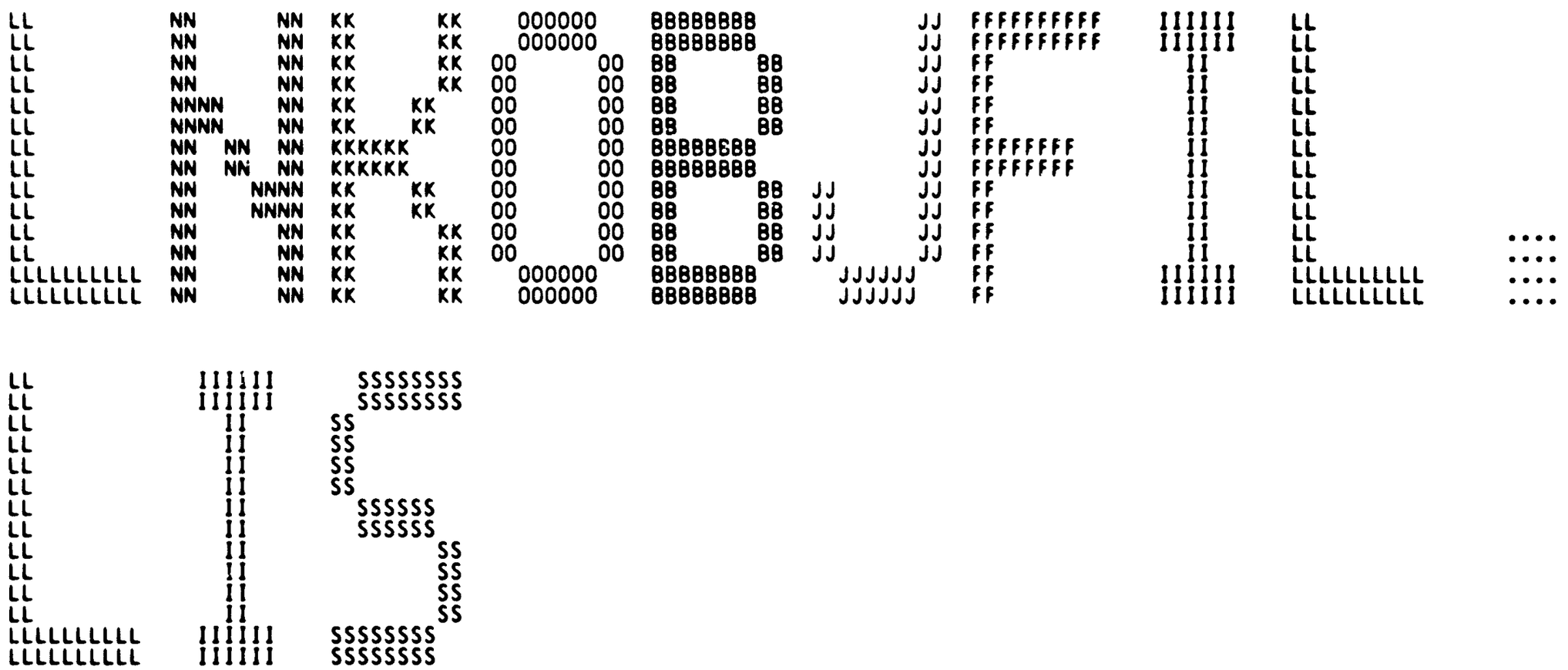
```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL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


LL NN NN KK KK 000000 BBBBBBBB JJ FFFFFFFFFF IIIIII LL
LL NN NN KK KK 000000 BBBBBBBB JJ FFFFFFFFFF IIIIII LL
LL NN NN KK KK 00 00 BB BB JJ FF II LL
LL NN NN KK KK 00 00 BB BB JJ FF II LL
LL NNNN NN KK KK 00 00 BB BB JJ FF II LL
LL NNNN NN KK KK 00 00 BB BB JJ FF II LL
LL NN NN NN KKKKK 00 00 BBBBBBBB JJ FFFFFFFF II LL
LL NN NN NN KKKKK 00 00 BBBBBBBB JJ FFFFFFFF II LL
LL NN NNNN KK KK 00 00 BB BB JJ JJ FF II LL
LL NN NNNN KK KK 00 00 BB BB JJ JJ FF II LL
LL NN NN KK KK 00 00 BB BB JJ JJ FF II LL
LL NN NN KK KK 00 00 BB BB JJ JJ FF II LL
LLLLLLLLLL NN NN KK KK 000000 BBBBBBBB JJJJJJ FF IIIIII LLLLLLLLLL
LLLLLLLLLL NN NN KK KK 000000 BBBBBBBB JJJJJJ FF IIIIII LLLLLLLLLL
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;  1          0001  0 module lnk_objfil
;  2          0002  0                 (ident = 'V04-000'
;  3          0003  0                 ,addressing_mode
;  4          0004  0                         (external = general)
;  5          0005  0                 ) =
;  6          0006  1 begin
;  7          0007  1 !
;  8          0008  1 !
;  9          0009  1 !*****************************************************************************
; 10          0010  1 !*                                                                           *
; 11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
; 12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 13          0013  1 !*  ALL RIGHTS RESERVED.                                                     *
; 14          0014  1 !*                                                                           *
; 15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 20          0020  1 !*  TRANSFERRED.                                                             *
; 21          0021  1 !*                                                                           *
; 22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 24          0024  1 !*  CORPORATION.                                                             *
; 25          0025  1 !*                                                                           *
; 26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 28          0028  1 !*                                                                           *
; 29          0029  1 !*                                                                           *
; 30          0030  1 !*****************************************************************************
; 31          0031  1 !
; 32          0032  1 !
; 33          0033  1 !
; 34          0034  1 !
; 35          0035  1 !
; 36          0036  1 !++
; 37          0037  1 !
; 38          0038  1 ! MODULE:        LNK_OBJFIL
; 39          0039  1 !
; 40          0040  1 ! FACILITY:      LINKER
; 41          0041  1 !
; 42          0042  1 ! ABSTRACT:      ROUTINES TO OPEN AND READ ALL RECORDS OF OBJECT FILES
; 43          0043  1 !
; 44          0044  1 ! HISTORY:
; 45          0045  1 !
; 46          0046  1 !       VERSION:        X01.00
; 47          0047  1 !
; 48          0048  1 !       AUTHOR: T.J. PORTER 03-MAY-77
; 49          0049  1 !
; 50          0050  1 ! MODIFIED BY:
; 51          0051  1 !
; 52          0052  1 !       V03-022 ADE0004         Alan D. Eldridge        17-Jul-1984
; 53          0053  1 !               Search default cluster when looking for a library to
; 54          0054  1 !               close as well as the cluster list.
; 55          0055  1 !
; 56          0056  1 !       V03-021 ADE0003         Alan D. Eldridge        11-Jul-1984
; 57          0057  1 !               Fix infinite loop which occurred in selected cases when
```

```
;    58          0058  1 !                the librarian had too many files open.
;    59          0059  1 !
;    60          0060  1 !        V03-020 ADE0002         Alan D. Eldridge        30-Apr-1984
;    61          0061  1 !                Signal proper RMS error status upon failure to open a
;    62          0062  1 !                shared image.
;    63          0063  1 !
;    64          0064  1 !        V03-019 ADE0001         Alan D. Eldridge         7-Mar-1984
;    65          0065  1 !                Close a file and retry on a LBR$_TOOMNYLIB error from
;    66          0066  1 !                LBR$OPEN.
;    67          0067  1 !
;    68          0068  1 !        V03-018 JWT0113         Jim Teague              20-Apr-1983
;    69          0069  1 !                Call $getjpi to get number of open files left.
;    70          0070  1 !
;    71          0071  1 !        V03-017 JWT0057         Jim Teague              30-Sep-1982
;    72          0072  1 !                Fixed bug in LNK$CLOSEFILE logic.
;    73          0073  1 !
;    74          0074  1 !        V03-016 JWT0054         Jim Teague              15-Sep-1982
;    75          0075  1 !                Fix another bug in LNK$CLOSEFILE logic.  The initial
;    76          0076  1 !                input file processing can now more intelligently specify
;    77          0077  1 !                a file to close when necessary.  Also streamline LNK$NXTFIL.
;    78          0078  1 !
;    79          0079  1 !        V03-015 JWT0051         Jim Teague              11-Aug-1982
;    80          0080  1 !                Fix bug in LNK$CLOSEFILE logic.
;    81          0081  1 !
;    82          0082  1 !        V03-014 JWT0044         Jim Teague              30-Jul-1982
;    83          0083  1 !                Open file performance boost.
;    84          0084  1 !
;    85          0085  1 !        V03-013 JWT0033         Jim Teague              25-May-1982
;    86          0086  1 !                Clear SQO bit if FDB has any non-TIR object modules
;    87          0087  1 !
;    88          0088  1 !--
```

```
;    90          0089  1 !
;    91          0090  1 !++
;    92          0091  1 !
;    93          0092  1 ! FUNCTIONAL DESCRIPTION:
;    94          0093  1 !
;    95          0094  1 !       THIS MODULE CONTAINS A ROUTINE TO OPEN THE NEXT
;    96          0095  1 !       OBJECT MODULE FILE AND ANOTHER TO READ THE NEXT
;    97          0096  1 !       RECORD FROM THE CURRENTLY OPEN FILE. IT ALSO CONTAINS THE
;    98          0097  1 !       ROUTINE TO CLOSE THE CURRENTLY OPEN FILE.
;    99          0098  1 !
;   100          0099  1 !               LNK$NXTFIL
;   101          0100  1 !               LNK$NXTREC(RECADRS,RECLNG)
;   102          0101  1 !               LNK$CLOSCURFIL()
;   103          0102  1 !
;   104          0103  1 !       IF THERE ARE NO MORE FILES OR RECORDS, ROUTINES HAVE THE VALUE FALSE.
;   105          0104  1 !       IF FILE IS SUCCESSFULLY OPENED (BY FILE ID USING LINKED LIST OF FDB'S)
;   106          0105  1 !       LNK$NXTFIL HAS THE VALUE TRUE AND THE ADDRESS OF THE
;   107          0106  1 !       FILE DESCRIPTOR BLOCK IS PLACED IN LNK$GL_CURFIL.
;   108          0107  1 !
;   109          0108  1 !       THERE IS NO RETURN ON THE FOLLOWING ERROR CONDITIONS:
;   110          0109  1 !               OPEN FAILURE
;   111          0110  1 !               CONNECT FAILURE
;   112          0111  1 !
;   113          0112  1 !       CALLING SEQUENCE:
;   114          0113  1 !               LNK$NXTFIL()
;   115          0114  1 !               LNK$NXTREC(RECADRS,RECLNG)
;   116          0115  1 !               LNK$CLOSCURFIL()
;   117          0116  1 !       WHERE:
;   118          0117  1 !               RECADRS = ADDRESS OF LONGWORD TO RECEIVE THE
;   119          0118  1 !                         ADDRESS OF NEXT RECORD.
;   120          0119  1 !               RECLNG  = ADDRESS OF A WORD TO RECEIVE LENGTH
;   121          0120  1 !                         OF THE RECORD (IN BYTES).
;   122          0121  1 !       THE ROUTINE RETURNS THE VALUE TRUE, UNLESS END OF FILE
;   123          0122  1 !       IS DETECTED, WHEN IT RETURNS THE VALUE FALSE.
;   124          0123  1 !--
;   125          0124  1 !
;   126          0125  1 library
;   127          0126  1         'SYS$LIBRARY:STARLET.L32';                      ! SYSTEM STRUCTURE DEFINITIONS
;   128          0127  1 require
;   129          0128  1         'PREFIX';
;   130          0243  1 library
;   131          0244  1         'DATBAS';                                       ! INTERNAL DATA BASE DEFINITIONS
;   132          0245  1 !
;   133          0246  1 forward routine
;   134          0247  1     tran_next_lib,                                      ! RETURN NEXT DEFAULT LIBRARY NAME
;   135          0248  1     lnk$closefile       : novalue,                      ! SEARCH FOR A FILE TO CLOSE
;   136          0249  1     lnk$openlib;                                        ! OPEN LIBRARY
;   137          0250  1
;   138          0251  1 external routine
;   139          0252  1     lnk$alloblk         : novalue,                      ! DYNAMIC MEMORY ALLOCATOR
;   140          0253  1     lnk$dealblk         : novalue,                      ! AND DEALLOCATOR
;   141          0254  1     lnk$allocluster     : novalue,                      ! ALLOCATE CLUSTER DESCRIPTORS
;   142          0255  1     lnk$allofdb         : novalue,                      ! ALLOCATE FILE DESCRIPTOR BLOCKS
;   143          0256  1     lnk$filnamdsc,                                      ! RETURN STRING DESCRIPTOR FOR FILENAME
;   144          0257  1     sys$fao,                                            ! FORMATTED ASCII OUTPUT
;   145          0258  1     lbr$lookup_key,                                     ! LOOKUP KEY IN MODULE
;   146          0259  1     lbr$set_module,                                     ! READ MODULE HEADER
```

```
;  147     0260  1      lbr$find,                                   ! POSITION TO READ MODULE
;  148     0261  1      lbr$ini_control,                            ! INITIALIZE TO READ LIBRARY
;  149     0262  1      lbr$get_record,                             ! READ RECORD FROM LIBRARY
;  150     0263  1      lbr$open,                                   ! OPEN LIBRARY
;  151     0264  1      lbr$close;                                  ! CLOSE LIBRARY
;  152     0265  1
;  153     0266  1  external literal
;  154     0267  1      lbr$_toomnylib,                             ! TOO MANY .OLB's FOR LIBRARIAN
;  155     0268  1      lin$_closein,                               ! CLOSE ERROR MESSAGE
;  156     0269  1      lin$_format,                                ! FORMAT ERROR ON A FILE
;  157     0270  1      lin$_libfind,                               ! FAILED TO GET (BY RFA) CORRECT LIBR RECORD
;  158     0271  1      lin$_notobjlib,                             ! FILE NOT OBJECT LIBRARY
;  159     0272  1      lin$_notimglib,                             ! FILE NOT SHAREABLE IMAGE STB LIBRARY
;  160     0273  1      lin$_openin,                                ! FAILED TO OPEN INPUT FILE
;  161     0274  1      lin$_readerr;                               ! READ ERROR
;  162     0275  1
;  163     0276  1  external
;  164     0277  1      lnk$gt_jpilst,
;  165     0278  1      lnk$gl_filesleft,
;  166     0279  1      lbr$gl_control      : ref block [,byte],    ! CURRENT LIBRARY CONTROL ADDRESS
;  167     0280  1      lbr$gl_rmsstv,                              ! RMS STV FROM LIBRARY OPERATIONS
;  168     0281  1      lnk$gl_ulibmask     : bitvector [],         ! ENABLE BITS FOR USER LIBRARIES
;  169     0282  1      lnk$gl_ctlmsk       : block [,byte],        ! CONTROL FLAGS
;  170     0283  1      lnk$gl_clulst       : vector [2],           ! CLUSTER DESCRIPTOR LISTHEAD
;  171     0284  1      lnk$gl_lastclu      : ref block [,byte],    ! LAST CLUSTER IN LIST DESCR.
