ; 2716
                                       08   AA  9F 002C7        PUSHAB  LNK$GL_OMDTREE                            ;
                   00000000G    00          07  FB 002CA        CALLS   #7, LIB$INSERT_TREE                       ;
                                            67  D4 002D1        CLRL    (R7)                                      ; 2718
                                50     F8   A9  9E 002D3        MOVAB   OBMODESC+32, R0                           ; 2720
                                50     FC   A9  D1 002D7        CMPL    OBMODESC+36, R0                           ;
                                            05  12 002DB        BNEQ    42$                                       ;
                           24   A7     20   A7  9E 002DD        MOVAB   32(R7), 36(R7)                            ; 2722
                                13     8C   A9  E9 002E2 42$:   BLBC    OMD_HAS_DBG, 43$                          ; 2729
                                50     D0   AA  3C 002E6        MOVZWL  LNK$GW_PSCDST, R0                         ; 2732
                                51     EA   A9  3C 002EA        MOVZWL  OBMODESC+18, R1                           ;
                                50          51  C0 002EE        ADDL2   R1, R0                                    ;
       D0   AA                  50          01  A1 002F1        ADDW3   #1, R0, LNK$GW_PSCDST                     ;
                                       8C   A9  94 002F6        CLRB    OMD_HAS_DBG                               ; 2733
                                50     D8   A9  D0 002F9 43$:   MOVL    OBMODESC, CURCONTRIBLK                    ; 2739
                                            0D  13 002FD 44$:   BEQL    45$                                       ; 2741
                                51     04   A0  D0 002FF        MOVL    4(CURCONTRIBLK), NEXTCONTRIBLK            ; 2743
```

```
                        04   A0          57  D0 00303          MOVL    R7, 4(CURCONTRIBLK)                    ; 2744
                             50          51  D0 00307          MOVL    NEXTCONTRIBLK, CURCONTRIBLK            ; 2745
                                         F1  11 0030A          BRB     44$                                    ; 2741
              56        00               00  2C 0030C 45$:     MOVC5   #0, (SP), #0, OBJDESCSIZE, OBMODESC    ; 2748
                                    D8   A9      00311
                        FC   A9     F8   A9  9E 00313          MOVAB   OBMODESC+32, OBMODESC+36               ; 2749
                        A0   A9   0800   8F  B0 00318          MOVW    #2048, MAXRECLNG                       ; 2750
                             50          01  D0 0031E          MOVL    #1, R0                                 ; 2751
                                             04 00321          RET
                                         50  D4 00322 46$:     CLRL    R0                                     ; 2752
                                             04 00324          RET

; Routine Size:  805 bytes,    Routine Base:  $CODE$ + 126E
```

```
;  2276          2753  1 routine countdbg =
;  2277          2754  2     begin
;  2278          2755  2 !
;  2279          2756  2 !       ROUTINE TO COUNT THE NUMBER OF DEBUG (TYPES 4 AND 5) RECORDS SEEN AND TO
;  2280          2757  2 !       TOTAL THE NUMBER OF BYTES CONTAINED FOR USE AS AN ESTIMATE OF THE
;  2281          2758  2 !       SIZE OF DEBUG SYMBOL TABLE TO BE WRITTEN AT END OF BINARY
;  2282          2759  2 !       IF THE DEBUGGER IS LINKED IN. ALSO IF AN EXECUTABLE IMAGE
;  2283          2760  2 !       WITH THE DEBUGGER IS BEING PRODUCED, TURN OFF THE NO BINARY FLAG FOR
;  2284          2761  2 !       THIS MODULE (PROVIDED IT IS NOT THE DEBUGGER ITSELF).
;  2285          2762  2 !
;  2286          2763  2
;  2287          2764  2     if not seqchk ()                              ! CHECK LEGAL SEQUENCE
;  2288          2765  2     then
;  2289          2766  2         return false;
;  2290          2767  2
;  2291          2768  2     if .obmodesc [omd$v_debuger]                  ! IF THIS IS THE DEBUGER ITSELF
;  2292          2769  2     then
;  2293          2770  2         return true;                              ! IGNORE ANY DEBUG RECORDS
;  2294          2771  2
;  2295          2772  2     if not .omd_has_dbg and .lnk$gl_ctlmsk [lnk$v_dbg]  ! IF DBG REC FLAG IS CLEAR AND THE
;  2296          2773  2     then                                          !  DEBUGGER IS BEING INCLUDED
;  2297          2774  3         begin                                     !  IN THE IMAGE,
;  2298          2775  3         omd_has_dbg = true;                       !  THEN SET FLAG NOW AND
;  2299          2776  3         lnk$gl_omddst = .lnk$gl_omddst + 1;       ! ADD TO COUNT OF DBG OBJ MODS
;  2300          2777  2         end;
;  2301          2778  2
;  2302          2779  2     lnk$gl_dbgestim = .lnk$gl_dbgestim + .reclng;       ! ADD IN THE LENGTH OF THIS RECORD
;  2303          2780  2     lnk$gw_dbgrecs = .lnk$gw_dbgrecs + 1;         ! AND COUNT THE RECORD
;  2304          2781  2
;  2305          2782  2     if .lnk$gl_ctlmsk [lnk$v_dbg]                 ! IF IT CONTAINS THE DEBUGGER IT NOW HAS
;  2306          2783  2         or .lnk$gl_ctlmsk [lnk$v_trace]           !
;  2307          2784  2     then
;  2308          2785  2         obmodesc [omd$v_nobin] = false;           ! BINARY WE CARE ABOUT
;  2309          2786  2
;  2310          2787  2     return true;                                  ! AND THAT IS ALL
;  2311          2788  1     end;


                                001C 00000 COUNTDBG:
                                            .WORD   Save R2,R3,R4                                   ;  2753
                    54 00000000'    EF  9E 00002    MOVAB   LNK$GL_OMDDST, R4                       ;
                    53 00000000G    00  9E 00009    MOVAB   LNK$GL_CTLMSK, R3                       ;
                    52 00000000'    EF  9E 00010    MOVAB   OBMODESC+20, R2                         ;
            ED37    CF              00  FB 00017    CALLS   #0, SEQCHK                              ;  2764
                    2D              50  E9 0001C    BLBC    R0, 4$                                  ;
      25            62              05  E0 0001F    BBS     #5, OBMODESC+20, 3$                     ;  2768
                    0A      A0      A2  E8 00023    BLBS    OMD_HAS_DBG, 1$                         ;  2772
      06            63              06  E1 00027    BBC     #6, LNK$GL_CTLMSK, 1$                   ;
        A0          A2              01  90 0002B    MOVB    #1, OMD_HAS_DBG                         ;  2775
                                    64  D6 0002F    INCL    LNK$GL_OMDDST                           ;  2776
                    50      B8      A2  3C 00031 1$:MOVZWL  RECLNG, R0                              ;  2779
            38      A4              50  C0 00035    ADDL2   R0, LNK$GL_DBGESTIM                     ;
                            40      A4  B6 00039    INCW    LNK$GW_DBGRECS                          ;  2780
      05            63              06  E0 0003C    BBS     #6, LNK$GL_CTLMSK, 2$                   ;  2782
```

```
                03      02  A3              02  E1 00040          BBC     #2, LNK$GL_CTLMSK+2, 3$          ; 2783
                            62              02  8A 00045 2$:      BICB2   #2, OBMODESC+20                  ; 2785
                            50              01  D0 00048 3$:      MOVL    #1, R0                           ; 2787
                                                04 0004B          RET                                      ;
                                            50  D4 0004C 4$:      CLRL    R0                               ; 2788
                                                04 0004E          RET                                      ;

; Routine Size:  79 bytes,    Routine Base:  $CODE$ + 1593
```

```
;  2313          2789  1 routine compare_idc (entityname, node, idcrec) =
;  2314          2790  2     begin
;  2315          2791  2 !
;  2316          2792  2 ! COMPARE AN ENTITY NAME WITH THE CURRENT NODE
;  2317          2793  2 !
;  2318          2794  2
;  2319          2795  2     map
;  2320          2796  2         entityname : ref vector [, byte],
;  2321          2797  2         node : ref block [, byte],
;  2322          2798  2         idcrec : ref block [, byte];
;  2323          2799  2
;  2324          2800  2     return ch$compare (.entityname [0], entityname [1], .node [idcd$b_namlng], node [idcd$t_name], 0)
;  2325          2801  1     end;


                                        001C 00000 COMPARE_IDC:
                                                   .WORD   Save R2,R3,R4                                   ;  2789
                                52      04   AC  D0 00002          MOVL    ENTITYNAME, R2                  ;  2800
                                53           62  9A 00006          MOVZBL  (R2), R3                        ;
                                50      08   AC  D0 00009          MOVL    NODE, R0                        ;
                                51      1E   A0  9A 0000D          MOVZBL  30(R0), R1                      ;
                                54           01  D0 00011          MOVL    #1, R4                          ;
        51          00      01  A2           53  2D 00014          CMPC5   R3, 1(R2), #0, R1, 31(R0)       ;
                                         1F  A0      0001A                                                  ;
                                             03  1A 0001C          BGTRU   1$                              ;
                                54           01  D9 0001E          SBWC    #1, R4                          ;
                                50           54  D0 00021 1$:      MOVL    R4, R0                          ;
                                             04  00024              RET                                     ;  2801

; Routine Size:  37 bytes,    Routine Base:  $CODE$ + 15E2
```

```
;  2327     2802  1 routine alloc_idc (entityname, retnodeadr, idcrec) =
;  2328     2803  2     begin
;  2329     2804  2 !
;  2330     2805  2 ! ALLOCATE AND FILL IN A NEW IDCD BLOCK
;  2331     2806  2 !
;  2332     2807  2     map
;  2333     2808  2         entityname : ref vector [, byte],       ! POINTS TO ASCIC NAME
;  2334     2809  2         retnodeadr : ref vector [, long],
;  2335     2810  2         idcrec : ref block [, byte];            ! THE IDC GSD SUB-RECORD
;  2336     2811  2     bind
;  2337     2812  2         identstring = idcrec [idc$b_namlng] + 1 + .idcrec [idc$b_namlng] : vector [, byte],
;  2338     2813  2                                                 ! IDENT STRING IN RECORD
;  2339     2814  2         binaryident = identstring [1],          ! ADDR OF BINARY IDENT
;  2340     2815  2         objectname = identstring [1] + .identstring [0] : vector [, byte];      ! ADDR OF OBJECT NAME STRING
;  2341     2816  2
;  2342     2817  2     local
;  2343     2818  2         ptr,
;  2344     2819  2         idsize,
;  2345     2820  2         newnode : ref block [, byte];
;  2346     2821  2
;  2347     2822  2 !
;  2348     2823  2 ! COMPUTE SIZE OF IDENT STRING
;  2349     2824  2 !
;  2350     2825  2     idsize = 0;
;  2351     2826  2
;  2352     2827  2     if not .idcrec [idc$v_binident] then idsize = .identstring [0] + 1;
;  2353     2828  2
;  2354     2829  2 !
;  2355     2830  2 ! ALLOCATE A NEW IDCD BLOCK
;  2356     2831  2 !
;  2357     2832  2     lnk$alloblk (idcd$c_size + .entityname [0] + .idsize + .objectname [0] + 1, newnode);
;  2358     2833  2                                                 !+1 FOR SIZE OF OBJECTNAME IN ASCIC STRING
;  2359     2834  2     newnode [idcd$w_flags] = .idcrec [idc$w_flags];     ! COPY FLAGS FROM IDC SUBRECORD
;  2360     2835  2     newnode [idcd$l_defomd] = .lnk$gw_nmodules; ! SET DEFINING MODULE NUMBER
;  2361     2836  2     newnode [idcd$l_deffdb] = .lnk$gl_curfil;   !  DEFINING FILE FDB ADDRESS
;  2362     2837  2     newnode [idcd$b_idlng] = .identstring [0];  ! LENGTH OF IDENT STRING
;  2363     2838  2     newnode [idcd$b_objlng] = .objectname [0];  ! LENGTH OF OBJECT NAME STRING
;  2364     2839  2     newnode [idcd$b_namlng] = .idcrec [idc$b_namlng];   ! ENTITY NAME LENGTH
;  2365     2840  2     ptr = ch$move (.idcrec [idc$b_namlng],      ! COPY IN THE ENTITY NAME
;  2366     2841  2         idcrec [idc$b_namlng] + 1, newnode [idcd$t_name]);
;  2367     2842  2     newnode [idcd$l_objnam] = .ptr;             ! SET POINTER TO OBJECT NAME
;  2368     2843  2     ptr = ch$move (.objectname [0] + 1, objectname [0], .ptr);  ! MOVE IN THE OBJECT NAME
;  2369     2844  2
;  2370     2845  2     if .newnode [idcd$v_binident]               ! IF BINARY IDENT, SET IN THE IDENT
;  2371     2846  2     then
;  2372     2847  2         newnode [idcd$l_ident] = .binaryident
;  2373     2848  2     else
;  2374     2849  3         begin
;  2375     2850  3         newnode [idcd$l_ident] = .ptr;          ! ASCIC IDENT, SET POINTER TO IDENT STRING
;  2376     2851  3         ch$move (.identstring [0] + 1, identstring [0], .ptr);  !  AND COPY IN THE IDENT
;  2377     2852  2         end;
;  2378     2853  2
;  2379     2854  2 !
;  2380     2855  2 ! RETURN NODE ADDRESS TO CALLER
;  2381     2856  2 !
;  2382     2857  2     retnodeadr [0] = .newnode;
;  2383     2858  2     return true
```

```
; 2384          2859  1     end;


                                        03FC 00000 ALLOC_IDC:
                                                         .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9          ; 2802
                     5E              04  C2 00002        SUBL2    #4, SP                                ;
                     52         0C   AC  D0 00005        MOVL     IDCREC, R2                            ; 2812
                     50         03   A2  9A 00009        MOVZBL   3(R2), R0                             ;
                     57         04 A240  9E 0000D        MOVAB    4(R2)[R0], R7                         ;
                     59              67  9A 00012        MOVZBL   (R7), R9                              ; 2815
                                     51  D4 00015        CLRL     IDSIZE                                ; 2825
                     04         01   A2  E8 00017        BLBS     1(R2), 1$                             ; 2827
                     51         01   A9  9E 0001B        MOVAB    1(R9), IDSIZE                         ;
                                     5E  DD 0001F 1$:    PUSHL    SP                                    ; 2832
                     50         04   BC  9A 00021        MOVZBL   @ENTITYNAME, R0                       ;
                     50              51  C0 00025        ADDL2    IDSIZE, R0                            ;
                     58         01 A947  9A 00028        MOVZBL   1(R9)[R7], R8                         ;
                                20 A840  9F 0002D        PUSHAB   32(R8)[R0]                            ;
          00000000G  00              02  FB 00031        CALLS    #2, LNK$ALLOBLK                       ;
                     56              6E  D0 00038        MOVL     NEWNODE, R6                           ; 2834
                0A   A6         01   A2  B0 0003B        MOVW     1(R2), 10(R6)                         ;
                0C   A6 00000000'    EF  3C 00040        MOVZWL   LNK$GW_NMODULES, 12(R6)               ; 2835
                10   A6 00000000G    00  D0 00048        MOVL     LNK$GL_CURFIL, 16(R6)                 ; 2836
                14   A6              59  90 00050        MOVB     R9, 20(R6)                            ; 2837
                15   A6              58  90 00054        MOVB     R8, 21(R6)                            ; 2838
                1E   A6         03   A2  90 00058        MOVB     3(R2), 30(R6)                         ; 2839
                     50         03   A2  9A 0005D        MOVZBL   3(R2), R0                             ; 2840
     1F   A6    04   A2              50  28 00061        MOVC3    R0, 4(R2), 31(R6)                     ; 2841
                1A   A6              53  D0 00067        MOVL     PTR, 26(R6)                           ; 2842
                     50         01   A8  9E 0006B        MOVAB    1(R8), R0                             ; 2843
          63    01 A947              50  28 0006F        MOVC3    R0, 1(R9)[R7], (PTR)                  ;
                     07         0A   A6  E9 00075        BLBC     10(R6), 2$                            ; 2845
                16   A6         01   A7  D0 00079        MOVL     1(R7), 22(R6)                         ; 2847
                                     0C  11 0007E        BRB      3$                                    ;
                16   A6              53  D0 00080 2$:    MOVL     PTR, 22(R6)                           ; 2850
                     50         01   A9  9E 00084        MOVAB    1(R9), R0                             ; 2851
          63         67              50  28 00088        MOVC3    R0, (R7), (PTR)                       ;
                08   BC              56  D0 0008C 3$:    MOVL     R6, @RETNODEADR                       ; 2857
                     50              01  D0 00090        MOVL     #1, R0                                ; 2858
                                         04 00093        RET                                            ; 2859

