F 9

```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
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        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
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
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
```

```
**FILE**ID**LNKPROSHR                                              M  7







LL          NN      NN  KK      KK  PPPPPPPP    RRRRRRRR        OOOOOO      SSSSSSSS  HH      HH  RRRRRRRR
LL          NN      NN  KK      KK  PPPPPPPP    RRRRRRRR        OOOOOO      SSSSSSSS  HH      HH  RRRRRRRR
LL          NN      NN  KK      KK  PP      PP  RR      RR  OO      OO  SS          HH      HH  RR      RR
LL          NN      NN  KK      KK  PP      PP  RR      RR  OO      OO  SS          HH      HH  RR      RR
LL          NNNN    NN  KK    KK    PP      PP  RR      RR  OO      OO  SS          HH      HH  RR      RR
LL          NNNN    NN  KK    KK    PP      PP  RR      RR  OO      OO  SS          HH      HH  RR      RR
LL          NN  NN  NN  KKKKKK      PPPPPPPP    RRRRRRRR    OO      OO    SSSSSS    HHHHHHHHHH  RRRRRRRR
LL          NN  NN  NN  KKKKKK      PPPPPPPP    RRRRRRRR    OO      OO    SSSSSS    HHHHHHHHHH  RRRRRRRR
LL          NN    NNNN  KK    KK    PP          RR  RR      OO      OO          SS  HH      HH  RR  RR
LL          NN    NNNN  KK    KK    PP          RR  RR      OO      OO          SS  HH      HH  RR  RR
LL          NN      NN  KK      KK  PP          RR    RR    OO      OO          SS  HH      HH  RR    RR          ....
LL          NN      NN  KK      KK  PP          RR    RR    OO      OO          SS  HH      HH  RR    RR          ....
LLLLLLLLLL  NN      NN  KK      KK  PP          RR      RR    OOOOOO    SSSSSSSS    HH      HH  RR      RR      ....
LLLLLLLLLL  NN      NN  KK      KK  PP          RR      RR    OOOOOO    SSSSSSSS    HH      HH  RR      RR      ....

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
```

```
0 module lnk_procshrim                          ! PROCESS SHAREABLE IMAGES ON PASS 1
0                  (ident = 'V04-000'
0                  ,addressing_mode
0                           (external    = general
0                           ,nonexternal = long_relative
0                           )
0                  ) =
1 begin
1 !
1 !*****************************************************************************
1 !*                                                                           *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
1 !*  ALL RIGHTS RESERVED.                                                     *
1 !*                                                                           *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
1 !*  TRANSFERRED.                                                             *
1 !*                                                                           *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
1 !*  CORPORATION.                                                             -
1 !*                                                                           *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
1 !*                                                                           *
1 !*                                                                           *
1 !*****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     LINKER
1 !
1 ! ABSTRACT:     THIS MODULE CONTAINS THE ROUTINES TO READ SHAREABLE
1 !                       IMAGES ON PASS 1.
1 !
1 !
1 ! ENVIRONMENT:  VMS NATIVE MODE
1 !
1 ! AUTHOR:       T.J. PORTER, CREATION DA:E: 15-DEC-77
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-008 ADE0003         Alan D. Eldridge        7-Aug-1984
1 !               Propagate the DZRO bit from the shareable image ISD to the
1 !               corresponding ISD in the new image.  It was being cleared
1 !               unconditionally.
1 !
1 !       V03-007 JWT0192         Jim Teague              2-Aug-1984
1 !               Unfortunately, push came to shove, and the global
1 !               isd search optimization has been tossed.
1 !
1 !       V03-006 JWT0189         Jim Teague              25-Jul-1984
1 !               Make some changes to accomodate demand-zero image
```

```
;   58        0058  1 !             sections in shareable images.  The minimum image
;   59        0059  1 !             file VBN for the symbol table is now 2, instead of 3.
;   60        0060  1 !             Why?  Consider the case of a shareable image with
;   61        0061  1 !             nothing BUT a demand zero image section...
;   62        0062  1 !
;   63        0063  1 !     V03-005 ADE0002         Alan D. Eldridge        10-Jul-1984
;   64        0064  1 !             Fix null arguments passed in SIGNAL due to editting errors.
;   65        0065  1 !
;   66        0066  1 !     V03-004 ADE0001         Alan D. Eldridge        26-Feb-1984
;   67        0067  1 !             Don't erase CLU$L_GSMATCH.  It is no longer re-used as
;   68        0068  1 !             something else in pass 2, and it is referenced to correctly
;   69        0069  1 !             perform the GSMATCH checking in pass one.
;   70        0070  1 !
;   71        0071  1 !     V03-003 JWT0152         Jim Teague               8-Feb-1984
;   72        0072  1 !             Long names for global image section descriptors.
;   73        0073  1 !
;   74        0074  1 !     V03-002 JWT0111         Jim Teague              14-Apr-1983
;   75        0075  1 !             Don't pull in shareable images that are in global
;   76        0076  1 !             isds unless they're based.
;   77        0077  1 !
;   78        0078  1 !     V03-001 JWT0044         Jim Teague              30-Jul-1982
;   79        0079  1 !             Open file performance boost.
;   80        0080  1 !
;   81        0081  1 !--
;   82        0082  1
;   83        0083  1 !
;   84        0084  1 ! TABLE OF CONTENTS:
;   85        0085  1 !
;   86        0086  1 forward routine
;   87        0087  1     readnextblock;                        ! READS THE NEXT HEADER BLOCK OF IMAGE
;   88        0088  1
;   89        0089  1 !
;   90        0090  1 ! INCLUDE FILES:
;   91        0091  1 !
;   92        0092  1 library 'LIBL32';                         ! VMS SYSTEM STRUCTURE DEFINITIONS
;   93        0093  1 require 'PREFIX';                         ! USEFUL GENERAL MACROS
;   94        0208  1 library 'DATBAS';                         ! INTERNAL DATA BASE DEFINITIONS
;   95        0209  1 require 'ISGENC';                         ! IMAGE SECTION PARAMETERS
;   96        0593  1
;   97        0594  1 !
;   98        0595  1 ! MACROS:
;   99        0596  1 !
;  100        0597  1 !     NONE
;  101        0598  1 !
;  102        0599  1 ! EQUATED SYMBOLS:
;  103        0600  1 !
;  104        0601  1 !     NONE
;  105        0602  1 !
;  106        0603  1 ! EXTERNAL REFERENCES:
;  107        0604  1 !
;  108        0605  1
;  109        0606  1 external routine
;  110        0607  1     lnk$alloblk         : novalue,        ! DYNAMIC MEMORY ALLOCATOR
;  111        0608  1     lnk$allocluster,                      ! ALLOCATE CLUSTER DESCRIPTOR
;  112        0609  1     lnk$closefile       : novalue,        ! CLOSE CURRENT INPUT FILE
;  113        0610  1     lnk$pointobj        : novalue,        ! POINTS TO NEW PLACE IN FILE
;  114        0611  1     lnk$addimage,                         ! PROCESS SHAREABLE IMAGE
;  115          0612  1       lnk$procsobj;                                   ! PROCESSES OBJECT MODULES (I.E. THE GST)
;  116          0613  1
;  117          0614  1 external literal
;  118          0615  1       lin$_badimghdr,                                 . BAD IMAGE HEADER ERROR MESSAGE
;  119          0616  1       lin$_basshrbel,                                 ! BASED SHAREABLE IMAGE BELOW BASE=
;  120          0617  1       lin$_confbasadr,                                ! CONFLICTING BASE ADDRESSES FOR SHR IMGS
;  121          0618  1       lin$_idmismch,                                  ! GSMATCH MISMATCH WITH SHR IMG STB LIBRARY
;  122          0619  1       lin$_imgbased,                                  . ATTEMPT TO RE-BASE A NON-PIC IMAGE
;  123          0620  1       lin$_noimgfil,                                  ! NO IMAGE FILE CREATED
;  124          0621  1       lin$_nonpicimg,                                 ! SHAREABLE IMAGE IS NON-PIC
;  125          0622  1       lin$_relink,                                    ! RELINK DUE TO COPYALWAYS SECTION
;  126          0623  1       lin$_readerr;                                   ! READ ERROR
;  127          0624  1
;  128          0625  1 external
;  129          0626  1       lnk$gl_shrclstrs,                               ! COUNT OF NUMBER OF SHAREABLE IMAGE CLUSTERS (PIC AND NON-P
;  130          0627  1       lnk$al_rab          : block [, byte],           ! RAB FOR OPEN IMAGE FILE
;  131          0628  1       lnk$gw_nisects      : word,                     ! IMAGE SECTION ACCUMULATOR
;  132          0629  1       lnk$gw_shrisects    : word,                     ! NUMBER OF SHAREABLE IMAGE ISECTS
;  133          0630  1       lnk$gb_pass         : byte,                     ! CURRENT PASS
;  134          0631  1       lnk$gl_curfil       : ref block [, byte],       ! CURRENT FILE POINTER
;  135          0632  1       lnk$gl_curclu       : ref block [, byte],       ! CURRENT CLUSTER DESCRIPTOR
;  136          0633  1       lnk$gl_defclu       : block [, byte],           ! DEFAULT_CLUSTER DESCRIPTOR
;  137          0634  1       lnk$gl_lastclu      : ref block [, byte],       ! POINTER TO LAST CLUSTER DESCRIPTOR
;  138          0635  1       lnk$gl_ctlmsk       : block [, byte],           ! CONTROL FLAGS
;  139          0636  1       lnk$gl_objrecs ;                                ! COUNT OF OBJECT RECORDS
;  140          0637  1
;  141          0638  1 !
;  142          0639  1 ! MODULE OWN STORAGE:
;  143          0640  1 !
;  144          0641  1 global
;  145          0642  1       lnk$gl_gsbufdsc     : vector [3],               ! BUFFER DESCRIPTOR FOR COPIED GLOBAL SECTIONS
;  146          0643  1       lnk$gl_imgrecs ;                                ! COUNT OF IMAGE RECORDS
;  147          0644  1
;  148          0645  1 own
;  149          0646  1       curisdseq           : vector [4, byte]          ! SUFFIX TO SCLUSTER NAME GIVING GBL ISD NAME
;  150          0647  1                           initial (%ascii '_000'),!
;  151          0648  1       hdrblkcnt,                                      ! NUMBER REMAINING HEADER BLOCKS
;  152          0649  ;       headerblock ;                                   ! CURRENT HEADER VBN
;  153          0650  1
;  154          0651  1
```

```
1 global routine lnk$procshrim (modrfa) =          !
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !                         THIS ROUTINE IS CALLED TO PROCESS SHAREABLE IMAGES ON PASS
1 !       ONE OF THE LINK. IT READS AND VALIDATES EVERY BLOCK OF THE IMAGE HEADER
1 !       BUILDING THE LIST OF IMAGE SECTION DESCRIPTORS. AFTER COMPLETING THE HEADER
1 !       IT SETS THE RAB POINTING TO THE SYMBOL TABLE PART OF THE IMAGE FILE AND
1 !       CALLS LNK$PROCSOBJ TO DO THE PASS ONE OBJECT MODULE PROCESSING
1 !       OF THE SYMBOL TABLE.
1 !                         THE SYMBOL TABLE OF A SHAREABLE IMAGE CONTAINS ALL THE UNIVERSAL
1 !       SYMBOLS DEFINED FOR THE IMAGE AND IS IN THE OBJECT MODULE FORMAT.
1 !
1 ! FORMAL PARAMETERS:
1 !       MODRFA IS A POINTER TO THE 6 BYTE RFA OF THE SHAREABLE IMAGE IF
1 !       IT IS IN A LIBRARY. IF NOT A LIBRARY SHAREABLE IMAGE THE ARGUMENT
1 !       DOES NOT EXIST.
1 !
1 ! IMPLICIT INPUTS:
1 !       SEE ABOVE EXTERNAL DECLARATIONS.
1 !       IN ADDITION THE FILE CONTAINING THE IMAGE IS ALREADY OPEN
1 !       FOR BLOCK READ OPERATIONS.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       SEE ABOVE GLOBAL DECLARATIONS.