;  172     0285  1      lnk$gl_curclu       : ref block [,byte],    ! CURRENT CLUSTER POINTER
;  173     0286  1      lnk$gl_defclu       : block [,byte],        ! DEFAULT CLUSTER
;  174     0287  1      lnk$gw_nudfsyms     : word,                 ! NUMBER OF UNDEFINED SYMBOLS
;  175     0288  1      lnk$gb_pass         : byte;                 ! THE PASS CURRENTLY EXECUTING
;  176     0289  1
;  177     0290  1  global literal
;  178     0291  1      lnk$c_objmbc = 5;                           ! MULTI BLOCK COUNT FOR OBJ FILES
;  179     0292  1
;  180     0293  1  global
;  181     0294  1      lnk$gl_open_lbr : initial (0),              ! NUMBER OF CURRENTLY OPEN LIBRARIES
;  182     0295  1      lnk$gl_objrecs,                             ! ACCUMULATED OBJECT RECORDS READ
;  183     0296  1      lnk$gl_record,                              ! RECORD NUMBER THIS FILE
;  184   P 0297  1      lnk$al_rab      : $rab (rac = seq           ! RECORD ACCESS BLOCK SPECIFYING
;  185   P 0298  1                             ,rop = loc          ! SEQUENTIAL,LOCATE MODE
;  186   P 0299  1                             ,mbc = lnk$c_objmbc ! WITH MULTIPLE BLOCKS
;  187     0300  1                             ),
;  188     0301  1      lnk$gl_curomd : ref block [,byte],          ! POINTER TO CURRENT OBJ MOD. DESCRIPTOR
;  189     0302  1      lnk$gl_curfil : ref block [,byte];          ! POINTER TO CURRENT FDB
;  190     0303  1
;  191     0304  1  own
;  192     0305  1      openclus      : ref block [,byte],          ! PTR TO CURRENT CLUSTER FOR OPEN FILE SEARCHES
;  193     0306  1      openfdb       : ref block [,byte],          ! PTR TO CURRENT CLUSTER FDB FOR OPEN FILE SEARCHES
;  194     0307  1      start_fdb,                                  ! STARTING POINT IN FDB SEARCH
;  195     0308  1      input_fab     : block [fab$c_bln, byte],    ! FAB BLOCK FOR INPUT FILES
;  196     0309  1      intfiltbl     : quadvector [2]              ! TABLE OF INTERNAL FILES
;  197     0310  1                      initial (stringdesc ('STARLET')     ! SYSTEM DEFAULT OBJECT LIBRARY
;  198     0311  1                               ,stringdesc ('IMAGELIB')   ! LIBRARY OF SHAREABLE IMAGE SYMBOL TABLES
;  199     0312  1                               ),
;  200     0313  1      syslibdefext : descriptor ('SYS$LIBRARY:.OLB'),  ! DEFAULT FILENAME FOR STARLET AND IMAGELIB
;  201     0314  1      curdefext    : block [dsc$c_s_bln,byte],     ! STRING DESCRIPTOR FOR CURRENT DEFAULT STRING
;  202     0315  1      libdefext    : descriptor ('SYS$DISK:.OLB'), ! DEFAULT FILENAME FOR USER LIBRARY
;  203     0316  1      shrdefext    : descriptor ('SYS$LIBRARY:.EXE'),  ! DEFAULT FILENAME FOR SHAREABLE IMAGES
```

```
;  204          0317  1     libnamefao   : descriptor ('LNK$LIBRARY_!UW'),      ! FAO CONTROL STRING TO CREATE USER LIBRARY NAMES
;  205          0318  1     previous_fdb : ref block [,byte],                   ! POINTER TO PREVIOUS FDB.
;  206          0319  1     lastuserclu  : ref block [,byte],                   ! POINTER TO LAST USER CLUSTER (OR DEFAULT)
;  207          0320  1     nextintfil   : byte initial (3),                    ! FOUR INTERNAL FILES
;  208          0321  1     userlibno,                                          ! NUMBER OF NEXT DEFAULT USER LIBRARY
;  209          0322  1     first_time   : byte initial (true),                 ! INITIALLY FIRST TIME THROUGH
;  210          0323  1     default_lib  : byte initial (false),                ! AND NOT PROCESSING DEFAULT LIBRARIES
;  211          0324  1     deflibacmode : byte initial (0),                    ! ACCESS MODE INDEX
;  212          0325  1     deflibdsbmsk : vector [4,byte] initial (%x'060503'),! DISABLE TABLE SEARCH BIT MASK
;  213          0326  1     resultstring : vector [nam$c_maxrss,byte],          ! RESULT OF TRANSLATING LOGICAL NAME
;  214          0327  1     libnamdescr  : vector [2]                           ! DESCRIPTOR OF RESULTANT NAME BUFFER
;  215          0328  1                    initial (nam$c_maxrss,resultstring);
```

```
;  217          0329  1 global routine lnk$nxtfil =
;  218          0330  2 begin
;  219          0331  2 !
;  220          0332  2 !       CLOSE THE PREVIOUSLY OPEN FILE AND THEN
;  221          0333  2 !       IF THERE ARE ANY MORE FILES IN THE TABLE GO OPEN THE
;  222          0334  2 !       NEXT AND PUT ADDRESS OF ITS DESCRIPTOR IN CURRENT SLOT. IF
;  223          0335  2 !       NONE, RETURN FALSE.
;  224          0336  2 !
;  225          0337  2     routine openit =
;  226          0338  3         begin
;  227          0339  3         !
;  228          0340  3         !   THIS ROUTINE ATTEMPTS TO OPEN FILES AND RETURNS THE
;  229          0341  3         !   STATUS OF THE OPEN
;  230          0342  3         !
;  231          0343  3         local    openok;
;  232          0344  3
;  233          0345  3         if .lnk$gl_curfil [fdb$w_ifi] neq 0                     ! IF IFI IS INITIALIZED, THEN
;  234          0346  3         then return true                                        !  THE FILE IS ALREADY OPEN
;  235          0347  4         else begin                                              ! FILE IS NOT OPEN, SO
;  236          0348  4              openok = $open (fab=input_fab);                    !  TRY OPENING THE FILE
;  237          0349  4              if not .openok                                     ! IF OPEN FAILED AND THIS IS AND
;  238          0350  4              then if .lnk$gl_curclu [clu$v_intclu]              !  THIS IS INTERNALLY MATERIALIZED
;  239          0351  5                   then begin                                    !  CLUSTER, THEN TRY SYS$LIBRARY:
;  240          0352  5                        input_fab [fab$b_dns] = .shrdefext [0];
;  241          0353  5                        input_fab [fab$l_dna] = .shrdefext [1];
;  242          0354  5                        openok = $open (fab=input_fab);  ! TRY AGAIN
;  243          0355  4                        end;
;  244          0356  4              return (.openok);
;  245          0357  3              end;                                               ! RETURN OPEN STATUS
;  246          0358  2         end;                                                    ! OF OPENIT


                                                        .TITLE  LNK_OBJFIL
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    00  54  45  4C  52  41  54  53  00000 P.AAA:   .ASCII  \STARLET\<0>
                                    42  49  4C  45  47  41  4D  49  00008 P.AAB:   .ASCII  \IMAGELIB\
4C  4F  2E  3A  59  52  41  52  42  49  4C  24  53  59  53  00010 P.AAC:   .ASCII  \SYS$LIBRARY:.OLB\
                                                        42  0001F
00  00  42  4C  4F  2E  3A  4B  53  49  44  24  53  59  53  00020 P.AAD:   .ASCII  \SYS$DISK:.OLB\<0><0><0>
                                                        00  0002F
58  45  2E  3A  59  52  41  52  42  49  4C  24  53  59  53  00030 P.AAE:   .ASCII  \SYS$LIBRARY:.EXE\
                                                        45  0003F
57  55  21  5F  59  52  41  52  42  49  4C  24  4B  4E  4C  00040 P.AAF:   .ASCII  \LNK$LIBRARY_!UW\<0>
                                                        00  0004F

                                                        .PSECT  $OWN$,NOEXE,2

                                                00000 OPENCLUS:
                                                        .BLKB   4
                                                00004 OPENFDB:.BLKB   4
                                                00008 START_FDB:
                                                        .BLKB   4
                                                0000C INPUT_FAB:
                                                        .BLKB   80
```

```
00000007  0005C INTFILTBL:
                        .LONG   7
00000000' 00060         .ADDRESS P.AAA
00000008  00064         .LONG   8
00000000' 00068         .ADDRESS P.AAB
00000010  0006C SYSLIBDEFEXT:
                        .LONG   16
00000000' 00070         .ADDRESS P.AAC
          00074 CURDEFEXT:
                        .BLKB   8
0000000D  0007C LIBDEFEXT:
                        .LONG   13
00000000' 00080         .ADDRESS P.AAD
00000010  00084 SHRDEFEXT:
                        .LONG   16
00000000' 00088         .ADDRESS P.AAE
0000000F  0008C LIBNAMEFAO:
                        .LONG   15
00000000' 00090         .ADDRESS P.AAF
          00094 PREVIOUS_FDB:
                        .BLKB   4
          00098 LASTUSERCLU:
                        .BLKB   4
       03 0009C NEXTINTFIL:
                        .BYTE   3
          0009D         .BLKB   3
          000A0 USERLIBNO:
                        .BLKB   4
       01 000A4 FIRST_TIME:
                        .BYTE   1
       00 000A5 DEFAULT_LIB:
                        .BYTE   0
       00 000A6 DEFLIBACMODE:
                        .BYTE   0
          000A7         .BLKB   1
00060503  000A8 DEFLIBDSBMSK:
                        .LONG   394499
          000AC RESULTSTRING:
                        .BLKB   255
          001AB         .BLKB   1
000000FF  001AC LIBNAMDESCR:
                        .LONG   255
00000000' 001B0         .ADDRESS RESULTSTRING

                        .PSECT  $GLOBAL$,NOEXE,2

00000000  00000 LNK$GL_OPEN_LBR::
                        .LONG   0
          00004 LNK$GL_OBJRECS::
                        .BLKB   4
          00008 LNK$GL_RECORD::
                        .BLKB   4
       01 0000C LNK$AL_RAB::
                        .BYTE   1
       44 0000D         .BYTE   68
     0000 0000E         .WORD   0
00010000  00010         .LONG   65536
```

```
00000000  00014          .LONG   0
00000000  00018          .LONG   0
    0000# 0001C          .WORD   0[3]
     0000 00022          .WORD   0
00000000  00024          .LONG   0
     0000 00028          .WORD   0
       00 0002A          .BYTE   0
       00 0002B          .BYTE   0
     0000 0002C          .WORD   0
     0000 0002E          .WORD   0
00000000  00030          .LONG   0
00000000  00034          .LONG   0
00000000  00038          .LONG   0
00000000  0003C          .LONG   0
       00 00040          .BYTE   0
       00 00041          .BYTE   0
       00 00042          .BYTE   0
       05 00043          .BYTE   5
00000000  00044          .LONG   0
00000000  00048          .LONG   0
00000000  0004C          .LONG   0
          00050 LNK$GL_CUROMD::
                         .BLKB   4
          00054 LNK$GL_CURFIL::
                         .BLKB   4

                LNK$C_OBJMBC==      5
                         .EXTRN  LNK$ALLOBLK, LNK$DEALBLK
                         .EXTRN  LNK$ALLOCLUSTER
                         .EXTRN  LNK$ALLOFDB, LNK$FILNAMDSC
                         .EXTRN  SYS$FAO, LBR$LOOKUP_KEY
                         .EXTRN  LBR$SET_MODULE, LBR$FIND
                         .EXTRN  LBR$INI_CONTROL
                         .EXTRN  LBR$GET_RECORD, LBR$OPEN
                         .EXTRN  LBR$CLOSE, LBR$_TOOMNYLIB
                         .EXTRN  LIN$_CLOSEIN, LIN$_FORMAT
                         .EXTRN  LIN$_LIBFIND, LIN$_NOTOBJLIB
                         .EXTRN  LIN$_NOTIMGLIB, LIN$_OPENIN
                         .EXTRN  LIN$_READERR, LNK$GT_JPILST
                         .EXTRN  LNK$GL_FILESLEFT
                         .EXTRN  LBR$GL_CONTROL, LBR$GL_RMSSTV
                         .EXTRN  LNK$GL_ULIBMASK
                         .EXTRN  LNK$GL_CTLMSK, LNK$GL_CLULST
                         .EXTRN  LNK$GL_LASTCLU, LNK$GL_CURCLU
                         .EXTRN  LNK$GL_DEFCLU, LNK$GW_NUDFSYMS
                         .EXTRN  LNK$GB_PASS, SYS$OPEN

                         .PSECT  $CODE$,NOWRT,2

                000C 00000 OPENIT: .WORD   Save R2,R3                      ; 0337
53 00000000G   00   9E 00002       MOVAB   SYS$OPEN, R3                   ;
52       0000'  CF   9E 00009       MOVAB   INPUT_FAB, R2                  ;
50       0000'  CF   D0 0000E       MOVL    LNK$GL_CURFIL, R0              ; 0345
         0024   A0   B5 00013       TSTW    36(R0)                         ;
                04   13 00016       BEQL    1$                             ;
50               01   D0 00018       MOVL    #1, R0                         ; 0347
                     04 0001B       RET                                    ;
```

```
                                                52  DD 0001C 1$:     PUSHL   R2                                              ; 0348
                                          63        01  FB 0001E         CALLS   #1, SYS$OPEN                                    ;
                                          1B        50  E8 00021         BLBS    OPENOK, 2$                                      ; 0349
                                          51 000000000G   00  D0 00024         MOVL    LNK$GL_CURCLU, R1                       ; 0350
                                    0F    59  A1        01  E1 00028         BBC     #1, 89(R1), 2$                          ;
                                          35  A2    78  A2  90 00030         MOVB    SHRDEFEXT, INPUT_FAB+53                 ; 0352
                                          30  A2    7C  A2  D0 00035         MOVL    SHRDEFEXT+4, INPUT_FAB+48               ; 0353
                                                52  DD 0003A         PUSHL   R2                                              ; 0354
                                          63        01  FB 0003C         CALLS   #1, SYS$OPEN                                    ;
                                                04 0003F 2$:     RET                                                     ; 0358

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 0000

;  247          0359  2 !
;  248          0360  2 !
;  249          0361  2 ! Main body of LNK$NXTFIL
;  250          0362  2 !
;  251          0363  2 !
;  252          0364  2 local
;  253          0365  2         current_ifi,
;  254          0366  2         turn_sqo_off    : byte,                         ! SHOULD SQO BE SHUT OFF?
;  255          0367  2         auxfnb          : ref block [,byte],            ! POINTER TO AUXILIARY FILE NAME BLOCK
;  256          0368  2         rms_stv,                                        ! STV RETURNED ON RMS OPERATION
;  257          0369  2         errorcode       : block [4,byte],               !
;  258          0370  2         open_sts;                                       ! ERROR CODE ON OPEN
;  259          0371  2
;  260          0372  2 if .lnk$gl_curfil eql 0                                 ! IF NO CURRENT FILE
;  261          0373  3 then begin                                              ! START AT TOP OF LIST
;  262          0374  3      lnk$gl_curclu = .lnk$gl_clulst [0];                ! OF CLUSTERS
;  263          0375  3      previous_fdb  =  lnk$gl_curclu [clu$l_fstfdb];     ! SETTING PREVIOUS TO ITS LISTHEAD
;  264          0376  3      end
;  265          0377  3 else begin
;  266          0378  3      if  .lnk$gl_curfil [fdb$v_imglib]                  ! ALWAYS CLOSE SHR IMG LIBS IMMEDIATELY
;  267          0379  3      or  .lnk$gb_pass  eql 2                            ! DONE WITH FILE IF THIS IS PASS 2
;  268          0380  3      then lnk$closefile (.lnk$gl_curfil);               ! SO CLOSE THIS FILE
;  269          0381  3
;  270          0382  3      if  .lnk$gb_pass eql 1                             ! IF THIS IS PASS 1
;  271          0383  3      then lnk$gl_curfil [fdb$v_p1] = true;              !  THEN FLAG FILE PROCESSED IN PASS 1
;  272          0384  3
;  273          0385  3      if  .lnk$gl_curfil [fdb$v_libr]                    ! IF THAT LAST FILE WAS A LIBRARY
;  274          0386  3      and .lnk$gl_curfil [fdb$l_omdlst] eql 0            ! AND NO MODULES WERE LOADED
;  275          0387  4      then begin                                         ! WE DO NOT NEED IT ANYMORE
;  276          0388  4           lnk$closefile (.lnk$gl_curfil);                    ! SO CLOSE IT AND
;  277          0389  4           previous_fdb [fdb$l_nxtfdb] = .lnk$gl_curfil [fdb$l_nxtfdb];  ! REMOVE IT FROM THE LIST
;  278          0390  4
;  279          0391  4           if .previous_fdb [fdb$l_nxtfdb] eql 0          ! AND IF LAST ON THIS
;  280          0392  4           then lnk$gl_curclu [clu$l_lstfdb] = .previous_fdb;     ! CLUSTER LIST, FIX EOL POINTER
;  281          0393  4
;  282          0394  4           !**THIS IS A PROBLEM...THE BLOCK CAN'T BE DEALLOCATED!!!