; Routine Size:  148 bytes,    Routine Base:  $CODE$ + 1607
```

```
;  2386     2860  1 routine randentity =
;  2387     2861  2     begin
;  2388     2862  2 !
;  2389     2863  2 ! THIS ROUTINE PROCESSES THE IDC GSD RECORD.
;  2390     2864  2 !
;  2391     2865  2     local
;  2392     2866  2         status,
;  2393     2867  2         length,
;  2394     2868  2         defomd : ref block [, byte],            ! DEFINING OMD ADDRESS
;  2395     2869  2         deffdb : ref block [, byte],            ! DEFINING FDB ADDRESS
;  2396     2870  2         identstring : ref vector [, byte],      ! POINTER TO IDENT STRING
;  2397     2871  2         objectname : ref vector [, byte],       ! POINTER TO OBJECT NAME STRING
;  2398     2872  2         entry : ref block [, byte];             ! ADDRESS OF ENTRY
;  2399     2873  2
;  2400     2874  2     bind
;  2401     2875  2         idcrec = objvec [.gsdoffset] : block [, byte];  ! POINT TO THE IDC SUBRECORD
;  2402     2876  2
;  2403     2877  2 !
;  2404     2878  2 ! CHECK THE LENGTH OF ALL THE NAME STRINGS
;  2405     2879  2 !
;  2406     2880  2 !
;  2407     2881  2 ! THE ENTITY NAME
;  2408     2882  2 !
;  2409     2883  2
;  2410     2884  2     if .idcrec [idc$b_namlng] eql 0 or .idcrec [idc$b_namlng] gtru sym$c_maxlng
;  2411     2885  2     then
;  2412     2886  3         begin
;  2413     2887  3         signal (lin$_illnamelen, 6, lnk$gt_entity, idcrec [idc$b_namlng], .idcrec [idc$b_namlng],
;  2414     2888  3             sym$c_maxlng, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2415     2889  3         return false
;  2416     2890  2         end;
;  2417     2891  2
;  2418     2892  2 !
;  2419     2893  2 ! THE IDENT STRING
;  2420     2894  2 !
;  2421     2895  2     identstring = idcrec [idc$b_namlng] + 1 + .idcrec [idc$b_namlng];
;  2422     2896  2
;  2423     2897  2     if .identstring [0] eql 0 or .identstring [0] gtru sym$c_maxlng
;  2424     2898  2     then
;  2425     2899  3         begin
;  2426     2900  3         signal (lin$_illnamelen, 6, lnk$gt_ident, identstring [0], .identstring [0], sym$c_maxlng,
;  2427     2901  3             obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2428     2902  3         return false
;  2429     2903  2         end;
;  2430     2904  2
;  2431     2905  2 !
;  2432     2906  2 ! AND THE OBJECT NAME
;  2433     2907  2 !
;  2434     2908  2     objectname = identstring [1] + .identstring [0];
;  2435     2909  2
;  2436     2910  2     if .objectname [0] eql 0 or .objectname [0] gtru sym$c_maxlng
;  2437     2911  2     then
;  2438     2912  3         begin
;  2439     2913  3         signal (lin$_illnamelen, 6, lnk$gt_objnam, objectname [0], .objectname [0], sym$c_maxlng,
;  2440     2914  3             obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2441     2915  3         return false
;  2442     2916  2         end;
```

```
; 2443          2917  2
; 2444          2918  2 !
; 2445          2919  2 ! UPDATE GSD OFFSET INTO RECORD
; 2446          2920  2 !
; 2447          2921  2     length = objectname [1] + .objectname [0] - idcrec;
; 2448          2922  2     gsdoffset = .gsdoffset + .length;
; 2449          2923  2 !
; 2450          2924  2 ! ENTRY APPEARS GOOD.  LET'S ENTER IT
; 2451          2925  2 !
; 2452          2926  2
; 2453          2927  3     if (status = lib$insert_tree (lnk$gl_entitree, idcrec [idc$b_namlng], %ref (0), compare_idc, alloc_idc,
; 2454          2928  2         entry, idcrec)) eql lib$_keyalrins
; 2455          2929  2     then
; 2456          2930  3         begin
; 2457          2931  3         lib$lookup_tree (lnk$gl_omdtree, .entry [idcd$l_defomd],        ! FIND DEFINING MODULE OMD
; 2458          2932  3             lnk$compare_omd, defomd);
; 2459          2933  3         defomd = .defomd [node$l_ptr];          ! REALLY POINT TO THE OMD
; 2460          2934  3         deffdb = .entry [idcd$l_deffdb];        ! GET THE DEFINING FDB
; 2461          2935  3
; 2462          2936  4         if (.idcrec [idc$v_binident] neq .entry [idcd$v_binident])
; 2463          2937  4                                                 ! IF BOTH ARE NOT THE SAME (ASCIC OR BINARY)
; 2464          2938  4             or (.idcrec [idc$v_binident]        ! OR BINARY IDENT WITH DIFFERENT MATCH CONTROL
; 2465          2939  4             and (.idcrec [idc$v_idmatch] neq .entry [idcd$v_idmatch]))
; 2466          2940  3         then
; 2467          2941  3             signal ((lin$_entidmtcht and not sts$m_severity) or .idcrec [idc$v_errsev], 5,
; 2468          2942  3                 idcrec [idc$b_namlng],          ! ISSUE WARNING
; 2469          2943  3                 obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename], defomd [omd$b_namlng],
; 2470          2944  3                 deffdb [fdb$q_filename]);
; 2471          2945  3
; 2472          2946  3         if not ch$eql (.objectname [0], objectname [1],         ! CHECK THAT OBJECT NAMES THE SAME
; 2473          2947  3                 .entry [idcd$b_objlng], .entry [idcd$l_objnam] + 1)
; 2474          2948  3         then
; 2475          2949  3             signal ((lin$_entidmtcho and not sts$m_severity) or .idcrec [idc$v_errsev], 7, objectname [0],
; 2476          2950  3                 idcrec [idc$b_namlng], obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename],
; 2477          2951  3                 .entry [idcd$l_objnam], defomd [omd$b_namlng], deffdb [fdb$q_filename]);
; 2478          2952  3
; 2479          2953  3         if not .entry [idcd$v_binident]         ! NOW CHECK THE IDENT MATCH
; 2480          2954  3         then
; 2481          2955  4             begin
; 2482          2956  4
; 2483          2957  4             if not ch$eql (.identstring [0], identstring [1],   ! CHECK ASCII IDENTS
; 2484          2958  4                     .entry [idcd$b_idlng], .entry [idcd$l_ident] + 1, 0)
; 2485          2959  4             then
; 2486          2960  4                 signal ((lin$_entidmtch and not sts$m_severity) or .idcrec [idc$v_errsev], 8, objectname [0]
; 2487          2961  4                     identstring [0], idcrec [idc$b_namlng], obmodesc [omd$b_namlng],
; 2488          2962  4                     lnk$gl_curfil [fdb$q_filename], .entry [idcd$l_ident], defomd [omd$b_namlng],
; 2489          2963  4                     deffdb [fdb$q_filename]);
; 2490          2964  4
; 2491          2965  4             end
; 2492          2966  3         else
; 2493          2967  4             begin
; 2494          2968  4 !
; 2495          2969  4 ! CHECK BINARY IDENTS
; 2496          2970  4 !
; 2497          2971  4
; 2498          2972  4             bind
; 2499          2973  4                 ident1 = identstring [1] : block [, byte],
```

```
;  2500          2974  4                ident2 = entry [idcd$l_ident] : block [, byte];
;  2501          2975  4
;  2502          2976  4            local
;  2503          2977  4                ident1majid,
;  2504          2978  4                ident1minid,
;  2505          2979  4                ident2majid,
;  2506          2980  4                ident2minid;
;  2507          2981  4
;  2508          2982  4            ident1majid = .ident1 [gmt$b_majorid];      ! BLISS GENERATES BETTER CODE THIS WAY
;  2509          2983  4            ident1minid = .ident1 [gmt$b_minorid];
;  2510          2984  4            ident2majid = .ident2 [gmt$b_majorid];
;  2511          2985  4            ident2minid = .ident2 [gmt$b_minorid];
;  2512          2986  4
;  2513          2987  4            if .ident1majid neq .ident2majid    ! MAJOR IDS MUST BE EQUAL
;  2514          2988  5                or (if .entry [idcd$v_idmatch] eql idc$c_leq    ! COMPARE MINOR IDS BASED ON MATCH CONTROL
;  2515          2989  5                then .ident2minid gtru .ident1minid else .ident2minid neq .ident1minid)
;  2516          2990  4            then
;  2517          2991  4                signal ((lin$_entidmtchb and not sts$m_severity) or .idcrec [idc$v_errsev], 10,
;  2518          2992  4                    objectname [0], .ident1majid, .ident1minid, idcrec [idc$b_namlng],
;  2519          2993  4                    obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename], .ident2majid, .ident2minid,
;  2520          2994  4                    defomd [omd$b_namlng], deffdb [fdb$q_filename]);
;  2521          2995  4
;  2522          2996  3            end;
;  2523          2997  3
;  2524          2998  2        end;
;  2525          2999  2
;  2526          3000  2    return true
;  2527          3001  1    end;


                                 0FFC 00000 RANDENTITY:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2860
                      5E                10  C2 00002          SUBL2   #16, SP                               ;
                      52 00000000'      EF  3C 00005          MOVZWL  GSDOFFSET, R2                         ; 2875
                      52 00000000'      EF  C0 0000C          ADDL2   OBJVEC, R2                            ;
                      5B          03    A2  9E 00013          MOVAB   3(R2), R11                            ; 2884
                                        6B  95 00017          TSTB    (R11)                                 ;
                                        05  13 00019          BEQL    1$                                    ;
                      1F                6B  91 0001B          CMPB    (R11), #31                            ;
                                        1D  1B 0001E          BLEQU   2$                                    ;
7E 00000000G          00                14  C1 00020 1$:      ADDL3   #20, LNK$GL_CURFIL, -(SP)             ; 2888
                          00000000'     EF  9F 00028          PUSHAB  OBMODESC+40                           ;
                                        1F  DD 0002E          PUSHL   #31                                   ;
                      7E                6B  9A 00030          MOVZBL  (R11), -(SP)                          ;
                                        5B  DD 00033          PUSHL   R11                                   ;
                          00000000'     EF  9F 00035          PUSHAB  LNK$GT_ENTITY                         ; 2887
                                        5F  11 0003B          BRB     6$                                    ; 2888
                      50                6B  9A 0003D 2$:      MOVZBL  (R11), R0                             ; 2895
                      58          04 A042   9E 00040          MOVAB   4(R0)[R2], IDENTSTRING                ;
                                        68  95 00045          TSTB    (IDENTSTRING)                         ; 2897
                                        05  13 00047          BEQL    3$                                    ;
                      1F                68  91 00049          CMPB    (IDENTSTRING), #31                    ;
                                        1D  1B 0004C          BLEQU   4$                                    ;
7E 00000000G          00                14  C1 0004E 3$:      ADDL3   #20, LNK$GL_CURFIL, -(SP)             ; 2901
```

```
                    00000000'   EF   9F 00056          PUSHAB  OBMODESC+40                     ;
                                1F   DD 0005C          PUSHL   #31                             ;
                          7E        68   9A 0005E      MOVZBL  (IDENTSTRING), -(SP)            ;
                                    58   DD 00061      PUSHL   IDENTSTRING                     ;
                    00000000'   EF   9F 00063          PUSHAB  LNK$GT_IDENT                    ; 2900
                                    31   11 00069      BRB     6$                              ; 2901
                 04   AE            68   9A 0006B 4$:  MOVZBL  (IDENTSTRING), 4(SP)           ; 2908
50               04   AE            01   C1 0006F      ADDL3   #1, 4(SP), R0                   ;
59               58                 50   C1 00074      ADDL3   R0, IDENTSTRING, OBJECTNAME     ;
                                    69   95 00078      TSTB    (OBJECTNAME)                    ; 2910
                                    05   13 0007A      BEQL    5$                              ;
                          1F        69   91 0007C      CMPB    (OBJECTNAME), #31               ;
                                    2D   1B 0007F      BLEQU   7$                              ;
7E 00000000G              00        14   C1 00081 5$:  ADDL3   #20, LNK$GL_CURFIL, -(SP)       ; 2914
                    00000000'   EF   9F 00089          PUSHAB  OBMODESC+40                     ;
                                1F   DD 0008F          PUSHL   #31                             ;
                          7E        69   9A 00091      MOVZBL  (OBJECTNAME), -(SP)             ;
                                    59   DD 00094      PUSHL   OBJECTNAME                      ;
                    00000000'   EF   9F 00096          PUSHAB  LNK$GT_OBJNAM                   ; 2913
                                    06   DD 0009C 6$:  PUSHL   #6                              ; 2914
                    00000000G   8F   DD 0009E          PUSHL   #LIN$_ILLNAMELEN                ;
   00000000G        00               08   FB 000A4      CALLS   #8, LIB$SIGNAL                  ;
                                01AD   31   000AB      BRW     17$                             ; 2915
                          53        69   9A 000AE 7$:  MOVZBL  (OBJECTNAME), R3                ; 2921
50                        59        53   C1 000B1      ADDL3   R3, OBJECTNAME, R0              ;
                          50        52   C2 000B5      SUBL2   R2, R0                          ;
                                    50   D6 000B8      INCL    LENGTH                          ;
   00000000'        EF               50   A0 000BA      ADDW2   LENGTH, GSDOFFSET               ; 2922
                                    52   DD 000C1      PUSHL   R2                              ; 2927
                       0C   AE       9F   000C3       PUSHAB  ENTRY                           ;
                     FEA2   CF       9F   000C6       PUSHAB  ALLOC_IDC                       ;
                     FE79   CF       9F   000CA       PUSHAB  COMPARE_IDC                     ;
                       10   AE       D4   000CE       CLRL    16(SP)                          ;
                       10   AE       9F   000D1       PUSHAB  16(SP)                          ;
                                    5B   DD 000D4      PUSHL   R11                             ;
                    00000000'   EF   9F 000D6          PUSHAB  LNK$GL_ENTITREE                 ;
   00000000G        00               07   FB 000DC      CALLS   #7, LIB$INSERT_TREE             ;
   00000000G        8F               50   D1 000E3      CMPL    STATUS, #LIB$_KEYALRINS         ; 2928
                                    03   13 000EA      BEQL    8$                              ;
                                0168   31   000EC      BRW     16$                             ;
                       0C   AE       9F   000EF 8$:    PUSHAB  DEFOMD                          ; 2931
                     EEEC   CF       9F   000F2       PUSHAB  LNK$COMPARE_OMD                 ;
                          54   10   AE   D0   000F6     MOVL    ENTRY, R4                       ;
                               0C   A4   DD   000FA     PUSHL   12(R4)                          ;
                    00000000'   EF   9F 000FD          PUSHAB  LNK$GL_OMDTREE                  ;
   00000000G        00               04   FB 00103      CALLS   #4, LIB$LOOKUP_TREE             ;
                          50        0C   AE   D0   0010A     MOVL    DEFOMD, R0              ; 2933
                 0C   AE             0A   A0   D0   0010E     MOVL    10(R0), DEFOMD          ;
                          55        10   A4   D0   00113     MOVL    16(R4), DEFFDB          ; 2934
                          5A        01   A2   9E   00117     MOVAB   1(R2), R10              ; 2936
50               0A   A4                  6A   8D   0011B     XORB3   (R10), 10(R4), R0       ;
                          0D             50   E8   00120     BLBS    R0, 9$                  ;
                          38             6A   E9   00123     BLBC    (R10), 10$              ; 2938
50               0A   A4                  6A   8D   00126     XORB3   (R10), 10(R4), R0       ; 2939
                          06             50   93   0012B     BITB    R0, #6                  ;
                                         2E   13   0012E     BEQL    10$                     ;
                       14   A5   9F   00130 9$:    PUSHAB  20(DEFFDB)                      ; 2944
```