1 !       IN ADDITION ALL IMAGE SECTIONS FOUND IN THIS SHAREABLE IMAGE
1 !       ARE APPENDED TO THE LIST AND THE GST HAS BEEN PROCESSED AS AN OBJECT
1 !       MODULE. THAT IS ALL UNIVERSAL SYMBOLS ARE IN THE LINKER SYMBOL TABLE.
1 !
1 ! ROUTINE VALUE:
1 !       RETURNS VALUE TRUE IF SUCCESSFULLY PROCESSED, ELSE FALSE
1 !
1 ! SIDE EFFECTS:
1 !       THE ROUTINE DOES NOT RETURN IF A FATAL ERROR IS DETECTED.
1 !
1 !--
2 begin
2 builtin
2       actualcount ;                            ! GETS COUNT OF ARGUMENTS
2 local
2       nxtisdoff,                               ! OFFSET TO NEXT ISD
2       blockoffset,                             ! OFFSET IN FILE TO FIRST HEADER BLOCK
2       isectident,                              ! MAJOR AND MINOR ID FROM HEADER
2       firstisdvpg,                             ! VPG OF FIRST ISECT
2       symdbgdatdsc    : ref block [, byte],    ! POINTER TO SYMBOL TABLE DESCRIPTOR
2       gstrecs,                                 ! NUMBER OF RECORDS IN SYMBOL TABLE
2       ownclu          : ref block [, byte],    ! POINTER TO OWNING CLUSTER DESCRIPTOR
2       ownfdb          : ref block [, byte],    !  AND IT'S FILE DESCRIPTOR BLOCK
2       cludesc         : ref block [, byte],    ! POINTER TO CREATED CLUSTER DESCRIPTOR
2       curhdrisd       : ref block [, byte],    ! POINTER TO CURRENT ISD IN HEADER
2       curisd          : ref block [, byte],    ! POINTER TO CURRENT ISD BEING BUILT
2       newhdrisd       : ref block [, byte],    ! POINTER TO HEADER PART OF CURRENT ISD BEING BUILT
2       firstisd        : ref block [, byte],    ! POINTER TO FIRST ISECT IN CLUSTER
2       gstvbn          : vector [2, long],      ! RFA OF GST
2       iafva,                                   ! RELATIVE VA OF FIXUP SECTION
2       saverecount,                             ! SAVED RECORD COUNT WHILE IN OBJPS1
2       maxisdvbn ;                              ! LAST IMAGE VBN + 1
2
```

```
;  213          0709  2 map
;  214          0710  2         modrfa : ref block [, byte] ;                          ! POINTER TO BLOCK
;  215          0711  2
;  216          0712  2 bind
;  217          0713  2         header = lnk$al_rab [rab$l_ubf] : ref block [, byte] ;  ! POINTER TO BLOCK BUFFER
;  218          0714  2
;  219          0715  2 if actualcount () eql 0                                        ! IF CALLED WITH NO ARGUMENTS
;  220          0716  2 then
;  221          0717  2     blockoffset = 0                                            ! HEADER AT START OF FILE
;  222          0718  2 else
;  223          0719  2     blockoffset = .modrfa [rfa$l_vbn] - 1 ;                    ! OTHERWISE GET OFFSET.
;  224          0720  2
;  225          0721  2 lnk$gl_shrclstrs = .lnk$gl_shrclstrs + 1 ;                     ! COUNT THIS SHAREABLE IMAGE CLUSTER
;  226          0722  2
;  227          0723  2 cludesc      =  0 ;                                            ! NO CREATED CLUSTER YET
;  228          0724  2 hdrblkcnt    =  1 ;                                            ! MUST BE AT LEAST ONE BLOCK
;  229          0725  2 nxtisdoff    = -1 ;                                            ! NEXT ISD IS ON NEXT BLOCK
;  230          0726  2 maxisdvbn    =  0 ;                                            ! RESET LAST BLOCK OF IMAGE
;  231          0727  2 headerblock = .blockoffset ;                                   ! AND SET FOR FIRST BLOCK READ
;  232          0728  2
;  233          0729  2 ch$fill (%c'0', 3, curisdseq [1]) ;                            ! INITIALIZE THE ISD NAME SUFFIX
;  234          0730  2
;  235          0731  2 while .nxtisdoff eql -1                                        ! WHILE THERE ARE MORE
;  236          0732  3 do  begin                                                      ! HEADER BLOCKS
;  237          0733  3     if not readnextblock ()                                    ! GET THE NEXT ONE
;  238          0734  3     then
;  239          0735  3         signal_stop (lin$_badimghdr, 1                         ! AND IF UNSUCCESSFUL, FATAL IMAGE
;  240          0736  3                     ,lnk$gl_curfil [fdb$q_filename]
;  241          0737  3                     ) ;
;  242          0738  4     if .headerblock eql (.blockoffset + 1)                     ! IF THE FIRST
;  243          0739  3     then
;  244          0740  4         begin                                                  ! HAVE SOME VALIDATION TO DO
;  245          0741  4         !
;  246          0742  4         !   VALIDATE IMAGE HEADER
;  247          0743  4         !
;  248          0744  4         if .header [ihd$b_imgtype] neq ihd$k_shr              ! CHECK IT IS A SHAREABLE IMAGE
;  249          0745  4         or .header [ihd$w_majorid] neq ihd$k_majorid          ! MAJOR HEADER ID MUST MATCH
;  250          0746  6         or (.(header [ihd$w_minorid])<0, 8, 0> eql (ihd$k_minorid and %x'FF')
;  251          0747  5              and
;  252          0748  5             .(header [ihd$w_minorid])<8, 8, 0> gtru (ihd$k_minorid and %x'FF00')/256
;  253          0749  5            )
;  254          0750  5         or .header [ihd$w_size] gtru maxu ((.header [ihd$w_patchoff] ! THE HEADER FIXED PART
;  255          0751  5                                             + ihp$k_length)        ! MUST BE LESS THAN A BLOCK AND MUST
;  256          0752  4                                            ,ihd$k_maxlength
;  257          0753  4                                            )
;  258          0754  4         or (hdrblkcnt    = .header [ihd$b_hdrblkcnt] - 1) lss 0  ! CONTAIN PATCH AREA. 0 TO
;  259          0755  5         or (symdbgdatdsc = .header + .header [ihd$w_symdbgoff])  ! 127 MORE BLOCKS. GET THE
;  260          0756  5                            gequ (.header + .header [ihd$w_size]) ! GST DESCRIPTOR WHICH MUST BE CONTAINED
;  261          0757  4         or (gstrecs      = .symdbgdatdsc [ihs$w_gstrecs]) lssu 3 ! IN HEADER. MUST BE AT LEAST 3 RECORDS
;  262          0758  5         or (gstvbn [0]   = .symdbgdatdsc [ihs$l_gstvbn])         ! AND MUST BE BEYOND THE HEADER BLOCKS
;  263          0759  5                            lequ (.hdrblkcnt + 1)
;  264          0760  4         then
;  265          0761  4             signal_stop (lin$_badimghdr, 1                     ! ANY ABOVE NOT TRUE, FATAL IMAGE
;  266          0762  4                         ,lnk$gl_curfil [fdb$q_filename]        !  HEADER ERROR
;  267          0763  4                         ) ;
;  268          0764  5         if not (lnk$gl_curclu [clu$v_pic] = .header [ihd$v_picimg]) ! EXTRACT THE PIC BIT AND IF NON-PIC
;  269          0765  4         then
;  270        0766  5          begin
;  271        0767  5          lnk$gl_ctlmsk [lnk$v_picimg] = false ;              ! THIS IMAGE IS ALSO NON-PIC
;  272        0768  5
;  273        0769  5          if          .lnk$gl_curclu [clu$v_usrbased]         ! IF NON-PIC IMAGE BASED BY USER
;  274        0770  5             and not .lnk$gl_curclu [clu$v_intclu]           !  AND NOT AN INTERNALLY CREATED CLUSTER
;  275        0771  5          then
;  276        0772  6              begin
;  277        0773  6              signal  (lin$_imgbased, 1                       !  THEN THAT'S AN ERROR, BECAUSE WE CAN'T
;  278        0774  6                      ,lnk$gl_curfil [fdb$q_filename]         !  RELOCATE IT
;  279        0775  6                      ) ;
;  280        0776  6              lnk$gl_curclu [clu$v_usrbased] = false ;
;  281        0777  6              lnk$gl_curclu [clu$l_usrbase] = 0 ;
;  282        0778  5              end ;
;  283        0779  5          end
;  284        0780  4      else
;  285        0781  4          if .lnk$gl_curclu [clu$v_usrbased]                  ! BUT IF CLUSTER BASED BY OPTION
;  286        0782  4          then
;  287        0783  4              lnk$gl_ctlmsk [lnk$v_picimg] = false ;
;  288        0784  4
;  289        0785  4      if          .lnk$gl_ctlmsk [lnk$v_shr]                  ! IF CREATING A SHAREABLE IMAGE
;  290        0786  4          and not .lnk$gl_ctlmsk [lnk$v_ubased]               !  AND IMAGE NOT BASED BY USER
;  291        0787  4          and not .lnk$gl_curclu [clu$v_pic]                  !  AND THIS IMAGE JUST MADE IT NON-PIC
;  292        0788  4      then
;  293        0789  4          signal  (lin$_nonpicimg, 1                          ! THEN TELL USER ABOUT IT (INFO)
;  294        0790  4                  ,lnk$gl_curfil [fdb$q_filename]
;  295        0791  4                  ) ;
;  296        0792  5      if (lnk$gl_curclu [clu$v_matchctl] =                    ! EXTRACT THE GLOBAL SECTION MATCH CONTROL
;  297        0793  4              .header [ihd$v_matchctl]) eql isd$k_matnev
;  298        0794  4      then
;  299        0795  4          lnk$gl_curclu [clu$v_copy] = true ;                 ! MAKING PRIVATE COPY IF MATCH NEVER
;  300        0796  4
;  301        0797  4      isectident = .header [ihd$l_ident] ;                    ! NOW THE IDENT TO WHICH IT APPLIES
;  302        0798  4      iafva = 0 ;
;  303        0799  4
;  304        0800  5      if (.header + .header [ihd$w_activoff])
;  305        0801  4          gtru header [ihd$l_iafva]                           ! IF IMAGE HAS FIXUP VA OFFSET
;  306        0802  4      then
;  307        0803  5          begin
;  308        0804  5          iafva = .header [ihd$l_iafva] ;                     !  THEN PICK IT UP
;  309        0805  5          if .iafva eql 0
;  310        0806  5          then
;  311        0807  5              lnk$gl_curclu [clu$v_prefixup] = true ;         ! FIXVA THERE, BUT 0, SO ASSUME NOT THERE
;  312        0808  5          end
;  313        0809  4      else
;  314        0810  4          lnk$gl_curclu [clu$v_prefixup] = true ;             ! NO FIXUP VA, FLAG OLD IMAGE
;  315        0811  4
;  316        0812  4      lnk$gl_imgrecs = .lnk$gl_imgrecs + .hdrblkcnt + 1 ;     ! COUNT HEADER BLOCKS AS RECORDS
;  317        0813  4      curhdrisd = .header + .header [ihd$w_size] ;            ! POINT TO FIRST ISD IN HEADER
;  318        0814  4      !
;  319        0815  4      !
;  320        0816  4      !   CHECK GSMATCH OF IMAGE AGAINST GSMATCH FOUND IN SHAREABLE IMAGE
;  321        0817  4      !   LIBRARY IF THIS IMAGE FOUND IN A LIBRARY
;  322        0818  4      !