;  283          0395  4           !***  LNK$DEALBLK (FDB$C_SIZE+NAM$C_BLN,.LNK$GL_CURFIL); ! AND DISPOSE OF ITS DESCRIPTOR
;  284          0396  4
;  285          0397  4           end
;  286          0398  3      else previous_fdb = .lnk$gl_curfil;                ! IF NOT EMPTY LIB, IT BECOMES THE PREVIOUS
;  287          0399  3
;  288          0400  3      if .lnk$gl_curfil [fdb$w_ifi] neq 0                !  IF FILE LEFT OPEN,
;  289          0401  3      then $disconnect (rab=lnk$al_rab);                 !   THEN DISCONNECT ITS RAB
```

```
;  290     0402  2      end;
;  291     0403  2
;  292     0404  2 while (lnk$gl_curfil = .previous_fdb [fdb$l_nxtfdb]) eql 0      ! GET NEXT FILE ON THIS CLUSTER LIST
;  293     0405  3 or    (.lnk$gl_curfil [fdb$v_p1] and .lnk$gb_pass eql 1)
;  294     0406  2 do if   .lnk$gl_curclu [clu$l_nxtclu] eql 0                    ! BUT IF THE END AND LAST CLUSTER
;  295     0407  3    and (.lnk$gb_pass eql 1 and .nextintfil neq 0)              ! OR THE DEFAULT CLUSTER ON
;  296     0408  3                                                               ! AND WE HAVE NOT PROCESSED DEFAULT LIBRARIE
;  297     0409  2    and  .lnk$gw_nudfsyms neq 0                                 ! AND THERE ARE STILL UNDEFINED SYMBOLS
;  298     0410  4    and (((.nextintfil eql 3) and .lnk$gl_ctlmsk [lnk$v_usrlib])
;  299     0411  3         or
;  300     0412  4         ((.nextintfil gtr 0) and .lnk$gl_ctlmsk [lnk$v_syslib])
;  301     0413  3        )
;  302     0414  3    then begin
;  303     0415  3         local  fileflags;
;  304     0416  3
;  305     0417  3         fileflags = fdb$m_libr or fdb$m_libsrch;               ! SET DEFAULT FILE FLAGS
;  306     0418  3         if .first_time                                         ! IF THIS IS THE FIRST TIME
;  307     0419  3         or .default_lib                                        ! OR IF PROCESSING USER LIBRARIES
;  308     0420  4         then begin
;  309     0421  4              first_time = false;                               ! NO LONGER FIRST TIME
;  310     0422  4              if not .lnk$gl_ctlmsk [lnk$v_usrlib]              ! IF USER LIBRARIES ARE DISABLED
;  311     0423  4              or not  tran_next_lib ()                          ! OR NO MORE DEFAULT LIBRARIES
;  312     0424  5              then begin
;  313     0425  5                   nextintfil = 2;                              ! JUST SKIP IT
;  314     0426  5                   default_lib = false;                         ! DISABLE USER LIBRARIES
;  315     0427  5                   lnk$gl_ctlmsk [lnk$v_usrlib] = false;
;  316     0428  5                   if not .lnk$gl_ctlmsk [lnk$v_syslib]         ! IF SYSLIB DISABLED
;  317     0429  6                   then  begin
;  318     0430  6                         lnk$gl_curfil = 0;                     ! RETURN NO MORE FILES
;  319     0431  6                         return false;
;  320     0432  5                         end;
;  321     0433  5                   end
;  322     0434  5              else begin
;  323     0435  5                   default_lib = true;                          ! FLAG PROCESSING DEFAULT LIBRARIES
;  324     0436  5                   userlibno   = .userlibno + 1;                ! SET FOR NEXT LIBRARY
;  325     0437  5                   nextintfil  = 3;                             ! ENSURE INDEX IS CORRECT
;  326     0438  4                   end;
;  327     0439  3              end;
;  328     0440  3
;  329     0441  3         if .nextintfil eql 2                                   ! IF READY TO PROCESS IMAGELIB
;  330     0442  3         then if not .lnk$gl_ctlmsk [lnk$v_sysshr]              ! IF /NOSYSSHR WAS SPECIFIED
;  331     0443  3              then nextintfil = 1;                              !  THEN SKIP SEARCH OF IMAGELIB
;  332     0444  3
;  333     0445  4         if (.default_lib or .nextintfil eql 1)                 ! IF DOING A DEFAULT OBJECT LIBRARY
;  334     0446  4         then begin                                             ! THEN WE NEED A DIFFERENT CLUSTER
;  335     0447  4              lnk$gl_curclu = lnk$gl_defclu;                    ! SET SEARCH TO KNOWN USER CLUSTER
;  336     0448  4              if  .lnk$gl_defclu [clu$l_fstfdb] eql 0           ! IF NO FILES IN THE CLUSTER
;  337     0449  4              and .lnk$gl_defclu [clu$l_nxtclu] eql 0           !  AND IT HAS NO CLUSTER FOLLOWING IT
;  338     0450  4              and .lnk$gl_lastclu neq lnk$gl_defclu             !  AND IT ISN'T THE LAST CLUSTER
;  339     0451  5              then begin                                        ! THEN WE MUST LINK INTO CLUSTER LIST
;  340     0452  5                   lnk$gl_lastclu [clu$l_nxtclu] = lnk$gl_defclu;
;  341     0453  5                   lnk$gl_lastclu = lnk$gl_defclu;
;  342     0454  4                   end;
;  343     0455  3              end;
;  344     0456  3
;  345     0457  3         previous_fdb  = .lnk$gl_curclu [clu$l_lstfdb];         ! PREVIOUS FILE IS LAST
;  346     0458  3         lnk$allofdb (previous_fdb [fdb$l_nxtfdb]);             ! ALLOCATE A FILE DESCRIPTOR
```

```
;  347          0459  3          lnk$gl_curfil = .previous_fdb [fdb$l_nxtfdb];                ! AND MAKE IT CURRENT
;  348          0460  3          lnk$gl_curclu [clu$l_lstfdb]  = .lnk$gl_curfil;              ! AND SET NEW LAST FDB
;  349          0461  3          lnk$gl_curfil [fdb$b_filflgs] = .fileflags;                  ! SET APPROPRIATE FLAGS
;  350          0462  3
;  351          0463  3          if .default_lib                                              ! IF THIS IS THE DEFAULT USER LIBRARY
;  352          0464  4          then begin
;  353          0465  4               ch$move (dsc$c_s_bln,libnamdescr,lnk$gl_curfil [fdb$w_usrnamlen]);
;  354          0466  4               ch$move (dsc$c_s_bln,libdefext,  lnk$gl_curfil [fdb$w_defnamlen]);
;  355          0467  4               end
;  356          0468  4          else begin
;  357          0469  4               ch$move (dsc$c_s_bln,intfiltbl [.nextintfil-1],lnk$gl_curfil [fdb$w_usrnamlen]);
;  358          0470  4               ch$move (dsc$c_s_bln,syslibdefext,lnk$gl_curfil [fdb$w_defnamlen]);
;  359          0471  3               end;
;  360          0472  3
;  361          0473  3          if not .default_lib
;  362          0474  3          then lnk$gl_ctlmsk [lnk$v_intfil] = true;                    ! RECORD WE ARE DOING THIS
;  363          0475  3          nextintfil = .nextintfil - 1;                                ! REDUCE FILE NUMBER
;  364          0476  3          exitloop;
;  365          0477  3          end
;  366          0478  3     else begin                                                        ! IF NOT LAST CLUSTER
;  367          0479  3          lnk$gl_curclu = .lnk$gl_curclu [clu$l_nxtclu];               ! MOVE TO NEXT
;  368          0480  3          if .lnk$gl_curclu eql 0
;  369          0481  4          then begin
;  370          0482  4               lnk$gl_ctlmsk [lnk$v_intfil] = false;                   ! TURN OFF INTERNAL FLAG
;  371          0483  4               lnk$gl_curfil = 0;
;  372          0484  4               return false;                                           ! AND RETURN NO MORE FILES
;  373          0485  3               end;
;  374          0486  3          previous_fdb = lnk$gl_curclu [clu$l_fstfdb];                 ! RESET TO TOP OF FILE LIST
;  375          0487  2          end;
;  376          0488  2
;  377          0489  2 auxfnb       = lnk$gl_curfil [fdb$t_auxfnb];                         ! POINT TO AUXILIARY FILENMAME BLOCK
;  378          0490  2 turn_sqo_off = .lnk$gb_pass eql 2  and  .lnk$gl_curfil [fdb$v_omdnobin];
;  379          0491  2
;  380        P 0492  2 $fab_init (fab = input_fab
;  381        P 0493  2            ,fac = get
;  382        P 0494  2            ,fop = sqo
;  383        P 0495  2            ,mrs = obj$c_maxrecsiz
;  384        P 0496  2            ,nam = .auxfnb
;  385          0497  2            );
;  386          0498  2
;  387          0499  2 if .turn_sqo_off
;  388          0500  3 then begin                                                           ! TIR RECORDS, OPEN WITHOUT SQO...
;  389          0501  3      lnk$closefile (.lnk$gl_curfil);
;  390          0502  3      input_fab [fab$v_sqo] = false;
;  391          0503  2      end;
;  392          0504  2
;  393          0505  2 current_ifi = .lnk$gl_curfil [fdb$w_ifi];
;  394          0506  2
;  395          0507  2 if .auxfnb [nam$w_fid_num] eql 0                                     ! FILE APPENDED TO THE LIST OR FILE HAS NEVE
;  396          0508  3 then begin
;  397          0509  3      input_fab [fab$b_fns] = .lnk$gl_curfil [fdb$w_usrnamlen];       ! IT HAS NEVER BEEN OPENED, SO SET
;  398          0510  3      input_fab [fab$l_fna] = .lnk$gl_curfil [fdb$l_usrnamadr];       ! THE FILE NAME LENGTH AND ADDRESS
;  399          0511  3      input_fab [fab$b_dns] = .lnk$gl_curfil [fdb$w_defnamlen];       ! SET DEFAULT NAME STRING SIZE
;  400          0512  3      input_fab [fab$l_dna] = .lnk$gl_curfil [fdb$l_defnamadr];       ! AND ADDRESS
;  401          0513  3      end                                                             ! OTHERWISE, CAN'T USE NAME BLOCK OPEN
;  402          0514  2 else input_fab [fab$v_nam] = true;                                   ! OTHERWISE SET OPEN BY FILE ID FLAG
;  403          0515  2
```

```
;  404     0516  2 if .lnk$gl_curfil [fdb>v_libr]                                  ! IF FILE IS A LIBRARY
;  405     0517  3 then begin                                                       ! THEN OPEN DIFFERENTLY
;  406     0518  5      if (open_sts = (if .current_ifi eql 0
;  407     0519  5                      then lnk$openlib (.auxfnb)
;  408     0520  5                      else true
;  409     0521  4         )            )
;  410     0522  4      then begin
;  411     0523  4           if  .lnk$gl_ctlmsk [lnk$v_intfil]                     ! IF THIS IS THE INTERNAL SHAREABLE IMAGE ST
;  412     0524  4           and .nextintfil eql 1                                 !  (REMEMBER NEXTINTFIL HAS BEEN DECREMENTED
;  413     0525  4           then if not .lnk$gl_curfil [fdb$v_imglib]             !  BUT DID NOT TURN OUT TO BE SHAREABLE IMAG
;  414     0526  5                then begin
;  415     0527  5                     signal (lin$_notimglib, 1                   ! ISSUE THE MESSAGE
;  416     0528  5                             ,lnk$gl_curfil [fdb$q_filename]
;  417     0529  5                             );
;  418     0530  5                     lnk$closefile (.lnk$gl_curfil);             ! CLOSE THE FILE
;  419     0531  5                     lnk$gl_curfil [fdb$v_libr] = true;          ! FORCE DESCRIPTOR TO EVAPORATE
;  420     0532  5                     return lnk$nxtfil ();                       ! RECURSE TO DISPOSE OF DESCRIPTOR AND RETUR
;  421     0533  4                     end;
;  422     0534  4           lnk$gl_record = 0;                                    ! INIT RECORD COUNTER
;  423     0535  4           return true;
;  424     0536  3           end;
;  425     0537  3      rms_stv = .lbr$gl_rmsstv;                                  ! THERE WAS AN ERROR. SET STV
;  426     0538  3      end
;  427     0539  3 else begin
;  428     0540  3      if .lnk$gl_curfil [fdb$v_shr]                              ! IF A SHAREABLE IMAGE FILE SET TO OPEN FOR
;  429     0541  4      then begin
;  430     0542  4           input_fab [fab$v_sqo]    = false;                     ! NOT SEQUENTIAL ONLY
;  431     0543  4           input_fab [fab$v_bro]    = true;                      ! RECORD AND BLOCK OPERATIONS
;  432     0544  4           input_fab [fab$v_upi]    = true;                      ! SHARED ACCESS
;  433     0545  4           input_fab [fab$v_shrget] = true;                      ! SET FOR SHARED GETS
;  434     0546  4           input_fab [fab$v_shrput] = true;                      ! AND PUTS (LINKER DOESN'T, BUT MUST SET SO
;  435     0547  3           end;                                                  !  CAN READ IT IN CASE INSTALLED /WRITE)
;  436     0548  3
;  437     0549  3      if .current_ifi eql 0
;  438     0550  4      then begin
;  439     0551  4           $getjpi (itmlst=lnk$gt_jpilst);
;  440     0552  4           if  .lnk$gl_filesleft [eq 3
;  441     0553  4           then lnk$closefile ();                                ! THEN CLOSE DOWN A FILE
;  442     0554  4
;  443     0555  4           open_sts =  openit ();                                !  AND TRY AGAIN
;  444     0556  4           rms_stv  = .input_fab [fab$l_stv] ;                   ! PICKUP SECONDARY STATUS
;  445     0557  4           end
;  446     0558  3      else open_sts = true ;
;  447     0559  3
;  448     0560  3      if .open_sts                                               ! IF SUCCESSFUL
;  449     0561  4      then begin
;  450     0562  4           if .lnk$gl_curfil [fdb$w_ifi] eql 0
;  451     0563  5           then  begin
;  452     0564  5                 lnk$gl_curfil [fdb$w_ifi] = .input_fab [fab$w_ifi];     ! SAVE IFI FOR LATER CLOSE
;  453     0565  5                 ch$move  (dsc$c_s_bln, lnk$filnamdsc (input_fab)        ! SAVE THE RESULTANT FILE NAME
;  454     0566  5                          ,lnk$gl_curfil [fdb$q_filename]                !  IN CASE OF ERROR LATER
;  455     0567  5                          );
;  456     0568  5                 end
;  457     0569  4           else input_fab [fab$w_ifi] = .lnk$gl_curfil [fdb$w_ifi];      !IF FILE OPEN, USE ITS IFI
;  458     0570  4
;  459     0571  4           if .lnk$al_rab [rab$l_ubf] eql 0                          ! ALLOCATE A USER BUFFER (IF NECESSARY)
;  460     0572  5           then  begin                                               ! WITH A SIZE WHICH IS
```

```
;  461     0573  5                lnk$al_rab [rab$w_usz] = obj$c_maxrecsiz;        !  MAX ALLOWABLE OBJECT RECORD
;  462     0574  5                lnk$allOblk (.lnk$al_rab [rab$w_usz]             ! FOR RMS TO USE ON RECORDS THAT
;  463     0575  5                          , lnk$al_rab [rab$l_ubf]);             ! CROSS BLOCK BOUNDARIES
;  464     0576  4                end;                                             ! AND FOR READING LIBRARY HEADERS
;  465     0577  4
;  466     0578  4            lnk$al_rab [rab$l_fab] = input_fab;                  ! SET FAB POINTER
;  467     0579  4            lnk$al_rab [rab$w_isi] = 0;                          ! ENSURE IT IS RE-USABLE RAB
;  468     0580  5            if (open_sts = $connect (rab=lnk$al_rab);            ! AND ATTEMPT TO CONNECT IT
;  469     0581  5                rms_stv  = .lnk$al_rab [rab$l_stv];
;  470     0582  5                .open_sts
;  471     0583  5               )
;  472     0584  5            then  begin                                          ! AND IF SUCCESSFUL
;  473     0585  5                  if .lnk$gl_curfil [fdb$v_shr]                  ! FINALLY IF THIS IS A SHAREABLE IMAGE
;  474     0586  5                                                                 ! TELL RMS WHAT I KNOW ABOUT THE
;  475     0587  6                  then begin                                     ! THE RECORDS OF THE FILE I.E.