```
                7E      10   AE              28  C1 00133          ADDL3   #40, DEFOMD, -(SP)                                  ; 2943
                7E 00000000G 00              14  C1 00138          ADDL3   #20, LNK$GL_CURFIL, -(SP)
                                00000000'    EF  9F 00140          PUSHAB  OBMODESC+40
                                             5B  DD 00146          PUSHL   R11                                                 ; 2944
                                             05  DD 00148          PUSHL   #5
          50    6A            03             03  EF 0014A          EXTZV   #3, #3, (R10), R0                                   ; 2941
                7E            50 00000000*   8F  C9 0014F          BISL3   #<LIN$_ENTIDMTCHT&-8>, R0, -(SP)
                   00000000G  00             07  FB 00157          CALLS   #7, LIB$SIGNAL                                      ; 2944
                              50       15    A4  9A 0015E 10$:     MOVZBL  21(R4), R0                                          ; 2947
                              56       1A    A4  D0 00162          MOVL    26(R4), R6
          50    00       01    A9           53  2D 00166          CMPC5   R3, 1(OBJECTNAME), #0, R0, 1(R6)                    ; 2946
                                       01    A6      0016C
                                             32  13 0016E          BEQL    11$
                                       14    A5  9F 00170          PUSHAB  20(DEFFDB)                                          ; 2951
                7E      10   AE              28  C1 00173          ADDL3   #40, DEFOMD, -(SP)
                                             56  DD 0017?          PUSHL   R6
                7E 00000000G 00              14  C1 0017A          ADDL3   #20, LNK$GL_CURFIL, -(SP)                           ; 2950
                                00000000'    EF  9F 00182          PUSHAB  OBMODESC+40
                                     0A00    8F  BB 00188          PUSHR   #^M<R9,R11>                                         ; 2951
                                             07  DD 0018C          PUSHL   #7
          50    6A            03             03  EF 0018E          EXTZV   #3, #3, (R10), R0                                   ; 2949
                7E            50 00000000*   8F  C9 00193          BISL3   #<LIN$_ENTIDMTCHO&-8>, R0, -(SP)
                   00000000G  00             09  FB 0019B          CALLS   #9, LIB$SIGNAL                                      ; 2951
                              57       16    A4  9E 001A2 11$:     MOVAB   22(R4), R7                                          ; 2958
                              48       0A    A4  E8 001A6          BLBS    10(R4), 12$
                              50       14    A4  9A 001AA          MOVZBL  20(R4), R0
                              56             67  D0 001AE          MOVL    (R7), R6
          50    00       01    A8           04  AE  2D 001B1      CMPC5   4(SP), 1(IDENTSTRING), #0, R0, 1(R6)                 ; 2957
                                       01    A6      001B8
                                             61  13 001BA          BEQL    14$
                                       14    A5  9F 001BC          PUSHAB  20(DEFFDB)                                          ; 2963
                7E      10   AE              28  C1 001BF          ADDL3   #'0, DEFOMD, -(SP)                                  ; 2962
                                             56  DD 001C4          PUSHL   R6                                                  ; 2963
                7E 00000000G 00              14  C1 001C6          ADDL3   #20, LNK$GL_CURFIL, -(SP)                           ; 2962
                                00000000'    EF  9F 001CE          PUSHAB  OBMODESC+40                                         ; 2961
                                     0900    8F  BB 001D4          PUSHR   #^M<R8,R11>                                         ; 2963
                                             59  DD 001D8          PUSHL   OBJECTNAME
                                             08  DD 001DA          PUSHL   #8
          50    6A            03             03  EF 001DC          EXTZV   #3, #3, (R10), R0                                   ; 2960
                7E            50 00000000*   8F  C9 001E1          BISL3   #<LIN$_ENTIDMTCH&-8>, R0, -(SP)
                   00000000G  00             0A  FB 001E9          CALLS   #10, LIB$SIGNAL                                     ; 2963
                                             65  11 001F0          BRB     16$                                                 ; 2953
                              50       01    A8  9E 001F2 12$:     MOVAB   1(IDENTSTRING), R0                                  ; 2973
                              53       03    A0  9A 001F6          MOVZBL  3(R0), IDENT1MAJID                                  ; 2982
          51    60            18             00  EF 001FA          EXTZV   #0, #24, (R0), IDENT1MINID                          ; 2983
                              52       03    A7  9A 001FF          MOVZBL  3(R7), IDENT2MAJID                                  ; 2984
          50    67            18             00  EF 00203          EXTZV   #0, #24, (R7), IDENT2MINID                          ; 2985
                              52             53  D1 00208          CMPL    IDENT1MAJID, IDENT2MAJID                            ; 2987
                                             12  12 0020B          BNEQ    15$
                              06       0A    A4  93 0020D          BITB    10(R4), #6                                          ; 2988
                                             07  12 00211          BNEQ    13$
                              51             50  D1 00213          CMPL    IDENT2MINID, IDENT1MINID                            ; 2989
                                             3F  1B 00216          BLEQU   16$
                                             05  11 00218          BRB     15$
                              51             50  D1 0021A 13$:     CMPL    IDENT2MINID, IDENT1MINID
                                             38  13 0021D 14$:     BEQL    16$
                                       14    A5  9F 0021F 15$:     PUSHAB  20(DEFFDB)                                          ; 2994
```

```
                7E        10    AE              28  C1 00222          ADDL3   #40, DEFOMD, -(SP)                                  ;
                                                50  DD 00227          PUSHL   IDENT2MINID                                         ;
                                                52  DD 00229          PUSHL   IDENT2MAJID                                         ;
                7E 00000000G    00              14  C1 0022B          ADDL3   #20, LNK$GL_CURFIL, -(SP)                        ; 2993
                                    00000000'   EF  9F 00233          PUSHAB  OBMODESC+40                                         ;
                                          0802  8F  BB 00239          PUSHR   #^M<R1,R11>                                         ; 2994
                                                53  DD 0023D          PUSHL   IDENT1MAJID                                         ;
                                                59  DD 0023F          PUSHL   OBJECTNAME                                          ;
                                                0A  DD 00241          PUSHL   #10                                                 ;
        50      6A              03              03  EF 00243          EXTZV   #3, #3, (R10), R0                                   ; 2991
                7E              50 00000000*    8F  C9 00248          BISL3   #<LIN$_ENTIDMISCHB-8>, R0, -(SP)                    ;
                   00000000G    00              0C  FB 00250          CALLS   #12, LIB$SIGNAL                                     ; 2994
                                50              01  D0 00257 16$:     MOVL    #1, R0                                              ; 3000
                                                    04 0025A          RET                                                         ;
                                                50  D4 0025B 17$:     CLRL    R0                                                  ; 3001
                                                    04 0025D          RET                                                         ;

; Routine Size:  606 bytes,    Routine Base:  $CODE$ + 169B
```

```
1 routine compare_env (envname, node, envrec) =
2     begin
2 !
2 !       COMPARE AN ENVIRONMENT NAME WITH THE CURRENT NODE IN TREE
2 !
2     map
2         envname : ref vector [, byte],
2         node : ref block [, byte];
2     bind
2         envdesc = .node + node$c_short : block [, byte];
2
2     return ch$compare (.envname [0], envname [1], .envdesc [nvd$b_namlng], envdesc [nvd$t_name], 0)
1     end;
```

```
                                        001C 00000 COMPARE_ENV:
                                                          .WORD   Save R2,R3,R4                      ; 3002
            50      08  AC           0A  C1 00002         ADDL3   #10, NODE, R0                      ; 3011
                        52      04   AC  D0 00007         MOVL    ENVNAME, R2                        ; 3013
                        53           62  9A 0000B         MOVZBL  (R2), R3                           ;
                        51      12   A0  9A 0000E         MOVZBL  18(R0), R1                         ;
                        54           01  D0 00012         MOVL    #1, R4                             ;
    51      00      01  A2           53  2D 00015         CMPC5   R3, 1(R2), #0, R1, 19(R0)          ;
                                13   A0     0001B                                                    ;
                                     03  1A 0001D         BGTRU   1$                                 ;
                        54           01  D9 0001F         SBWC    #1, R4                             ;
                        50           54  D0 00022 1$:     MOVL    R4, R0                             ;
                                         04 00025         RET                                        ; 3014
```

; Routine Size: 38 bytes,   Routine Base: $CODE$ + 18F9

```
;  2543          3015  1 routine alloc_env (envname, retnodeadr, envrec) =
;  2544          3016  2     begin
;  2545          3017  2 !
;  2546          3018  2 !       ALLOCATE A NEW ENVIRONMENT DESCRIPTOR BLOCK
;  2547          3019  2 !
;  2548          3020  2     map
;  2549          3021  2         envname : ref vector [, byte],
;  2550          3022  2         retnodeadr : ref vector [, long],
;  2551          3023  2         envrec : ref block [, byte];
;  2552          3024  2
;  2553          3025  2     local
;  2554          3026  2         newnode : ref block [, byte],
;  2555          3027  2         envdesc : ref block [, byte];
;  2556          3028  2
;  2557          3029  2     lnk$alloblk (node$c_short + nvd$c_size + .envname [0], newnode);
;  2558          3030  2     envdesc = .newnode + node$c_short;
;  2559          3031  2     ch$fill (0, nvd$c_size, .envdesc);
;  2560          3032  2     envdesc [nvd$l_omdnum] = .lnk$gw_nmodules;
;  2561          3033  2     ch$move (.envname [0] + 1, envname [0], envdesc [nvd$b_namlng]);
;  2562          3034  2     retnodeadr [0] = .newnode;
;  2563          3035  2     return true
;  2564          3036  1     end;


                                                007C 00000 ALLOC_ENV:
                                                           .WORD   Save R2,R3,R4,R5,R6                          ;  3015
                                    5E              04  C2 00002           SUBL2   #4, SP                       ;
                                                    5E  DD 00005           PUSHL   SP                           ;  3029
                                    7E        04    BC  9A 00007           MOVZBL  @ENVNAME, -(SP)              ;
                                    6E              1D  C0 0000B           ADDL2   #29, (SP)                    ;
                        00000000G   00              02  FB 0000E           CALLS   #2, LNK$ALLOBLK              ;
                              56    6E              0A  C1 00015           ADDL3   #10, NEWNODE, ENVDESC        ;  3030
                    13        00    6E              00  2C 00019           MOVC5   #0, (SP), #0, #19, (ENVDESC) ;  3031
                                                    66      0001E          ;                                    ;
                                0C  A6 00000000'    EF  3C 0001F           MOVZWL  LNK$GW_NMODULES, 12(ENVDESC) ;  3032
                                    50        04    BC  9A 00027           MOVZBL  @ENVNAME, R0                 ;  3033
                                                    50  D6 0002B           INCL    R0                           ;
                          12    A6        04    BC      50  28 0002D       MOVC3   R0, @ENVNAME, 18(ENVDESC)    ;
                                0C  BC              6E  D0 00033           MOVL    NEWNODE, @RETNODEADR         ;  3034
                                    50              01  D0 00037           MOVL    #1, R0                       ;  3035
                                                        04 0003A           RET                                  ;  3036

; Routine Size:  59 bytes,    Routine Base:  $CODE$ + 191F
```

```
;  2566          3037  1 routine insudfenv (envnode) : novalue =
;  2567          3038  2     begin
;  2568          3039  2 !
;  2569          3040  2 !       THIS ROUTINE ENTERS AN ENVIRONMENT INTO THE UNDEFINED
;  2570          3041  2 !       ENVIRONMENT LIST.
;  2571          3042  2 !
;  2572          3043  2     map
;  2573          3044  2         envnode : ref block [, byte];
;  2574          3045  2     builtin
;  2575          3046  2         insque;
;  2576          3047  2
;  2577          3048  2     bind
;  2578          3049  2         envdesc = .envnode + node$c_short : block [, byte];
;  2579          3050  2
;  2580          3051  2     local
;  2581          3052  2         ch_result,
;  2582          3053  2         nxtenv : ref block [, byte];            ! POINTER TO SYMBOL VALUE BLOCK
;  2583          3054  2
;  2584          3055  2 !
;  2585          3056  2 !       FIND THE SPOT TO INSERT THE NEW ENVIRONMENT IN THE UNDEFINED LIST
;  2586          3057  2 !
;  2587          3058  2     nxtenv = lnk$gl_udfenv;                     ! SCAN THE UNDEFINED LIST
;  2588          3059  2
;  2589          3060  2     while (nxtenv = .nxtenv [nvd$l_udflink]) neq lnk$gl_udfenv do
;  2590          3061  3         begin
;  2591          3062  3
;  2592          3063  4         if (ch_result = ch$compare (.nxtenv [nvd$b_namlng],     ! IF GREATER
;  2593          3064  3                 nxtenv [nvd$t_name], .envdesc [nvd$b_namlng], envdesc [nvd$t_name])) gtr 0
;  2594          3065  3         then
;  2595          3066  3             exitloop                            ! THEN WE ARE ALL DONE
;  2596          3067  3         else
;  2597          3068  3
;  2598          3069  3             if .ch_result eql 0                 ! IF EQUAL
;  2599          3070  3             then
;  2600          3071  3                 return;                         ! THEN ALREADY IN LIST, SO RETURN
;  2601          3072  3
;  2602          3073  2         end;
;  2603          3074  2
;  2604          3075  2     insque (envdesc [nvd$l_udflink],            ! INSERT IN UNDEFINED LIST
;  2605          3076  2         .nxtenv [nvd$l_udblink]);
;  2606          3077  2     lnk$gw_nudfenvs = .lnk$gw_nudfenvs + 1;     ! COUNT THE UNDEFINED ENVIRONMENT
;  2607          3078  2     return;
;  2608          3079  1     end;


                                01FC 00000 INSUDFENV:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 3037
                    58 00000000'  EF  9E 00002     MOVAB   LNK$GL_UDFENV, R8                ;
            56      04    AC      0A  C1 00009     ADDL3   #10, ENVNODE, R6                 ; 3049
                          54      68  9E 0000E     MOVAB   LNK$GL_UDFENV, NXTENV            ; 3058
                          54      64  D0 00011 1$: MOVL    (NXTENV), NXTENV                 ; 3060
                          50      68  9E 00014     MOVAB   LNK$GL_UDFENV, R0                ;
                          50      54  D1 00017     CMPL    NXTENV, R0                       ;
                                  20  13 0001A     BEQL    3$                               ;
```

```
                                51      12    A4  9A 0001C          MOVZBL  18(NXTENV), R1                          ; 3063
                                50      12    A6  9A 00020          MOVZBL  18(R6), R0                              ; 3064
                                55            01  D0 00024          MOVL    #1, R5                                  ;
        50              00      13    A4      51  2D 00027          CMPC5   R1, 19(NXTENV), #0, R0, 19(R6)          ;
                                          13    A6     0002D                                                        ;
                                              03  1A 0002F          BGTRU   2$                                      ;
                                55            01  D9 00031          SBWC    #1, R5                                  ;
                                57            55  D0 00034 2$:      MOVL    R5, CH_RESULT                           ;
                                              03  14 00037          BGTR    3$                                      ;
                                              D6  12 00039          BNEQ    1$                                      ; 3069
                                                  04 0003B          RET                                             ; 3071
                                04    B4        66  0E 0003C 3$:      INSQUE  (R6), @4(NXTENV)                        ; 3076
                                          40    A8  B6 00040          INCW    LNK$GW_NUDFENVS                         ; 3077
                                                  04 00043          RET                                             ; 3079