;  323        0819  4      if      .lnk$gl_curclu [clu$l_gsmatch] neq 0            ! IF FOUND IN SHAREABLE IMAGE LIB
;  324        0820  4         and .lnk$gl_curclu [clu$l_gsmatch] neq .isectident   !  AND IMAGE/LIBRARY MISMATCH
;  325        0821  4      then
;  326        0822  5          begin
;  327          0823  5           bind    libident = lnk$gl_curclu [clu$l_gsmatch] : block [, byte],
;  328          0824  5                   imgident = isectident                     : block [, byte] ;
;  329          0825  5
;  330          0826  5           signal (lin$_idmismch, 6                          ! WARN USER ABOUT IDENT PROBLEM
;  331          0827  5                   ,.imgident [gmt$b_majorid],.imgident [gmt$b_minorid]
;  332          0828  5                   ,lnk$gl_curfil [fdb$q_filename], .libident [gmt$b_majorid],.libident [gmt$b_minorid]
;  333          0829  5                   ,lnk$gl_curfil [fdb$q_libnamdsc]
;  334          0830  5                   ) ;
;  335          0831  4           end ;
;  336          0832  4       end                                                   ! ALL DONE WITH FIRST BLOCK
;  337          0833  3   else
;  338          0834  3       curhdrisd = .header ;                                 ! OTHER BLOCKS ARE ALL ISD
;  339          0835  3
;  340          0836  3   !
;  341          0837  3   !  PROCESS IMAGE SECTION DESCRIPTORS
;  342          0838  3   !
;  343          0839  3   while (nxtisdoff = .(curhdrisd [isd$w_size])<0, 16, 1>) gtr 0  ! WHILE MORE ISD'S ON THIS BLOCK
;  344          0840  4   do  begin
;  345          0841  4       if .curhdrisd [isd$b_type] neq isd$k_usrstack           ! BEGIN LOOP THAT PROCESSES EACH
;  346          0842  4       then
;  347          0843  5           begin                                               ! IGNORING STACK SECTION
;  348          0844  5           if .curhdrisd [isd$v_copyalway]                     ! IF THIS IS COPY ALWAYS SECTION
;  349          0845  5           then
;  350          0846  5               signal_stop (lin$_relink, 2                     ! THEN ISSUE MESSAGE AND GIVE UP NOW
;  351          0847  5                       ,lnk$gl_curfil [fdb$q_filename]
;  352          0848  5                       ,lnk$gl_curfil [fdb$q_filename]
;  353          0849  5                       ) ;
;  354          0850  5
;  355          0851  5           if .curhdrisd [isd$v_gbl]                              ! IF GLOBAL ISECT
;  356          0852  5           then
;  357          0853  6               begin                                              !  THEN CHECK FROM SAME SHAREABLE IMAGE
;  358          0854  6               local   found,
;  359          0855  6                       shrdesc : block [dsc$c_s_bln, byte] ;
;  360          0856  6               bind
;  361          0857  6                       gblnam  = curhdrisd [isd$t_gblnam] : vector [, byte] ;
;  362          0858  6
;  363          0859  6               shrdesc [dsc$a_pointer] =  gblnam [1] ;            ! COMPLETE SECTION NAME DESCRIPTOR
;  364          0860  6               shrdesc [dsc$w_length]  = .gblnam [0] - suffix_size ;   ! FIND LENGTH OF GLOBAL SECTION
;  365          0861  6                                                                  ! SECTION NAME IN ISECT DESCRIPTOR
;  366          0862  6               if .cludesc eql 0                                  ! IF NO CREATED CLUSTERS YET
;  367          0863  6               or not ch$eql (.shrdesc [dsc$w_length], gblnam [1] ! OR THIS IS FOR A DIFFERENT CLUSTER
;  368          0864  6                       ,.shrdesc [dsc$w_length], cludesc [clu$t_name]
;  369          0865  6                       )
;  370          0866  6               then
;  371          0867  7                   begin
;  372          0868  7                   lnk$addimage (shrdesc, 0, cludesc, found) ;     ! ADD IMAGE TO THE CLUSTER LIST
;  373          0869  7                   if .cludesc neq 0 and not .found
;  374          0870  7                   then
;  375          0871  8                       begin
;  376          0872  8                       cludesc [clu$v_prefixup] = .lnk$gl_curclu [clu$v_prefixup] ;
;  377          0873  8                                                                  ! COPY PREFIXUP FLAG FROM PARENT CLUSTE
;  378          0874  8                       cludesc [clu$l_ownclu]   = .lnk$gl_curclu ;  ! SET OWNING CLUSTER DESCRIPTOR
;  379          0875  8
;  380          0876  8                       if not .lnk$gl_curclu [clu$v_pic]           ! IF THIS IMAGE IS NON-PIC
;  381          0877  8                       then
;  382          0878  9                           begin
;  383          0879  9                           cludesc [clu$v_based] = true ;                  ! FLAG CLUSTER AS BASED
; 10          cludesc [clu$l_base] = (if   .curhdrisd [isd$v_based]
; 10                                  then .curhdrisd [isd$v_vpg]^9
; 10                                  else .lnk$gl_curclu [clu$l_base] +
; 10                                       .lnk$gl_curclu [clu$l_cluoff]
;  9                                  ) ;
;  9          lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
;  9                                         .curhdrisd [isd$w_pagcnt]*512 ;
;  8          end ;
;  8      end
;  7  else
;  8      begin
;  9      if (.lnk$gl_curclu [clu$v_based]                 ! IF THIS CLUSTER IS BASED
;  9              and not
;  9          .lnk$gl_curclu [clu$v_pic]
;  9         )
;  8      or  .lnk$gl_curclu [clu$v_usrbased]
;  8      then
;  9          begin
;  9          ownclu = .cludesc [clu$l_ownclu] ;           ! GET OWNING CLUSTER POINTER
;  9
;  9          if .ownclu neq 0
;  9          then
;  9              lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
;  9                                              .curhdrisd[isd$w_pagcnt]*512 ;
;  9
;  9          if  .ownclu neq 0                            ! IF THERE IS AN OWNING CLUSTER
;  9              and
;  9              .ownclu [clu$v_based]                    !  WHICH IS ALSO BASED
;  9              and
;  9              .cludesc [clu$l_base] neq .curhdrisd [isd$v_vpg]^9
;  9                                                       ! AND WANTS THIS CLUSTER AT A
;  9                                                       ! DIFFERENT PLACE
;  9          then
; 10              begin
; 10              ownfdb = .ownclu [clu$l_fstfdb] ;        ! GET FIRST FILE DESCRIPTOR BLOCK
; 10              signal  (lin$_confbasadr, 5
; 10                      ,cludesc [clu$b_namlng]
; 10                      ,.curhdrisd [isd$v_vpg]^9, lnk$gl_curfil [fdb$q_filename]
; 10                      ,.cludesc [clu$l_base], ownfdb [fdb$q_filename]
; 10                      ,lin$_noimgfil
; 10                      ) ;
; 10              lnk$gl_ctlmsk [lnk$v_image] = false ;    ! DON'T MAKE A NON-RUNNABLE IMAGE
; 10              end
;  9          else
; 10              begin
; 10              if not .lnk$gl_curclu [clu$v_usrbased]
; 10              then
; 11                  begin
; 11                  if not .cludesc [clu$v_based]        ! IF CLUSTER NOT ALREADY BASED
; 11                  then
; 11                      lnk$gw_shriscts = .lnk$gw_shriscts - .cludesc [clu$l_nisects] ;
; 11
; 11                  cludesc [clu$v_based] = true ;       ! FLAG AS BASED
; 11                  cludesc [clu$l_base] = .curhdrisd [isd$v_vpg]^9 ;
; 10                  end ;
;  9              end ;
;  8          end ;
;  441          0937  7                         end ;
;  442          0938  7                     end
;  443          0939  6                 else
;  444          0940  7                     begin
;  445          0941  7                     if .lnk$gl_curclu [clu$v_based] or .lnk$gl_curclu [clu$v_usrbased]
;  446          0942  7                     then
;  447          0943  7                         lnk$gl_curclu [clu$l_cluoff] =  .lnk$gl_curclu [clu$l_cluoff] +
;  448          0944  7                                                         .curhdrisd [isd$w_pagcnt] * 512 ;
;  449          0945  7
;  450          0946  6                     end ;
;  451          0947  5                 end ;
;  452          0948  5
;  453          0949  5
;  454          0950  5             if not .curhdrisd[isd$v_gbl]
;  455          0951  5             then
;  456          0952  6                 begin
;  457          0953  6                 lnk$alloblk (isd$c_size, curisd) ;                       ! ALLOCATE A DESCRIPTOR
;  458          0954  6                 curisd [isl$l_nxtisd] = 0 ;                              ! CURRENT IS LAST
;  459          0955  6                 curisd [isl$l_bufadr] = 0 ;                              ! NO BUFFER FOR IT YET
;  460          0956  6                 curisd [isl$l_bufend] = 0 ;                              ! ...
;  461          0957  6                 curisd [isl$l_cludsc] = .lnk$gl_curclu ;                 ! SET POINTER TO CLUSTER DESCRIPTOR
;  462          0958  6                 newhdrisd = curisd [isl$t_hdrisd] ;                      ! POINT TO PART DESTINED FOR HEADER
;  463          0959  6                 ch$copy (.curhdrisd [isd$w_size]
;  464          0960  6                         , curhdrisd [isd$w_size], 0                      ! COPY THE ISD FROM THE
;  465          0961  6                         , isd$k_maxlenglbl, curisd [isl$t_hdrisd]        ! HEADER TO DESCRIPTOR 0 FILLED
;  466          0962  6                         ) ;
;  467          0963  6                 if .lnk$gl_curclu [clu$l_fstisd] eql 0                   ! IF THIS IS THE FIRST
;  468          0964  6                 then
;  469          0965  7                     begin
;  470          0966  7                     if .lnk$gl_curclu [clu$v_usrbased]                   ! IF BASED BY USER
;  471          0967  7                     then
;  472          0968  8                         begin
;  473          0969  8                         if .lnk$gl_curclu [clu$v_based]                  ! IF CLUSTER IS ALSO BASED
;  474          0970  8                             and
;  475          0971  8                             .lnk$gl_curclu [clu$l_base] neq .lnk$gl_curclu [clu$l_usrbase]
;  476          0972  8                         then
;  477          0973  9                             begin
;  478          0974  9                             signal ( lin$_confbasadr, 5
;  479          0975  9                                    , lnk$gl_curclu [clu$b_namlng]
;  480          0976  9                                    ,.lnk$gl_curclu [clu$l_base], lnk$gl_curfil [fdb$q_filename]
;  481          0977  9                                    ,.lnk$gl_curclu [clu$l_usrbase], $descriptor ('Options file')
;  482          0978  9                                    , lin$_noimgfil
;  483          0979  9                                    ) ;
;  484          0980  9                             lnk$gl_ctlmsk [lnk$v_image] = false ;
;  485          0981  8                             end ;
;  486          0982  8
;  487          0983  8                         lnk$gl_curclu [clu$l_base]    = .lnk$gl_curclu [clu$l_usrbase] ;
;  488          0984  8                         lnk$gl_curclu [clu$l_usrbase] = 0 ;