;  476     0588  6                       input_fab [fab$v_esc] = true;             ! RMS $MODIFY FUNCTION WHICH WILL TELL
;  477     0589  6                       input_fab [fab$l_ctx] = rme$c_setrfm;     ! RMS THAT THIS IS A VARIABLE
;  478     0590  6                       input_fab [fab$b_rfm] = fab$c_var;        ! LENGTH RECORD FILE, EVEN THOUGH THE
;  479     0591  7                       if (open_sts = $modify (fab=input_fab);   ! THE RECORDS OF THE FILE ARE FIXED
;  480     0592  7                           rms_stv  = .input_fab [fab$l_stv];
;  481     0593  7                          .open_sts
;  482     0594  7                          )
;  483     0595  7                       then (lnk$gl_record = 0; return true)     ! AND THAT IS ALL IF IT SUCCEEDS
;  484     0596  6                       end
;  485     0597  5                  else (lnk$gl_record = 0; return true);
;  486     0598  4                end;
;  487     0599  3            end;                                                 ! IF ANY OF THE ABOVE FAILS
;  488     0600  2        end;                                                     ! FILE WAS NOT A LIBRARY
;  489     0601  2
;  490     0602  2 lnk$closefile (.lnk$gl_curfil);                                 ! ATTEMPT CLOSE
;  491     0603  2                                                                 ! ISSUE MESSAGES
;  492     0604  2 errorcode = lin$_openin;                                        ! SET ERROR CODE
;  493     0605  2
;  494     0606  2 if .lnk$gl_ctlmsk [lnk$v_intfil] or .default_lib                ! IF AN INTERNAL FILE
;  495     0607  2 then errorcode [sts$v_severity] = sts$k_info                    ! THEN NOT FATAL
;  496     0608  2 else errorcode [sts$v_severity] = sts$k_severe;                 ! OTHERWISE IT IS
;  497     0609  2
;  498     0610  2 signal (.errorcode,1                                           ! FIRST THE LINK PASS NUMBER
;  499     0611  2         ,lnk$filnamdsc (input_fab)
;  500     0612  2         ,.open_sts, .rms_stv
;  501     0613  2         );
;  502     0614  2 if .lnk$gl_curclu [clu$v_intclu]                                ! IF THIS IS INTERNALLY CREATED CLUSTER
;  503     0615  3 then begin
;  504     0616  3      lnk$gl_curclu [clu$l_shrlst]  = 0;                         ! CLEAR INFO LEFT AROUND FOR CONSISTENCY CHE
;  505     0617  3      lnk$gl_curclu [clu$l_adrcnt]  = 0;
;  506     0618  3      lnk$gl_curclu [clu$l_adrleft] = 0;
;  507     0619  2      end;
;  508     0620  2
;  509     0621  2 !
;  510     0622  2 !    Now, if we bombed out while trying to open an internally materialized
;  511     0623  2 !    shareable image, we'd better get rid of its cluster descriptor, too
;  512     0624  2 !
;  513     0625  2 if   .lnk$gl_ctlmsk [lnk$v_intfil]                              ! FILE IS INTERNALLY MATERIALIZED
;  514     0626  2 and  .lnk$gl_curfil [fdb$v_shr]                                 ! AND IS SHAREABLE
;  515     0627  2 and  .lnk$gl_curclu [clu$v_intclu]                              ! CLUSTER IS INTERNALLY MATERIALIZED
;  516     0628  2 and  .lnk$gl_curclu [clu$v_shrimg]                              ! AND IS SHRIMG CLUSTER
;  517     0629  3 then begin
```

```
;  518          0630  3      bind   nxtclu  = lnk$gl_curclu [clu$l_nxtclu]   : ref block [,byte],
;  519          0631  3             prevclu = lnk$gl_curclu [clu$l_prevclu]  : ref block [,byte] ;
;  520          0632  3
;  521          0633  3      if  .prevclu neq 0                                ! IF NOT FIRST CLUSTER,
;  522          0634  3      then prevclu [clu$l_nxtclu] = .nxtclu;            ! THEN SET PREVIOUS CLUSTER'S
;  523          0635  3                                                        ! FWD PTR TO THE NEXT CLUSTER
;  524          0636  3      if  .nxtclu  neq 0                                ! IF NOT LAST CLUSTER,
;  525          0637  3      then nxtclu [clu$l_prevclu] = .prevclu;           ! THEN SET THE NEXT CLUSTER'S
;  526          0638  2      end;                                              ! BACK PTR TO PREVIOUS CLUSTER
;  527          0639  2
;  528          0640  2 lnk$gl_curfil [fdb$v_libr] = true;                     ! UNCONDITIONALLY MAKE IT A LIBRARY SO THAT
;  529          0641  2                                                        !  IT'S DESCRIPTOR WILL BE EVAPORATED
;  530          0642  2 return lnk$nxtfil ();                                  ! RECURSING TO DISPOSE OF ITS DESCRIPTOR
;  531          0643  2
;  532          0644  1 end;                                                   ! OF LNK$NXTFIL ROUTINE


                                                        $RMS_PTR=           INPUT FAB
                                                                .EXTRN  SYS$DISCONNECT, SYS$GETJPI
                                                                .EXTRN  SYS$CONNECT, SYS$MODIFY

                                                 0FFC 00000             .ENTRY  LNK$NXTFIL, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0329
                                                                                R10,R11                                     ;
                                       5B 00000000G  00  9E 00002       MOVAB   LNK$GL_CTLMSK, R11                          ;
                                       5A      0000' CF  9E 00009       MOVAB   LNK$GL_CURFIL, R10                          ;
                                       59      0000' CF  9E 0000E       MOVAB   NEXTINTFIL, R9                              ;
                                       50            6A  D0 00013       MOVL    LNK$GL_CURFIL, R0                           ; 0372
                                                     16  12 00016       BNEQ    1$                                          ;
                                 00000000G  00 00000000G  00  D0 00018       MOVL    LNK$GL_CLULST, LNK$GL_CURCLU            ; 0374
                              F8 A9 00000000G  00         08  C1 00023       ADDL3   #8, LNK$GL_CURCLU, PREVIOUS_FDB     ; 0375
                                                     65  11 0002C       BRB     7$                                          ; 0372
                                       09        0B      A0  E8 0002E 1$:    BLBS    11(R0), 2$                          ; 0378
                                       02 00000000G      00  91 00032       CMPB    LNK$GB_PASS, #2                         ; 0379
                                                     07  12 00039       BNEQ    3$                                          ;
                                                     50  DD 0003B 2$:    PUSHL   R0                                      ; 0380
                                    0000V CF             01  FB 0003D       CALLS   #1, LNK$CLOSEFILE                       ;
                                       01 00000000G      00  91 00042 3$:    CMPB    LNK$GB_PASS, #1                         ; 0382
                                                     07  12 00049       BNEQ    4$                                          ;
                                       50            6A  D0 0004B       MOVL    LNK$GL_CURFIL, R0                           ; 0383
                                    0B   A0               02  88 0004E       BISB2   #2, 11(R0)                              ;
                                       50            6A  D0 00052 4$:    MOVL    LNK$GL_CURFIL, R0                           ; 0385
                                 23    0A   A0             01  E1 00055       BBC     #1, 10(R0), 5$                          ;
                                                 04      A0  D5 0005A       TSTL    4(R0)                                   ; 0386
                                                     1E  12 0005D       BNEQ    5$                                          ;
                                                     50  DD 0005F       PUSHL   R0                                      ; 0388
                                    0000V CF             01  FB 00061       CALLS   #1, LNK$CLOSEFILE                       ;
                                       51        F8      A9  D0 00066       MOVL    PREVIOUS_FDB, R1                        ; 0389
                                       61        00      BA  D0 0006A       MOVL    @LNK$GL_CURFIL, (R1)                    ;
                                                     11  12 0006E       BNEQ    6$                                          ; 0391
                                       50 00000000G      00  D0 00070       MOVL    LNK$GL_CURCLU, R0                       ; 0392
                                    0C   A0               51  D0 00077       MOVL    R1, 12(R0)                              ;
                                                     04  11 0007B       BRB     6$                                          ; 0385
                                    F8   A9               50  D0 0007D 5$:    MOVL    R0, PREVIOUS_FDB                        ; 0398
                                       50            6A  D0 00081 6$:    MOVL    LNK$GL_CURFIL, R0                       ; 0400
                                                 24      A0  B5 00084       TSTW    36(R0)                                  ;
                                                     0A  13 00087       BEQL    7$                                          ;
```

```
                                B8   AA   9F 00089          PUSHAB  LNK$AL_RAB                                      ; 0401
    00000000G                   00        01 FB 0008C          CALLS   #1, SYS$DISCONNECT                          ;
                                6A        F8 B9 D0 00093 7$:   MOVL    @PREVIOUS_FDB, LNK$GL_CURFIL                ; 0404
                                          14 13 00097          BEQL    10$                                         ;
                                50             6A D0 00099    MOVL    LNK$GL_CURFIL, R0                           ; 0405
03          0B                  A0             01 E0 0009C    BBS     #1, 11(R0), 9$                              ;
                                          0150 31 000A1 8$:   BRW     27$                                         ;
                                01 00000000G   00 91 000A4 9$:   CMPB    LNK$GB_PASS, #1                             ;
                                          F4   12 000AB          BNEQ    8$                                          ;
                                50 00000000G   00 D0 000AD 10$:  MOVL    LNK$GL_CURCLU, R0                           ; 0406
                                          60   D5 000B4          TSTL    (R0)                                        ;
                                          28   12 000B6          BNEQ    12$                                         ;
                                01 00000000G   00 91 000B8      CMPB    LNK$GB_PASS, #1                             ; 0407
                                          1F   12 000BF          BNEQ    12$                                         ;
                                          69   95 000C1          TSTB    NEXTINTFIL                                  ;
                                          1B   13 000C3          BEQL    12$                                         ;
                                   00000000G   00 B5 000C5      TSTW    LNK$GW_NUDFSYMS                             ; 0409
                                          1?   13 000CB          BEQL    12$                                         ;
                                03             69 91 000CD      CMPB    NEXTINTFIL, #3                              ; 0410
                                               05 12 000D0      BNEQ    11$                                         ;
0C          02                  AB             06 E0 000D2      BBS     #6, LNK$GL_CTLMSK+2, 13$                    ;
                                               69 95 000D7 11$: TSTB    NEXTINTFIL                                  ; 0412
                                               05 13 000D9      BEQL    12$                                         ;
03          01                  AB             02 E0 000DB      BBS     #2, LNK$GL_CTLMSK+1, 13$                    ;
                                          00F0 31 000E0 12$:   BRW     24$                                         ;
                                52        82   8F 9A 000E3 13$:   MOVZBL  #130, FILEFLAGS                          ; 0417
                                04        08   A9 E8 000E7      BLBS    FIRST_TIME, 14$                             ; 0418
                                2D        09   A9 E9 000EB      BLBC    DEFAULT_LIB, 17$                            ; 0419
                                          08   A9 94 000EF 14$:   CLRB    FIRST_TIME                                  ; 0421
08          02                  AB             06 E1 000F2      BBC     #6, LNK$GL_CTLMSK+2, 15$                    ; 0422
          0000V                 CF             00 FB 000F7      CALLS   #0, TRAN_NEXT_LIB                           ; 0423
                                13             50 E8 000FC      BLBS    R0, 16$                                     ;
                                69             02 90 000FF 15$:   MOVB    #2, NEXTINTFIL                              ; 0425
                                          09   A9 94 00102      CLRB    DEFAULT_LIB                                 ; 0426
                                02             40 8F 8A 00105   BICB2   #64, LNK$GL_CTLMSK+2                        ; 0427
0D          01                  AB             02 E0 0010A      BBS     #2, LNK$GL_CTLMSK+1, 17$                    ; 0428
                                          00D5 31 0010F          BRW     25$                                         ; 0430
                09              A9             01 90 00112 16$:   MOVB    #1, DEFAULT_LIB                             ; 0435
                                          04   A9 D6 00116      INCL    USERLIBNO                                   ; 0436
                                69             03 90 00119      MOVB    #3, NEXTINTFIL                              ; 0437
                                02             69 91 0011C 17$:   CMPB    NEXTINTFIL, #2                              ; 0441
                                               08 12 0011F      BNEQ    18$                                         ;
03          02                  AB             04 E0 00121      BBS     #4, LNK$GL_CTLMSK+2, 18$                    ; 0442
                                69             01 90 00126      MOVB    #1, NEXTINTFIL                              ; 0443
                                05        09   A9 E8 00129 18$:   BLBS    DEFAULT_LIB, 19$                            ; 0445
                                01             69 91 0012D      CMPB    NEXTINTFIL, #1                              ;
                                               44 12 00130      BNEQ    20$                                         ;
    000000000G                  00 00000000G   00 9E 00132 19$:   MOVAB   LNK$GL_DEFCLU, LNK$GL_CURCLU                ; 0447
                                   00000000G   00 D5 00136      TSTL    LNK$GL_DEFCLU+8                             ; 0448
                                               31 12 00143      BNEQ    20$                                         ;
                                   00000000G   00 D5 00145      TSTL    LNK$GL_DEFCLU                               ; 0449
                                               29 12 0014B      BNEQ    20$                                         ;
                                50 00000000G   00 9E 0014D      MOVAB   LNK$GL_DEFCLU, R0                           ; 0450
                                50 00000000G   00 D1 00154      CMPL    LNK$GL_LASTCLU, R0                          ;
                                               19 13 0015B      BEQL    20$                                         ;
                                50 00000000G   00 D0 0015D      MOVL    LNK$GL_LASTCLU, R0                          ; 0452
                                60 00000000G   00 9E 00164      MOVAB   LNK$GL_DEFCLU, (R0)                         ;
```

```
                  00000000G  00 00000000G  00  9E 0016B       MOVAB   LNK$GL_DEFCLU, LNK$GL_LASTCLU          ; 0453
                             50 00000000G  00  D0 00176 20$:  MOVL    LNK$GL_CURCLU, R0                      ; 0457
                         F8  A9        0C  A0  D0 0017D       MOVL    12(R0), PREVIOUS_FDB                   ;
                                       F8  A9  DD 00182       PUSHL   PREVIOUS_FDB                           ; 0458
                  00000000G  00            01  FB 00185       CALLS   #1, LNK$ALLOFDB                        ;
                             6A        F8  B9  D0 0018C       MOVL    @PREVIOUS_FDB, LNK$GL_CURFIL           ; 0459
                             50 00000000G  00  D0 00190       MOVL    LNK$GL_CURCLU, R0                      ; 0460
                             56            6A  D0 00197       MOVL    LNK$GL_CURFIL, R6                      ;
                         0C  A0            56  D0 0019A       MOVL    R6, 12(R0)                             ;
                         0A  A6            52  90 0019E       MOVB    FILEFLAGS, 10(R6)                      ; 0461
                             0F        09  A9  E9 001A2       BLBC    DEFAULT_LIB, 21$                       ; 0463
          0C     A6    0110  C9            08  28 001A6       MOVC3   #8, LIBNAMDESCR, 12(R6)                ; 0465
          14     A6      E0  A9            08  28 001AD       MOVC3   #8, LIBDEFEXT, 20(R6)                  ; 0466
                                           12  11 001B3       BRB     22$                                    ; 0463
                             50            69  9A 001B5 21$:  MOVZBL  NEXTINTFIL, R0                         ; 0469
                                   B8 A940     7F 001B8       PUSHAQ  INTFILTBL-8[R0]                        ;
          0C     A6      9E                08  28 001BC       MOVC3   #8, @(SP)+, 12(R6)                     ;
          14     A6      D0  A9            08  28 001C1       MOVC3   #8, SYSLIBDEFEXT, 20(R6)               ; 0470
                             04        09  A9  E8 001C7 22$:  BLBS    DEFAULT_LIB, 23$                       ; 0473
                         01  AB            08  88 001CB       BISB2   #8, LNK$GL_CTLMSK+1                    ; 0474
                                           69  97 001CF 23$:  DECB    NEXTINTFIL                             ; 0475
                                           21  11 001D1       BRB     27$                                    ; 0414
                  00000000G  00            60  D0 001D3 24$:  MOVL    (R0), LNK$GL_CURCLU                    ; 0479
                             50 00000000G  00  D0 001DA       MOVL    LNK$GL_CURCLU, R0                      ; 0480