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 195A

; 2609          3080  1
```

```
;2611     3081  1 global routine lnk$fndenvmap (envnum) =
;2612     3082  2     begin
;2613     3083  2 !
;2614     3084  2 !       THIS ROUTINE RETURNS THE ADDRESS OF THE MAPPING
;2615     3085  2 !       TABLE ENTRY FOR ENVIRONMENT NUMBER ''ENVNUM'' IN
;2616     3086  2 !       THE CURRENT MODULE.
;2617     3087  2 !       THE ENVIRONMENT MAPPING TABLE IS AN ARRAY POINTED TO BY
;2618     3088  2 !       THE MODULE DESCRIPTOR BLOCK. IT IS INITIALLY
;2619     3089  2 !       ALLOCATED WITH SPACE SUFFICIENT FOR 256
;2620     3090  2 !       ENTRIES.  IF MORE THAN 256 ENVIRONMENTS ARE ENCOUNTERED DURING
;2621     3091  2 !       THE PROCESSING OF THE OBJECT MODULE, THE INITIAL TABLE IS
;2622     3092  2 !       COPIED OUT TO ANOTHER BLOCK, AND THE MAP TABLE ORIGINAL MAP TABLE
;2623     3093  2 !       BECOMES A TABLE OF TABLE ADDRESSES.
;2624     3094  2 !
;2625     3095  2     literal
;2626     3096  2         tablesize = 256*pmt$c_size;             ! SIZE IN BYTES OF A BLOCK
;2627     3097  2     local
;2628     3098  2         blockoff,                               ! OFFSET OF ENTRY IN EXTENDED BLOCK
;2629     3099  2         first_entry : ref block [, byte],
;2630     3100  2         first_256 : ref block [, byte],         ! address of block of first 256
;2631     3101  2         mapent : ref block [, byte],            ! ADDRESS IN PSECT MAPPING TABLE
;2632     3102  2         mapoff : ref block [, byte];            ! ADDRESS OF EXTENDED MAPPING BLOCK
;2633     3103  2
;2634     3104  2     if .lnk$gl_curomd [omd$l_envmap] eql 0      ! IF NO TABLE ALLOCATED YET
;2635     3105  2     then
;2636     3106  3         begin
;2637     3107  3         lnk$alloblk (tablesize, lnk$gl_curomd [omd$l_envmap]);  ! ALLOCATE IT NOW
;2638     3108  3         ch$fill (0, tablesize, .lnk$gl_curomd [omd$l_envmap]);
;2639     3109  2         end;
;2640     3110  2
;2641     3111  2     mapent = .lnk$gl_curomd [omd$l_envmap] + .envnum*pmt$c_size;
;2642     3112  2                                                 ! GET ENTRY ADDRESS IN BASE MAPPING TABLE
;2643     3113  2
;2644     3114  2     if not .lnk$gl_curomd [omd$v_e256]          ! IF NOT INTO EXTENDED MAP TABLE
;2645     3115  2         and .envnum lequ 255                    !  AND LEQU 255 ENVIRONMENTS
;2646     3116  2     then
;2647     3117  2         return .mapent
;2648     3118  2 !
;2649     3119  2 ! EXTENDED ENVIRONMENT ENTRY
;2650     3120  2 !
;2651     3121  2     else
;2652     3122  3         begin
;2653     3123  3         blockoff = (.envnum mod 256)*pmt$c_size;        ! OFFSET OF ENTRY WITHIN EXTENDED TABLE
;2654     3124  3         mapent = .lnk$gl_curomd [omd$l_envmap] + ((.envnum )/256)*pmt$c_size;
;2655     3125  3                                                 ! GET ADDRESS OF EXTENDED TABLE ADDRESS
;2656     3126  3
;2657     3127  3         if ..mapent neq 0                       ! IF EXTENDED TABLE PRESENT
;2658     3128  3             and .lnk$gl_curomd [omd$v_e256]     !  AND THERE REALLY IS AN EXTENDED TABLE
;2659     3129  3         then
;2660     3130  3             return ..mapent + .blockoff         !  THEN RETURN ENTRY ADDRESS
;2661     3131  3         else
;2662     3132  4             begin
;2663     3133  4             lnk$alloblk (tablesize, mapoff);    ! ALLOCATE EXTENDED TABLE BLOCK
;2664     3134  4
;2665     3135  4             if not .lnk$gl_curomd [omd$v_e256]  ! IF THIS IS THE FIRST ONE
;2666     3136  4             then
;2667     3137  5                 begin
```

```
;  2668          3138  5                first_entry = lnk$gl_curomd[omd$l_envmap];
;  2669          3139  5                lnk$allOblk(tablesize,first_256);
;  2670          3140  5                ch$move (tablesize, .lnk$gl_curomd [omd$l_envmap], .first_256); ! AND COPY BASE TABLE OUT
;  2671          3141  5                ch$fill (0, tablesize, .lnk$gl_curomd [omd$l_envmap]);  ! ZERO BASE TABLE
;  2672          3142  5                first_entry[pmt$l_secpmt] = .first_256;
;  2673          3143  5                lnk$gl_curomd [omd$v_e256] = true;       ! FLAG INTO EXTENDED TABLE
;  2674          3144  4                end;
;  2675          3145  4
;  2676          3146  4            ch$fill (0, tablesize, .mapoff);     !  ZERO NEW BLOCK
;  2677          3147  4
;  2678          3148  4            !
;  2679          3149  4            ! POINT ENTRY IN BASE TABLE TO NEW BLOCK
;  2680          3150  4            !
;  2681          3151  4            mapent [pmt$l_secpmt] = .mapoff;
;  2682          3152  4            return .mapoff + .blockoff
;  2683          3153  3            end;
;  2684          3154  3
;  2685          3155  2        end;
;  2686          3156  2
;  2687          3157  1    end;


                                    0FFC 00000          .ENTRY  LNK$FNDENVMAP, Save R2,R3,R4,R5,R6,R7,R8,-      ; 3081
                                                                R9,R10,R11                                          ;
                              5B 00000000G   00  9E 00002       MOVAB   LNK$ALLOBLK, R11                            ;
                              5A 00000000G   00  9E 00009       MOVAB   LNK$GL_CUROMD, R10                          ;
                              5E             08  C2 00010       SUBL2   #8, SP                                      ;
                              50             6A  D0 00013       MOVL    LNK$GL_CUROMD, R0                           ; 3104
                                       18    A0  D5 00016       TSTL    24(R0)                                      ;
                                             17  12 00019       BNEQ    1$                                          ;
                                       18    A0  9F 0001B       PUSHAB  24(R0)                                      ; 3107
                                 7E    0800  8F  3C 0001E       MOVZWL  #2048, -(SP)                                ;
                                 6B          02  FB 00023       CALLS   #2, LNK$ALLOBLK                             ;
                                 50          6A  D0 00026       MOVL    LNK$GL_CUROMD, R0                           ; 3108
0800    8F                 00    6E          00  2C 00029       MOVC5   #0, (SP), #0, #2048, @24(R0)                ;
                                       18    B0     00030                                                        ;
                                 51          6A  D0 00032 1$:   MOVL    LNK$GL_CUROMD, R1                           ; 3111
                                 50    04    AC  D0 00035       MOVL    ENVNUM, R0                                  ;
                                 56    18 B140  7E 00039       MOVAQ   @24(R1)[R0], MAPENT                         ;
                                 0D    15    A1  E8 0003E       BLBS    21(R1), 2$                                  ; 3114
                    000000FF     8F          50  D1 00042       CMPL    R0, #255                                    ; 3115
                                             04  1A 00049       BGTRU   2$                                          ;
                                 50          56  D0 0004B       MOVL    MAPENT, R0                                  ; 3122
                                             04  0004E       RET                                                 ;
      7E         00              50          01  7A 0004F 2$:   EMUL    #1, R0, #0, -(SP)                           ; 3123
      52         52              8E 00000100 8F  7B 00054       EDIV    #256, (SP)+, R2, R2                         ;
                 59              52          03  78 0005D       ASHL    #3, R2, BLOCKOFF                            ;
                                 50 00000100 8F  C6 00061       DIVL2   #256, R0                                    ; 3124
                                 56    18 B140  7E 00068       MOVAQ   @24(R1)[R0], MAPENT                         ;
                                             66  D5 0006D       TSTL    (MAPENT)                                    ; 3127
                                             09  13 0006F       BEQL    3$                                          ;
                                 05    15    A1  E9 00071       BLBC    21(R1), 3$                                  ; 3128
                                 50          66  C1 00075       ADDL3   BLOCKOFF, (MAPENT), R0                      ; 3130
                                             04  00079       RET                                                 ; 3132
```

```
                                                  5E  DD 0007A 3$:     PUSHL   SP                                      ; 3133
                                          7E      8F  3C 0007C         MOVZWL  #2048, -(SP)                            ;
                                          6B      02  FB 00081         CALLS   #2, LNK$ALLOBLK                         ;
                                          57      6A  D0 00084         MOVL    LNK$GL_CUROMD, R7                       ; 3135
                                          2A  15  A7  E8 00087         BLBS    21(R7), 4$                              ;
                                          57      18  C0 0008B         ADDL2   #24, FIRST_ENTRY                        ; 3138
                                              04  AE  9F 0008E         PUSHAB  FIRST_256                               ; 3139
                                          7E  0800    8F  3C 00091     MOVZWL  #2048, -(SP)                            ;
                                          6B      02  FB 00096         CALLS   #2, LNK$ALLOBLK                         ;
                                          58      6A  D0 00099         MOVL    LNK$GL_CUROMD, R8                       ; 3140
                                  04  BE      18  B8  0800    8F  28 0009C     MOVC3   #2048, @24(R8), @FIRST_256      ;
    0800    8F      00          6E              00  2C 000A4         MOVC5   #0, (SP), #0, #2048, @24(R8)            ; 3141
                                                18  B8      000AB     ;
                                          67      04  AE  D0 000AD     MOVL    FIRST_256, (FIRST_ENTRY)                ; 3142
                                              15  A8          01  88 000B1     BISB2   #1, 21(R8)                      ; 3143
    0800    8F      00          6E              00  2C 000B5 4$:     MOVC5   #0, (SP), #0, #2048, @MAPOFF            ; 3146
                                                00  BE      000BC     ;
                                          66          6E  D0 000BE     MOVL    MAPOFF, (MAPENT)                        ; 3151
                          50              6E          59  C1 000C1     ADDL3   BLOCKOFF, MAPOFF, R0                    ; 3152
                                                      04  000C5         RET                                             ; 3157

; Routine Size:  198 bytes,    Routine Base:  $CODE$ + 199E

; 2688          3158  1
```

```
; 2690          3159  1 routine proenv =
; 2691          3160  1 !
; 2692          3161  1 !       THIS ROUTINE PROCESSES ENVIRONMENT DEFINITION/REFERENCE
; 2693          3162  1 !       SUBRECORDS.
; 2694          3163  1 !
; 2695          3164  2     begin
; 2696          3165  2     bind
; 2697          3166  2         envrec = objvec [.gsdoffset] : block [, byte];
; 2698          3167  2     builtin
; 2699          3168  2         remque;
; 2700          3169  2     local
; 2701          3170  2         ptr,
; 2702          3171  2         envmap : ref block [, byte],
; 2703          3172  2         envnode : ref block [, byte],
; 2704          3173  2         envdesc : ref block [, byte];
; 2705          3174  2
; 2706          3175  2     if .envrec [env$b_namlng] eql 0 or .envrec [env$b_namlng] gtru sym$c_maxlng
; 2707          3176  2     then
; 2708          3177  3         (signal (lin$_illnamelen, 6, lnk$gt_envstring, envrec [env$b_namlng], .envrec [env$b_namlng],
; 2709          3178  2             sym$c_maxlng, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]); return false);
; 2710          3179  2
; 2711          3180  2 !
; 2712          3181  2 ! INSERT ENVIRONMENT RECORD INTO ENVIRONMENT TREE
; 2713          3182  2 !
; 2714          3183  2     lib$insert_tree (lnk$gl_envtree, envrec [env$b_namlng], %ref (0), compare_env, alloc_env,
; 2715          3184  2                      envnode, envrec);
; 2716          3185  2     envdesc = .envnode + node$c_short;
; 2717          3186  2 !
; 2718          3187  2 ! PUT ENVIRONMENT INTO THE ENVIRONMENT MAP TABLE
; 2719          3188  2 !
; 2720          3189  2     obmodesc [omd$v_noenv] = false;
; 2721          3190  2     obmodesc [omd$w_hienv] = .obmodesc [omd$w_hienv] + 1;
; 2722          3191  2     envmap = lnk$fndenvmap (.obmodesc [omd$w_hienv]);
; 2723          3192  2     envmap [pmt$l_pscdes] = .envnode;
; 2724          3193  2
; 2725          3194  2 !IF .ENVMAP[PMT$L_SYMLST] NEQ 0                ! SYMBOLS DEFINED BEFORE THE ENVIRONMENT?
; 2726          3195  2 !THEN BEGIN                                !**ONLY NEED TO ASSOCIATE SYMBOLS WITH ENVIRONMENT FOR CROSS REF
; 2727          3196  2 !     TRUE
; 2728          3197  2 !     END;
; 2729          3198  2 !
; 2730          3199  2 ! IF THIS IS A DEFINITION, REMOVE FROM UNDEFINED LIST IF IT'S
; 2731          3200  2 ! ON IT.  IF IT'S THE FIRST REFERENCE TO THIS ENVIRONMENT,
; 2732          3201  2 ! THEN PUT IT ON THE UNDEFINED LIST.
; 2733          3202  2 !
; 2734          3203  2
; 2735          3204  2     if .envrec [env$v_def]
; 2736          3205  2     then
; 2737          3206  3         begin
; 2738          3207  3
; 2739          3208  3         if .envdesc [nvd$l_udflink] neq 0
; 2740          3209  3         then
; 2741          3210  4             (remque (envdesc [nvd$l_udflink], ptr);
; 2742          3211  3             lnk$gw_nudfenvs = .lnk$gw_nudfenvs - 1);
; 2743          3212  3
; 2744          3213  3         envdesc [nvd$v_def] = true;
; 2745          3214  3
; 2746          3215  3         if .envmap[pmt$l_symlst] neq 0
```

```
;  2747          3216  3          then
;  2748          3217  4              begin
;  2749          3218  4              envdesc[nvd$l_symtbl] = .envmap[pmt$l_symlst];
;  2750          3219  4              envmap[pmt$l_symlst] = 0;
;  2751          3220  3              end;
;  2752          3221  3          end
;  2753          3222  2      else
;  2754          3223  2
;  2755          3224  2          if not .envdesc [nvd$v_def] and .envdesc [nvd$l_udflink] eql 0
;  2756          3225  2          then
;  2757          3226  2              insudfenv (.envnode);
;  2758          3227  2
;  2759          3228  2      gsdoffset = .gsdoffset + envrec [env$t_name] - envrec [env$b_gsdtyp] + .envrec [env$b_namlng];
;  2760          3229  2      return true
;  2761          3230  1      end;
```

```
                                    003C 00000 PROENV:  .WORD   Save R2,R3,R4,R5                        ; 3159
                              55     FEB5   CF   9E 00002        MOVAB   ALLOC_ENV, R5
                              54 00000000'  EF   9E 00007        MOVAB   GSDOFFSET, R4
                              5E            08   C2 0000E        SUBL2   #8, SP
                              53            64   3C 00011        MOVZWL  GSDOFFSET, R3                  ; 3166
                              53       0C   A4   C0 00014        ADDL2   OBJVEC, R3
                              50       05   A3   9A 00018        MOVZBL  5(R3), R0                      ; 3175
                                            05   13 0001C        BEQL    1$
                              1F            50   91 0001E        CMPB    R0, #31
                                            2A   1B 00021        BLEQU   2$
                 7E 00000000G  00           14   C1 00023 1$:    ADDL3   #20, LNK$GL_CURFIL, -(SP)      ; 3178
                                    64      A4   9F 0002B        PUSHAB  OBMODESC+40
                                    1F      DD   9F 0002E        PUSHL   #31
                                    50      DD   9F 00030        PUSHL   R0
                                       05   A3   9F 00032        PUSHAB  5(R3)                          ; 3177
                                 00000000'  EF   9F 00035        PUSHAB  LNK$GT_ENVSTRING
                                            06   DD 00038        PUSHL   #6                             ; 3178
                                 00000000G  8F   DD 0003D        PUSHL   #LIN$_ILLNAMELEN
                    00000000G  00           08   FB 00043        CALLS   #8, LIB$SIGNAL
                                         0084   31 0004A        BRW     6$
                                            53   DD 0004D 2$:    PUSHL   R3                              ; 3183
                                       08   AE   9F 0004F        PUSHAB  ENVNODE
                                            55   DD 00052        PUSHL   R5
                                       DA   A5   9F 00054        PUSHAB  COMPARE_ENV
                                       10   AE   D4 00057        CLRL    16(SP)
                                       10   AE   9F 0005A        PUSHAB  16(SP)
                                       05   A3   9F 0005D        PUSHAB  5(R3)
                                 00000000'  EF   9F 00060        PUSHAB  LNK$GL_ENVTREE
                    00000000G  00           07   FB 00066        CALLS   #7, LIB$INSERT_TREE
                52       04   AE           0A   C1 0006D        ADDL3   #10, ENVNODE, ENVDESC          ; 3185
                         50   A4      80   8F   8A 00072        BICB2   #128, OBMODESC+20              ; 3189
                                      52   A4   B6 00077        INCW    OBMODESC+22                    ; 3190
                                   7E 52   A4   3C 0007A        MOVZWL  OBMODESC+22, -(SP)             ; 3191
                              7F   A5           01   FB 0007E        CALLS   #1, LNK$FNDENVMAP
                                   60      04   AE   D0 00082        MOVL    ENVNODE, (ENVMAP)          ; 3192
                                   20      01   A3   E9 00086        BLBC    1(R3), 4$                  ; 3204
                                               62   D5 0008A        TSTL    (ENVDESC)                  ; 3208
```