;  489          0985  8                         lnk$gl_curclu [clu$v_based]   = true ;
;  490          0986  8                         end
;  491          0987  7                     else
;  492          0988  8                         begin
;  493          0989  8                         if .lnk$gl_curclu [clu$v_based]                  ! IF BASED DUE TO ANOTHER IMAGE
;  494          0990  8                             and not                                      ! BASING IT AND IT TURNED OUT TO
;  495          0991  8                             .lnk$gl_curclu [clu$v_pic]                   ! BE NON-PI(
;  496          0992  8                         then
;  497          0993  9                             begin
;  498          0994  9            if .lnk$gl_curclu [clu$l_base] neq .newhdrisd [isd$v_vpg]^9
;  499          0995  9            then
;  500          0996 10                begin
;  501          0997 10                ownclu = .lnk$gl_curclu [clu$l_ownclu] ;
;  502          0998 10                ownfdb = .ownclu [clu$l_fstfdb] ;
;  503          0999 10                signal  ( lin$_confbasadr, 5
;  504          1000 10                        , lnk$gl_curclu [clu$b_namlng]
;  505          1001 10                        ,.newhdrisd [isd$v_vpg]^9, lnk$gl_curfil [fdb$q_filename]
;  506          1002 10                        ,.lnk$gl_curclu [clu$l_base], ownfdb [fdb$q_filename]
;  507          1003 10                        , lin$_noimgfil
;  508          1004 10                        ) ;
;  509          1005 10                lnk$gl_ctlmsk [lnk$v_image] = false ;
;  510          1006 10                end
;  511          1007  9            end
;  512          1008  8        else
;  513          1009  8            if not .lnk$gl_curclu [clu$v_pic]
;  514          1010  8            then                                        ! THEN EXTRACT BASE VPN
;  515          1011  9                begin
;  516          1012  9                lnk$gl_curclu [clu$l_base]  = .newhdrisd [isd$v_vpg]^9 ;
;  517          1013  9                lnk$gl_curclu [clu$v_based] = true ;
;  518          1014  8                end ;
;  519          1015  7        end ;
;  520          1016  7    firstisd = .newhdrisd ;                             ! POINT TO FIRST ISECT IN CLUSTER
;  521          1017  6    end ;
;  522          1018  6
;  523          1019  6lnk$gl_curclu [clu$l_nisects] = .lnk$gl_curclu [clu$l_nisects] + 1 ;
;  524          1020  6                                                        ! COUNT ISECT IN CLUSTER
;  525          1021  6
;  526          1022  7    begin
;  527          1023  7    bind   lastisd = lnk$gl_curclu [clu$l_lstisd]       ! POINTER TO LAST ISD IN CLUSTER
;  528          1024  7                   : ref block [, byte] ;
;  529          1025  7
;  530          1026  7    lastisd [isl$l_nxtisd] = .curisd ;                  ! PUT AT END OF LIST
;  531          1027  7    lastisd = .curisd ;                                 ! AND MAKE CURRENT THE NEW LAST
;  532          1028  6    end ;
;  533          1029  6
;  534          1030  6
;  535          1031  6if .lnk$gl_curclu [clu$v_pic]                           ! IF A PIC CLUSTER
;  536          1032  6then
;  537          1033  7    begin
;  538          1034  7    if .newhdrisd eql .firstisd                         ! IF FIRST ISECT THIS CLUSTER
;  539          1035  7    then
;  540          1036  7        firstisdvpg = .newhdrisd [isd$v_vpg] ;          ! THEN SAVE VPG OF FIRST ISECT
;  541          1037  7
;  542          1038  7    newhdrisd [isd$v_vpg] = .newhdrisd [isd$v_vpg] -    ! THEN SUBTRACT OUT THE BASE
;  543          1039  7                            .firstisdvpg ;              ! FOR LATER RE-LOCATION (NEEDED
;  544          1040  7                                                        ! FOR OLD IMAGES LINKED @200)
;  545          1041  7
;  546          1042  7    if  not .lnk$gl_curclu [clu$v_based]
;  547          1043  7    then lnk$gw_shriscts = .lnk$gw_shriscts + 1 ;       ! COUNT IT
;  548          1044  6    end ;
;  549          1045  6
;  550          1046  6lnk$gl_curclu [clu$l_pages] = .lnk$gl_curclu [clu$l_pages] + .newhdrisd [isd$w_pagcnt] ;
;  551          1047  6                                                        ! ACCUMULATE THE PAGES REQUIRED
;  552          1048  6
;  553          1049  6if (ownclu = .lnk$gl_curclu [clu$l_ownclu]) eql 0
;  554          1050  6then
;  555        1051  6          lnk$gl_curclu [clu$l_cluoff] =  .lnk$gl_curclu [clu$l_cluoff] +
;  556        1052  6                                          .newhdrisd [isd$w_pagcnt]*512
;  557        1053  6      else
;  558        1054  6          ownclu [clu$l_cluoff] = .ownclu [clu$l_cluoff] + .newhdrisd [isd$w_pagcnt]*512 ;
;  559        1055  6
;  560        1056  6      if .curhdrisd [isd$v_wrt]                              ! IF SECTION IS WRITEABLE
;  561        1057  6          and not
;  562        1058  6          .curhdrisd [isd$v_crf]                             !  AND NOT COPY-ON-REF
;  563        1059  6      then
;  564        1060  6          lnk$gl_curclu [clu$v_wrt] = true ;                 !  THEN REMEMBER FOR LNKIMGOUT
;  565        1061  6
;  566        1062  6      if .lnk$gl_curclu [clu$v_based]                        ! IF CLUSTER IS BASED THEN FLAG
;  567        1063  6      then
;  568        1064  6          newhdrisd [isd$v_based] = true ;                   ! IN CLUSTER DESCRIPTOR ALSO
;  569        1065  6
;  570        1066  6      if not .curhdrisd [isd$v_dzro]                         ! PROVIDED NOT A DEMAND ZERO ISD
;  571        1067  6         and .curhdrisd [isd$l_vbn] neq 0                    !  AND SECTION IS PRESENT IN IMAGE
;  572        1068  6      then
;  573        1069  6          if .curhdrisd [isd$l_vbn] gequ .maxisdvbn          ! IF IMAGE SECTION GOES BEYOND LAST
;  574        1070  6          then
;  575        1071  6              maxisdvbn = .curhdrisd [isd$l_vbn] +           ! BLOCK OF LAST ISECT, THEN
;  576        1072  6                          .curhdrisd [isd$w_pagcnt] ;        ! COMPUTE NEW MAX VBN IN USE
;  577        1073  6
;  578        1074  6      if not .curhdrisd [isd$v_gbl]                          ! IF IMAGE SECTION IS NOT
;  579        1075  6      then                                                   !   ALREADY GLOBAL
;  580        1076  7          begin
;  581        1077  7          local gblsect_namlng ;
;  582        1078  7
;  583        1079  7          gblsect_namlng = .lnk$gl_curclu[clu$b_namlng] + suffix_size ;
;  584        1080  7          !
;  585        1081  7          !   The size of this global isd   =   length of private isd
;  586        1082  7          !                                   + length of gblsect ident
;  587        1083  7          !                                   + length of gblsect name count byte
;  588        1084  7          !                                   + length of gblsect name
;  589        1085  7          !
;  590        1086  7          newhdrisd [isd$w_size] = isd$k_lenpriv + .gblsect_namlng + 5 ; ! SET SIZE AND
;  591        1087  7          newhdrisd [isd$v_gbl]  = true ;                                ! MAKE IT GLOBAL NOW
;  592        1088  7                                                                        !  GLOBAL ISDS
;  593        1089  7          decr i from 3 to 1                                            ! COMPUTE ISD NAME BY
;  594        1090  8          do  begin
;  595        1091  8              if (curisdseq [.i] = .curisdseq [.i] + 1) gtru %c'9'      ! INCREMENTING THE SUFFIX
;  596        1092  8              then
;  597        1093  8                  curisdseq [.i] = %c'0'
;  598        1094  8              else
;  599        1095  8                  exitloop ;
;  600        1096  7              end ;
;  601        1097  7
;  602        1098  7          (newhdrisd [isd$t_gblnam])<0, 8, 0> = .gblsect_namlng ;
;  603        1099  7
;  604        1100  7                                                             ! COPY THE CLUSTER
;  605        1101  7                                                             ! NAME CONCATENATED WITH
;  606        1102  7          ch$copy (.lnk$gl_curclu [clu$b_namlng]
;  607        1103  7                  ,lnk$gl_curclu [clu$t_name], 4              ! THE SEQUENTIAL NUMBER
;  608        1104  7                  ,curisdseq [0], 0, .gblsect_namlng          ! OF THE SECTION AND ZERO
;  609        1105  7                  ,newhdrisd [isd$t_gblnam] + 1
;  610        1106  7                  ) ;                                         ! FILL THEN SET MATCH CONTROL
;  611        1107  7          newhdrisd [isd$v_matchctl] = .lnk$gl_curclu [clu$v_matchctl] ;
```

```
                        newhdrisd [isd$l_ident]    = .isectident ;          ! AND THE MATCH CONTROL IDENT
                        end ;

                    if not .curhdrisd [isd$v_dzro]                          ! IF NOT DEMAND ZERO AND
                       and .lnk$gl_curclu [clu$v_copy]                      ! SHAREABLE IMAGE TO BE COPIED
                       and .curhdrisd [isd$l_vbn] neq 0                     ! AND SECTION IS PRESENT IN THIS
                    then                                                    ! ONE
                        begin
                        if .curhdrisd [isd$w_pagcnt] gtru .lnk$gl_gsbufdsc [0]      ! MAXIMIZE THE SIZE OF
                        then                                                         ! BUFFER WE'LL NEED
                            lnk$gl_gsbufdsc [0] = .curhdrisd [isd$w_pagcnt] ;
                        end ;

                    if .iafva neq 0                                         ! IF THERE IS A FIXUP SECTION
                       and
                       (.newhdrisd [isd$v_vpg]^9 eql .iafva)                !  AND THIS IS IT
                    then
                        begin
                        lnk$gl_curclu [clu$l_fixisd] = .curisd ;            !  THEN REMEMBER IT FOR LATER
                        newhdrisd [isd$v_fixupvec] = true ;                 ! FLAG FIXUP SECTION IN ISD
                        end ;                                               ! UNCONDITIONALLY

                    lnk$gw_nisects = .lnk$gw_nisects + 1 ;                  ! COUNT THAT IMAGE SECTION
                    end ;                                                   ! OF LOCAL ISECT

                if (curhdrisd = .curhdrisd + .nxtisdoff) gtru (.header + 510) ! CHECK IT WAS COMPLETELY
                then                                                        ! CONTAINED BY THE CURRENT BLOCK
                     signal_stop (lin$_badimghdr, 1                         ! IF NOT TRUE, FATAL IMAGE
                                 ,lnk$gl_curfil [fdb$q_filename]            !  HEADER ERROR
                                 ) ;
                    end                                                     ! END OF ISECTION LOOP
                else
                    !
                    !   IMAGE SECTION WAS A STACK ISD, JUST SKIP IT BUT MAKE SURE IT IS CONTAINED BY THE
                    !   CURRENT BLOCK -- ISSUE ERROR AND QUIT IF NOT
                    !