                                           09  12 001E1       BNEQ    26$                                    ;
                         01  AB            08  8A 001E3       BICB2   #8, LNK$GL_CTLMSK+1                    ; 0482
                                           6A  D4 001E7 25$:  CLRL    LNK$GL_CURFIL                          ; 0483
                                         0242  31 001E9       BRW     54$                                    ; 0484
                         F8  A9        08  A0  9E 001EC 26$:  MOVAB   8(R0), PREVIOUS_FDB                    ; 0486
                                         FE9F  31 001F1       BRW     7$                                     ; 0406
                             57            6A  D0 001F4 27$:  MOVL    LNK$GL_CURFIL, R7                      ; 0489
                             56        26  A7  9E 001F7       MOVAB   38(R7), AUXFNB                         ;
                                           50  D4 001FB       CLRL    R0                                     ; 0490
                             02 00000000G  00  91 001FD       CMPB    LNK$GB_PASS, #2                        ;
                                           02  12 00204       BNEQ    28$                                    ;
                                           50  D6 00206       INCL    R0                                     ;
      51          0B     A7        01      02  EF 00208 28$:  EXTZV   #2, #1, 11(R7), R1                     ;
                                   51      51  D2 0020E       MCOML   R1, R1                                 ;
                         58        50      51  8B 00211       BICB3   R1, R0, TURN_SQO_OFF                   ;
    0050  8F             00        6E      00  2C 00215       MOVC5   #0, (SP), #0, #80, $RMS_PTR            ; 0497
                                   FF70    C9     0021C                                                      ;
                       FF70  C9    5003    8F  B0 0021F       MOVW    #20483, $RMS_PTR                       ;
                       FF74  C9      40    8F  9A 00226       MOVZBL  #64, $RMS_PTR+4                        ;
                         86  A9            02  90 0022C       MOVB    #2, $RMS_PTR+22                        ;
                         8F  A9            02  90 00230       MOVB    #2, $RMS_PTR+31                        ;
                         98  A9            56  D0 00234       MOVL    AUXFNB, $RMS_PTR+40                    ;
                         A6  A9    0800    8F  B0 00238       MOVW    #2048, $RMS_PTR+54                     ;
                             0D            58  E9 0023E       BLBC    TURN_SQO_OFF, 29$                      ; 0499
                                           57  DD 00241       PUSHL   R7                                     ; 0501
                      0000V  CF            01  FB 00243       CALLS   #1, LNK$CLOSEFILE                      ;
                       FF74  C9      40    8F  8A 00248       BICB2   #64, INPUT_FAB+4                       ; 0502
                             50            6A  D0 0024E 29$:  MOVL    LNK$GL_CURFIL, R0                      ; 0505
                             52      24    A0  3C 00251       MOVZWL  36(R0), CURRENT_IFI                    ;
                                     24    A6  B5 00255       TSTW    36(AUXFNB)                             ; 0507
                                           16  12 00258       BNEQ    30$                                    ;
                         A4  A9      0C    A0  90 0025A       MOVB    12(R0), INPUT_FAB+52                   ; 0509
                         9C  A9      10    A0  D0 0025F       MOVL    16(R0), INPUT_FAB+44                   ; 0510
```

```
                A5   A9        14   A0  90 00264          MOVB    20(R0), INPUT_FAB+53          ; 0511
                A0   A9        18   A0  D0 00269          MOVL    24(R0), INPUT_FAB+48          ; 0512
                                    05  11 0026E          BRB     31$                           ; 0507
              FF77   C9              01  88 00270 30$:     BISB2   #1, INPUT_FAB+7               ; 0514
50              0A   A0              01  E1 00275 31$:     BBC     #1, 10(R0), 38$               ; 0516
                                    52  D5 0027A          TSTL    CURRENT_IFI                   ; 0518
                                    0C  12 0027C          BNEQ    32$
                                    56  DD 0027E          PUSHL   AUXFNB                        ; 0519
              0000V  CF             01  FB 00280          CALLS   #1, LNK$OPENLIB
                     56             50  D0 00285          MOVL    R0, OPEN_STS
                                    03  11 00288          BRB     33$
                     56             01  D0 0028A 32$:     MOVL    #1, OPEN_STS                  ; 0518
                     30             56  E9 0028D 33$:     BLBC    OPEN_STS, 36$
03              01   AB              03  E0 00290          BBS     #3, LNK$GL_CTLMSK+1, 35$      ; 0523
                                 0109  31 00295 34$:     BRW     46$
                     01             69  91 00298 35$:     CMPB    NEXTINTFIL, #1                ; 0524
                                    F8  12 0029B          BNEQ    34$
                     50             6A  D0 0029D          MOVL    LNK$GL_CURFIL, R0             ; 0525
                     F1        0B   A0  E8 002A0          BLBS    11(R0), 34$
                               14   A0  9F 002A4          PUSHAB  20(R0)                        ; 0528
                                    01  DD 002A7          PUSHL   #1
                          00000000G  8F  DD 002A9          PUSHL   #LIN$_NOTIMGLIB
    00000000G   00                  03  FB 002AF          CALLS   #3, LIB$SIGNAL
                                    6A  DD 002B6          PUSHL   LNK$GL_CURFIL                 ; 0530
              0000V  CF             01  FB 002B8          CALLS   #1, LNR$CLOSEFILE
                                 0161  31 002BD          BRW     53$                           ; 0531
                     57 00000000G   00  D0 002C0 36$:     MOVL    LBR$GL_RMSSTV, RMS_STV        ; 0537
                                 00DE  31 002C7 37$:     BRW     47$                           ; 0516
0C              0A   A0              02  E1 002CA 38$:     BBC     #2, 10(R0), 39$               ; 0540
              FF74   C9        40    8F  8A 002CF          BICB2   #64, INPUT_FAB+4              ; 0542
                86   A9      4340    8F  A8 002D5          BISW2   #17216, INPUT_FAB+22          ; 0546
                                    52  D5 002DB 39$:     TSTL    CURRENT_IFI                   ; 0549
                                    32  12 002DD          BNEQ    41$
                                    7E  7C 002DF          CLRQ    -(SP)                         ; 0551
                                    7E  D4 002E1          CLRL    -(SP)
                          00000000G   00  9F 002E3          PUSHAB  LNK$GT_JPILST
                                    7E  7C 002E9          CLRQ    -(SP)
                                    7E  D4 002EB          CLRL    -(SP)
    00000000G   00                  07  FB 002ED          CALLS   #7, SYS$GETJPI
                     03 00000000G   00  D1 002F4          CMPL    LNK$GL_FILESLEFT, #3          ; 0552
                                    05  14 002FB          BGTR    40$
              0000V  CF             00  FB 002FD          CALLS   #0, LNK$CLOSEFILE             ; 0553
              FCB9   CF             00  FB 00302 40$:     CALLS   #0, OPENIT                    ; 0555
                     56             50  D0 00307          MOVL    R0, OPEN_STS
                     57        FF7C C9  D0 0030A          MOVL    INPUT_FAB+12, RMS_STV         ; 0556
                                    03  11 0030F          BRB     42$                           ; 0549
                     56             01  D0 00311 41$:     MOVL    #1, OPEN_STS                  ; 0558
                     B0             56  E9 00314 42$:     BLBC    OPEN_STS, 37$                 ; 0560
                     50             6A  D0 00317          MOVL    LNK$GL_CURFIL, R0             ; 0562
                               24   A0  B5 0031A          TSTW    36(R0)
                                    1B  12 0031D          BNEQ    43$
                     24   A0   FF72  C9  B0 0031F          MOVW    INPUT_FAB+2, 36(R0)           ; 0564
                               FF70  C9  9F 00325          PUSHAB  INPUT_FAB                     ; 0565
    00000000G   00                  01  FB 00329          CALLS   #1, LNK$FILNAMDSC
                     51             6A  D0 00330          MOVL    LNK$GL_CURFIL, R1             ; 0566
14  A1               60             08  28 00333          MOVC3   #8, (R0), 20(R1)
                                    06  11 00338          BRB     44$                           ; 0562
```

```
                        FF72   C9      24   A0   B0 00033A 43$:    MOVW    36(R0), INPUT_FAB+2                    ; 0569
                                       DC   AA   D5 00340  44$:    TSTL    LNK$AL_RAB+36                          ; 0571
                                       14   12   00343            BNEQ    45$
                          D8   AA    0800   8F   B0 00345         MOVW    #2048, LNK$AL_RAB+32                   ; 0573
                                       DC   AA   9F 0034B         PUSHAB  LNK$AL_RAB+36                          ; 0575
                               7E      D8   AA   3C 0034E         MOVZWL  LNK$AL_RAB+32, -(SP)                   ; 0574
                00000000G      00           02   FB 00352         CALLS   #2, LNR$ALLOBLK
                          F4   AA    FF70   C9   9E 00359 45$:    MOVAB   INPUT_FAB, LNK$AL_RAB+60               ; 0578
                                       BA   AA   B4 0035F         CLRW    LNK$AL_RAB+2                           ; 0579
                                       B8   AA   9F 00362         PUSHAB  LNK$AL_RAB                             ; 0580
                00000000G      00           01   FB 00365         CALLS   #1, SYS$CONNECT
                               56           50   D0 0036C         MOVL    R0, OPEN_STS
                               57      C4   AA   D0 0036F         MOVL    LNK$AL_RAB+12, RMS_STV                 ; 0581
                               32           56   E9 00373         BLBC    OPEN_STS, 47$                          ; 0582
                               50           6A   D0 00376         MOVL    LNK$GL_CURFIL, R0                      ; 0585
                23        0A   A0           02   E1 00379         BBC     #2, 10(R0), 46$
                          FF77   C9           08   88 0037E         BISB2   #8, INPUT_FAB+7                        ; 0588
                            88   A9           01   D0 00383         MOVL    #1, INPUT_FAB+24                       ; 0589
                            8F   A9           02   90 00387         MOVB    #2, INPUT_FAB+31                       ; 0590
                                       FF70   C9   9F 0038B         PUSHAB  INPUT_FAB                              ; 0591
                00000000G      00           01   FB 0038F         CALLS   #1, SYS$MODIFY
                               56           50   D0 00396         MOVL    R0, OPEN_STS
                               57      FF7C C9   D0 00399         MOVL    INPUT_FAB+12, RMS_STV                  ; 0592
                               07           56   E9 0039E         BLBC    OPEN_STS, 47$                          ; 0593
                                       B4   AA   D4 003A1 46$:    CLRL    LNK$GL_RECORD                          ; 0597
                               50           01   D0 003A4         MOVL    #1, R0
                                            04   003A7            RET
                                            6A   DD 003A8 47$:    PUSHL   LNK$GL_CURFIL                          ; 0602
                          0000V CF           01   FB 003AA         CALLS   #1, LNR$CLOSEFILE
                               52 00000000G  8F   D0 003AF         MOVL    #LIN$_OPENIN, ERRORCODE                ; 0604
                23        01   AB           03   E0 003B6         BBS     #3, LNK$GL_CTLMSK+1, 48$               ; 0606
                               07      09   A9   E9 003BB         BLBC    DEFAULT_LIB, 49$
          52              03   00           03   F0 003BF 48$:    INSV    #3, #0, #3, ERRORCODE                  ; 0607
                                            05   11 003C4         BRB     50$
          52              03   00           04   F0 003C6 49$:    INSV    #4, #0, #3, ERRORCODE                  ; 0608
                               7E           56   7D 003CB 50$:    MOVQ    OPEN_STS, -(SP)                        ; 0612
                                       FF70   C9   9F 003CE         PUSHAB  INPUT_FAB                              ; 0611
                00000000G      00           01   FB 003D2         CALLS   #1, LNK$FILNAMDSC
                                            50   DD 003D9         PUSHL   R0
                                            01   DD 003DB         PUSHL   #1                                     ; 0610
                                            52   DD 003DD         PUSHL   ERRORCODE
                00000000G      00           05   FB 003DF         CALLS   #5, LIB$SIGNAL
                               50 00000000G  00   D0 003E6         MOVL    LNK$GL_CURCLU, R0                      ; 0614
                06        59   A0           01   E1 003ED         BBC     #1, 89(R0), 51$
                                       30   A0   7C 003F2         CLRQ    48(R0)                                 ; 0616
                                       38   A0   D4 003F5         CLRL    56(R0)                                 ; 0618
                24        01   AB           03   E1 003F8 51$:    BBC     #3, LNK$GL_CTLMSK+1, 53$               ; 0625
                               51           6A   D0 003FD         MOVL    LNK$GL_CURFIL, R1                      ; 0626
                1C        0A   A1           02   E1 00400         BBC     #2, 10(R1), 53$
                17        59   A0           01   E1 00405         BBC     #1, 89(R0), 53$                        ; 0627
                12        58   A0           02   E1 0040A         BBC     #2, 88(R0), 53$                        ; 0628
                               51      04   A0   D0 0040F         MOVL    4(R0), R1                              ; 0633
                                            03   13 00413         BEQL    52$
                               61           60   D0 00415         MOVL    (R0), (R1)                             ; 0634
                               50           60   D0 00418 52$:    MOVL    (R0), R0                               ; 0636
                                            04   13 0041B         BEQL    53$
                                   04       51   D0 0041D         MOVL    R1, 4(R0)                              ; 0637
```

```
                    50              6A  D0 00421 53$:   MOVL    LNK$GL_CURFIL, R0                       ; 0640
                  0A    A0          02  88 00424         BISB2   #2, 10(R0)                              ;
                FBD3    CF          00  FB 00428         CALLS   #0, LNK$NXTFIL                          ; 0642
                                        04 0042D         RET                                             ;
                                    50  D4 0042E 54$:     CLRL    R0                                      ; 0644
                                        04 00430         RET                                             ;

; Routine Size:  1073 bytes,    Routine Base:  $CODE$ + 0040
```

```
;  534          0645  1 global routine lnk$nxtrec (recdesc) =
;  535          0646  1 !
;  536          0647  1 !       READ NEXT RECORD FROM CURRENTLY OPEN FILE.
;  537          0648  1 !       THIS ROUTINE ACQUIRES RECORDS SEQUENTIALLY
;  538          0649  1 !       UNTIL END OF FILE IS DETECTED, IN WHICH
;  539          0650  1 !       CASE IT HAS THE VALUE FALSE. SUCCESSFUL READS RETURN
;  540          0651  1 !       THE ADDRESS AND LENGTH OF NEXT RECORD AND THE ROUTINE
;  541          0652  1 !       HAS THE VALUE TRUE
;  542          0653  1 !
;  543          0654  2 begin
;  544          0655  2 map
;  545          0656  2         recdesc         : ref block [,byte];
;  546          0657  2 local
;  547          0658  2         stvcode,
;  548          0659  2         readerror,                                      ! READ ERROR CODE
;  549          0660  2         bufdesc         : block [dsc$c_s_bln, byte];    ! DESCRIPTOR FOR BUFFER
;  550          0661  2
;  551          0662  2 bind auxfnb = lnk$gl_curfil [fdb$t_auxfnb]              ! REFERENCE THE AUX. FILE NAME BLOCK PART
;  552          0663  2                 : block [nam$c_bln, byte];
;  553          0664  2 !
;  554          0665  2 if .lnk$gl_curfil [fdb$v_libr]                         ! IF READING A LIBRARY FILE
;  555          0666  3 then begin
;  556          0667  3      bufdesc [dsc$w_length]  = .lnk$al_rab [rab$w_usz]; ! SET UP DESCRIPTOR FOR BUFFER
;  557          0668  3      bufdesc [dsc$a_pointer] = .lnk$al_rab [rab$l_ubf];
;  558          0669  3
;  559          0670  3      readerror = lbr$get_record ( %ref(.lnk$gl_curfil [fdb$w_ifi]), bufdesc, bufdesc);  ! READ A RECORD
;  560          0671  3
;  561          0672  3      lnk$al_rab [rab$l_stv]  = .lbr$gl_rmsstv;
;  562          0673  3      lnk$al_rab [rab$w_rsz]  = .bufdesc [dsc$w_length];  ! SET INFO INTO RAB
;  563          0674  3      lnk$al_rab [rab$l_rbf]  = .bufdesc [dsc$a_pointer]; !...
;  564          0675  3      end
;  565          0676  2 else readerror = $get (rab=lnk$al_rab);                 ! NOT LIBRARY, JUST DO A GET
;  566          0677  2
;  567          0678  2 if   not .readerror                                     ! IF THERE WAS AN ERROR OTHER THAN
;  568          0679  2 then if  .readerror neq rms$_eof                        ! AND IF AN ERROR OTHER THAN
;  569          0680  3      then begin                                         ! END OF FILE
;  570          0681  3           stvcode = .lnk$al_rab [rab$l_stv];            ! SAVE STV OVER CLOSE
;  571          0682  3           lnk$closefile (.lnk$gl_curfil);               ! ATTEMPT TO CLOSE IT
;  572          0683  3           signal ( lin$_readerr, 1                      ! ISSUE READ ERROR MESSAGE
;  573          0684  3                  , lnk$gl_curfil [fdb$q_filename]
;  574          0685  3                  ,.readerror, .stvcode
;  575          0686  3                  );
;  576          0687  3           return false;
;  577          0688  3           end
;  578          0689  2      else return false;                                 ! ALL DONE IF END OF FILE
;  579          0690  2
;  580          0691  2 lnk$gl_objrecs = .lnk$gl_objrecs + 1;                   !