```
                                  09  13 0008C          BEQL    3$                          ;
                 51               62  0F 0008E          REMQUE  (ENVDESC), PTR              ; 3210
                       00000000'  EF  B7 00091          DECW    LNK$GW_NUDFENVS             ; 3211
         10      A2               01  88 00097 3$:      BISB2   #1, 16(ENVDESC)             ; 3213
                             04   A0  D5 0009B          TSTL    4(ENVMAP)                   ; 3215
                                  19  13 0009E          BEQL    5$                          ;
         08      A2          04   A0  D0 000A0          MOVL    4(ENVMAP), 8(ENVDESC)       ; 3218
                             04   A0  D4 000A5          CLRL    4(ENVMAP)                   ; 3219
                                  0F  11 000A8          BRB     5$                          ; 3204
                 0B          10   A2  E8 000AA 4$:      BLBS    16(ENVDESC), 5$             ; 3224
                                  62  D5 000AE          TSTL    (ENVDESC)                   ;
                                  07  12 000B0          BNEQ    5$                          ;
                             04   AE  DD 000B2          PUSHL   ENVNODE                     ; 3226
         3B      A5               01  FB 000B5          CALLS   #1, INSUDFENV               ;
                 50               64  3C 000B9 5$:      MOVZWL  GSDOFFSET, R0               ; 3228
                 50               53  C0 000BC          ADDL2   R3, R0                      ;
                 50               53  C2 000BF          SUBL2   R3, R0                      ;
                 51          05   A3  9A 000C2          MOVZBL  5(R3), R1                   ;
                 50               51  C0 000C6          ADDL2   R1, R0                      ;
    64           50               06  A1 000C9          ADDW3   #6, R0, GSDOFFSET           ;
                 50               01  D0 000CD          MOVL    #1, R0                      ; 3229
                                      04 000D0          RET                                 ;
                                  50  D4 000D1 6$:      CLRL    R0                          ; 3230
                                      04 000D3          RET                                 ;

; Routine Size:  212 bytes,    Routine Base:  $CODE$ + 1A64
```

```
;  2763          3231  1 routine localsymbols =
;  2764          3232  2 begin
;  2765          3233  2 local
;  2766          3234  2         length;
;  2767          3235  2 bind
;  2768          3236  2         symbolrec = objvec [.gsdoffset] : block [, byte];
;  2769          3237  2
;  2770          3238  2 entrymask = 0;                                          ! ZERO THE ENTRY MASK SINCE NONE EXISTS
;  2771          3239  2
;  2772          3240  2 if not .symbolrec [lsy$v_def]                          ! IF NOT DEFINED
;  2773          3241  3 then begin
;  2774          3242  3      symbolstring = symbolrec [lsrf$b_namlng];          ! POINT TO THE SYMBOL STRING
;  2775          3243  3      length       = symbolrec [lsrf$t_name] - symbolrec [lsrf$t_start] + .symbolrec [lsrf$b_namlng];
;  2776          3244  3      end
;  2777          3245  3 else begin
;  2778          3246  3      symbolstring = symbolrec [lsdf$b_namlng];          ! POINT TO THE SYMBOL
;  2779          3247  3      length       = symbolrec [lsdf$t_name] - symbolrec [lsdf$t_start] + .symbolrec [lsdf$b_namlng];
;  2780          3248  2      end;
;  2781          3249  2
;  2782          3250  2 if not prosymbol (0)                                   ! GO PROCESS THIS SYMBOL
;  2783          3251  2 then return false;                                    ! AND EXIT IF FAILURE
;  2784          3252  2
;  2785          3253  2 gsdoffset = .gsdoffset + .length;                     ! UPDATE THE GSD OFFSET FOR NEXT
;  2786          3254  2 return true
;  2787          3255  1 end;


                                000C 00000 LOCALSYMBOLS:
                                           .WORD   Save R2,R3                          ;  3231
                    53 00000000'   EF  9E  00002           MOVAB   GSDOFFSET, R3
                    52             63  3C  00009           MOVZWL  GSDOFFSET, R2       ;  3236
                    52       0C    A3  C0  0000C           ADDL2   OBJVEC, R2
                             02    A3  B4  00010           CLRW    ENTRYMASK           ;  3238
                16  02       A2        01  E0  00013       BBS     #1, 2(R2), 1$       ;  3240
                    10       A3  06    A2  9E  00018       MOVAB   6(R2), SYMBOLSTRING ;  3242
                50             52        52  C3  0001D     SUBL3   R2, R2, R0          ;  3243
                               51  06    A2  9A  00021     MOVZBL  6(R2), R1
                               50        51  C0  00025     ADDL2   R1, R0
                               52  07    A0  9E  00028     MOVAB   7(R0), LENGTH
                                         14  11  0002C     BRB     2$                  ;  3240
                    10       A3  0C    A2  9E  0002E 1$:   MOVAB   12(R2), SYMBOLSTRING;  3246
                50             52        52  C3  00033     SUBL3   R2, R2, R0          ;  3247
                               51  0C    A2  9A  00037     MOVZBL  12(R2), R1
                               50        51  C0  0003B     ADDL2   R1, R0
                               52  0D    A0  9E  0003E     MOVAB   13(R0), LENGTH
                                         7E  D4  00042 2$:   CLRL    -(SP)               ;  3250
                F116           CF        01  FB  00044     CALLS   #1, PROSYMBOL
                               07        50  E9  00049     BLBC    R0, 3$
                               63        52  A0  0004C     ADDW2   LENGTH, GSDOFFSET   ;  3253
                               50        01  D0  0004F     MOVL    #1, R0              ;  3254
                                         04  00052         RET
                                         50  D4  00053 3$:   CLRL    R0                  ;  3255
                                             04  00055     RET
```

```
; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 1B38
```

```
;  2789          3256  1 routine localentpnt =
;  2790          3257  2 begin
;  2791          3258  2 local
;  2792          3259  2         length;
;  2793          3260  2 bind
;  2794          3261  2         symbolrec = objvec [.gsdoffset] : block [, byte];
;  2795          3262  2
;  2796          3263  2 entrymask    = .symbolrec [lepm$w_mask];                          ! EXTRACT THE ENTRY POINT MASK
;  2797          3264  2 symbolstring =  symbolrec [lepm$b_naming];                       ! POINT TO THE SYMBOL
;  2798          3265  2 length       =  symbolrec [lepm$t_name] - symbolrec [lepm$t_start] + .symbolrec [lepm$b_naming];
;  2799          3266  2
;  2800          3267  2 if not prosymbol (1)                                           ! GO PROCESS THE SYMBOL
;  2801          3268  2 then return false;                                            ! AND EXIT IF FAILURE
;  2802          3269  2
;  2803          3270  2 gsdoffset = .gsdoffset + .length;                            ! ELSE UPDATE THE OFFSET FOR NEXT
;  2804          3271  2 return true
;  2805          3272  1 end;

                                000C 00000 LOCALENTPNT:
                                           .WORD   Save R2,R3                                          ; 3256
                     53 00000000'  EF  9E 00002    MOVAB   GSDOFFSET, R3                            ;
                     52            63  3C 00009    MOVZWL  GSDOFFSET, R2                            ; 3261
                     52        0C  A3  C0 0000C    ADDL2   OBJVEC, R2                               ;
                 02  A3        0C  A2  B0 00010    MOVW    12(R2), ENTRYMASK                        ; 3263
                 10  A3        0E  A2  9E 00015    MOVAB   14(R2), SYMBOLSTRING                     ; 3264
              50             52      52  C3 0001A    SUBL3   R2, R2, R0                               ; 3265
                             51        0E  A2  9A 0001E    MOVZBL  14(R2), R1                               ;
                             50            51  C0 00022    ADDL2   R1, R0                                   ;
                             52        0F  A0  9E 00025    MOVAB   15(R0), LENGTH                           ;
                                           01  DD C0029    PUSHL   #1                                       ; 3267
                           F0D9  CF        01  FB 0002B    CALLS   #1, PROSYMBOL                            ;
                                 07        50  E9 00030    BLBC    R0, 1$                                   ;
                                 63        52  A0 00033    ADDW2   LENGTH, GSDOFFSET                        ; 3270
                                 50        01  D0 00036    MOVL    #1, R0                                   ; 3271
                                           04  00039    RET                                              ;
                                           50  D4 0003A 1$:    CLRL    R0                                       ; 3272
                                           04  0003C    RET                                              ;

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 1B8E
```

```
; 2807          3273  1 routine localprocedef =
; 2808          3274  2     begin
; 2809          3275  2 !
; 2810          3276  2 !       PROCESS PROCEDURE DEFINTION OF MODULE LOCAL SYMBOL
; 2811          3277  2 !       SEE PROCEDEF FOR AN EXPLANATION OF WHAT A PROCEDURE
; 2812          3278  2 !       DEFINITION LOOKS LIKE.
; 2813          3279  2 !
; 2814          3280  2
; 2815          3281  2     local
; 2816          3282  2         entflag,
; 2817          3283  2         argvaldata : ref vector [, byte],
; 2818          3284  2         argcount;
; 2819          3285  2
; 2820          3286  2 !
; 2821          3287  2
; 2822          3288  2     if not localentpnt () then return false;    ! PROCESS THE ENTRY POINT SYMBOL PART
; 2823          3289  2
; 2824          3290  2     entflag = .symtabent neq 0;                 ! SEE IF ENTERED
; 2825          3291  3     begin
; 2826          3292  3
; 2827          3293  3     bind
; 2828          3294  3         formals = objvec [.gsdoffset] : block [, byte]; ! THE FORMAL DESCRIPTION
; 2829          3295  3
; 2830          3296  3     if .formals [fml$b_minargs] gtru .formals [fml$b_maxargs]   ! IF THERE IS AN INCONSISTENT
; 2831          3297  3     then
; 2832          3298  4         begin                                   ! DESCRIPTION OF FORMALS
; 2833          3299  4         signal (lin$_illfmlcnt, 5,              ! OUTPUT ERROR MESSAGE
; 2834          3300  4             .formals [fml$b_minargs], .formals [fml$b_maxargs], symentnam [snb$b_namlng],
; 2835          3301  4             obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
; 2836          3302  4         return false;
; 2837          3303  3         end;
; 2838          3304  3
; 2839          3305  3     gsdoffset = .gsdoffset + fml$c_size;        ! UPDATE RECORD POINTER
; 2840          3306  3
; 2841          3307  3     if .entflag                                 ! DELETE OLD VALIDATAION DATA
; 2842          3308  3         and (argvaldata = .symtabent [sym$l_valdata]) neq 0
; 2843          3309  3     then
; 2844          3310  4         begin                                   ! MUST BE A NEW DEFINITION
; 2845          3311  4         lnk$dealblk (.argvaldata [0], argvaldata [0]);  ! SO DISPOSE OF OLD DATA
; 2846          3312  4         symtabent [sym$l_valdata] = 0;          ! AND FORGET IT
; 2847          3313  3         end;
; 2848          3314  3
; 2849          3315  3     if (argcount = .formals [fml$b_maxargs]) eql 0      ! IF MAXIMUM NUMBER ARGUMENTS
; 2850          3316  3     then
; 2851          3317  3         return true;                            ! IS ZERO, ALL DONE
; 2852          3318  3
; 2853          3319  3     if .entflag
; 2854          3320  3     then
; 2855          3321  4         begin
; 2856          3322  4         lnk$alloblk (.argcount + 2, symtabent [sym$l_valdata]); ! ALLOCATE A VALIDATION DATA BLOCK
; 2857          3323  4         argvaldata [0] = .argcount + 2;         ! SET ITS SIZE
; 2858          3324  4         argvaldata [1] = .formals [fml$b_minargs];      ! SAVE MINIMUM ARGUMENT SPECIFICATION
; 2859          3325  3         end;
; 2860          3326  3
; 2861          3327  3     incr i from 1 to .argcount                  ! BEGIN A LOOP THROUGH
; 2862          3328  3         do
; 2863          3329  4         begin                                   ! ALL FORMAL ARGUMENT DESCRIPTORS
```

```
;  2864          3330  4
;  2865          3331  4          bind
;  2866          3332  4              argdesc = objvec [.gsdoffset] : block [, byte];     ! POINT TO NEXT DESCRIPTOR
;  2867          3333  4
;  2868          3334  4          if .entflag then argvaldata [.i + 1] = .argdesc [arg$v_passmech];
;  2869          3335  4
;  2870          3336  4                                                  ! EXTRACT THE PASSING MECHANISM
;  2871          3337  4          gsdoffset = .gsdoffset + .argdesc [arg$b_bytecnt] +     ! AND UPDATE RECORD POINTER
;  2872          3338  4          arg$c_size;
;  2873          3339  3          end;
;  2874          3340  3
;  2875          3341  3      return true;
;  2876          3342  2      end;
;  2877          3343  1      end;


                                        007C 00000 LOCALPROCEDEF:
                                                   .WORD   Save R2,R3,R4,R5,R6                          ; 3273
          56 00000000'  EF  9E 00002               MOVAB   GSDOFFSET, R6
    B6    AF            00  FB 00009               CALLS   #0, LOCALENTPNT                              ; 3288
          3F            50  E9 0000D               BLBC    R0, 2$
                        50  D4 00010               CLRL    R0                                           ; 3290
                  14    A6  D5 00012               TSTL    SYMTABENT
                        02  13 00015               BEQL    1$
                        50  D6 00017               INCL    R0
                    55        50  D0 00019 1$:     MOVL    R0, ENTFLAG
                    53        66  3C 0001C         MOVZWL  GSDOFFSET, R3                                ; 3294
                    53    0C  A6  C0 0001F         ADDL2   OBJVEC, R3
                01  A3        63  91 00023         CMPB    (R3), 1(R3)                                  ; 3296
                              28  1B 00027         BLEQU   3$
7E 00000000G  00              14  C1 00029         ADDL3   #20, LNK$GL_CURFIL, -(SP)                    ; 3301
                          64  A6  9F 00031         PUSHAB  OBMODESC+40
7E      18    A6              04  C1 00034         ADDL3   #4, SYMENTNAM, -(SP)                         ; 3300
              7E          01  A3  9A 00039         MOVZBL  1(R3), -(SP)                                 ; 3301
              7E              63  9A 0003D         MOVZBL  (R3), -(SP)
                              05  DD 00040         PUSHL   #5
                00000000G     8F  DD 00042         PUSHL   #LIB$_ILLFMLCNT
   00000000G    00            07  FB 00048         CALLS   #7, LIB$SIGNAL
                              74  11 0004F 2$:     BRB     10$                                          ; 3302
                    66        02  A0 00051 3$:     ADDW2   #2, GSDOFFSET                                ; 3305
                    1D        55  E9 00054         BLBC    ENTFLAG, 4$                                  ; 3307
                    50    14  A6  D0 00057         MOVL    SYMTABENT, R0                                ; 3308
                    52    18  A0  D0 0005B         MOVL    24(R0), ARGVALDATA
                              13  13 0005F         BEQL    4$
                              52  DD 00061         PUSHL   ARGVALDATA                                   ; 3311
                          7E      62  9A 00063     MOVZBL  (ARGVALDATA), -(SP)
   00000000G    00            02  FB 00066         CALLS   #2, LNK$DEALBLK
                    50    14  A6  D0 0006D         MOVL    SYMTABENT, R0                                ; 3312
                          18  A0  D4 00071         CLRL    24(R0)
                    54    01  A3  9A 00074 4$:     MOVZBL  1(R3), ARGCOUNT                              ; 3315
                              47  13 00078         BEQL    9$
                    1A            55  E9 0007A     BLBC    ENTFLAG, 5$                                  ; 3319
7E      14    A6              18  C1 0007D         ADDL3   #24, SYMTABENT, -(SP)                        ; 3322
                          02  A4  9F 00082         PUSHAB  2(ARGCOUNT)
```

