```
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL              YYY         YYY  ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL              YYY         YYY              ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY              ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY              ZZZ
AAA         AAA  NNNNNN      NNN  AAA         AAA  LLL                 YYY   YYY              ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY              ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY              ZZZ
AAA         AAA  NNN   NNN   NNN  AAA         AAA  LLL                    YYY              ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY           ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY           ZZZ
AAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAA  LLL                    YYY           ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLL                    YYY        ZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
AAA         AAA  NNN         NNN  AAA         AAA  LLLLLLLLLLLLLLL        YYY        ZZZZZZZZZZZZZZZ
```

```
28      **FILE**ID**OBJINPUT
29
75
80
```

```
  000000    BBBBBBBB          JJ    IIIIII    NN      NN  PPPPPPPP    UU      UU  TTTTTTTTTT
  000000    BBBBBBBB          JJ    IIIIII    NN      NN  PPPPPPPP    UU      UU  TTTTTTTTTT
00      00  BB      BB        JJ      II      NN      NN  PP      PP  UU      UU      TT
00      00  BB      BB        JJ      II      NN      NN  PP      PP  UU      UU      TT
00      00  BB      BB        JJ      II      NNNN    NN  PP      PP  UU      UU      TT
00      00  BB      BB        JJ      II      NNNN    NN  PP      PP  UU      UU      TT
00      00  BBBBBBBB          JJ      II      NN  NN  NN  PPPPPPPP    UU      UU      TT
00      00  BBBBBBBB          JJ      II      NN  NN  NN  PPPPPPPP    UU      UU      TT
00      00  BB      BB  JJ    JJ      II      NN    NNNN  PP          UU      UU      TT
00      00  BB      BB  JJ    JJ      II      NN    NNNN  PP          UU      UU      TT
00      00  BB      BB  JJ    JJ      II      NN      NN  PP          UU      UU      TT          ....
00      00  BB      BB  JJ    JJ      II      NN      NN  PP          UU      UU      TT          ....
  000000    BBBBBBBB      JJJJJJ    IIIIII    NN      NN  PP          UUUUUUUUUU      TT          ....
  000000    BBBBBBBB      JJJJJJ    IIIIII    NN      NN  PP          UUUUUUUUUU      TT          ....
```

```
LL              IIIIII      SSSSSSSS
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 %title 'OBJINPUT - Handle Object Files & Libraries'
;  2          0002  0         module objinput (
;  3          0003  1                          ident='V04-000') = begin
;  4          0004  1
;  5          0005  1 !
;  6          0006  1 !*****************************************************************************
;  7          0007  1 !*                                                                           *
;  8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 10          0010  1 !*  ALL RIGHTS RESERVED.                                                     *
; 11          0011  1 !*                                                                           *
; 12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 13          0013  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE          *
; 14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER     *
; 15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 16          0016  1 !*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
; 17          0017  1 !*  TRANSFERRED.                                                             *
; 18          0018  1 !*                                                                           *
; 19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE    *
; 20          0020  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT        *
; 21          0021  1 !*  CORPORATION.                                                             *
; 22          0022  1 !*                                                                           *
; 23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS     *
; 24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 25          0025  1 !*                                                                           *
; 26          0026  1 !*                                                                           *
; 27          0027  1 !*****************************************************************************
; 28          0028  1 !
; 29          0029  1
; 30          0030  1 !++
; 31          0031  1 ! Facility:     VAX/VMS Analyze Facility, Handle Object Files & Libraries
; 32          0032  1 !
; 33          0033  1 ! Abstract:     This module is responsible for handling file specs from
; 34          0034  1 !               the command line, and reading data from object files and
; 35          0035  1 !               libraries.
; 36          0036  1 !
; 37          0037  1 !
; 38          0038  1 ! Environment:
; 39          0039  1 !
; 40          0040  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 8 January 1981
; 41          0041  1 !
; 42          0042  1 ! Modified By:
; 43          0043  1 !
; 44          0044  1 !       V03-005 BLS0286         Benn Schreiber          20-MAR-1984
; 45          0045  1 !               Correct 004.
; 46          0046  1 !
; 47          0047  1 !       V03-004 LJA0113         Laurie J. Anderson      24-Feb-1984
; 48          0048  1 !               Add new related file parsing arguments to LIB$FIND_FILE
; 49          0049  1 !               to make search lists behave properly.
; 50          0050  1 !
; 51          0051  1 !       V03-003 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
; 52          0052  1 !               Change the message prefix to ANLOBJ$_ to ensure that
; 53          0053  1 !               message symbols are unique across all ANALYZEs.  This
; 54          0054  1 !               is necessitated by the new merged message files.
; 55          0055  1 !
; 56          0056  1 !       V03-002 PCA0022         Paul Anagnostopoulos    24-Mar-1982
; 57          0057  1 !               Signal errors using the correct STV values.
```

```
;   58          0058  1 !
;   59          0059  1 !       V03-001 PCA0013         Paul Anagnostopoulos    22-Mar-1982
;   60          0060  1 !               Use the resultant spec rather than the wildcard spec
;   61          0061  1 !               when complaining about a file to be analyzed.
;   62          0062  1 !--
```