                    if (curhdrisd = .curhdrisd + .nxtisdoff) gtru (.header + 510)
                    then
                        signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename]) ;

            end ;                                               ! END OF BLOCK LOOP
        end ;

    if .maxisdvbn gtru .gstvbn [0] or .hdrblkcnt neq 0             ! AND THIS SHOULD ALSO POINT TO GST
                                                                   ! AND NO HEADER BLOCKS REMAIN
    then                                                                ! IF EITHER ABOVE NOT TRUE,
        signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename]) ;   ! FATAL IMAGE HEADER ERROR

    if     .lnk$gl_curclu [clu$v_based]                                 ! IF THIS CLUSTER IS BASED
       and .lnk$gl_defclu [clu$v_based]                                 ! AND BASE= IN OPTION FILE
       and .lnk$gl_curclu [clu$l_base] lequ .lnk$gl_defclu [clu$l_base]     !  AND THIS IMAGE IS BELOW IT
    then
        signal  (lin$_basshrbel, 3, lnk$gl_curfil [fdb$q_filename]          !  THEN WARN USER SHR IMG
                ,.lnk$gl_curclu [clu$l_base], .lnk$gl_defclu [clu$l_base]   !  BELOW BASE=
                ) ;

    gstvbn [0]  = .gstvbn [0] + .blockoffset ;                          ! RELOCATE DOWN FILE IF NECESSARY
```

```
2 gstvbn [1]  =  0 ;                                                       ! CLEAR THE BYTE OFFSET IN THE RFA
2 saverecount = .lnk$gl_objrecs ;                                          ! SAVE CURRENT RECORD COUNT
2 lnk$pointobj (gstvbn) ;                                                  ! POINT TO GST
2
2 if not lnk$procsobj (gstvbn)                                             ! AND GO PROCESS IT LIKE AN
2 then
2 return false ;                                                           ! OBJECT RETURNING IF ERROR
2
2 saverecount = .lnk$gl_objrecs - .saverecount ;                           ! GET NUMBER PROCESSED
2
2 if .saverecount neq .gstrecs                                             ! AND IF NOT CORRECT NUMBER
2 then
2     signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename]) ;   ! FATAL IMAGE HEADER ERROR
2
2 lnk$gl_imgrecs = .lnk$gl_imgrecs + .saverecount ;                        ! ACCUMULATE RECORD COUNT
2 return true                                                              ! ALL DONE SO RETURN SUCCESS
1 end ;                                                                    ! End of LNK$PROCSHRIM


                                                              .TITLE  LNK_PROCSHRIM
                                                              .IDENT  \V04-000\

                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

65 6C 69 66 20 73 6E 6F 69 74 70 4F  00000 P.AAB:  .ASCII  \Options file\                        ;
                          0000000C  0000C P.AAA:  .LONG   12                                     ;
                         00000000'  00010         .ADDRESS P.AAB                                 ;

                                                  .PSECT  $OWN$,NOEXE,2

                       30 30 30 5F  00000 CURISDSEQ:
                                                  .ASCII  \_000\                                 ;
                                    00004 HDRBLKCNT:
                                                  .BLKB   4
                                    00008 HEADERBLOCK:
                                                  .BLKB   4

                                                  .PSECT  $GLOBAL$,NOEXE,2

                                    00000 LNK$GL_GSBUFDSC::
                                                  .BLKB   12
                                    0000C LNK$GL_IMGRECS::
                                                  .BLKB   4

                                          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$K_MINFILL==       2
                                                  .EXTRN  LNK$ALLOBLK, LNK$ALLOCLUSTER
                                                  .EXTRN  LNK$CLOSEFILE, LNK$POINTOBJ
                                                  .EXTRN  LNK$ADDIMAGE, LNK$PROCSOBJ
                                                  .EXTRN  LIN$_BADIMGHDR, LIN$_BASSHRBEL
```

```
                                                                   .EXTRN   LIN$_CONFBASADR
                                                                   .EXTRN   LIN$_IDMISMCH, LIN$_IMGBASED
                                                                   .EXTRN   LIN$_NOIMGFIL, LIN$_NONPICIMG
                                                                   .EXTRN   LIN$_RELINK, LIN$_READERR
                                                                   .EXTRN   LNK$GL_SHRCLSTRS
                                                                   .EXTRN   LNK$AL_RAB, LNK$GW_NISECTS
                                                                   .EXTRN   LNK$GW_SHRISCTS
                                                                   .EXTRN   LNK$GB_PASS, LNK$GL_CURFIL
                                                                   .EXTRN   LNK$GL_CURCLU, LNK$GL_DEFCLU
                                                                   .EXTRN   LNK$GL_LASTCLU, LNK$GL_CTLMSK
                                                                   .EXTRN   LNK$GL_OBJRECS

                                                                   .PSECT   $CODE$,NOWRT,2

                                                0FFC    00000      .ENTRY   LNK$PROCSHRIM, Save R2,R3,R4,R5,R6,R7,R8,-      ; 0652
                                                                            R9,R10,R11                                      ;
                                   5E        A8   AE 9E 00002      MOVAB    -88(SP), SP                                     ;
                                                  6C 95 00006      TSTB     (AP)                                            ; 0715
                                                  04 12 00008      BNEQ     1$                                              ;
                                                  59 D4 0000A      CLRL     BLOCKOFFSET                                     ; 0717
                                                  05 11 0000C      BRB      2$                                              ;
                      59        04 BC             01 C3 0000E 1$:  SUBL3    #1, @MODRFA, BLOCKOFFSET                        ; 0719
                                      00000000G   00 D6 00013 2$:  INCL     LNK$GL_SHRCLSTRS                                ; 0721
                                             40   AE D4 00019      CLRL     CLUDESC                                         ; 0723
                         00000000' EF             01 D0 0001C      MOVL     #1, HDRBLKCNT                                   ; 0724
                                1C AE             01 CE 00023      MNEGL    #1, NXTISDOFF                                   ; 0725
                                             24   AE D4 00027      CLRL     MAXISDVBN                                       ; 0726
                         00000000' EF             59 D0 0002A      MOVL     BLOCKOFFSET, HEADERBLOCK                        ; 0727
00000000'   EF        18           00  00303030   8F F0 00031      INSV     #3158064, #0, #24, CURISDSEQ+1                  ; 0729
                                2C AE        01   A9 9E 0003E      MOVAB    1(R9), 44(SP)                                   ; 0738
                          FFFFFFFF 8F        1C   AE D1 00043 3$:  CMPL     NXTISDOFF, #-1                                  ; 0731
                                                  03 13 0004B      BEQL     4$                                              ;
                                                0613 31 0004D      BRW      59$                                             ;
                         00000000V EF             00 FB 00050 4$:  CALLS    #0, READNEXTBLOCK                               ; 0733
                                   17             50 E8 00057      BLBS     R0, 5$                                          ;
                      7E 00000000G 00             14 C1 0005A      ADDL3    #20, LNK$GL_CURFIL, -(SP)                       ; 0736
                                                  01 DD 00062      PUSHL    #1                                              ;
                                      00000000G   8F DD 00064      PUSHL    #LIN$_BADIMGHDR                                 ;
                         00000000G 00             03 FB 0006A      CALLS    #3, LIB$STOP                                    ;
                                   50 00000000G   00 D0 00071 5$:  MOVL     HEADER, R0                                      ; 0744
                                2C AE 00000000'   EF D1 00078      CMPL     HEADERBLOCK, 44(SP)                             ; 0738
                                                  03 13 00080      BEQL     6$                                              ;
                                                01C2 31 00082      BRW      17$                                             ;
                                   02        11   A0 91 00085 6$:  CMPB     17(R0), #2                                      ; 0744
                                                  71 12 00089      BNEQ     9$                                              ;
                              3230 8F        0C   A0 B1 0008B      CMPW     12(R0), #12848                                  ; 0745
                                                  69 12 00091      BNEQ     9$                                              ;
                                   30        0E   A0 91 00093      CMPB     14(R0), #48                                     ; 0746
                                                  06 12 00097      BNEQ     7$                                              ;
                                   35        0F   A0 91 00099      CMPB     15(R0), #53                                     ; 0748
                                                  5D 1A 0009D      BGTRU    9$                                              ;
                                   51        08   A0 3C 0009F 7$:  MOVZWL   8(R0), R1                                       ; 0750
                                   51             2C C0 000A3      ADDL2    #44, R1                                         ;
                          000000A8 8F             51 D1 000A6      CMPL     R1, #168                                        ;
                                                  04 1E 000AD      BGEQU    8$                                              ;
                                   51        A8   8F 9A 000AF      MOVZBL   #168, R1                                        ;
            51        60           10             00 ED 000B3 8$:  CMPZV    #0, #16, (R0), R1                               ;
```

```
                                                     42  1A 000B8          BGTRU   9$                                           ;
                            00000000'  EF       10   A0  9A 000BA          MOVZBL  16(R0), HDRBLKCNT                            ; 0754
                                          00000000'  EF  D7 000C2          DECL    HDRBLKCNT                                    ;
                                                     32  19 000C8          BLSS    9$                                           ;
                                       58       04   A0  3C 000CA          MOVZWL  4(R0), SYMDBGDATDSC                          ; 0755
                                       58            50  C0 000CE          ADDL2   R0, SYMDBGDATDSC                             ;
                                       51            60  3C 000D1          MOVZWL  (R0), R1                                     ; 0756
                                       50            51  C0 000D4          ADDL2   R1, R0                                       ;
                                       50            58  D1 000D7          CMPL    SYMDBGDATDSC, R0                             ;
                                                     20  1E 000DA          BGEQU   9$                                           ;
                                  28   AE       0A   A8  3C 000DC          MOVZWL  10(SYMDBGDATDSC), GSTRECS                    ; 0757
                                       03       28   AE  D1 000E1          CMPL    GSTRECS, #3                                  ;
                                                     15  1F 000E5          BLSSU   9$                                           ;
                                       50       04   A8  D0 000E7          MOVL    4(SYMDBGDATDSC), R0                          ; 0758
                                  50   AE            50  D0 000EB          MOVL    R0, GSTVBN                                   ;
                         51 00000000'  EF            01  C1 000EF          ADDL3   #1, HDRBLKCNT, R1                            ; 0759
                                       51            50  D1 000F7          CMPL    R0, R1                                       ;
                                                     17  1A 000FA          BGTRU   10$                                          ;
                         7E 00000000G  00            14  C1 000FC 9$:      ADDL3   #20, LNK$GL_CURFIL, -(SP)                    ; 0762
                                                     01  DD 00104          PUSHL   #1                                           ;
                                          00000000G  8F  DD 00106          PUSHL   #LIN$_BADIMGHDR                              ;
                            00000000G  00            03  FB 0010C          CALLS   #3, LIB$STOP                                 ;
                                       50 00000000G  00  D0 00113 10$:     MOVL    LNK$GL_CURCLU, R0                            ; 0764
                                       51 00000000G  00  D0 0011A          MOVL    HEADER, R1                                   ;
           51       20   A1                          01  03  EF 00121          EXTZV   #3, #1, 32(R1), R1                           ;
      58   A0            01                          03  51  F0 00127          INSV    R1, #3, #1, 88(R0)                           ;
                                       38            51  E8 0012D          BLBS    R1, 11$                                      ;
                            00000000G  00            02  8A 00130          BICB2   #2, LNK$GL_CTLMSK+2                          ; 0767
                         38       59   A0            02  E1 00137          BBC     #2, 89(R0), 12$                              ; 0769
                         33       59   A0            01  E0 0013C          BBS     #1, 89(R0), 12$                              ; 0770
                         7E 00000000G  00            14  C1 00141          ADDL3   #20, LNK$GL_CURFIL, -(SP)                    ; 0774
                                                     01  DD 00149          PUSHL   #1                                           ;
                                          00000000G  8F  DD 0014B          PUSHL   #LIN$_IMGBASED                               ;
                            00000000G  00            03  FB 00151          CALLS   #3, LIB$SIGNAL                               ;
                                       50 00000000G  00  D0 00158          MOVL    LNK$GL_CURCLU, R0                            ; 0776
                                  59   A0            04  8A 0015F          BICB2   #4, 89(R0)                                   ;
                                                3C   A0  D4 00163          CLRL    60(R0)                                       ; 0777
                                                     0C  11 00166          BRB     12$                                          ; 0764
                         07       59   A0            02  E1 00168 11$:     BBC     #2, 89(R0), 12$                              ; 0781
                            00000000G  00            02  8A 0016D          BICB2   #2, LNK$GL_CTLMSK+2                          ; 0783
                         2B 00000000G  00            02  E1 00174 12$:     BBC     #2, LNK$GL_CTLMSK, 13$                       ; 0785
                         23 00000000G  00            03  E0 0017C          BBS     #3, LNK$GL_CTLMSK+3, 13$                     ; 0786