```
                 00000000G  00          02  FB 00085          CALLS   #2, LNK$ALLOBLK                          ;
                            50      02  A4  9E 0008C          MOVAB   2(ARGCOUNT), R0                         ; 3323
                            62          50  90 00090          MOVB    R0, (ARGVALDATA)                        ;
                         01 A2          63  90 00093          MOVB    (R3), 1(ARGVALDATA)                     ; 3324
                                        51  D4 00097 5$:       CLRL    I                                       ; 3332
                                        22  11 00099          BRB     8$                                      ;
                            50          66  3C 0009B 6$:       MOVZWL  GSDOFFSET, R0                           ;
                            50      0C  A6  C0 0009E          ADDL2   OBJVEC, R0                              ;
                            0A          55  E9 000A2          BLBC    ENTFLAG, 7$                             ; 3334
          53        60      02          00  EF 000A5          EXTZV   #0, #2, (R0), R3                        ;
                         01 A142        53  90 000AA          MOVB    R3, 1(I)[ARGVALDATA]                    ;
                            53          66  3C 000AF 7$:       MOVZWL  GSDOFFSET, R3                           ; 3337
                            50      01  A0  9A 000B2          MOVZBL  1(R0), R0                               ;
                            50          53  C0 000B6          ADDL2   R3, R0                                  ;
                        66  50          02  A1 000B9          ADDW3   #2, R0, GSDOFFSET                       ;
                        DA  51          54  F3 000BD 8$:       AOBLEQ  ARGCOUNT, I, 6$                         ; 3327
                            50          01  D0 000C1 9$:       MOVL    #1, R0                                  ; 3341
                                            04 000C4          RET                                             ;
                                        50  D4 000C5 10$:      CLRL    R0                                      ; 3343
                                            04 000C7          RET                                             ;

; Routine Size:  200 bytes,    Routine Base:  $CODE$ + 1BCB
```

```
;  2879          3344  1 routine progsd =
;  2880          3345  2     begin
;  2881          3346  2 !
;  2882          3347  2
;  2883          3348  2 !++
;  2884          3349  2 !       VERIFY GSD RECORDS AND DISPATCH ON THE FOUR SUB-TYPES:
;  2885          3350  2 !               (0) P-SECTION DEFINITION
;  2886          3351  2 !               (1) SYMBOL DEFINITION/REFERENCE
;  2887          3352  2 !               (2) ENTRY POINT DEFINITION
;  2888          3353  2 !               (3) PROCEDURE DECLARATION
;  2889          3354  2 !--
;  2890          3355  2
;  2891          3356  2 !
;  2892          3357  2
;  2893          3358  2     bind
;  2894          3359  2         gsddispatch = plit (
;  2895          3360  2                           propsectdef,
;  2896          3361  2                           symbols,
;  2897          3362  2                           entpnts,
;  2898          3363  2                           procedef,
;  2899          3364  2                           symbols,
;  2900          3365  2                           entpnts,
;  2901          3366  2                           procedef,
;  2902          3367  2                           randentity,
;  2903          3368  2                           proenv,
;  2904          3369  2                           localsymbols,
;  2905          3370  2                           localentpnt,
;  2906          3371  2                           localprocedef,
;  2907          3372  2                           propsectdef) : vector;
;  2908          3373  2
;  2909          3374  2 !
;  2910          3375  2
;  2911          3376  2     local
;  2912          3377  2         gsdtype;
;  2913          3378  2
;  2914          3379  2 !
;  2915          3380  2
;  2916          3381  2     if not seqchk () then return false;
;  2917          3382  2
;  2918          3383  2     gsdoffset = obj$c_suotyp;
;  2919          3384  2 !
;  2920          3385  2
;  2921          3386  2     while .gsdoffset lssu .reclng do
;  2922          3387  3         begin
;  2923          3388  3
;  2924          3389  3         if (gsdtype = .objvec [.gsdoffset]) gequ .gsddispatch [-1]
;  2925          3390  3         then
;  2926          3391  4             begin
;  2927          3392  4             signal (lin$_gsdtyp, 3, .gsdtype, obmodesc [omd$b_namlng], lnk$gl_curfil [fdb$q_filename]);
;  2928          3393  4             return false;
;  2929          3394  4             end
;  2930          3395  3         else
;  2931          3396  4             begin
;  2932          3397  6             wordpsectgsd = ((.gsdtype gequ gsd$c_symw)  ! TEST FOR WORD OF PSECT NUMBER
;  2933          3398  4             and (.gsdtype lequ gsd$c_prow));
;  2934          3399  6             !clsymgsd = ((.gsdtype gequ gsd$c_lsy)      ! TEST FOR LOCAL SYMBOL
;  2935          3400  4             and (.gsdtype lequ gsd$c_lpro));
```

```
 4
 4          if not (.gsddispatch [.gsdtype]) () then return false;
 4
 3          end;
 3
 2       end;                                        ! OF LOOP ON A CONCATENATED GSD RECORD
 2
 2    return true                                    ! ALL DONE
 1    end;                                           ! OF PROGSD


                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        0006F     .BLKB   1
                              0000000D  00070     .LONG   13
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00074 P.AAK:  .ADDRESS PROPSECTDEF, SYMBOLS, ENTPNTS, PROCEDEF, -   ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0008C                  SYMBOLS, ENTPNTS, PROCEDEF, RANDENTITY, -   ;
                              00000000' 000A4                  PROENV, LOCALSYMBOLS, LOCALENTPNT, -       ;
                                                               LOCALPROCEDEF, PROPSECTDEF                 ;

                                        GSDDISPATCH=       P.AAK


                                                  .PSECT  $CODE$,NOWRT,2

                                  001C 00000 PROGSD: .WORD   Save R2,R3,R4                        ; 3344
              54 00000000'    EF    9E 00002         MOVAB   GSDOFFSET, R4                        ;
       E645   CF              00    FB 00009         CALLS   #0, SEQCHK                           ; 3381
              7C              50    E9 0000E         BLBC    R0, 7$                               ;
              64              01    B0 00011         MOVW    #1, GSDOFFSET                        ; 3383
         08   A4              64    B1 00014 1$:     CMPW    GSDOFFSET, RECLNG                    ; 3386
                              75    1E 00018         BGEQU   8$                                   ;
              50              64    3C 0001A         MOVZWL  GSDOFFSET, R0                        ; 3389
              50         0C   A4    C0 0001D         ADDL2   OBJVEC, R0                           ;
              52              60    9A 00021         MOVZBL  (R0), GSDTYPE                        ;
    00000000' EF              52    D1 00024         CMPL    GSDTYPE, GSDDISPATCH-4               ;
                              1E    1F 0002B         BLSSU   2$                                   ;
 7E 00000000G 00              14    C1 0002D         ADDL3   #20, LNK$GL_CURFIL, -(SP)            ; 3392
                         64   A4    9F 00035         PUSHAB  OBMODESC+40                          ;
                              52    DD 00038         PUSHL   GSDTYPE                              ;
                              03    DD 0003A         PUSHL   #3                                   ;
                   00000000G  8F    DD 0003C         PUSHL   #LIN$_GSDTYP                         ;
    00000000G 00              05    FB 00042         CALLS   #5, LIB$SIGNAL                       ;
                              48    11 00049         BRB     9$                                   ; 3393
                              51    D4 0004B 2$:     CLRL    R1                                   ; 3397
              04              52    D1 0004D         CMPL    GSDTYPE, #4                          ;
                              02    1F 00050         BLSSU   3$                                   ;
                              51    D6 00052         INCL    R1                                   ;
                              50    D4 00054 3$:     CLRL    R0                                   ; 3398
              06              52    D1 00056         CMPL    GSDTYPE, #6                          ;
                              02    1A 00059         BGTRU   4$                                   ;
                              50    D6 0005B         INCL    R0                                   ;
              53              51    D2 0005D 4$:     MCOML   R1, R3                               ;
   FA   A4    50              53    8B 00060         BICB3   R3, R0, WORDPSECTGSD                 ;
                              51    D4 00065         CLRL    R1                                   ; 3399
              09              52    D1 00067         CMPL    GSDTYPE, #9                          ;
```

```
                                 02  1F 0006A          BLSSU   5$                                          ;
                                 51  D6 0006C          INCL    R1                                          ;
                                 50  D4 0006E 5$:      CLRL    R0                                          ; 3400
                           0B    52  D1 00070          CMPL    GSDTYPE, #11                                ;
                                 02  1A 00073          BGTRU   6$                                          ;
                                 50  D6 00075          INCL    R0                                          ;
                           53    51  D2 00077 6$:      MCOML   R1, R3                                      ;
                     F9    A4    50  53  8B 0007A      BICB3   R3, R0, LCLSYMGSD                           ;
                           50 00000000'EF42  D0 0007F   MOVL    GSDDISPATCH[GSDTYPE], R0                    ; 3402
                           60        00  FB 00087      CALLS   #0, (R0)                                    ;
                           87        50  E8 0008A      BLBS    R0, 1$                                      ;
                                     04  11 000P: 7$:   BRB     9$                                          ;
                           50        01  D0 000oF 8$:   MOVL    #1, R0                                      ; 3408
                                     04  00092          RET                                                 ;
                                 50  D4 00093 9$:      CLRL    R0                                          ; 3409
                                     04  00095          RET                                                 ;

; Routine Size:  150 bytes,    Routine Base:  $CODE$ + 1C93

; 2945          3410  1
```

```
;  2947          3411  1 global routine lnk$procsobj (modrfa) =
;  2948          3412  1 !
;  2949          3413  1 ! READ AND PROCESS ALL REQUIRED OBJECT MODULE RECORDS OF THE FILE JUST OPENED
;  2950          3414  1 ! THAT IS, KEEP READING RECORDS TO END OF FILE EXCEPT IF THIS FILE IS A LIBRARY,
;  2951          3415  1 ! STOP AT THE FIRST END OF MODULE RECORD.
;  2952          3416  1 !
;  2953          3417  1 ! MODRFA EXISTS IN THE CASE OF A LIBRARY MODULE OR SHAREABLE IMAGE AND
;  2954          3418  1 ! IS A POINTER TO THE RECORD FILE ADDRESS OF THE MODULE.
;  2955          3419  1 !
;  2956          3420  2 begin
;  2957          3421  2 map
;  2958          3422  2         modrfa : ref block [, byte] ;                          ! REALLY A POINTER
;  2959          3423  2 bind
;  2960          3424  2         auxfnb = lnk$gl_curfil [fdb$t_auxfnb] : block [, byte]; ! AUXILIRIARY FILENAME BLOCK
;  2961          3425  2
;  2962          3426  2 while lnk$nxtrec (objrecdesc)                                 ! IF THERE IS ANOTHER RECORD
;  2963          3427  3 do  begin
;  2964          3428  3     local       status ;
;  2965          3429  3
;  2966          3430  3     if .reclng gtru .maxreclng                               ! AND ITS LENGTH IS ILLEGAL
;  2967          3431  4     then begin
;  2968          3432  4          if .mhdseen
;  2969          3433  4          then signal (lin$_illreclen, 3                      ! TELL MODULE NAME IF IT IS KNOWN
;  2970          3434  4                      ,.reclng, obmodesc [omd$b_namlng]
;  2971          3435  4                      ,lnk$gl_curfil [fdb$q_filename]
;  2972          3436  4                      )
;  2973          3437  4          else signal (lin$_reclng, 2                         ! OUTPUT AN ERROR AND QUIT ON SEVERE ERROR
;  2974          3438  4                      ,lnk$gl_curfil [fdb$q_filename]
;  2975          3439  4                      ,.reclng
;  2976          3440  4                      ) ;
;  2977          3441  4          end
;  2978          3442  4     else begin                                               ! LENGTH OF RECORD OR SO
;  2979          3443  4          lastrectyp = .currectyp ;                           ! COPY OLD CURRENT TO LAST TYPE
;  2980          3444  4          currectyp  = .objrec [obj$b_rectyp] ;               ! AND GET NEW TYPE
;  2981          3445  4          if .currectyp lssu .recdispatch [-1]                ! CHECK IT IS WITH RANGE
;  2982          3446  5          and not (not .mhdseen and (.currectyp neq obj$c_hdr)) ! MUST BE "hdr" IF HEADER NOT SEEN YET
;  2983          3447  4          then (.recdispatch [.currectyp]) ()                 ! DISPATCH TO RECORD SPECIFIC ROUTINE
;  2984          3448  5          else begin
;  2985          3449  5               if .mhdseen
;  2986          3450  5               then signal (lin$_illrectyp, 4                 ! TELL MODULE NAME IF KNOWN
;  2987          3451  5                           ,.currectyp, obmodesc [omd$b_namlng]
;  2988          3452  5                           ,.lnk$gl_record, lnk$gl_curfil [fdb$q_filename]
;  2989          3453  5                           )
;  2990          3454  5               else signal (lin$_rectyp, 3                    ! ELSE OUTPUT ERROR
;  2991          3455  5                           ,lnk$gl_curfil [fdb$q_filename]
;  2992          3456  5                           ,.lnk$gl_record, .currectyp
;  2993          3457  5                           ) ;
;  2994          3458  4               end ;
;  2995          3459  3          end ;
;  2996          3460  3
;  2997          3461  4     if ((.currectyp eql obj$c_eom)   or   (.currectyp eql obj$c_eomw))  ! IF THAT RECORD WAS AN EOM
;  2998          3462  3         and                                                  ! OR CONTAINS A SHAREABLE IMAGE
;  2999          3463  4         (.lnk$gl_curfil [fdb$v_libr] or .lastobjmod [omd$v_shrimg])     ! AND THE FILE IS A LIBRARY
;  3000          3464  3     then
;  3001          3465  4         begin                                                ! THEN THE FILE IS DONE
;  3002          3466  4         lastobjmod [omd$l_modvbn] = .modrfa [rfa$l_vbn] ;    ! AS SOON AS WE EXTRACT THE RFA
;  3003          3467  4         lastobjmod [omd$w_bytoff] = .modrfa [rfa$w_offset] ; ! OF THE MODULE
```

```
;3004     3468  4          return true ;
;3005     3469  3          end ;
;3006     3470  2      end ;                                                  ! OF RECORDS LOOP
;3007     3471  2
;3008     3472  2 !
;3009     3473  2 !    ALL OBJECT RECORDS HAVE BEEN READ AND PROCESSED
;3010     3474  2 !
;3011     3475  3 if  (.currectyp neq obj$c_eom)
;3012     3476  3 and (.currectyp neq obj$c_eomw)
;3013     3477  3 then begin
;3014     3478  3      local      eomrec : block [eom$c_eommin, byte] ;
;3015     3479  3
;3016     3480  3      signal (lin$_noeom, 2
;3017     3481  3             ,obmodesc [omd$b_namlng]                        ! IF DID NOT END WITH EOM, THEN WARNING
;3018     3482  3             ,lnk$gl_curfil [fdb$q_filename]
;3019     3483  3             ) ;
;3020     3484  3      eomrec [eom$b_rectyp] = obj$c_eom ;                     ! CREATE FAKE EOM RECORD
;3021     3485  3      eomrec [eom$b_comcod] = eom$c_error;
;3022     3486  3      reclng    = 2 ;                                         ! SET LENGTH OF RECORD
;3023     3487  3      objrec    = eomrec ;                                    ! AND IT'S ADDRESS
;3024     3488  3      currectyp = obj$c_eom ;                                 ! SET RECORD TYPE
;3025     3489  3      proeom (1) ;                                            ! FINISH EOM PROCESSING
;3026     3490  3      mhdseen   = false ;                                     ! RESET MHD FLAGS
;3027     3491  3      lnmseen   = false ;
;3028     3492  2      end ;
;3029     3493  2
;3030     3494  2 return true ;                                                ! FINALLY RETURN AFTER NO MORE
;3031     3495  1 end ;                                                        ! OF PASS 1 OBJ READING ROUTINE
```

```
                                 003C 00000          .ENTRY  LNK$PROCSOBJ, Save R2,R3,R4,R5          ;3411
                    55 00000000G  00   9E 00002      MOVAB   LIB$SIGNAL, R5                          ;
                    54 00000000G  00   9E 00009      MOVAB   LNK$GL_CURFIL, R4                       ;
                    53 00000000'  EF   9E 00010      MOVAB   CURRECTYP, R3                           ;
                    5E            04   C2 00017      SUBL2   #4, SP                                  ;
                          0A      A3   9F 0001A 1$:   PUSHAB  OBJRECDESC                             ;3426
          00000000G 00            01   FB 0001D      CALLS   #1, LNK$NXTREC                          ;
                    03            50   E8 00024      BLBS    R0, 2$                                  ;
                                00BE   31 00027      BRW     13$                                     ;
                    51      0A    A3   3C 0002A 2$:   MOVZWL  RECLNG, R1                            ;3430
                    51      06    A3   B1 0002E      CMPW    MAXRECLNG, R1                           ;
                                  28   1E 00032      BGEQU   4$                                      ;
          50        64            14   C1 00034      ADDL3   #20, LNK$GL_CURFIL, R0                  ;3435
                    11      FD    A3   E9 00038      BLBC    MHDSEEN, 3$                             ;3432
                                  50   DD 0003C      PUSHL   R0                                      ;3435
                          66      A3   9F 0003E      PUSHAB  OBMODESC+40                             ;3434
                                  51   DD 00041      PUSHL   R1                                      ;3435
                                  03   DD 00043      PUSHL   #3                                      ;
                       00000000G  8F   DD 00045      PUSHL   #LIN$_ILLRECLEN                         ;
                                  67   11 0004B      BRB     8$                                      ;
                                  03   BB 0004D 3$:   PUSHR   #^M<R0,R1>                             ;3438
                                  02   DD 0004F      PUSHL   #2                                      ;
                       00000000G  8F   DD 00051      PUSHL   #LIN$_RECLNG                            ;
                    65            04   FB 00057      CALLS   #4, LIB$SIGNAL                          ;
```