;  581          0692  2 lnk$gl_record  = .lnk$gl_record + 1;
;  582          0693  2
;  583          0694  2 recdesc [dsc$w_length]  = .lnk$al_rab [rab$w_rsz];      ! RETURN LENGTH AND ADDRESS OF THIS
;  584          0695  2 recdesc [dsc$a_pointer] = .lnk$al_rab [rab$l_rbf];      ! SUCCESSFULLY OBTAINED RECORD
;  585          0696  2
;  586          0697  2 return true                                             ! AND ALL IS DONE
;  587          0698  1 end;                                                    ! OF LNK$NXTREC ROUTINE
```

```
                                                        .EXTRN  SYS$GET

                                                        .ENTRY  LNK$NXTREC, Save R2,R3,R4                       ; 0645
                        54      0000'   CF   9E 00002  MOVAB   LNK$GL_CURFIL, R4                               ;
                        5E              0C   C2 00007  SUBL2   #12, SP                                         ;
                        50              64   D0 0000A  MOVL    LNK$GL_CURFIL, R0                               ; 0662
36              0A      A0              01   E1 0000D  BBC     #1, 10(R0), 1$                                  ; 0665
                04      AE      D8      A4   B0 00012  MOVW    LNK$AL_RAB+32, BUFDESC                          ; 0667
                08      AE      DC      A4   D0 00017  MOVL    LNK$AL_RAB+36, BUFDESC+4                        ; 0668
                                04      AE   9F 0001C  PUSHAB  BUFDESC                                         ; 0670
                                08      AE   9F 0001F  PUSHAB  BUFDESC                                         ;
                        08      AE      24   A0 3C 00022  MOVZWL  36(R0), 8(SP)                                ;
                                        08   AE 9F 00027  PUSHAB  8(SP)                                         ;
    00000000G           00              03   FB 0002A  CALLS   #3, LBR$GET_RECORD                              ;
                        53              50   D0 00031  MOVL    R0, READERROR                                   ;
                C4      A4  00000000G   00   D0 00034  MOVL    LBR$GL_RMSSTV, LNK$AL_RAB+12                    ; 0672
                DA      A4      04      AE   B0 0003C  MOVW    BUFDESC, LNK$AL_RAB+34                          ; 0673
                E0      A4      08      AE   D0 00041  MOVL    BUFDESC+4, LNK$AL_RAB+40                        ; 0674
                                        0D   11 00046  BRB     2$                                              ; 0665
                                B8      A4   9F 00048  1$:     PUSHAB  LNK$AL_RAB                              ; 0676
    00000000G           00              01   FB 0004B  CALLS   #1, SYS$GET                                     ;
                        53              50   D0 00052  MOVL    R0, READERROR                                   ;
                        2D              53   E8 00055  2$:     BLBS    READERROR, 3$                           ; 0678
    0001827A            8F              53   D1 00058  CMPL    READERROR, #98938                               ; 0679
                                        3B   13 0005F  BEQL    4$                                              ;
                        52      C4      A4   D0 00061  MOVL    LNK$AL_RAB+12, STVCODE                          ; 0681
                                        64   DD 00065  PUSHL   LNK$GL_CURFIL                                   ; 0682
            0000V       CF              01   FB 00067  CALLS   #1, LNK$CLOSEFILE                               ;
                                        52   DD 0006C  PUSHL   STVCODE                                         ; 0685
                                        53   DD 0006E  PUSHL   READERROR                                       ;
7E                      64              14   C1 00070  ADDL3   #20, LNK$GL_CURFIL, -(SP)                     ; 0684
                                        01   DD 00074  PUSHL   #1                                              ;
                    00000000G           8F   DD 00076  PUSHL   #LIN$_READERR                                  ;
    00000000G           00              05   FB 0007C  CALLS   #5, LIB$SIGNAL                                  ;
                                        17   11 00083  BRB     4$                                              ; 0689
                                B0      A4   D6 00085  3$:     INCL    LNK$GL_OBJRECS                          ; 0691
                                B4      A4   D6 00088          INCL    LNK$GL_RECORD                           ; 0692
                        50      04      AC   D0 0008B          MOVL    RECDESC, R0                             ; 0694
                        60      DA      A4   B0 0008F          MOVW    LNK$AL_RAB+34, (R0)                     ;
            04          A0      E0      A4   D0 00093          MOVL    LNK$AL_RAB+40, 4(R0)                    ; 0695
                        50              01   D0 00098          MOVL    #1, R0                                  ; 0697
                                        04   0009B          RET                                             ;
                                        50   D4 0009C  4$:     CLRL    R0                                      ; 0698
                                        04   0009E          RET                                             ;

; Routine Size:  159 bytes,    Routine Base:  $CODE$ + 0471
```

```
;  589          0699  1 global routine lnk$pointobj(modrfa) : novalue =
;  590          0700  2 begin
;  591          0701  2 !
;  592          0702  2 !++
;  593          0703  2 !
;  594          0704  2 !       THIS ROUTINE IS CALLED DURING LIBRARY OR SHAREABLE IMAGE FILE READING TO POINT TO
;  595          0705  2 !       THE REQUIRED PARTITION IN THE FILE GIVEN BY ITS RFA.
;  596          0706  2 !       FOR A LIBRARY THE NEXT GET WILL READ THE LIBRARY MODULE'S HEADER RECORD.
;  597          0707  2 !       FOR A SHAREABLE IMAGE FILE MERELY DO A FIND BY RFA SO THAT NO RECORD IS
;  598          0708  2 !       SKIPPED. I.E. NEXT GET GETS RECORD POINTED TO.
;  599          0709  2 !
;  600          0710  2 map
;  601          0711  2         modrfa          : ref block [,byte];                    ! POINTER TO RFA BLOCK
;  602          0712  2 local
;  603          0713  2         status,
;  604          0714  2         hdrdesc         : block [dsc$c_s_bln,byte],             ! STRING DESCRIPTOR FOR MODULE HEADER
;  605          0715  2         libmodhdr       : ref block [,byte],                    ! POINTER TO LIBRARY MODULE HEADER
;  606          0716  2         reclng;                                                 ! LENGTH OF RECORD
;  607          0717  2
;  608          0718  2 bind auxfnb = lnk$gl_curfil [fdb$t_auxfnb]                       ! REFERENCE THE AUXILIARY FILE NAME BLOCK PA
;  609          0719  2                         : block[nam$c_bln,byte];
;  610          0720  2
;  611          0721  2 if .lnk$gl_curfil [fdb$v_libr]                                  ! IF THIS IS A LIBRARY FILE
;  612          0722  3 then begin
;  613          0723  3      local       modnamdesc  : block[dsc$c_s_bln,byte],
;  614          0724  3                  header      : block[lbr$c_maxhdrsiz,byte];
;  615          0725  3
;  616          0726  3      if .lnk$gb_pass eql 1                                      ! ONLY READ HEADER ON PASS 1
;  617          0727  4      then begin
;  618          0728  4           hdrdesc [dsc$w_length]  = lbr$c_maxhdrsiz;            ! SET UP DESCRIPTOR FOR HEADER BUFFER
;  619          0729  4           hdrdesc [dsc$a_pointer] = header;
;  620          0730  5           if not (status = lbr$set_module (%ref (.lnk$gl_curfil [fdb$w_ifi]), .modrfa, hdrdesc))
;  621          0731  5           then  begin
;  622          0732  5                 lnk$al_rab [rab$l_sts] = .status;             ! ERROR - SET CODES INTO RAB
;  623          0733  5                 lnk$al_rab [rab$l_stv] = .lbr$gl_rmsstv;
;  624          0734  5                 end
;  625          0735  5           else  begin
;  626          0736  5                 if .header [mhd$v_selsrc]                     ! IF MODULE IS SELECTIVELY SEARCHED
;  627          0737  5                 then lnk$gl_curfil [fdb$v_selser] = true
;  628          0738  5                 else lnk$gl_curfil [fdb$v_selser] = false;
;  629          0739  5                 return true;
;  630          0740  4                 end;
;  631          0741  4           end
;  632          0742  4      else begin                                                 ! PASS 2 -- NEED TO FIND MODULE
;  633          0743  4           status = lbr$find (%ref (.lnk$gl_curfil[fdb$w_ifi]), .modrfa);! POSITION TO READ MODULE
;  634          0744  4           if .status then return true;                          ! ALL DONE IF SUCCESSFUL
;  635          0745  4
;  636          0746  4           modnamdesc [dsc$w_length]  = .lnk$gl_curomd [omd$b_namlng];   ! IT APPEARS TO HAVE DISAPPEARED.
;  637          0747  4           modnamdesc [dsc$a_pointer] =  lnk$gl_curomd [omd$t_name];     !  TRY A LOOKUP IN CASE IT WAS MEREL
;  638          0748  4           status = lbr$lookup_key (%ref (.lnk$gl_curfil [fdb$w_ifi]), modnamdesc, .modrfa) ;
;  639          0749  4           if .status  then return true;
;  640          0750  4
;  641          0751  4           lnk$al_rab [rab$l_sts] = .status;                     ! ERROR--SET CODES IN RAB
;  642          0752  4           lnk$al_rab [rab$l_stv] = .lbr$gl_rmsstv;
;  643          0753  3           end;
;  644          0754  3      end
;  645          0755  3 else begin                                                     ! FILE IS NOT A LIBRARY
```

```
;  646     0756  3      lnk$al_rab [rab$b_rac]  =  rab$c_rfa;                    ! SET TO ACCESS BY RFA
;  647     0757  3      lnk$al_rab [rab$l_rfa0] = .modrfa [rfa$l_vbn];          ! SET UP RFA
;  648     0758  3      lnk$al_rab [rab$w_rfa4] = .modrfa [rfa$w_offset];
;  649     0759  3      status = $find (rab=lnk$al_rab);                        ! PERFORM THE FIND
;  650     0760  3      lnk$al_rab [rab$b_rac]  = rab$c_seq;                    ! RESET TO SEQUENTIAL ACCESS
;  651     0761  3      if .status then return true;                            ! ALL DONE IF GOOD FIND
;  652     0762  2      end;
;  653     0763  2
;  654     0764  2 signal  ( lin$_libfind, 4, .modrfa [rfa$l_vbn]               ! REPORT ERROR IF FALL THROUGH
;  655     0765  2         ,.modrfa [rfa$w_offset], lnk$gl_curomd [omd$b_namlng]
;  656     0766  2         , lnk$gl_curfil [fdb$g_filename], lin$_format, 0
;  657     0767  2         ,.status,.lnk$al_rab [rab$l_stv]
;  658     0768  2         );
;  659     0769  2 return;                                                       ! AND ALL SET
;  660     0770  1 end;


                                                        .EXTRN  SYS$FIND

                                                000C 00000      .ENTRY  LNK$POINTOBJ, Save R2,R3                ; 0699
                                  53    0000'   CF   9E 00002  MOVAB   LNK$GL_CURFIL, R3                       ;
                                  5E    FF6C    CE   9E 00007  MOVAB   -148(SP), SP                            ;
                                  51            63   D0 0000C  MOVL    LNK$GL_CURFIL, R1                       ; 0718
                                  52      04    AC   D0 0000F  MOVL    MODRFA, R2                              ; 0730
                           03       0A    A1        01   E0 00013  BBS     #1, 10(R1), 1$                          ; 0721
                                                0085   31 00018  BRW     5$
                                  01 00000000G  00   91 0001B 1$:  CMPB    LNK$GB_PASS, #1                         ; 0726
                                                32   12 00022  BNEQ    3$
                                 F8    AD       80   8F 9B 00024  MOVZBW  #128, HDRDESC                         ; 0728
                                 FC    AD       04   AE 9E 00029  MOVAB   HEADER, HDRDESC+4                     ; 0729
                                                F8   AD 9F 0002E  PUSHAB  HDRDESC                               ; 0730
                                                52   DD 00031  PUSHL   R2
                                       08    AE       24   A1 3C 00033  MOVZWL  36(R1), 8(SP)
                                                08   AE 9F 00038  PUSHAB  8(SP)
                                00000000G       00        03   FB 0003B  CALLS   #3, LBR$SET_MODULE
                                                4D        50   E9 00042  BLBC    STATUS, 4$
                                                51        63   D0 00045  MOVL    LNK$GL_CURFIL, R1             ; 0737
                                                05        14   AE E9 00048  BLBC    HEADER+16, 2$                 ; 0736
                                          0A    A1             08   88 0004C  BISB2   #8, 10(R1)                 ; 0737
                                                               04   00050  RET
                                          0A    A1             08   8A 00051  2$:  BICB2   #8, 10(R1)           ; 0738
                                                               04   00055  RET                                   ; 0739
                                                               52   DD 00056  3$:  PUSHL   R2                    ; 0743
                                          04    AE       24   A1 3C 00058  MOVZWL  36(R1), 4(SP)
                                                04   AE 9F 0005D  PUSHAB  4(SP)
                                00000000G       00        02   FB 00060  CALLS   #2, LBR$FIND
                                                7E        50   E8 00067  BLBS    STATUS, 7$                    ; 0744
                                                51   FC   A3   D0 0006A  MOVL    LNK$GL_CUROMD, R1             ; 0746
                                          F0    AD       28   A1 9B 0006E  MOVZBW  40(R1), MODNAMDESC
                                          F4    AD       29   A1 9E 00073  MOVAB   41(R1), MODNAMDESC+4         ; 0747
                                                               52   DD 00078  PUSHL   R2                        ; 0748
                                                          F0   AD 9F 0007A  PUSHAB  MODNAMDESC
                                                51             63   D0 0007D  MOVL    LNK$GL_CURFIL, R1
                                          08    AE       24   A1 3C 00080  MOVZWL  36(R1), 8(SP)
                                                08   AE 9F 00085  PUSHAB  8(SP)
                                00000000G       00        03   FB 00088  CALLS   #3, LBR$LOOKUP_KEY
```

```
                 56              50  E8 0008F        BLBS    STATUS, 7$                            ; 0749
          C0     A3              50  D0 00092  4$:   MOVL    STATUS, LNK$AL_RAB+8                  ; 0751
          C4     A3  00000000G   00  D0 00096        MOVL    LBR$GL_RMSSTV, LNK$AL_RAB+12          ; 0752
                                 1D  11 0009E        BRB     6$                                    ; 0721
          D6     A3              02  90 000A0  5$:   MOVB    #2, LNK$AL_RAB+30                     ; 0756
          C8     A3              62  D0 000A4        MOVL    (R2), LNK$AL_RAB+16                   ; 0757
          CC     A3          04  A2  B0 000A8        MOVW    4(R2), LNK$AL_RAB+20                  ; 0758
                             B8  A3  9F 000AD        PUSHAB  LNK$AL_RAB                            ; 0759
      00000000G  00              01  FB 000B0        CALLS   #1, SYS$FIND                          ;
                             D6  A3  94 000B7        CLRB    LNK$AL_RAB+30                         ; 0760
                 2B              50  E8 000BA        BLBS    STATUS, 7$                            ; 0761
                             C4  A3  DD 000BD  6$:   PUSHL   LNK$AL_RAB+12                         ; 0767
                                 50  DD 000C0        PUSHL   STATUS                                ;
                                 7E  D4 000C2        CLRL    -(SP)                                 ; 0766
                     00000000G   8F  DD 000C4        PUSHL   #LIN$_FORMAT                          ;
   7E            63              14  C1 000CA        ADDL3   #20, LNK$GL_CURFIL, -(SP)             ;
   7E     FC     A3              28  C1 000CE        ADDL3   #40, LNK$GL_CUROMD, -(SP)             ; 0765
                 7E          04  A2  3C 000D3        MOVZWL  4(R2), -(SP)                          ; 0766
                                 62  DD 000D7        PUSHL   (R2)                                  ;
                                 04  DD 000D9        PUSHL   #4                                    ;
                     00000000G   8F  DD 000DB        PUSHL   #LIN$_LIBFIND                         ;
      00000000G  00              0A  FB 000E1        CALLS   #10, LIB$SIGNAL                       ;
                                     04 000E8  7$:   RET                                           ; 0770

; Routine Size:  233 bytes,    Routine Base:  $CODE$ + 0510
```

```
;  662          0771  1 global routine lnk$closefile ( fdbclose ) : novalue =
;  663          0772  2 begin
;  664          0773  2 !
;  665          0774  2 ! THIS ROUTINE WILL LOCATE A FILE FOR CLOSING -- IF THE FDB
;  666          0775  2 !       IS EXPLICITLY SPECIFIED BY BEING PASSED, THEN THAT FILE IS
;  667          0776  2 !       CLOSED, OTHERWISE THE NEXT OPEN FILE LOCATED FROM THE
;  668          0777  2 !       OPENCLUS AND OPENFDB POINTERS IS CLOSED.
;  669          0778  2 !
;  670          0779  2
;  671          0780  2
;  672          0781  2         routine closeit (fdbclose) : novalue =
;  673          0782  3         begin
;  674          0783  3         !
;  675          0784  3         ! ROUTINE TO CLOSE A FILE GIVEN ITS FDB
;  676          0785  3         !