                                       50 00000000G  00  D0 00184          MOVL    LNK$GL_CURCLU, R0                            ; 0787
                         17       58   A0            03  E0 0018B          BBS     #3, 88(R0), 13$                              ;
                         7E 00000000G  00            14  C1 00190          ADDL3   #20, LNK$GL_CURFIL, -(SP)                    ; 0790
                                                     01  DD 00198          PUSHL   #1                                           ;
                                          00000000G  8F  DD 0019A          PUSHL   #LIN$_NONPICIMG                              ;
                            00000000G  00            03  FB 001A0          CALLS   #3, LIB$SIGNAL                               ;
                                       51 00000000G  00  D0 001A7 13$:     MOVL    LNK$GL_CURCLU, R1                            ; 0792
                                       50 00000000G  00  D0 001AE          MOVL    HEADER, R0                                   ; 0793
           52       23   A0                          03  00  EF 001B5          EXTZV   #0, #3, 35(R0), R2                           ;
      59   A1            03                          05  52  F0 001BB          INSV    R2, #5, #3, 89(R1)                           ;
                                       03            52  D1 001C1          CMPL    R2, #3                                       ;
                                                     04  12 001C4          BNEQ    14$                                          ;
                                  58   A1            10  88 001C6          BISB2   #16, 88(R1)                                  ; 0795
                                  38   AE       24   A0  D0 001CA 14$:     MOVL    36(R0), ISECTIDENT                           ; 0797
                                                20   AE  D4 001CF          CLRL    IAFVA                                        ; 0798
                            52         02   A0 3C 001D2         MOVZWL   2(R0), R2                                   ; 0800
                            52              50 C0 001D6         ADDL2    R0, R2                                      ;
                            53         2C   A0 9E 001D9         MOVAB    44(R0), R3                                  ; 0801
                            53              52 D1 001DD         CMPL     R2, R3                                      ;
                                            07 1B 001E0         BLEQU    15$                                         ;
                        20  AE         2C   A0 D0 001E2         MOVL     44(R0), IAFVA                               ; 0804
                                            04 12 001E7         BNEQ     16$                                         ; 0805
                        59  A1              01 88 001E9 15$:    BISB2    #1, 89(R1)                                  ; 0810
            52   00000000'  EF  00000000'   EF C1 001ED 16$:    ADDL3    HDRBLKCNT, LNK$GL_IMGRECS, R2               ; 0812
                 00000000'  EF         01   A2 9E 001F9         MOVAB    1(R2), LNK$GL_IMGRECS                       ;
                            56              60 3C 00201         MOVZWL   (R0), CURHDRISD                             ; 0813
                            56              50 C0 00204         ADDL2    R0, CURHDRISD                               ;
                            51       0084   C1 9E 00207         MOVAB    132(R1), R1                                 ; 0819
                                            61 D5 0020C         TSTL     (R1)                                        ;
                                            3A 13 0020E         BEQL     18$                                         ;
                        38  AE              61 D1 00210         CMPL     (R1), ISECTIDENT                            ; 0820
                                            34 13 00214         BEQL     18$                                         ;
                            50  00000000G   00 D0 00216         MOVL     LNK$GL_CURFIL, R0                           ; 0829
                                       1C   A0 9F 0021D         PUSHAB   28(R0)                                      ;
  7E        61                18              00 EF 00220         EXTZV    #0, #24, (R1), -(SP)                        ;
                            7E         03   A1 9A 00225         MOVZBL   3(R1), -(SP)                                ;
                                       14   A0 9F 00229         PUSHAB   20(R0)                                      ; 0828
  7E    48  AE                18              00 EF 0022C         EXTZV    #0, #24, IMGIDENT, -(SP)                    ; 0829
                            7E         4F   AE 9A 00232         MOVZBL   IMGIDENT+3, -(SP)                           ;
                                            06 DD 00236         PUSHL    #6                                          ;
                                00000000G   8F DD 00238         PUSHL    #LIN$_IDMISMCH                              ;
                 00000000G  00              08 FB 0023E         CALLS    #8, LIB$SIGNAL                              ;
                                            03 11 00245         BRB      18$                                         ; 0738
                            56              50 D0 00247 17$:    MOVL     R0, CURHDRISD                               ; 0834
                        1C  AE              66 32 0024A 18$:    CVTWL    (CURHDRISD), NXTISDOFF                      ; 0839
                                            03 14 0024E         BGTR     19$                                         ;
                                          FDF0 31 00250         BRW      3$                                          ;
                        FD  8F         0B   A6 91 00253 19$:    CMPB     11(CURHDRISD), #253                         ; 0841
                                            03 12 00258         BNEQ     20$                                         ;
                                          03D7 31 0025A         BRW      57$                                         ;
                        14  AE         08   A6 9E 0025D 20$:    MOVAB    8(CURHDRISD), 20(SP)                        ; 0844
            1B          14  BE              08 E1 00262         BBC      #8, @20(SP), 21$                            ;
            50   00000000G  00              14 C1 00267         ADDL3    #20, LNK$GL_CURFIL, R0                      ; 0848
                                            50 DD 0026F         PUSHL    R0                                          ;
                                            50 DD 00271         PUSHL    R0                                          ;
                                            02 DD 00273         PUSHL    #2                                          ;
                                00000000G   8F DD 00275         PUSHL    #LIN$_RELINK                                ;
                 00000000G  00              04 FB 0027B         CALLS    #4, LIB$STOP                                ;
                            03         14   BE E8 00282 21$:    BLBS     @20(SP), 22$                                ; 0851
                                          0149 31 00286         BRW      37$                                         ;
                            50         14   A6 9E 00289 22$:    MOVAB    20(CURHDRISD), R0                           ; 0859
                        4C  AE         01   A0 9E 0028D         MOVAB    1(R0), SHRDESC+4                            ;
                        48  AE         14   A6 9B 00292         MOVZBW   20(CURHDRISD), SHRDESC                      ; 0860
                        48  AE         04   A2 00297            SUBW2    #4, SHRDESC                                 ;
                            51         40   AE D0 0029B         MOVL     CLUDESC, R1                                 ; 0862
                                            0C 13 0029F         BEQL     23$                                         ;
  5D        A1          01  A0         48   AE 29 002A1         CMPC3    SHRDESC, 1(R0), 93(R1)                      ; 0864
                                            03 12 002A8         BNEQ     23$                                         ;
                                          0102 31 002AA         BRW      34$                                         ;
                                       3C   AE 9F 002AD 23$:    PUSHAB   FOUND                                       ; 0868
                                       44   AE 9F 002B0         PUSHAB   CLUDESC                                     ;
                                            7E D4 002B3         CLRL     -(SP)                                       ;
```

```
                                                              D  9
LNK_PROCSHRIM                                                16-Sep-1984 00:30:18    VAX-11 Bliss-32 V4.0-742          Page  17
V04-000                                                      14-Sep-1984 12:40:34    [LINKER.SRC]LNKPROSHR.B32;2            (2)

                                                54   AE  9F 002B5        PUSHAB  SHRDESC                      ;
                          00000000G  00              04  FB 002B8        CALLS   #4, LNK$ADDIMAGE             ;
                                     50  00000000G   00  D0 002BF        MOVL    LNK$GL_CURCLU, R0            ; 0872
                                     51         58   A0  9E 002C6        MOVAB   88(R0), R1                   ;
                                     52         40   AE  D0 002CA        MOVL    CLUDESC, R2                  ; 0869
                                                     38  13 002CE        BEQL    28$                          ;
                                     34         3C   AE  E8 002D0        BLBS    FOUND, 28$                   ;
59     A2              01            00         01   A1  F0 002D4        INSV    1(R1), #0, #1, 89(R2)        ; 0872
                                 54  A2              50  D0 002DB        MOVL    R0, 84(R2)                   ; 0874
                       03            61              03  E1 002DF        BBC     #3, (R1), 25$                ; 0876
                                                   00E5  31 002E3 24$:   BRW     36$                          ;
                                 58  A2              01  88 002E6 25$:   BISB2   #1, 88(R2)                   ; 0879
                       0C        14  BE              09  E1 002EA        BBC     #9, @20(SP)  26$             ; 0880
       53        04    A6            17              00  EF 002EF        EXTZV   #0, #23, 4(CURHDRISD), R3    ; 0881
                       53            53              09  78 002F5        ASHL    #9, R3, R3                   ;
                                                     06  11 002F9        BRB     27$                          ;
                       53        4C  A0         20   A0  C1 002FB 26$:   ADDL3   32(R0), 76(R0), R3           ; 0883
                                 4C  A2              53  D0 00301 27$:   MOVL    R3, 76(R2)                   ; 0880
                                                   00B7  31 00305        BRW     35$                          ; 0886
                                     04              61  E9 00308 28$:   BLBC    (R1), 29$                    ; 0891
                       04            61              03  E1 0030B        BBC     #3, (R1), 30$                ; 0893
                       D0            61              0A  E1 0030F 29$:   BBC     #10, (R1), 24$               ; 0895
                                     6E         54   A2  D0 00313 30$:   MOVL    84(R2), OWNCLU               ; 0898
                                                     54  D4 00317        CLRL    R4                           ; 0900
                                                     6E  D5 00319        TSTL    OWNCLU                       ;
                                                     0E  13 0031B        BEQL    31$                          ;
                                                     54  D6 0031D        INCL    R4                           ;
                                     53         02   A6  3C 0031F        MOVZWL  2(CURHDRISD), R3             ; 0903
                       53            53              09  78 00323        ASHL    #9, R3, R3                   ;
                                 20  A0              53  C0 00327        ADDL2   R3, 32(R0)                   ;
                                     60              54  E9 0032B 31$:   BLBC    R4, 32$                      ; 0905
                       50            6E   00000058   8F  C1 0032E        ADDL3   #88, OWNCLU, R0              ; 0907
                                     55              60  E9 00336        BLBC    (R0), 32$                    ;
       53        04    A6            17              00  EF 00339        EXTZV   #0, #23, 4(CURHDRISD), R3    ; 0909
                       53            53              09  78 0033F        ASHL    #9, R3, R3                   ;
                                     53         4C   A2  D1 00343        CMPL    76(R2), R3                   ;
                                                     45  13 00347        BEQL    32$                          ;
                       50            6E              08  C1 00349        ADDL3   #8, OWNCLU, R0               ; 0914
                                 18  AE              60  D0 0034D        MOVL    (R0), OWNFDB                 ;
                                         00000000G   8F  DD 00351        PUSHL   #LIN$_NOIMGFIL               ; 0918
                       50        1C  AE              14  C1 00357        ADDL3   #20, OWNFDB, R0              ;
                                                     50  DD 0035C        PUSHL   R0                           ;
                                                4C   A2  DD 0035E        PUSHL   76(R2)                       ;
                       7E 00000000G  00              14  C1 00361        ADDL3   #20, LNK$GL_CURFIL, -(SP)    ; 0917
       50        04    A6            17              00  EF 00369        EXTZV   #0, #23, 4(CURHDRISD), R0    ;
                       7E            50              09  78 0036F        ASHL    #9, R0, -(SP)                ;
                                                5C   A2  9F 00373        PUSHAB  92(R2)                       ; 0916
                                                     05  DD 00376        PUSHL   #5                           ; 0918
                                         00000000G   8F  DD 00378        PUSHL   #LIN$_CONFBASADR             ;
                          00000000G  00              08  FB 0037E        CALLS   #8, LIB$SIGNAL               ;
                          00000000G  00              01  8A 00385        BICB2   #1, LNK$GL_CTLMSK            ; 0921
                                                     3D  11 0038C        BRB     36$                          ; 0905
                       39            61              0A  E0 0038E 32$:   BBS     #10, (R1), 36$               ; 0925
                                     08         58   A2  E8 00392        BLBS    88(R2), 33$                  ; 0928
                          00000000G  00         48   A2  A2 00396        SUBW2   72(R2), LNK$GW_SHRISCTS      ; 0930
                                 58  A2              01  88 0039E 33$:   BISB2   #1, 88(R2)                   ; 0932
       50        04    A6            17              00  EF 003A2        EXTZV   #0, #23, 4(CURHDRISD), R0    ; 0933
```

```
                   4C   A2            50            09 78 003A8       ASHL     #9, R0, 76(R2)                             ;
                                                    1C 11 003AD       BRB      36$                                        ; 0891
                                      50 00000000G  00 D0 003AF 34$:  MOVL     LNK$GL_CURCLU, R0                          ; 0941
                                      05        58  A0 E8 003B6       BLBS     88(R0), 35$                                ;
                        0C        59  A0            02 E1 003BA       BBC      #2, 89(R0), 36$                            ;
                                      53        02  A6 3C 003BF 35$:  MOVZWL   2(CURHDRISD), R3                           ; 0944
                        53            53            09 78 003C3       ASHL     #9, R3, R3                                 ;