```
                                 5B  11 0005A          BRB     9$                                      ; 3430
                              FF 63  90 0005C 4$:       MOVB    CURRECTYP, LASTRECTYP                   ; 3443
                              63 0E  B3 90 00060        MOVB    @OBJREC, CURRECTYP                      ; 3444
                              52     63 9A 00064        MOVZBL  CURRECTYP, R2                           ; 3445
                    00000000' EF     52 D1 00067        CMPL    R2, RECDISPATCH-4                       ;
                                     15 1E 0006E        BGEQU   6$                                      ;
                              04 FD  A3 E8 00070        BLBS    MHDSEEN, 5$                             ; 3446
                                     52 D5 00074        TSTL    R2                                      ;
                                     0D 12 00076        BNEQ    6$                                      ;
                    50 00000000'EF42 D0 00078 5$:       MOVL    RECDISPATCH[R2], R0                     ; 3447
                    60               00 FB 00080        CALLS   #0, (R0)                                ;
                                     32 11 00083        BRB     9$                                      ;
                 51                  64 C1 00085 6$:    ADDL3   #20, LNK$GL_CURFIL, R1                  ; 3452
                    50 00000000G     00 D0 00089        MOVL    LNK$GL_RECORD, R0                       ;
                    14 FD            A3 E9 00090        BLBC    MHDSEEN, 7$                             ; 3449
                                     03 BB 00094        PUSHR   #^M<R0,R1>                              ; 3452
                              66     A3 9F 00096        PUSHAB  OBMODESC+40                             ; 3451
                                     52 DD 00099        PUSHL   R2                                      ; 3452
                                     04 DD 0009B        PUSHL   #4                                      ;
                       00000000G     8F DD 0009D        PUSHL   #LIN$_ILLRECTYP                         ;
                    65               06 FB 000A3        CALLS   #6, LIB$SIGNAL                          ;
                                     0F 11 000A6        BRB     9$                                      ;
                                     05 BB 000A8 7$:    PUSHR   #^M<R0,R2>                              ; 3456
                                     51 DD 000AA        PUSHL   R1                                      ; 3455
                                     03 DD 000AC        PUSHL   #3                                      ;
                       00000000G     8F DD 000AE        PUSHL   #LIN$_RECTYP                            ;
                    65               05 FB 000B4 8$:    CALLS   #5, LIB$SIGNAL                          ;
                    03               63 91 000B7 9$:    CMPB    CURRECTYP, #3                           ; 3461
                                     08 13 000BA        BEQL    11$                                     ;
                    07               63 91 000BC        CMPB    CURRECTYP, #7                           ;
                                     03 13 000BF        BEQL    11$                                     ;
                                 FF56 31 000C1 10$:     BRW     1$                                      ;
                       50            64 D0 000C4 11$:   MOVL    LNK$GL_CURFIL, R0                       ; 3463
                 09 0A A0            01 E0 000C7        BBS     #1, 10(R0), 12$                         ;
                       50     F6     A3 D0 000CC        MOVL    LASTOBJMOD, R0                          ;
                 EC 14 A0            02 E1 000D0        BBC     #2, 20(R0), 10$                         ;
                       51     F6     A3 D0 000D5 12$:   MOVL    LASTOBJMOD, R1                          ; 3466
                       50     04     AC D0 000D9        MOVL    MODRFA, R0                              ;
                    0C A1             60 D0 000DD        MOVL    (R0), 12(R1)                            ;
                    10 A1     04      A0 B0 000E1        MOVW    4(R0), 16(R1)                           ; 3467
                                     36 11 000E6        BRB     14$                                     ; 3468
                    03               63 91 000E8 13$:   CMPB    CURRECTYP, #3                           ; 3475
                                     31 13 000EB        BEQL    14$                                     ;
                    07               63 91 000ED        CMPB    CURRECTYP, #7                           ; 3476
                                     2C 13 000F0        BEQL    14$                                     ;
                 7E                  64 C1 000F2        ADDL3   #20, LNK$GL_CURFIL, -(SP)               ; 3482
                              66     A3 9F 000F6        PUSHAB  OBMODESC+40                             ; 3481
                                     02 DD 000F9        PUSHL   #2                                      ; 3482
                       00000000G     8F DD 000FB        PUSHL   #LIN$_NOEOM                             ;
                    65               04 FB 00101        CALLS   #4, LIB$SIGNAL                          ;
                    6E     0203      8F B0 00104        MOVW    #515, EOMREC                            ; 3484
                 0A A3               02 B0 00109        MOVW    #2, RECLNG                              ; 3486
                 0E A3               6E 9E 0010D        MOVAB   EOMREC, OBJREC                          ; 3487
                    63               03 90 00111        MOVB    #3, CURRECTYP                           ; 3488
                                     01 DD 00114        PUSHL   #1                                      ; 3489
                F42A CF              01 FB 00116        CALLS   #1, PROEOM                              ;
                              FD     A3 B4 0011B        CLRW    MHDSEEN                                 ; 3490
```

```
                                50          01  D0 0011E 14$:   MOVL    #1, R0                          ; 3494
                                                04 00121        RET                                     ; 3495

; Routine Size:  290 bytes,    Routine Base:  $CODE$ + 1D29

; 3032          3496  1
```

```
;  3034     3497  1 global routine lnk$objpass1 (arglist) =
;  3035     3498  1 !
;  3036     3499  1 !       THIS IS THE MAIN DRIVING ROUTINE OF PASS ONE. IT OPENS
;  3037     3500  1 !       EACH FILE AND IF A LIBRARY, CALLS THE LIBRARY PROCESSING ROUTINE
;  3038     3501  1 !       OR IF NOT A LIBRARY ASSUMES IT IS A CONCATENATED OBJ FILE
;  3039     3502  1 !       AND CALLS THE OBJ MODULE PROCESSOR.
;  3040     3503  1 !       WHEN NO MORE FILES REMAIN, IF THERE ARE STILL UNDEFINED SYMBOLS
;  3041     3504  1 !       THE SYSTEM LIBRARY IS MATERAILIZED AS AN INPUT FILE AND
;  3042     3505  1 !       PROCESSED.
;  3043     3506  1 !
;  3044     3507  1 !       ARGLIST CONTAINS THE ADDRESS OF THE ARGUMENT LIST WITH WHICH THE LINKER
;  3045     3508  1 !       WAS ORIGINALLY CALLED. IT IS MERELY PASSED ON TO THE OBJECT
;  3046     3509  1 !       LIBRARY PROCESSOR SO IT CAN RE-CALL CLI TO OBTAIN THE NAMES (IF ANY)
;  3047     3510  1 !       OF MODULES TO BE EXPLICITLY EXTRACTED FROM THE LIBRARY.
;  3048     3511  1 !
;  3049     3512  2 begin
;  3050     3513  2 builtin
;  3051     3514  2         insque,                                 ! QUE INSERTION AND
;  3052     3515  2         remque;                                 ! QUE REMOVING INSTRUCTIONS
;  3053     3516  2 local
;  3054     3517  2         suffix,                                 ! UNDEFINED SYMBOL LIST SUFFIX POINTER
;  3055     3518  2         nxtsymnam : ref block [, byte],         ! NEXT SYMBOL NAME BLOCK POINTER
;  3056     3519  2         nxtsyment : ref block [, byte];         ! FOR SCANNING LIST OF UNDEFINED SYMBOLS
;  3057     3520  2
;  3058     3521  2 !
;  3059     3522  2 !   INSERT THE SYMBOL "SYS$K_VERSION" INTO THE SYMBOL TABLE AS AN UNDEFINED
;  3060     3523  2 !   WEAK ABSOLUTE REFERENCE WITH THE SPECIAL INTERNALLY CREATED SYMBOL
;  3061     3524  2 !   FLAG.
;  3062     3525  2 !
;  3063     3526  2 lnk$search (lnk$gt_sysver, symtabent, symentnam);          ! FIND WHERE IT GOES
;  3064     3527  2 lnk$insert (lnk$gt_sysver, symtabent, symentnam);          ! AND INSERT IT
;  3065     3528  2 symtabent [sym$v_weak]   = true;                          ! FLAG WEAK
;  3066     3529  2 symtabent [sym$v_intsym] = true;                          ! AND INTERNALLY CREATED
;  3067     3530  2 lnk$insudfsym (.symtabent);                               ! INSERT IN UDEFINED SYMBOL LIST
;  3068     3531  2 obmodesc [omd$l_lsterr]  = obmodesc [omd$l_errtxt];
;  3069     3532  2 lnk$gl_curomd            = obmodesc;                      ! POINT TO THE DESCRIPTOR
;  3070     3533  2
;  3071     3534  2 !
;  3072     3535  2 !   BEGIN THE LOOP WHICH OPENS EACH INPUT FILE
;  3073     3536  2 !
;  3074     3537  2 while lnk$nxtfil ()                                      ! WHILE THERE REMAIN
;  3075     3538  3 do  begin                                                ! MORE OBJ FILES
;  3076     3539  3     last_lnkopt_fdb = 0;                                ! Haven't seen linker opt recs in this file
;  3077     3540  3     lnk$gw_nfiles   = .lnk$gw_nfiles + 1;               ! COUNT THE FILE
;  3078     3541  3     lastobjmod      = lnk$gl_curfil [fdb$l_omdlst];     ! INITIALIZE LAST DESCRIPTOR TO LISTHEAD
;  3079     3542  3
;  3080     3543  4     if not (if  .lnk$gl_curfil [fdb$v_libr]             ! IF THIS INPUT FILE IS A LIBRARY
;  3081     3544  4             then lnk$procslib (.arglist)                ! GO PROCESS IT
;  3082     3545  4             else if  .lnk$gl_curfil [fdb$v_shr]         ! IF A SHAREABLE IMAGE
;  3083     3546  4                  then lnk$procshrim ()                  ! GO PROCESS IT
;  3084     3547  4                  else lnk$procsobj ()                   ! OTHERWISE ASSUME CONCATENATED OBJECT
;  3085     3548  4            )
;  3086     3549  3     then return false;                                  ! EXIT NOW ON FAILURE
;  3087     3550  3
;  3088     3551  3     lastobjmod [omd$l_nxtomd] = 0;                      ! END OF USER INPUT FILES
;  3089     3552  2     end;                                                ! END OF USER INPUT FILES
;  3090     3553  2
```

```
;  3091     3554  2 !
;  3092     3555  2 !   NOW CHECK THAT WE DID GET SOME MODULES AND IF SO
;  3093     3556  2 !   CHECK FOR UNDEFINED SYMBOLS.
;  3094     3557  2 !
;  3095     3558  2 lnk$gl_curomd = 0;                                          ! NO OBJECT MODULE NOW
;  3096     3559  2
;  3097     3560  2 if .lnk$gw_nmodules eql 0 then signal_stop (lin$_nomods);
;  3098     3561  2
;  3099     3562  2 if .lnk$gw_nudfsyms neq 0                                   ! IF THERE ARE STILL SOME UNDEFINED (STRONG
;  3100     3563  2 then signal (lin$_nudfsyms, 1, .lnk$gw_nudfsyms);           ! THEN REPORT THE NUMBER
;  3101     3564  2
;  3102     3565  2 while not remque (.lnk$gl_udflst, nxtsyment)                ! AND REMOVE EACH FROM TOP
;  3103     3566  3 do  begin                                                   ! OF THE LIST, OUTPUTTING
;  3104     3567  3     bind nxtsymnam = .nxtsyment - .nxtsyment [sym$b_naming] ! POINT TO SYMBOL NAME BLOCK
;  3105     3568  3                      - snb$c_fxdlen  : block [, byte];
;  3106     3569  3
;  3107     3570  3     if not .nxtsyment [sym$v_intsym]
;  3108     3571  4     then begin
;  3109     3572  4          if not .nxtsyment [sym$v_weak]                     ! THE SYMBOL NAME
;  3110     3573  4          then suffix = cstring ('')                         ! WITH NULL OR 'WEAK'
;  3111     3574  5          else begin
;  3112     3575  5               suffix = cstring (' (Weak Reference)');       ! SUFFIX DEPENDING ON THE REFERENCE
;  3113     3576  5               nxtsyment [sym$v_def] = true;                 ! BEING WEAK OR STRONG AND
;  3114     3577  4               end;                                          ! ENSURE WEAKS ARE NOW DEFINED
;  3115     3578  4
;  3116     3579  4          if .lnk$gw_nudfsyms neq 0                          ! NOW LIST THE SYMBOL PROVIDED
;  3117     3580  4          then signal (lin$_udfsym, 2                        ! NOT ALL WERE WEAK
;  3118     3581  4                      ,nxtsymnam [snb$b_naming], .suffix
;  3119     3582  4                      );
;  3120     3583  4
;  3121     3584  4          nxtsyment [sym$l_value] = 0;                       ! SET ITS VALUE TO 0
;  3122     3585  4          nxtsyment [sym$v_rel]   = false;                   !  AND MAKE IT ABSOLUTE
;  3123     3586  4
;  3124     3587  4          if not .lnk$gl_ctlmsk [lnk$v_brief]                ! PROVIDED THIS IS NOT A
;  3125     3588  4          and    .lnk$gl_ctlmsk [lnk$v_map]                  ! BRIEF MAP
;  3126     3589  4          then crf$insrtkey ( lnk$al_sytblfmt                ! INSERT THE KEY (SYMBOL) IN
;  3127     3590  4                            , nxtsymnam [snb$b_naming]       ! CREF'S TABLE SO THAT WE MAY
;  3128     3591  4                            , nxtsyment [sym$l_value]        ! MAP UNDEFINED SYMBOLS
;  3129     3592  4                            ,.nxtsyment [sym$w_flags]        ! AND IGNORE ERROR FROM CREF
;  3130     3593  4                            ) ;
;  3131     3594  4          end                                                ! NOT INTERNALLY CREATED SYMBOL
;  3132     3595  4     else begin
;  3133     3596  4          nxtsyment [sym$l_value] = 0;                       ! INTERNALLY CREATED--ZERO VALUE
;  3134     3597  4          nxtsyment [sym$v_rel]   = false;                   !  AND MAKE IT ABSOLUTE
;  3135     3598  3          end;
;  3136     3599  2     end;
;  3137     3600  2
;  3138     3601  2 !
;  3139     3602  2 !   LIST ANY UNDEFINED ENVIRONMENTS
;  3140     3603  2 !
;  3141     3604  2 if .lnk$gw_nudfenvs neq 0
;  3142     3605  3 then begin
;  3143     3606  3      local      envdesc : ref block [, byte];
;  3144     3607  3
;  3145     3608  3      signal (lin$_nudfenvs, 1, .lnk$gw_nudfenvs);
;  3146     3609  3
;  3147     3610  3      envdesc = .lnk$gl_udfenv [0];
```