;  677          0786  3         map     fdbclose  : ref block [,byte];
;  678          0787  3         bind    auxfnb    = fdbclose [fdb$t_auxfnb] : block [nam$c_bln,byte];
;  679          0788  3         local   input_fab : block [fab$c_bln,byte],
;  680          0789  3                 status ;
;  681          0790  3
;  682          0791  3         if .fdbclose [fdb$w_ifi] neq 0                          ! IF FILE IS OPEN
;  683          0792  4         then begin
;  684          0793  4              if not .fdbclose [fdb$v_libr]                      ! AND FILE IS NOT A LIBRARY
;  685          0794  5              then begin
;  686          0795  5                   $fab_init (fab=input_fab);
;  687          0796  5                   input_fab [fab$w_ifi] = .fdbclose [fdb$w_ifi];
;  688          0797  5                   auxfnb [nam$b_rss] = 0;
;  689          0798  5                   auxfnb [nam$b_ess] = 0;
;  690          0799  5                   status = $close (fab=input_fab);
;  691          0800  5                   end
;  692          0801  5              else begin                                         ! FILE IS A LIBRARY
;  693          0802  5                   status = lbr$close (%ref (.fdbclose [fdb$w_ifi]));
;  694          0803  5                   input_fab [fab$l_stv] = .lbr$gl_rmsstv;
;  695          0804  5                   if .status then lnk$gl_open_lbr = .lnk$gl_open_lbr - 1 ;
;  696          0805  4                   end;
;  697          0806  4
;  698          0807  4              if not .status
;  699          0808  4              then signal (lin$_closein,1, fdbclose [fdb$y_filename], .status, .input_fab [fab$l_stv]);
;  700          0809  4
;  701          0810  4              fdbclose [fdb$w_ifi] = 0;                          ! FLAG FILE CLOSED
;  702          0811  3              end;
;  703          0812  3         return;
;  704          0813  2         end;                                                    ! OF CLOSEIT


                                                .EXTRN  SYS$CLOSE

                                  01FC 00000 CLOSEIT:.WORD  Save R2,R3,R4,R5,R6,R7,R8                     ; 0781
                          5E      AC   AE  9E 00002        MOVAB   -84(SP), SP                            ;
                          56      04   AC  D0 00006        MOVL    FDBCLOSE, R6                           ; 0787
                          57      26   A6  9E 0000A        MOVAB   38(R6), R7                             ;
                          58      24   A6  3C 0000E        MOVZWL  36(R6), R8                             ; 0791
                                       6A  13 00012        BEQL    5$                                     ;
                    2D    0A    A6         01  E0 00014    BBS     #1, 10(R6), 1$                         ; 0793
    0050  8F        00          6E         00  2C 00019    MOVC5   #0, (SP), #0, #80, $RMS_PTR            ; 0795
                                       04  AE     00020                                                   ;
```

```
                     04   AE      5003   8F   B0 00022          MOVW    #20483, $RMS_PTR                                        ;
                     1A   AE             02   90 00028          MOVB    #2, $RMS_PTR+22                                         ;
                     23   AE             02   90 0002C          MOVB    #2, $RMS_PTR+31                                         ;
                     06   AE             58   B0 00030          MOVW    R8, INPUT_FAB+2                                         ; 0796
                                    02   A7   94 00034          CLRB    2(R7)                                                   ; 0797
                                    0A   A7   94 00037          CLRB    10(R7)                                                  ; 0798
                                    04   AE   9F 0003A          PUSHAB  INPUT_FAB                                               ; 0799
           00000000G   00           01   FB 0003D          CALLS   #1, SYS$CLOSE                                           ;
                                    1B   11 00044          BRB     2$                                                      ; 0793
                        6E          58   D0 00046 1$:       MOVL    R8, (SP)                                                ; 0802
                                    5E   DD 00049          PUSHL   SP                                                      ;
           00000000G   00           01   FB 0004B          CALLS   #1, LBR$CLOSE                                           ;
                     10   AE 00000000G   00   D0 00052          MOVL    LBR$GL_RMSSTV, INPUT_FAB+12                             ; 0803
                        07           50   E9 0005A          BLBC    STATUS, 3$                                              ; 0804
                                0000'   CF   D7 0005D          DECL    LNK$GL_OPEN_LBR                                         ;
                        17           50   E8 00061 2$:       BLBS    STATUS, 4$                                              ; 0807
                                 10   AE   DD 00064 3$:       PUSHL   INPUT_FAB+12                                            ; 0808
                                    50   DD 00067          PUSHL   STATUS                                                  ;
                                 14   A6   9F 00069          PUSHAB  20(R6)                                                  ;
                                    01   DD 0006C          PUSHL   #1                                                      ;
                            00000000G   8F   DD 0006E          PUSHL   #LIN$_CLOSEIN                                           ;
           00000000G   00           05   FB 00074          CALLS   #5, LIB$SIGNAL                                          ;
                                 24   A6   B4 00078 4$:       CLRW    36(R6)                                                  ; 0810
                                    04   00 0007E 5$:       RET                                                             ; 0813

; Routine Size:  127 bytes,    Routine Base:  $CODE$ + 05F9

;  705          0814  2
;  706          0815  2 !
;  707          0816  2 !
;  708          0817  2 ! Main body of LNK$CLOSEFILE
;  709          0818  2 !
;  710          0819  2 !
;  711          0820  2 map
;  712          0821  2         fdbclose : ref block [,byte];
;  713          0822  2 builtin
;  714          0823  2         nullparameter;
;  715          0824  2 label
;  716          0825  2         cluster;
;  717          0826  2
;  718          0827  2 if not nullparameter (1)                                  ! IF PARTICULAR FILE SPECIFIED
;  719          0828  2 then return (closeit (.fdbclose)) ;                       ! THEN CLOSE IT AND RETURN
;  720          0829  2
;  721          0830  2 if .openclus eql 0                                        ! IF OPEN CLUSTER LIST UNINITIALIZED,
;  722          0831  3 then begin                                               !   THEN DO THAT NOW
;  723          0832  3      openclus  = .lnk$gl_clulst [0];                     ! FIRST CLUSTER,
;  724          0833  3      openfdb   = .openclus [clu$l_fstfdb];               !  FIRST FDB IN FIRST CLUSTER
;  725          0834  2      end;
;  726          0835  2
;  727          0836  2 start_fdb = 0;                                          ! INITIALIZE STARTING FDB
;  728          0837  2
;  729          0838  2 cluster:
;  730          0839  3     begin
;  731          0840  3     while true do
;  732          0841  4         begin
;  733          0842  4         while   .openfdb neq 0 do                       ! WHILE NOT AT END OF FDB LIST FOR THIS CLUS
```

```
;  734          0843  4              if  .openfdb [fdb$w_ifi] eql 0                          !  IF FILE IS ALREADY CLOSED
;  735          0844  5              then begin                                              !
;  736          0845  5                   if  .start_fdb eql .openfdb                       ! BACK TO WHERE WE STARTED, NOTHING FOUND
;  737          0846  6                   then begin
;  738          0847  6                        if   .openclus eql lnk$gl_defclu             ! IF ALREADY AT THE DEFAULT CLUSTER
;  739          0848  7                        then begin                                   !  *** this should be an error report ***
;  740          0849  7                             openclus = 0;                           ! FORCE A RESTART FROM THE TOP
;  741          0850  7                             return;                                 ! GO AWAY FOR NOW
;  742          0851  7                             end
;  743          0852  7                        else begin
;  744          0853  7                             openclus =  lnk$gl_defclu;              ! ELSE, USE DEFAULT CLUSTER
;  745          0854  7                             openfdb  = .openclus [clu$l_fstfdb];    ! ALWAYS RESET FDB TO FIRST IN NEW CLUSTER
;  746          0855  7                             end
;  747          0856  6                        end
;  748          0857  6                   else begin
;  749          0858  6                        if  .start_fdb eql 0                        ! IF STARTING POINT HASN'T BEEN SET YET,
;  750          0859  6                        then start_fdb = .openfdb;                  !  THEN DO IT NOW
;  751          0860  6
;  752          0861  6                        openfdb = .openfdb [fdb$l_nxtfdb];          ! LOOK AT THE NEXT FDB FOR THIS CLUSTER
;  753          0862  6                        end
;  754          0863  5                   end
;  755          0864  4              else leave cluster;                                     ! WE HAVE FOUND AN OPEN FILE -- LEAVE
;  756          0865  4
;  757          0866  4            if  .openclus [clu$l_nxtclu] eql 0                        ! IF AT THE END OF THE CLUSTER LIST
;  758          0867  4            then openclus = .lnk$gl_clulst [0]                        ! START AT BEGINNING OF CLUSTER LIST AGAIN
;  759          0868  4            else openclus = .openclus [clu$l_nxtclu];                 ! ELSE MOVE ON TO NEXT CLUSTER IN LIST
;  760          0869  4
;  761          0870  4            openfdb = .openclus [clu$l_fstfdb];                       ! ALWAYS RESET FDB TO FIRST IN NEW CLUSTER
;  762          0871  3            end;
;  763          0872  2        end;                                                          ! OF CLUSTER
;  764          0873  2
;  765          0874  2 closeit (.openfdb);                                                  ! GOT A FILE, NOW CLOSE IT
;  766          0875  2 openfdb   = .openfdb [fdb$l_nxtfdb];
;  767          0876  2
;  768          0877  2 return;
;  769          0878  1 end;                                                                 ! OF LNK$CLOSEFILE



                                                001C 00000          .ENTRY  LNK$CLOSEFILE, Save R2,R3,R4                    ; 0771
                                    54 00000000G   00   9E 00002    MOVAB   LNK$GL_DEFCLU, R4                               ;
                                    53 00000000G   00   9E 00009    MOVAB   LNK$GL_CLULST, R3                               ;
                                    52       0000' CF   9E 00010    MOVAB   OPENCLOS, R2                                    ;
                                                   6C   95 00015    TSTB    (AP)                                            ; 0827
                                                   0E      00017    BEQL    1$                                              ;
                                             04    AC   D5 00019    TSTL    4(AP)                                           ;
                                                   09   13 0001C    BEQL    1$                                              ;
                                             04    AC   DD 0001E    PUSHL   FDBCLOSE                                        ; 0828
                                    FF5B     CF        01   FB 00021    CALLS   #1, CLOSEIT                                     ;
                                                       04   00026    RET                                                     ;
                                                   62   D5 00027 1$: TSTL    OPENCLUS                                        ; 0830
                                                   0B   12 00029    BNEQ    2$                                              ;
                                    62                 63   D0 0002B    MOVL    LNK$GL_CLULST, OPENCLUS                         ; 0832
                                    50                 62   D0 0002E    MOVL    OPENCLOS, R0                                    ; 0833
                                      04   A2        08   A0   D0 00031    MOVL    8(R0), OPENFDB                                  ;
```

```
                                08 A2  D4 00036 2$:     CLRL    START_FDB                       ; 0836
                             51 04 A2  D0 00039 3$:     MOVL    OPENFDB, R1                     ; 0842
                                2A 13  0003D            BEQL    7$                              ;
                             24 A1  B5 0003F            TSTW    36(R1)                          ; 0843
                                3D 12  00042            BNEQ    10$                             ;
                             51 08 A2  D1 00044         CMPL    START_FDB, R1                   ; 0845
                                10 12  00048            BNEQ    5$                              ;
                             50    64  9E 0004A         MOVAB   LNK$GL_DEFCLU, R0               ; 0847
                             50    62  D1 0004D         CMPL    OPENCLUS, R0                    ;
                                03 12  00050            BNEQ    4$                              ;
                                62  D4 00052            CLRL    OPENCLUS                        ; 0849
                                    04 00054            RET                                     ; 0848
                             62     64  9E 00055 4$:     MOVAB   LNK$GL_DEFCLU, OPENCLUS         ; 0853
                                1D 11  00058            BRB     9$                              ; 0854
                             08 A2  D5 0005A 5$:        TSTL    START_FDB                       ; 0858
                                04 12  0005D            BNEQ    6$                              ;
                08 A2           51  D0 0005F            MOVL    R1, START_FDB                   ; 0859
                04 A2           61  D0 00063 6$:        MOVL    (R1), OPENFDB                   ; 0861
                                D0 11  00067            BRB     3$                              ; 0844
                             50 00  B2  D0 00069 7$:    MOVL    @OPENCLUS, R0                   ; 0866
                                05 12  0006D            BNEQ    8$                              ;
                             62     63  D0 0006F         MOVL    LNK$GL_CLULST, OPENCLUS         ; 0867
                                03 11  00072            BRB     9$                              ;
                             62     50  D0 00074 8$:     MOVL    R0, OPENCLUS                    ; 0868
                             50     62  D0 00077 9$:     MOVL    OPENCLUS, R0                    ; 0870
                04 A2        08 A0  D0 0007A            MOVL    8(R0), OPENFDB                  ;
                                B8 11  0007F            BRB     3$                              ; 0840
                             04 A2  DD 00081 10$:       PUSHL   OPENFDB                         ; 0874
             FEF8 CF            01  FB 00084            CALLS   #1, CLOSEIT                     ;
                04 A2        04 B2  D0 00089            MOVL    @OPENFDB, OPENFDB               ; 0875
                                    04 0008E            RET                                     ; 0878

; Routine Size:  143 bytes,    Routine Base:  $CODE$ + 0678
```

```
;  771          0879  1 routine tran_next_lib =
;  772          0880  1 !
;  773          0881  1 !       THIS ROUTINE RETURNS THE VALUE TRUE IF THERE IS ANOTHER DEFAULT LIBRARY
;  774          0882  1 !       TO PROCESS AND FALSE IF NOT.  IF THERE IS ANOTHER LIBRARY, THE FILENAME
;  775          0883  1 !       IS SET UP IN THE VECTOR RESULTSTRING WITH ITS DESCRIPTOR IN LIBNAMDESCR.
;  776          0884  1 !
;  777          0885  2 begin
;  778          0886  2 bind
;  779          0887  2         maxliblng = %charcount ('LNK$LIBRARY_999');             ! MAX LIBRARY LOGICAL NAME LENGTH
;  780          0888  2 own
;  781          0889  2         lnklibnam : vector [%charcount('LNK$LIBRARY')+1, byte]  ! THE INITIAL DEFAULT LIBRARY NAME
;  782          0890  2                   initial (%ascii'LNK$LIBRARY');
;  783          0891  2 local
;  784          0892  2         trancode,
;  785          0893  2         deflibnam       : vector [maxliblng, byte],             ! TO RECEIVE THE FOPMATTED LOGICAL NAME
;  786          0894  2         namptr          : vector [2];                           ! DESCRIPTOR OF DEFLIBNAM
;  787          0895  2
;  788          0896  2 if .lnk$gl_ulibmask [2 - .deflibacmode] neq 0                    ! IF THIS LOGICAL NAME TABLE ENABLED
;  789          0897  3 then begin
;  790          0898  3      namptr[0] = maxliblng;                                     ! INIT THE DESCRIPTOR
;  791          0899  3      namptr[1] = deflibnam;
;  792          0900  3      if .userlibno eql 0                                        ! IF THIS IS FIRST IN THIS TABLE
;  793          0901  4      then begin
;  794          0902  4           namptr[0] = %charcount ('LNK$LIBRARY');               ! SET THE COPY LENGTH
;  795          0903  4           ch$move (.namptr [0], lnklibnam [0], deflibnam);      ! COPY THE NAME
;  796          0904  4           end
;  797          0905  3      else if not sys$fao(libnamefao,namptr,namptr,.userlibno)   ! OTHERWISE MUST FAO IT
;  798          0906  3           then return false;                                    ! AND GIVE UP IF IT FAILS
;  799          0907  3      !
;  800          0908  3      !    NOW TRANSLATE THE LOGICAL NAME
;  801          0909  3      !
;  802          0910  3      libnamdescr [0] = nam$c_maxrss;                            ! RESET STRING DESCRIPTOR
;  803        P 0911  3      trancode        = $trnlog  (lognam = namptr                ! TRANSLATE THE NAME
;  804        P 0912  3                                 ,dsbmsk =.deflibdsbmsk [.deflibacmode]
;  805        P 0913  3                                 ,rsllen = libnamdescr
;  806        P 0914  3                                 ,rslbuf = libnamdescr
;  807          0915  3                                 );
;  808          0916  4      if (.trancode and (.trancode neq ss$_notran))              ! IF GOT A GOOD NAME RETURN WITH IT
;  809          0917  3      then return true;
;  810          0918  2      end;
;  811          0919  2 !
;  812          0920  2 ! FAILED. MOVE ON TO NEXT GROUP
;  813          0921  2 !