4

7
8
5
7

8
9

0

1

```
;   64          0063  1 %sbttl 'Module Declarations'
2  ;   65          0064  1 !
5  ;   66          0065  1 ! Libraries and Requires:
;   67          0066  1 !
;   68          0067  1
;   69          0068  1 library 'starlet';
;   70          0069  1 require 'objexereq';
;   71          0505  1
;   72          0506  1 !
;   73          0507  1 ! Table of Contents:
;   74          0508  1 !
;   75          0509  1
;   76          0510  1 forward routine
;   77          0511  1         anl$open_next_object_file,
;   78          0512  1         anl$object_include,
;   79          0513  1         anl$get_object_record;
;   80          0514  1
;   81          0515  1 !
;   82          0516  1 ! External References:
;   83          0517  1 !
;   84          0518  1
;   85          0519  1 external routine
;   86          0520  1         anl$object_positionals,
;   87          0521  1         cli$get_value: addressing_mode(general),
;   88          0522  1         lbr$close: addressing_mode(general),
;   89          0523  1         lbr$get_index: addressing_mode(general),
6  ;   90          0524  1         lbr$get_record: addressing_mode(general),
7  ;   91          0525  1         lbr$ini_control: addressing_mode(general),
;   92          0526  1         lbr$lookup_key: addressing_mode(general),
8  ;   93          0527  1         lbr$open: addressing_mode(general),
;   94          0528  1         lib$find_file: addressing_mode(general),
9  ;   95          0529  1         lib$free_vm: addressing_mode(general),
;   96          0530  1         lib$get_vm: addressing_mode(general),
2  ;   97          0531  1         str$trim: addressing_mode(general);
4  ;   98          0532  1
;   99          0533  1 !
;  100          0534  1 ! Own Variables:
;  101          0535  1 !
;  102          0536  1 ! The following data is needed to keep track of what kind of file we
;  103          0537  1 ! are processing.
;  104          0538  1
;  105          0539  1 own
;  106          0540  1         own_described_buffer(resultant_spec,nam$c_maxrss),
;  107          0541  1         object_library: byte,
;  108          0542  1         library_index: long;
;  109          0543  1
;  110          0544  1 ! The following data structures are used to access and read records from
;  111          0545  1 ! a file we are to analyze.
;  112          0546  1
;  113          0547  1 own
;  114        P 0548  1         object_fab: $fab(fac=get,
;  115          0549  1                         shr=get),
;  116          0550  1
;  117          0551  1         own_described_buffer(object_buffer,obj$c_maxrecsiz),
;  118          0552  1
;  119        P 0553  1         object_rab: $rab(fab=object_fab,
;  120        P 0554  1                         rac=seq,
```

```
;  121        P 0555  1                       rop=loc,
;  122        P 0556  1                       ubf=object_buffer+8,
;  123          0557  1                       usz=obj$c_maxrecsiz);
```