```
;  3148          3611  3      while .envdesc neq lnk$gl_udfenv
;  3149          3612  4      do begin
;  3150          3613  4         signal (lin$_udfenv, 1, envdesc [nvd$b_namlng]);
;  3151          3614  4         envdesc = .envdesc [nvd$l_udflink];
;  3152          3615  3         end;
;  3153          3616  2      end;
;  3154          3617  2
;  3155          3618  2 !
;  3156          3619  2 !    LIST ANY UNDEFINED MODULE-LOCAL SYMBOLS
;  3157          3620  2 !
;  3158          3621  2 if .lnk$gw_nudflsyms neq 0
;  3159          3622  3 then begin
;  3160          3623  3      signal (lin$_nudflsyms, 1, .lnk$gw_nudflsyms);          ! TELL THAT THERE ARE UNDEF. LOCAL SYMBOLS
;  3161          3624  3
;  3162          3625  3      while not remque (.lnk$gl_udflsy, nxtsyment)            ! FOR EACH ONE OF THEM
;  3163          3626  4      do begin
;  3164          3627  4         bind nxtsymnam = .nxtsyment - .nxtsyment [sym$b_namlng] - snb$c_fxdlen : block [, byte];
;  3165          3628  4
;  3166          3629  4         if not .nxtsyment [sym$v_weak]                       ! SET SUFFIX TO SYMBOL MESSAGE
;  3167          3630  4         then suffix = cstring ('')
;  3168          3631  5         else begin
;  3169          3632  5              suffix = cstring (' (Weak Reference)');       ! IF IT'S WEAK, THEN SAY SO
;  3170          3633  5              nxtsyment [sym$v_def] = true;
;  3171          3634  4              end;
;  3172          3635  4
;  3173          3636  4         signal (lin$_udfsym, 2, nxtsymnam [snb$b_namlng], .suffix);     ! TELL USER OF UNDEFINED LOCAL SYMBO
;  3174          3637  4         nxtsyment [sym$l_value] = 0;                         ! DEFINE IT AS ABSOLUTE 0
;  3175          3638  4         nxtsyment [sym$v_rel]   = false;
;  3176          3639  3         end;
;  3177          3640  2      end;
;  3178          3641  2
;  3179          3642  2 if .lnk$gl_dbgtfps eql 0                                   ! IF NO DEBUGGER TRANSFER ADDRESS SEEN
;  3180          3643  2 then if lnk$search (lnk$gt_imgsta, symtabent, symentnam)     ! THEN LOOK UP SYS$IMGSTA
;  3181          3644  2      then if .symtabent [sym$v_def]                         ! AND IF DEFINED
;  3182          3645  3           then  begin
;  3183          3646  3                 lnk$gl_dbgtfr  = .symtabent [sym$l_value];    ! SET ITS ADDRESS
;  3184          3647  3                 lnk$gl_dbgtfps = 1;                           ! FLAG THERE IS A DEBUG XFR ADDRESS
;  3185          3648  2                 end;
;  3186          3649  2
;  3187          3650  2 !
;  3188          3651  2 !    DEFINE ANY PSECTS SPECIFIED IN A PSECT= OPTION THAT WERE NOT
;  3189          3652  2 !    SPECIFIED IN A COLLECT PSECT OPTION
;  3190          3653  2 !
;  3191          3654  2 lnk$gl_curclu = lnk$gl_defclu;                             ! SET TO DEFAULT CLUSTER
;  3192          3655  2 lib$traverse_tree (lnk$gl_pscdflst, checkpddentry);        ! TRAVERSE THE TREE
;  3193          3656  2
;  3194          3657  2 return true;
;  3195          3658  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 01  000A8 P.AAL:  .BYTE   1                                        ;
                                                 20  000A9         .ASCII  \ \                                       ;
                                                 11  000AA P.AAM:  .BYTE   17                                       ;
63  6E  65  72  65  66  65  52  20  6B  61  65  57  28  20  000AB         .ASCII  \ (Weak Reference)\                       ;
```

```
                                                  29  65  000BA                                              ;
                                                      01  000BC P.AAN:   .BYTE   1                           ;
                                                      20  000BD          .ASCII  \ \                         ;
                                                      11  000BE P.AAO:   .BYTE   17                          ;
63 6E 65 72 65 66 65 52 20 6B 61 65 57            28  20  000BF          .ASCII  \ (Weak Reference)\         ;
                                                  29  65  000CE                                              ;

                                                                         .PSECT  $CODE$,NOWRT,2

                                                   0FFC  00000          .ENTRY  LNK$OBJPASS1, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 3497
                                                                                 R10,R11                     ;
                           5B 00000000G     00    9E  00002          MOVAB   LNK$SEARCH, R11             ;
                           5A       E417    CF    9E  00009          MOVAB   LNK$INSUDFSYM, R10          ;
                           59 00000000G     00    9E  0000E          MOVAB   LIB$SIGNAL, R9              ;
                           58 00000000'     EF    9E  00015          MOVAB   LNK$GT_SYSVER, R8           ;
                           57 00000000'     EF    9E  0001C          MOVAB   LNK$GW_NUDFSYMS, R7         ;
                           56 00000000'     EF    9E  00023          MOVAB   SYMTABENT, R6               ;
                                   04       A6    9F  0002A          PUSHAB  SYMENTNAM                   ; 3526
                                            56    DD  0002D          PUSHL   R6                          ;
                                            58    DD  0002F          PUSHL   R8                          ;
                           6B               03    FB  00031          CALLS   #3, LNK$SEARCH              ;
                                   04       A6    9F  00034          PUSHAB  SYMENTNAM                   ; 3527
                                            56    DD  00037          PUSHL   R6                          ;
                                            58    DD  00039          PUSHL   R8                          ;
                 00000000G 00               03    FB  0003B          CALLS   #3, LNK$INSERT              ;
                           50               66    D0  00042          MOVL    SYMTABENT, R0               ; 3528
                       0A  A0     0401      8F    A8  00045          BISW2   #1025, 10(R0)               ; 3529
                                            50    DD  0004B          PUSHL   R0                          ; 3530
                           6A               01    FB  0004D          CALLS   #1, LNK$INSUDFSYM           ;
                       4C  A6       48      A6    9E  00050          MOVAB   OBMODESC+32, OBMODESC+36    ; 3531
                 00000000G 00       28      A6    9E  00055          MOVAB   OBMODESC, LNK$GL_CUROMD     ; 3532
                 00000000G 00               00    FB  0005D 1$:      CALLS   #0, LNK$NXTFIL              ; 3537
                           45               50    E9  00064          BLBC    R0, 7$                      ;
                                  0870      C6    D4  00067          CLRL    LAST_LNKOPT_FDB             ; 3539
                                    F4      A7    B6  0006B          INCW    LNK$GW_NFILES               ; 3540
                           50 00000000G     00    D0  0006E          MOVL    LNK$GL_CURFIL, R0           ; 3541
                       E0  A6       04      A0    9E  00075          MOVAB   4(R0), LASTOBJMOD           ;
              0C       0A  A0               01    E1  0007A          BBC     #1, 10(R0), 2$              ; 3543
                                    04      AC    DD  0007F          PUSHL   ARGLIST                     ; 3544
                 00000000G 00               01    FB  00082          CALLS   #1, LNK$PROCSLIB            ;
                                            0C    11  00089          BRB     3$                          ;
              0C       0A  A0               02    E1  0008B 2$:      BBC     #2, 10(R0), 4$              ; 3545
                 00000000G 00               00    FB  00090          CALLS   #0, LNK$PROCSHRIM           ; 3546
                           07               50    E9  00097 3$:      BLBC    R0, 5$                      ;
                                            0B    11  0009A          BRB     6$                          ;
                     1ABA  CA               00    FB  0009C 4$:      CALLS   #0, LNK$PROCSOBJ            ; 3547
                           03               50    E8  000A1 5$:      BLBS    R0, 6$                      ;
                                          0166    31  000A4          BRW     22$                         ;
                                    E0      B6    D4  000A7 6$:      CLRL    @LASTOBJMOD                 ; 3551
                                            B1    11  000AA          BRB     1$                          ; 3537
                              00000000G     00    D4  000AC 7$:      CLRL    LNK$GL_CUROMD               ; 3558
                                    F6      A7    B5  000B2          TSTW    LNK$GW_NMODULES             ; 3560
                                            0D    12  000B5          BNEQ    8$                          ;
                              00000000G     8F    DD  000B7          PUSHL   #LIN$_NOMODS                ;
                 00000000G 00               01    FB  000BD          CALLS   #1, LIB$STOP                ;
```

```
                    50              67  3C 000C4 8$:     MOVZWL  LNK$GW_NUDFSYMS, R0                     ; 3562
                                    0D  13 000C7         BEQL    9$
                                    50  DD 000C9         PUSHL   R0                                      ; 3563
                                    01  DD 000CB         PUSHL   #1
                    00000000G       8F  DD 000CD         PUSHL   #LIN$_NUDFSYMS
                    69              03  FB 000D3         CALLS   #3, LIB$SIGNAL
                    52          BA  B7  0F 000D6 9$:     REMQUE  @LNK$GL_UDFLST, NXTSYMENT               ; 3565
                                    6B  1D 000DA         BVS     14$
                    50          0F  A2  9A 000DC         MOVZBL  15(NXTSYMENT), R0                       ; 3567
              50    52              50  C3 000E0         SUBL3   R0, NXTSYMENT, R0
                    53          FB  A0  9E 000E4         MOVAB   -5(R0), R3                              ; 3568
                    54          0A  A2  9E 000E8         MOVAB   10(NXTSYMENT), R4                       ; 3570
              50    64              0A  E0 000EC         BBS     #10, (R4), 13$
                    06              64  E8 000F0         BLBS    (R4), 10$                               ; 3572
                    55          47  A8  9E 000F3         MOVAB   P.AAL, SUFFIX                           ; 3573
                                    07  11 000F7         BRB     11$
                    55          49  A8  9E 000F9 10$:     MOVAB   P.AAM, SUFFIX                           ; 3575
                    64              02  88 000FD         BISB2   #2, (R4)                                ; 3576
                                    67  B5 00100 11$:     TSTW    LNK$GW_NUDFSYMS                         ; 3579
                                    10  13 00102         BEQL    12$
                                    55  DD 00104         PUSHL   SUFFIX                                  ; 3581
                              04    A3  9F 00106         PUSHAB  4(R3)
                                    02  DD 00109         PUSHL   #2
                    00000000G       8F  DD 0010B         PUSHL   #LIN$_UDFSYM
                    69              04  FB 00111         CALLS   #4, LIB$SIGNAL
                                    62  D4 00114 12$:     CLRL    (NXTSYMENT)                             ; 3584
                    64              08  8A 00116         BICB2   #8, (R4)                                ; 3585
B5 00000000G        00              01  E0 00119         BBS     #1, LNK$GL_CTLMSK+1, 9$                 ; 3587
AD 00000000G        00              04  E1 00121         BBC     #4, LNK$GL_CTLMSK, 9$                   ; 3588
                    7E              64  3C 00129         MOVZWL  (R4), -(SP)                             ; 3592
                                    52  DD 0012C         PUSHL   NXTSYMENT                               ; 3591
                              04    A3  9F 0012E         PUSHAB  4(R3)                                   ; 3590
                    00000000G       00  9F 00131         PUSHAB  LNK$AL_SYTBLFMT                         ; 3589
00000000G           00              04  FB 00137         CALLS   #4, CRF$INSRTKEY                        ; 3591
                                    96  11 0013E         BRB     9$                                      ; 3570
                                    62  D4 00140 13$:     CLRL    (NXTSYMENT)                             ; 3596
                    64              08  8A 00142         BICB2   #8, (R4)                                ; 3597
                                    8F  11 00145         BRB     9$                                      ; 3565
                    50          06  A7  3C 00147 14$:     MOVZWL  LNK$GW_NUDFENVS, R0                     ; 3604
                                    2D  13 0014B         BEQL    16$
                                    50  DD 0014D         PUSHL   R0                                      ; 3608
                                    01  DD 0014F         PUSHL   #1
                    00000000G       8F  DD 00151         PUSHL   #LIN$_NUDFENVS
                    69              03  FB 00157         CALLS   #3, LIB$SIGNAL
                    54          C6  A7  D0 0015A         MOVL    LNK$GL_UDFENV, ENVDESC                  ; 3610
                    50          C6  A7  9E 0015E 15$:     MOVAB   LNK$GL_UDFENV, R0                       ; 3611
                    50              54  D1 00162         CMPL    ENVDESC, R0
                                    13  13 00165         BEQL    16$
                              12    A4  9F 00167         PUSHAB  18(ENVDESC)                             ; 3613
                                    01  DD 0016A         PUSHL   #1
                    00000000G       8F  DD 0016C         PUSHL   #LIN$_UDFENV
                    69              03  FB 00172         CALLS   #3, LIB$SIGNAL
                    54              64  D0 00175         MOVL    (ENVDESC), ENVDESC                      ; 3614
                                    E4  11 00178         BRB     15$                                     ; 3611
                    50          04  A7  3C 0017A 16$:     MOVZWL  LNK$GW_NUDFLSYMS, R0                    ; 3621
                                    48  13 0017E         BEQL    20$
                                    50  DD 00180         PUSHL   R0                                      ; 3623
```

```
                              01  DD 00182          PUSHL   #1                                        ;
              00000000G       8F  DD 00184          PUSHL   #LIN$_NUDFLSYMS                           ;
          69                  03  FB 0018A          CALLS   #3, LIB$SIGNAL                            ;
          52          CE      B7  0F 0018D 17$:     REMQUE  @LNK$GL_UDFLSY, NXTSYMENT                 ; 3625
                              35  1D 00191          BVS     20$                                       ;
          50          0F      A2  9A 00193          MOVZBL  15(NXTSYMENT), R0                         ; 3627
    50    52                  50  C3 00197          SUBL3   R0, NXTSYMENT, R0                         ;
          50                  05  C2 0019B          SUBL2   #5, R0                                    ;
          06          0A      A2  E8 0019E          BLBS    10(NXTSYMENT), 18$                        ; 3629
          55          5B      A8  9E 001A2          MOVAB   P.AAN, SUFFIX                             ; 3630
                              08  11 001A6          BRB     19$                                       ;
          55          5D      A8  9E 001A8 18$:     MOVAB   P.AAO, SUFFIX                             ; 3632
       0A A2                  02  88 001AC          BISB2   #2, 10(NXTSYMENT)                         ; 3633
                              55  DD 001B0 19$:     PUSHL   SUFFIX                                    ; 3636
                      04      A0  9F 001B2          PUSHAB  4(R0)                                     ;
                              02  DD 001B5          PUSHL   #2                                        ;
              00000000G       8F  DD 001B7          PUSHL   #LIN$_UDFSYM                              ;
          69                  04  FB 001BD          CALLS   #4, LIB$SIGNAL                            ;
                              62  D4 001C0          CLRL    (NXTSYMENT)                               ; 3637
       0A A2                  08  8A 001C2          BICB2   #8, 10(NXTSYMENT)                         ; 3638
                              C5  11 001C6          BRB     17$                                       ; 3625
                      DE      A7  D5 001C8 20$:     TSTL    LNK$GL_DBGTFPS                            ; 3642
                              21  12 001CB          BNEQ    21$                                       ;
                      04      A6  9F 001CD          PUSHAB  SYMENTNAM                                 ; 3643
                              56  DD 001D0          PUSHL   R6                                        ;
              00000000G       00  9F 001D2          PUSHAB  LNK$GT_IMGSTA                             ;
          6B                  03  FB 001D8          CALLS   #3, LNK$SEARCH                            ;
          10                  50  E9 001DB          BLBC    R0, 21$                                   ;
          50                  66  D0 001DE          MOVL    SYMTABENT, R0                             ; 3644
    08    0A  A0              01  E1 001E1          BBC     #1, 10(R0), 21$                           ;
          DA  A7              60  D0 001E6          MOVL    (R0), LNK$GL_DBGTFR                       ; 3646
          DE  A7              01  D0 001EA          MOVL    #1, LNK$GL_DBGTFPS                        ; 3647
 00000000G 00 00000000G       00  9E 001EE 21$:     MOVAB   LNK$GL_DEFCLU, LNK$GL_CURCLU              ; 3654
                      BC      AA  9F 001F9          PUSHAB  CHECKPDDENTRY                             ; 3655
              00000000G       00  9F 001FC          PUSHAB  LNK$GL_PSCDFLST                           ;
 00000000G 00                 02  FB 00202          CALLS   #2, LIB$TRAVERSE_TREE                     ;
          50                  01  D0 00209          MOVL    #1, R0                                    ; 3657
                                  04 0020C          RET                                               ;
                              50  D4 0020D 22$:     CLRL    R0                                        ; 3658
                                  04 0020F          RET                                               ;

; Routine Size:  528 bytes,    Routine Base:  $CODE$ + 1E4B


; 3196          3659  1
; 3197          3660  1
; 3198          3661  1
; 3199          3662  1 end
; 3200          3663  0 eludom


                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                         PSECT SUMMARY
;
```

```
;       Name                    Bytes                   Attributes
;
;  $GLOBAL$                       86  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                        2200  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                        208  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       8283  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                           Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       210         1        1000        00:01.9
;  _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1     538       185        34          28        00:00.8




;                                   COMMAND QUALIFIERS
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LNKOBJPS1/OBJ=OBJ$:LNKOBJPS1 MSRC$:LNKOBJPS1/UPDATE=(ENH$:LNKOBJPS1)

; Size:         8283 code + 2494 data bytes
; Run Time:        02:34.2
; Elapsed Time:    04:59.0
; Lines/CPU Min:    1425
; Lexemes/CPU-Min: 23122
; Memory Used:  540 pages
; Compilation Complete
```

LNKNXTOBJ
LIS

LNKOBJPSI
LIS

LNKOBJFIL
LIS

LNKOBJPS2
LIS

LNKOPTION
LIS