                                  20  A0            53 C0 003C7       ADDL2    R3, 32(R0)                                 ;
                                      03        14  BE E9 003CB 36$:  BLBC     @20(SP), 37$                               ; 0950
                                                  0262 31 003CF       BRW      57$                                        ;
                                                44  AE 9F 003D2 37$:  PUSHAB   CURISD                                     ; 0953
                                      7E        58  8F 9A 003D5       MOVZBL   #88, -(SP)                                 ;
                           00000000G  00            02 FB 003D9       CALLS    #2, LNK$ALLOBLK                            ;
                                      5B        44  AE D0 003E0       MOVL     CURISD, R11                                ; 0954
                                                    6B D4 003E4       CLRL     (R11)                                      ;
                                                08  AB 7C 003E6       CLRQ     8(R11)                                     ; 0955
                                      5A 00000000G  00 D0 003E9       MOVL     LNK$GL_CURCLU, R10                         ; 0957
                                  10  AB            5A D0 003F0       MOVL     R10, 16(R11)                               ;
                                      57        18  AB 9E 003F4       MOVAB    24(R11), NEWHDRISD                         ; 0958
    0040   8F           00            66            66 2C 003F8       MOVC5    (CURHDRISD), (CURHDRISD), #0, #64, 24(R11) ; 0961
                                                18  AB    003FF                                                           ;
                                                18  AA D5 00401       TSTL     24(R10)                                    ; 0963
                                                    03 13 00404       BEQL     38$                                        ;
                                                  00C8 31 00406       BRW      43$                                        ;
                                      51        58  AA 9E 00409 38$:  MOVAB    88(R10), R1                                ; 0966
                        52            61            0A E1 0040D       BBC      #10, (R1), 40$                             ;
                                      3A            61 E9 00411       BLBC     (R1), 39$                                  ; 0969
                                  3C  AA        4C  AA D1 00414       CMPL     76(R10), 60(R10)                           ; 0971
                                                    33 13 00419       BEQL     39$                                        ;
                                         00000000G  8F DD 0041B       PUSHL    #LIN$_NOIMGFIL                             ; 0976
                                         00000000'  EF 9F 00421       PUSHAB   P.AAA                                      ; 0977
                                                3C  AA DD 00427       PUSHL    60(R10)                                    ;
                        7E 00000000G  00            14 C1 0042A       ADDL3    #20, LNK$GL_CURFIL, -(SP)                  ; 0976
                                                4C  AA DD 00432       PUSHL    76(R10)                                    ;
                                                5C  AA 9F 00435       PUSHAB   92(R10)                                    ; 0975
                                                    05 DD 00438       PUSHL    #5                                         ; 0976
                                         00000000G  8F DD 0043A       PUSHL    #LIN$_CONFBASADR                           ;
                           00000000G  00            08 FB 00440       CALLS    #8, LIB$SIGNAL                             ;
                           00000000G  00            01 8A 00447       BICB2    #1, LNK$GL_CTLMSK                          ; 0980
                                      50 00000000G  00 D0 0044E 39$:  MOVL     LNK$GL_CURCLU, R0                          ; 0983
                                  4C  A0        3C  A0 D0 00455       MOVL     60(R0), 76(R0)                             ;
                                                3C  A0 D4 0045A       CLRL     60(R0)                                     ; 0984
                                  58  A0            01 88 0045D       BISB2    #1, 88(R0)                                 ; 0985
                                                    6A 11 00461       BRB      42$                                        ; 0966
                                      55            61 E9 00463 40$:  BLBC     (R1), 41$                                  ; 0989
                        63            61            03 E0 00466       BBS      #3, (R1), 42$                              ; 0991
           50      04   A7            17            00 EF 0046A       EXTZV    #0, #23, 4(NEWHDRISD), R0                  ; 0994
                        50            50            09 78 00470       ASHL     #9, R0, R0                                 ;
                                      50        4C  AA D1 00474       CMPL     76(R10), R0                                ;
                                                    53 13 00478       BEQL     42$                                        ;
                                      6E        54  AA D0 0047A       MOVL     84(R10), OWNCLU                            ; 0997
                        52            6E            08 C1 0047E       ADDL3    #8, OWNCLU, R2                             ; 0998
                                  18  AE            62 D0 00482       MOVL     (R2), OWNFDB                               ;
                                         00000000G  8F DD 00486       PUSHL    #LIN$_NOIMGFIL                             ; 1002
                        51        1C  AE            14 C1 0048C       ADDL3    #20, OWNFDB, R1                            ;
                                                    51 DD 00491       PUSHL    R1                                         ;
                                                4C  AA DD 00493       PUSHL    76(R10)                                    ;
```

```
                            7E 00000000G    00              14  C1 00496          ADDL3    #20, LNK$GL_CURFIL, -(SP)               ; 1001
                                                            50  DD C049E          PUSHL    R0                                      ; 1002
                                                      5C    AA  9F 004A0          PUSHAB   92(R10)                                 ; 1000
                                                            05  DD 004A3          PUSHL    #5                                      ; 1002
                                               00000000G    8F  DD 004A5          PUSHL    #LIN$_CONFBASADR                        ;
                               00000000G    00              08  FB 004AB          CALLS    #8, LIB$SIGNAL                          ;
                               00000000G    00              01  8A 004B2          BICB2    #1, LNK$GL_CTLMSK                       ; 1005
                                                            12  11 004B9          BRB      42$                                     ; 0993
                            0E              61              03  E0 004BB 41$:     BBS      #3, (R1), 42$                           ; 1009
            50        04    A7              17              00  EF 004BF          EXTZV    #0, #23, 4(NEWHDRISD), R0               ; 1012
                      4C    AA              50              09  78 004C5          ASHL     #9, R0, 76(R10)                         ;
                                            61              01  88 004CA          BISB2    #1, (R1)                                ; 1013
                                      30    AE              57  D0 004CD 42$:     MOVL     NEWHDRISD, FIRSTISD                     ; 1016
                                            5A 00000000G    00  D0 004D1 43$:     MOVL     LNK$GL_CURCLU, R10                      ; 1019
                                                      48    AA  D6 004D8          INCL     72(R10)                                 ;
                                      1C    BA              5B  D0 004DB          MOVL     R11, @28(R10)                           ; 1026
                                      1C    AA              5B  D0 004DF          MOVL     R11, 28(R10)                            ; 1027
                                      04    AE        58    AA  9E 004E3          MOVAB    88(R10), 4(SP)                          ; 1031
                            27        04    BE              03  E1 004E8          BBC      #3, @4(SP), 45$                         ;
                                      30    AE              57  D1 004ED          CMPL     NEWHDRISD, FIRSTISD                     ; 1034
                                                            07  12 004F1          BNEQ     44$                                     ;
      34    AE        04    A7              17              00  EF 004F3          EXTZV    #0, #23, 4(NEWHDRISD), FIRSTISDVPG      ; 1036
            50        04    A7              17              00  EF 004FA 44$:     EXTZV    #0, #23, 4(NEWHDRISD), R0               ; 1039
                                            50        34    AE  C2 00500          SUBL2    FIRSTISDVPG, R0                         ;
      04    A7              17              00              50  F0 00504          INSV     R0, #0, #23, 4(NEWHDRISD)               ;
                                            06        04    BE  E8 0050A          BLBS     @4(SP), 45$                             ; 1042
                                               00000000G    00  B6 0050E          INCW     LNK$GW_SHRISCTS                         ; 1043
                                            50        02    A7  3C 00514 45$:     MOVZWL   2(NEWHDRISD), R0                        ; 1046
                                      50    AA              50  C0 00518          ADDL2    R0, 80(R10)                             ;
                                            6E        54    AA  D0 0051C          MOVL     84(R10), OWNCLU                         ; 1049
                                                            0E  12 00520          BNEQ     46$                                     ;
                                            50        02    A7  3C 00522          MOVZWL   2(NEWHDRISD), R0                        ; 1052
                            50              50              09  78 00526          ASHL     #9, R0, R0                              ;
                                      20    AA              50  C0 0052A          ADDL2    R0, 32(R10)                             ;
                                                            0F  11 0052E          BRB      47$                                     ; 1051
                                            50        02    A7  3C 00530 46$:     MOVZWL   2(NEWHDRISD), R0                        ; 1054
                            50              50              09  78 00534          ASHL     #9, R0, R0                              ;
                            51              6E              20  C1 00538          ADDL3    #32, OWNCLU, R1                         ;
                                            61              50  C0 0053C          ADDL2    R0, (R1)                                ;
                            0A        14    BE              03  E1 0053F 47$:     BBC      #3, @20(SP), 48$                        ; 1056
                            05        14    BE              01  E0 00544          BBS      #1, @20(SP), 48$                        ; 1058
                                      04    BE        40    8F  88 00549          BISB2    #64, @4(SP)                             ; 1060
                                            04        04    BE  E9 0054E 48$:     BLBC     @4(SP), 49$                             ; 1062
                                      09    A7              02  88 00552          BISB2    #2, 9(NEWHDRISD)                        ; 1064
                            16        14    BE              02  E0 00556 49$:     BBS      #2, @20(SP), 50$                        ; 1066
                                                      0C    A6  D5 0055B          TSTL     12(CURHDRISD)                           ; 1067
                                                            11  13 0055E          BEQL     50$                                     ;
                                      24    AE        0C    A6  D1 00560          CMPL     12(CURHDRISD), MAXISDVBN                ; 1069
                                                            0A  1F 00565          BLSSU    50$                                     ;
                                            50        02    A6  3C 00567          MOVZWL   2(CURHDRISD), R0                        ; 1072
                                      24    AE        0C  B640  9E 0056B          MOVAB    @12(CURHDRISD)[R0], MAXISDVBN           ;
                                            79        14    BE  E8 00571 50$:     BLBS     @20(SP), 54$                            ; 1074
                                            50        5C    AA  9A 00575          MOVZBL   92(R10), GBLSECT_NAMLNG                 ; 1079
                                            50              04  C0 00579          ADDL2    #4, GBLSECT_NAMLNG                      ;
                            67              50              15  A1 0057C          ADDW3    #21, GBLSECT_NAMLNG, (NEWHDRISD)        ; 1086
                                      08    A7              01  88 00580          BISB2    #1, 8(NEWHDRISD)                        ; 1087
                                            51              03  D0 00584          MOVL     #3, I                                   ; 1089
```

```
                                        52 00000000'EF41  9A 00587 51$:    MOVZBL  CURISDSEQ[I], R2                              ; 1091
                                                      52  D6 0058F         INCL    R2                                            ;
                             00000000'EF41            52  90 00591         MOVB    R2, CURISDSEQ[I]                              ;
                                        39            52  D1 00599         CMPL    R2, #57                                       ;
                                                      0B  1B 0059C         BLEQU   52$                                           ;
                             00000000'EF41            30  90 0059E         MOVB    #48, CURISDSEQ[I]                             ; 1093
                                        DE            51  F5 005A6         SOBGTR  I, 51$                                        ; 1089
                                   14   A7            50  90 005A9 52$:    MOVB    GBLSECT_NAMLNG, 20(NEWHDRISD)                 ; 1098
                                   10   AE       5C   AA  9A 005AD         MOVZBL  92(R10), 16(SP)                               ; 1102
                                   0C   AE            50  D0 005B2         MOVL    GBLSECT_NAMLNG, 12(SP)                        ; 1104
                                   08   AE       15   A7  9E 005B6         MOVAB   21(R7), 8(SP)                                 ; 1105
      0C   AE             00       5D   AA       10   AE  2C 005BB         MOVC5   16(SP), 93(R10), #0, 12(SP), @8(SP)           ;
                                                 08   BE     005C3                                                               ;
                                                      16  18 005C5         BGEQ    53$                                           ;
                                   08   AE       10   AE  C0 005C7         ADDL2   16(SP), 8(SP)                                 ;
                                   0C   AE       10   AE  C2 005CC         SUBL2   16(SP), 12(SP)                                ;
      0^   AE             00 00000000'  EF            04  2C 005D1         MOVC5   #4, CURISDSEQ, #0, 12(SP), @8(SP)             ;
                                                 08   BE     005DB                                                               ;
           50        04   BE            03            0D  EF 005DD 53$:    EXTZV   #13, #3, @4(SP), R0                           ; 1107