;  814          0922  2 if (deflibacmode = .deflibacmode+1) gtr 2                       ! BUT IF WE ARE ALL DONE
;  815          0923  2 then return false;                                             ! THEN RETURN NO MORE
;  816          0924  2
;  817          0925  2 userlibno = 0;                                                 ! OTHERWISE RESET LIBRARY NUMBER
;  818          0926  2 return tran_next_lib ();                                       ! AND RECURSE TO RETURN NEXT LIBRARY
;  819          0927  1 end;


                                                        .PSECT  $OWN$,NOEXE,2

          00  59  52  41  52  42  49  4C  24  4B  4E  4C  001B4 LNKLIBNAM:
                                                        .ASCII  \LNK$LIBRARY\<0>                                ;
```

```
                                MAXLIBLNG=          15
                                        .EXTRN  SYS$TRNLOG

                                        .PSECT  $CODE$,NOWRT,2

                                007C 00000 TRAN_NEXT_LIB:
                                        .WORD   Save R2,R3,R4,R5,R6                     ; 0879
                             56   0000'  CF  9E 00002          MOVAB   DEFLIBACMODE, R6
                             5E           18  C2 00007          SUBL2   #24, SP
                             50           66  9A 0000A          MOVZBL  DEFLIBACMODE, R0   ; 0896
                          50 02           50  C3 0000D          SUBL3   R0, #2, R0
                51 00000000G  00  01      50  EF 00011          EXTZV   R0, #1, LNK$GL_ULIBMASK, R1
                                          61  13 0001A          BEQL    3$
                             6E           0F  D0 0001C          MOVL    #15, NAMPTR        ; 0898
                       04    AE     08    AE  9E 0001F          MOVAB   DEFLIBNAM, NAMPTR+4 ; 0899
                             50     FA    A6  D0 00024          MOVL    USERLIBNO, R0      ; 0900
                                          0C  12 00028          BNEQ    1$
                             6E           0B  D0 0002A          MOVL    #11, NAMPTR        ; 0902
                08    AE     010E  C6      6E  28 0002D          MOVC3   NAMPTR, LNKLIBNAM, DEFLIBNAM ; 0903
                                          15  11 00034          BRB     2$                 ; 0900
                                          50  DD 00036 1$:      PUSHL   R0                 ; 0905
                                    04    AE  9F 00038          PUSHAB  NAMPTR
                                    08    AE  9F 0003B          PUSHAB  NAMPTR
                                    E6    A6  9F 0003E          PUSHAB  LIBNAMEFAO
                          00000000G  00       04  FB 00041          CALLS   #4, SYS$FAO
                                     48       50  E9 00048          BLBC    R0, 4$
                              0106   C6    FF  8F  9A 0004B 2$:      MOVZBL  #255, LIBNAMDESCR  ; 0910
                                     50           66  9A 00051          MOVZBL  DEFLIBACMODE, R0   ; 0915
                                     7E       02 A640  9A 00054          MOVZBL  DEFLIBDSBMSK[R0], -(SP)
                                              7E  7C 00059          CLRQ    -(SP)
                                    0106   C6    9F 0005B          PUSHAB  LIBNAMDESCR
                                    0106   C6    9F 0005F          PUSHAB  LIBNAMDESCR
                                      14   AE    9F 00063          PUSHAB  NAMPTR
                          00000000G  00       06  FB 00066          CALLS   #6, SYS$TRNLOG
                                     0D       50  E9 0006D          BLBC    TRANCODE, 3$       ; 0916
                          00000629   8F       50  D1 00070          CMPL    TRANCODE, #1577
                                              04  13 00077          BEQL    3$
                                     50       01  D0 00079          MOVL    #1, R0             ; 0917
                                              04  0007C          RET
                                     50           66  9A 0007D 3$:      MOVZBL  DEFLIBACMODE, R0   ; 0922
                                              50  D6 00080          INCL    R0
                                     66       50  90 00082          MOVB    R0, DEFLIBACMODE
                                     02       50  D1 00085          CMPL    R0, #2
                                              09  14 00088          BGTR    4$
                                        FA    A6  D4 0008A          CLRL    USERLIBNO          ; 0925
                                FF6E  CF      00  FB 0008D          CALLS   #0, TRAN_NEXT_LIB  ; 0926
                                              04  00092          RET
                                              50  D4 00093 4$:      CLRL    R0                 ; 0927
                                              04  00095          RET

; Routine Size:  150 bytes,    Routine Base:  $CODE$ + 0707
```

```
;  821          0928  1 global routine lnk$openlib (nameblock, an_open_fdb) =
;  822          0929  1 !
;  823          0930  1 ! THIS ROUTINE OPENS A LIBRARY FILE USING THE LIBRARY ACCESS PROCEDURES.
;  824          0931  1 !
;  825          0932  2 begin
;  826          0933  2 map
;  827          0934  2         nameblock       : ref block [,byte] ;
;  828          0935  2 local
;  829          0936  2         status,
;  830          0937  2         libraryfunc     : initial (lbr$c_read),
;  831          0938  2         fdb_to_close    : initial (0),
;  832          0939  2         close_lib       : initial (false),
;  833          0940  2         init_lib        : initial (false),
;  834          0941  2         filnamdesc      : block [dsc$c_s_bln,byte],
;  835          0942  2         index ;
;  836          0943  2 builtin
;  837          0944  2         nullparameter ;
;  838          0945  2 bind
;  839          0946  2         fdbfilename = lnk$gl_curfil [fdb$q_filename] : block [,byte] ;
;  840          0947  2
;  841          0948  2
;  842          0949  2 if  .nameblock [nam$w_fid_num] neq 0                          ! IF FILE HAS BEEN OPENED
;  843          0950  2 and .lnk$gl_curfil [fdb$w_ifi] neq 0                         ! AND IS OPENED NOW
;  844          0951  2 then return true ;                                          ! THEN WE'RE DONE
;  845          0952  2
;  846          0953  2 if not nullparameter(2) then fdb_to_close = .an_open_fdb ;  ! DETERMINE FDB TO BE CLOSED NEXT
;  847          0954  2
;  848          0955  2 $getjpi (itmlst=lnk$gt_jpilst) ;                            ! WE MAY NEED TO CLOSE ONE SO THAT
;  849          0956  2 if .lnk$gl_filesleft leq 3  then close_lib = true ;         ! WE CAN OPEN ANOTHER
;  850          0957  2
;  851          0958  3 do  begin
;  852          0959  3     if  .close_lib
;  853          0960  4     then begin                                             ! THEN CLOSE A FILE AND TRY AGAIN
;  854          0961  4          lnk$closefile (.fdb_to_close) ;
;  855          0962  4          fdb_to_close = 0 ;
;  856          0963  4          end
;  857          0964  3     else close_lib = true ;                                ! CLOSE A FILE NEXT TIME THRU THE LOOP
;  858          0965  3     status = lbr$ini_control (index,libraryfunc,0,.nameblock) ; ! INITIALIZE INDEX
;  859          0966  3     end
;  860          0967  2 until .status neq lbr$_toomnylib ;
;  861          0968  2
;  862          0969  2 if not .status then return .status ;
;  863          0970  2 if .nameblock [nam$w_fid_num] eql 0                          ! IF FILE HAS NEVER BEEN OPENED...
;  864          0971  3 then begin
;  865          0972  3      status =  lbr$open (index, lnk$gl_curfil [fdb$q_usrnamdsc]
;  866          0973  3                        ,0,     lnk$gl_curfil [fdb$w_defnamlen]
;  867          0974  3                        ) ;
;  868          0975  3      fdbfilename [dsc$w_length]  = .nameblock [nam$b_rsl] ;  ! SET NAME DESCRIPTOR
;  869          0976  3      fdbfilename [dsc$a_pointer] = .nameblock [nam$l_rsa] ;
;  870          0977  3      end
;  871          0978  2 else status = lbr$open (index) ;                           ! IN PASS 2 CAN OPEN BY NAM BLOCK
;  872          0979  2
;  873          0980  2 if .status
;  874          0981  3 then begin
;  875          0982  3      lnk$gl_open_lbr                = .lnk$gl_open_lbr + 1 ;
;  876          0983  3      lnk$gl_curfil [fdb$w_ifi]      = .index ;             ! IF SUCCESSFUL OPEN, SET INDEX FOR CLOSE
;  877          0984  3      lbr$gl_control [lbr$v_locate] =  true ;               ! SET LOCATE MODE
```

```
;  878          0985  3
;  879          0986  3       if .lbr$gl_rmsstv eql lbr$c_typ_shstb                    ! IF SHAREABLE IMAGE SYMBOL TABLE LIBRARY
;  880          0987  3       then lnk$gl_curfil [fdb$v_imgli6] = true                 !  THEN MARK IT AS SUCH
;  881          0988  3       else if .lbr$gl_rmsstv neq lbr$c_typ_obj                 !  OTHERWISE IF NOT AN OBJECT LIBRARY
;  882          0989  3            then  signal_stop (lin$_notobjlib,1,fdbfilename) ;  !  THEN SIGNAL ERROR AND GIVE UP
;  883          0990  3
;  884          0991  3       if .lnk$al_rab [rab$w_usz] eql 0
;  885          0992  4       then begin
;  886          0993  4            lnk$al_rab [rab$w_usz] = obj$c_maxrecsiz ;
;  887          0994  4            lnk$alloblk (.lnk$al_rab [rab$w_usz], lnk$al_rab [rab$l_ubf]) ;
;  888          0995  3            end ;
;  889          0996  2       end ;
;  890          0997  2
;  891          0998  2 return .status
;  892          0999  1 end ;                                                         ! OF LNK$OPENLIB

                              01FC 00000          .ENTRY  LNK$OPENLIB, Save R2,R3,R4,R5,R6,R7,R8      ; 0928
58 00000000G   00   9E 00002  MOVAB   LBR$OPEN, R8                                                    ;
57      0000'  CF   9E 00009  MOVAB   LNK$GL_CURFIL, R7                                              ;
5E             0C   C2 0000E  SUBL2   #12, SP                                                        ;
               01   DD 00011  PUSHL   #1                                                            ; 0932
               53   7C 00013  CLRQ    CLOSE_LIB                                                     ;
               50   D4 00015  CLRL    INIT_LIB                                                      ;
50             67   D0 00017  MOVL    LNK$GL_CURFIL, R0                                             ; 0946
55          14   A0   9E 0001A  MOVAB   20(R0), R5                                                  ;
52          04   AC   D0 0001E  MOVL    NAMEBLOCK, R2                                               ; 0949
            24   A2   B5 00022  TSTW    36(R2)                                                      ;
               09   13 00025  BEQL    1$                                                            ;
            24   A0   B5 00027  TSTW    36(R0)                                                      ; 0950
               04   13 0002A  BEQL    1$                                                            ;
50             01   D0 0002C  MOVL    #1, R0                                                      ; 0951
               04   0002F  RET                                                                      ;
02             6C   91 00030  1$:     CMPB    (AP), #2                                            ; 0953
               09   1F 00033  BLSSU   2$                                                           ;
            08   AC   D5 00035  TSTL    8(AP)                                                       ;
               04   13 00038  BEQL    2$                                                            ;
54          08   AC   D0 0003A  MOVL    AN_OPEN_FDB, FDB_TO_CLOSE                                   ;
               7E   7C 0003E  2$:     CLRQ    -(SP)                                               ; 0955
               7E   D4 00040  CLRL    -(SP)                                                       ;
   00000000G   00   9F 00042  PUSHAB  LNK$GT_JPILST                                               ;
               7E   7C 00048  CLRQ    -(SP)                                                       ;
               7E   D4 0004A  CLRL    -(SP)                                                       ;
000000000G     00   07   FB 0004C  CALLS   #7, SYS$GETJPI                                         ;
               03 00000000G   00   D1 00053  CMPL    LNK$GL_FILESLEFT, #3                        ; 0956
               03   14 0005A  BGTR    3$                                                          ;
               53   01   D0 0005C  MOVL    #1, CLOSE_LIB                                         ;
               0B   53   E9 0005F  3$:     BLBC    CLOSE_LIB, 4$                                 ; 0959
               54   DD 00062  PUSHL   FDB_TO_CLOSE                                              ; 0961
    FE72       CF   01   FB 00064  CALLS   #1, LNR$CLOSEFILE                                      ;
               54   D4 00069  CLRL    FDB_TO_CLOSE                                              ; 0962
               03   11 0006B  BRB     5$                                                        ; 0959
               53   01   D0 0006D  4$:     MOVL    #1, CLOSE_LIB                                 ; 0964
               52   DD 00070  5$:     PUSHL   R2                                                ; 0965
```

```
                                 7E  D4 00072          CLRL    -(SP)                                                   ;
                              08 AE  9F 00074          PUSHAB  LIBRARYFUNC                                             ;
                              10 AE  9F 00077          PUSHAB  INDEX                                                   ;
                00000000G     00     04 FB 0007A       CALLS   #4, LBR$INI_CONTROL                                     ;
                               56    50 D0 00081       MOVL    R0, STATUS                                              ;
                00000000G      8F   56 D1 00084       CMPL    STATUS, #LBR$_TOOMNYLIB                                 ; 0967
                                    D2 13 0008B       BEQL    3$                                                      ;
                               2C   56 E9 0008D       BLBC    STATUS, 7$                                              ; 0969
                                 24 A2 B5 00090       TSTW    36(R2)                                                  ; 0970
                                    1E 12 00093       BNEQ    6$                                                      ;
                7E              67   14 C1 00095       ADDL3   #20, LNK$GL_CURFIL, -(SP)                               ; 0973
                                    7E D4 00099       CLRL    -(SP)                                                   ;
                7E              67   0C C1 0009B       ADDL3   #12, LNK$GL_CURFIL, -(SP)                               ; 0972
                              10 AE  9F 0009F          PUSHAB  INDEX                                                   ;
                               68    04 FB 000A2       CALLS   #4, LBR$OPEN                                            ; 0973
                               56    50 D0 000A5       MOVL    R0, STATUS                                              ;
                               65 03 A2 9B 000A8       MOVZBW  3(R2), (R5)                                             ; 0975
                          04   A5 04 A2 D0 000AC       MOVL    4(R2), 4(R5)                                            ; 0976
                                    09 11 000B1       BRB     7$                                                      ; 0970
                              04 AE  9F 000B3 6$:     PUSHAB  INDEX                                                   ; 0978
                               68    01 FB 000B6       CALLS   #1, LBR$OPEN                                            ;
                               56    50 D0 000B9       MOVL    R0, STATUS                                              ;
                               57    56 E9 000BC 7$:   BLBC    STATUS, 10$                                             ; 0980
                              AC     A7 D6 000BF       INCL    LNK$GL_OPEN_LBR                                         ; 0982
                               51    67 D0 000C2       MOVL    LNK$GL_CURFIL, R1                                       ; 0983
                          24   A1 04 AE B0 000C5       MOVW    INDEX, 36(R1)                                           ;
                               50 00000000G  00 D0 000CA MOVL   LBR$GL_CONTROL, R0                                     ; 0984
                          06   A0     01 88 000D1       BISB2   #1, 6(R0)                                               ;
                               50 00000000G  00 D0 000D5 MOVL   LBR$GL_RMSSTV, R0                                      ; 0986
                               05    50 D1 000DC       CMPL    R0, #5                                                  ;
                                    06 12 000DF       BNEQ    8$                                                      ;
                          0B   A1     01 88 000E1       BISB2   #1, 11(R1)                                              ; 0987
                                    16 11 000E5       BRB     9$                                                      ;
                               01    50 D1 000E7 8$:   CMPL    R0, #1                                                  ; 0988
                                    11 13 000EA       BEQL    9$                                                      ;
                                    55 DD 000EC       PUSHL   R5                                                      ; 0989
                                    01 DD 000EE       PUSHL   #1                                                      ;
                          00000000G     8F  DD 000F0  PUSHL   #LIN$_NOTOBJLIB                                         ;
                00000000G     00     03 FB 000F6       CALLS   #3, LIB$STOP                                            ;
                                 D8 A7 B5 000FD 9$:     TSTW    LNK$AL_RAB+32                                           ; 0991
                                    14 12 00100       BNEQ    10$                                                     ;
                          D8   A7  0800 8F B0 00102     MOVW    #2048, LNK$AL_RAB+32                                    ; 0993
                                 DC A7 9F 00108       PUSHAB  LNK$AL_RAB+36                                           ; 0994
                               7E  D8 A7 3C 0010B     MOVZWL  LNK$AL_RAB+32, -(SP)                                    ;
                00000000G     00     02 FB 0010F       CALLS   #2, LNK$ALLOBLK                                         ;
                               50    56 D0 00116 10$:  MOVL    STATUS, R0                                              ; 0998
                                    04 00119          RET                                                             ; 0999

; Routine Size:  282 bytes,    Routine Base:  $CODE$ + 079D

;  893          1000  1
;  894          1001  0 end eludom
```

```
                                .EXTRN  LIB$SIGNAL, LIB$STOP

;                                PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;   $GLOBAL$                       88  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                         448  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                         80  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       2231  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]STARLET.L32;1     9776       107         1         581        00:01.0
;   _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1   538        36         6          28        00:00.4


;                                COMMAND QUALIFIERS

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

; Size:          2231 code + 616 data bytes
; Run Time:        00:41.7
; Elapsed Time:    01:32.2
; Lines/CPU Min:    1442
; Lexemes/CPU-Min: 23402
; Memory Used:  339 pages
; Compilation Complete
```

0217 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LNKNXTOBJ
LIS

LNKOBJPS1
LIS

LNKOBJFIL
LIS