      08   A7             03            04            50  F0 005E3         INSV    R0, #4, #3, 8(NEWHDRISD)                      ;
                                   10   A7       38   AE  D0 005E9         MOVL    ISECTIDENT, 16(NEWHDRISD)                     ; 1108
                          1E       14   BE            02  E0 005EE 54$:    BBS     #2, @20(SP), 55$                              ; 1111
                          19       04   BE            04  E1 005F3         BBC     #4, @4(SP), 55$                               ; 1112
                                                 0C   A6  D5 005F8         TSTL    12(CURHDRISD)                                 ; 1113
                                                      14  13 005FB         BEQL    55$                                           ;
00000000'  EF        02   A6            10            00  ED 005FD         CMPZV   #0, #16, 2(CURHDRISD), LNK$GL_GSBUFDSC        ; 1116
                                                      08  1B 00607         BLEQU   55$                                           ;
                             00000000'  EF       02   A6  3C 00609         MOVZWL  2(CURHDRISD), LNK$GL_GSBUFDSC                 ; 1118
                                                 20   AE  D5 00611 55$:    TSTL    IAFVA                                         ; 1121
                                                      18  13 00614         BEQL    56$                                           ;
           50        04   A7            17            00  EF 00616         EXTZV   #0, #23, 4(NEWHDRISD), R0                     ; 1123
                          50            50            09  78 0061C         ASHL    #9, R0, R0                                    ;
                                   20   AE            50  D1 00620         CMPL    R0, IAFVA                                     ;
                                                      08  12 00624         BNEQ    56$                                           ;
                                   44   AA            5B  D0 00626         MOVL    R11, 68(R10)                                  ; 1126
                                   09   A7            04  88 0062A         BISB2   #4, 9(NEWHDRISD)                              ; 1127
                                           00000000G  00  B6 0062E 56$:    INCW    LNK$GJ_NISECTS                                ; 1130
                                        56       1C   AE  C0 00634 57$:    ADDL2   NXTISDOFF, CURHDRISD                          ; 1144
                          50 00000000G  00 000001FE   8F  C1 00638         ADDL3   #510, HEADER, R0                              ;
                                        50            56  D1 00644         CMPL    CURHDRISD, R0                                 ;
                                                      17  1B 00647         BLEQU   58$                                           ;
                          7E 00000000G  00            14  C1 00649         ADDL3   #20, LNK$GL_CURFIL, -(SP)                     ; 1146
                                                      01  DD 00651         PUSHL   #1                                            ;
                                           00000000G  8F  DD 00653         PUSHL   #LIN$_BADIMGHDR                               ;
                             00000000G  00            03  FB 00659         CALLS   #3, LIB$STOP                                  ;
                                                    FBE7  31 00660 58$:    BRW     18$                                           ; 0839
                                   50   AE       24   AE  D1 00663 59$:    CMPL    MAXISDVBN, GSTVBN                             ; 1151
                                                      08  1A 00668         BGTRU   60$                                           ;
                                           00000000'  EF  D5 0066A         TSTL    HDRBLKCNT                                     ;
                                                      17  13 00670         BEQL    61$                                           ;
                          7E 00000000G  00            14  C1 00672 60$:    ADDL3   #20, LNK$GL_CURFIL, -(SP)                     ; 1154
                                                      01  DD 0067A         PUSHL   #1                                            ;
                                           00000000G  8F  DD 0067C         PUSHL   #LIN$_BADIMGHDR                               ;
                             00000000G  00            03  FB 00682         CALLS   #3, LIB$STOP                                  ;
                                        50 00000000G  00  D0 00689 61$:    MOVL    LNK$GL_CURCLU, R0                             ; 1156
                                        31       58   A0  E9 00690         BLBC    88(R0), 62$                                   ;
                                        2A 00000000G  00  E9 00694         BLBC    LNK$GL_DEFCLU+88, 62$                         ; 1157
```

```
                    00000000G  00         4C   A0  D1 0069B        CMPL    76(R0), LNK$GL_DEFCLU+76                        ; 1158
                                               20  1A 006A3        BGTRU   62$                                             ;
                                00000000G  00  DD 006A5        PUSHL   LNK$GL_DEFCLU+76                                ; 1161
                                          4C   A0  DD 006AB        PUSHL   76(R0)                                          ;
                 7E 00000000G  00              14  C1 006AE        ADDL3   #20, LNK$GL_CURFIL, -(SP)                       ; 1160
                                               03  DD 006B6        PUSHL   #3                                              ;
                                00000000G  8F  DD 006B8        PUSHL   #LIN$_BASSHRBEL                                 ;
                    00000000G  00              05  FB 006BE        CALLS   #5, LIB$SIGNAL                                  ;
                           50  AE              59  C0 006C5 62$:   ADDL2   BLOCKOFFSET, GSTVBN                             ; 1164
                                          54   AE  D4 006C9        CLRL    GSTVBN+4                                        ; 1165
                             52 00000000G  00  D0 006CC        MOVL    LNK$GL_OBJRECS, SAVERECOUNT                     ; 1166
                                          50   AE  9F 006D3        PUSHAB  GSTVBN                                          ; 1167
                    00000000G  00              01  FB 006D6        CALLS   #1, LNK$POINTOBJ                                ;
                                          50   AE  9F 006DD        PUSHAB  GSTVBN                                          ; 1169
                    00000000G  00              01  FB 006E0        CALLS   #1, LNK$PROCSOBJ                                ;
                               30              50  E9 006E7        BLBC    R0, 64$                                         ;
                 52 00000000G  00              52  C3 006EA        SUBL3   SAVERECOUNT, LNK$GL_OBJRECS, SAVERECOUNT        ; 1173
                           28  AE              52  D1 006F2        CMPL    SAVERECOUNT, GSTRECS                            ; 1175
                                               17  13 006F6        BEQL    63$                                             ;
                 7E 00000000G  00              14  C1 006F8        ADDL3   #20, LNK$GL_CURFIL, -(SP)                       ; 1177
                                               01  DD 00700        PUSHL   #1                                              ;
                                00000000G  8F  DD 00702        PUSHL   #LIN$_BADIMGHDR                                 ;
                    00000000G  00              03  FB 00708        CALLS   #3, LIB$STOP                                    ;
                    00000000'  EF              52  C0 0070F 63$:   ADDL2   SAVERECOUNT, LNK$GL_IMGRECS                     ; 1179
                               50              01  D0 00716        MOVL    #1, R0                                          ; 1180
                                                   04 00719        RET                                                     ;
                                               50  D4 0071A 64$:   CLRL    R0                                              ; 1181
                                                   04 0071C        RET                                                     ;

; Routine Size:  1821 bytes,    Routine Base:  $CODE$ + 0000


;  686          1182  1
;  688          1183  1 routine readnextblock =                          ! READ NEXT BLOCK IF ANY
;  689          1184  1 !++
;  690          1185  1 !       THIS ROUTINE IS CALLED TO READ THE NEXT BLOCK OF THE IMAGE HEADER
;  691          1186  1 !       READ ERRORS ARE FATAL. A REQUEST TO READ ANOTHER BLOCK
;  692          1187  1 !       WHEN HDRBLKCNT IS ALREADY ZERO RETURNS FATAL.
;  693          1188  1 !       HDRBLKCNT IS DECREMENTED AFTER EACH READ AND HEADERBLOCK IS
;  694          1189  1 !       INCREMENTED BEFORE EACH READ.
;  695          1190  1 !
;  696          1191  1 !--
;  697          1192  2 begin
;  698          1193  2 local
;  699          1194  2         saveusz,
;  700          1195  2         readerror ;
;  701          1196  2
;  702          1197  2 if .hdrblkcnt leq 0                                        ! IF NO MORE BLOCKS
;  703          1198  2 then
;  704          1199  2     return false ;                                        ! RETURN FAILURE
;  705          1200  2
;  706          1201  2 saveusz     = .lnk$al_rab [rab$w_usz] ;                   ! SAVE USZ
;  707          1202  2 headerblock = .headerblock + 1 ;                          ! SET THE BLOCK TO READ
;  708          1203  2
;  709          1204  2 lnk$al_rab [rab$l_bkt] = .headerblock ;                   ! SET STARTING VBN
;  710          1205  2 lnk$al_rab [rab$w_usz] = 512 ;                            ! AND SET THE BYTE COUNT
;  711          1206  2
;  712          1207  3 if not (readerror = $read (rab = lnk$al_rab))             ! ATTEMPT TO READ LIBRARY, USING
;  713          1208  2 then
;  714          1209  3     begin                                                 ! BUFFER ALREADY SET UP
;  715          1210  3     signal (lin$_readerr, 1                               !
;  716          1211  3             ,lnk$gl_curfil [fdb$q_filename]
;  717          1212  3             ,.readerror, .lnk$al_rab [rab$l_stv]
;  718          1213  3             ) ;
;  719          1214  3     lnk$closefile (.lnk$gl_curfil) ;                      ! THE MESSAGES AND ATTEMPT TO
;  720          1215  3     lnk$al_rab [rab$w_usz] = .saveusz ;                   ! RESTORE USZ
;  721          1216  3     return false ;
;  722          1217  2     end ;
;  723          1218  2
;  724          1219  2 hdrblkcnt = .hdrblkcnt - 1 ;                              ! DECREMENT THE BLOCK COUNT
;  725          1220  2 lnk$al_rab [rab$w_usz] = .saveusz ;
;  726          1221  2 return true ;                                             ! AND ALL DONE SUCCESSFULLY
;  727          1222  1 end ;                                                     ! END OF READNEXTBLOCK


                                                     .EXTRN  SYS$READ

                                 003C 00000 READNEYTBLOCK:
                                                     .WORD   Save R2,R3,R4,R5                           ; 1183
                55 00000000G  00  9E 00002           MOVAB   LNK$GL_CURFIL, R5                          ;
                54 00000000'  EF  9E 00009           MOVAB   HDRBLKCNT, R4                              ;
                53 00000000G  00  9E 00010           MOVAB   LNK$AL_RAB+32, R3                          ;
                              64  D5 00017           TSTL    HDRBLKCNT                                  ; 1197
                              4C  15 00019           BLEQ    2$                                         ;
                52            63  3C 0001B           MOVZWL  LNK$AL_RAB+32, SAVEUSZ                     ; 1201
                        04    A4  D6 0001E           INCL    HEADERBLOCK                                ; 1202
            18  A3          04    A4  D0 00021       MOVL    HEADERBLOCK, LNK$AL_RAB+56                 ; 1204
                63        0200    8F  B0 00026       MOVW    #512, LNK$AL_RAB+32                        ; 1205
                            E0    A3  9F 0002B       PUSHAB  LNK$AL_RAB                                 ; 1207
                            00000000G  00            01  FB 0002E         CALLS    #1, SYS$READ                                ;
                                       26            50  E8 00035         BLBS     READERROR, 1$                               ;
                                                EC   A3  DD 00038         PUSHL    LNK$AL_RAB+12                               ; 1212
                                                     50  DD 0003B         PUSHL    READERROR                                   ;
                         7E            65            14  C1 0003D         ADDL3    #20, LNK$GL_CURFIL, -(SP)                   ; 1211
                                                     01  DD 00041         PUSHL    #1                                          ;
                                          00000000G  8F  DD 00043         PUSHL    #LIN$_READERR                               ;
                            00000000G  00            05  FB 00049         CALLS    #5, LIB$SIGNAL                              ;
                                                     65  DD 00050         PUSHL    LNK$GL_CURFIL                               ; 1214
                            00000000G  00            01  FB 00052         CALLS    #1, LNK$CLOSEFILE                           ;
                                       63            52  B0 00059         MOVW     SAVEUSZ, LNK$AL_RAB+32                      ; 1215
                                                     09  11 0005C         BRB      2$                                          ; 1216
                                                     64  D7 0005E 1$:     DECL     HDRBLKCNT                                   ; 1219
                                       63            52  B0 00060         MOVW     SAVEUSZ, LNK$AL_RAB+32                      ; 1220
                                       50            01  D0 00063         MOVL     #1, R0                                      ; 1221
                                                         04 00066         RET                                                  ;
                                                     50  D4 00067 2$:     CLRL     R0                                          ; 1222
                                                         04 00069         RET                                                  ;

; Routine Size:  106 bytes,    Routine Base:  $CODE$ + 071D


;  728          1223  1 end
;  729          1224  1
;  730          1225  0 eludom                                                        ! End of module



                                                                          .EXTRN  LIB$SIGNAL, LIB$STOP

;                                      PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;  $GLOBAL$                               16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                                  12  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                                 20  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                               1927  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        68         0      1000         00:02.0
;  _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1          538        41         7        28         00:00.8
;                                   COMMAND QUALIFIERS

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

; Size:          1927 code + 48 data bytes
; Run Time:         00:50.4
; Elapsed Time:     02:02.6
; Lines/CPU Min:     1457
; Lexemes/CPU-Min: 28711
; Memory Used:  563 pages
; Compilation Complete
```

0219 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LNKPROLIB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKVMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSO
LIS