9

8

```
99          ;  125          0558  1 %sbttl 'ANL$OPEN_NEXT_OBJECT_FILE - Right'
            ;  126          0559  1 !++
            ;  127          0560  1 ! Functional Description:
            ;  128          0561  1 !       This routine is called to open the next object file we are to analyze.
00          ;  129          0562  1 !       It handles multiple file specs, wildcarding, and object libraries.
            ;  130          0563  1 !
            ;  131          0564  1 ! Formal Parameters:
01          ;  132          0565  1 !       opened_spec     Address of descriptor of buffer in which to return
            ;  133          0566  1 !                       the spec of the file we open.  We set the length.
06          ;  134          0567  1 !
07          ;  135          0568  1 ! Implicit Inputs:
            ;  136          0569  1 !       global data
            ;  137          0570  1 !
            ;  138          0571  1 ! Implicit Outputs:
            ;  139          0572  1 !       global data
 1          ;  140          0573  1 !
            ;  141          0574  1 ! Returned Value:
            ;  142          0575  1 !       True if there is another object file, false otherwise.
            ;  143          0576  1 !
            ;  144          0577  1 ! Side Effects:
            ;  145          0578  1 !
            ;  146          0579  1 !--
            ;  147          0580  1
            ;  148          0581  1
            ;  149          0582  2 global routine anl$open_next_object_file(opened_spec) = begin
            ;  150          0583  2
            ;  151          0584  2 bind
            ;  152          0585  2         opened_spec_dsc = .opened_spec: descriptor;
            ;  153          0586  2
            ;  154          0587  2 own
            ;  155          0588  2         own_described_buffer(wildcard_spec,nam$c_maxrss),
            ;  156          0589  2         wildcard_context: long initial(0),
            ;  157          0590  2         all_modules: byte,
            ;  158          0591  2         module_list: ref blockvector[,obj$c_symsiz,byte],
            ;  159          0592  2         module_list_size: long,
            ;  160          0593  2         module_list_index: signed long,
            ;  161          0594  2         get_new_spec: long initial(true);
            ;  162          0595  2
            ;  163          0596  2 local
            ;  164          0597  2         stv: long,
            ;  165          0598  2         status: long;
```

```
;  167          0599  2 ! The following internal routine is called by the librarian when we need to
;  168          0600  2 ! scan an object library index.  We do this if the user asks us to analyze
;  169          0601  2 ! all the modules in the library.  The routine is called once for each module.
;  170          0602  2
;  171          0603  3 routine add_module_to_list(module_name) = begin
;  172          0604  3
;  173          0605  3 bind
;  174          0606  3         module_name_dsc = .module_name: descriptor;
;  175          0607  3
;  176          0608  3 ! Copy the module name into the next module_list entry.
;  177          0609  3
;  178          0610  3 ch$copy(.module_name_dsc[len],.module_name_dsc[ptr],
;  179          0611  3          ' ',obj$c_symsiz,module_list[.module_list_size,0,0,0,0]);
;  180          0612  3
;  181          0613  3 ! Increment the module list size.
;  182          0614  3
;  183          0615  3 increment (module_list_size);
;  184          0616  3
;  185          0617  3 return ss$_normal;
;  186          0618  3
;  187          0619  2 end;
```

```
;                                                       .TITLE  OBJINPUT OBJINPUT - Handle Object Files & Libra
                                                                ries
                                                       .IDENT  \V04-000\

                                                       .PSECT  $OWN$,NOEXE,2
000000FF  00000 RESULTANT_SPEC:
                        .LONG   255                                             ;
00000000' 00004         .ADDRESS RESULTANT_SPEC+8                               ;
          00008         .BLKB   255
          00107 OBJECT_LIBRARY:
                        .BLKB   1
          00108 LIBRARY_INDEX:
                        .BLKB   4
       03 0010C OBJECT_FAB:
                        .BYTE   3                                               ;
       50 0010D         .BYTE   80                                              ;
     0000 0010E         .WORD   0                                               ;
00000000  00110         .LONG   0                                               ;
00000000  00114         .LONG   0                                               ;
00000000  00118         .LONG   0                                               ;
00000000  0011C         .LONG   0                                               ;
     0000 00120         .WORD   0                                               ;
       02 00122         .BYTE   2                                               ;
       02 00123         .BYTE   2                                               ;
00000000  00124         .LONG   0                                               ;
       00 00128         .BYTE   0                                               ;
       00 00129         .BYTE   0                                               ;
       00 0012A         .BYTE   0                                               ;
       02 0012B         .BYTE   2                                               ;
00000000  0012C         .LONG   0                                               ;
00000000  00130         .LONG   0                                               ;
00000000  00134         .LONG   0                                               ;
00000000  00138         .LONG   0                                               ;
```

```
                     00000000  0013C          .LONG   0                                       ;
                           00  00140          .BYTE   0                                       ;
                           00  00141          .BYTE   0                                       ;
                         0000  00142          .WORD   0                                       ;
                     00000000  00144          .LONG   0                                       ;
                         0000  00148          .WORD   0                                       ;
                           00  0014A          .BYTE   0                                       ;
                           00  0014B          .BYTE   0                                       ;
\36                  00000000  0014C          .LONG   0                                       ;
                     00000000  00150          .LONG   0                                       ;
                         0000  00154          .WORD   0                                       ;
                           00  00156          .BYTE   0                                       ;
\5                         00  00157          .BYTE   0                                       ;
                     00000000  00158          .LONG   0                                       ;
\6                   00000800  0015C OBJECT_BUFFER:
                               .LONG   2048                                                    ;
\52                  00000000' 00160          .ADDRESS OBJECT_BUFFER+8                        ;
                               00164          .BLKB   2048
\53                        01  00964 OBJECT_RAB:
                               .BYTE   1
\54                        44  00965          .BYTE   68                                      ;
                         0000  00966          .WORD   0                                       ;
\55                  00010000  00968          .LONG   65536                                   ;
\56                  00000000  0096C          .LONG   0                                       ;
                     00000000  00970          .LONG   0                                       ;
                         0000# 00974          .WORD   0[3]                                    ;
\64                      0000  0097A          .WORD   0                                       ;
                     00000000  0097C          .LONG   0                                       ;
                         0000  00980          .WORD   0                                       ;
\2                         00  00982          .BYTE   0                                       ;
\3                         00  00983          .BYTE   0                                       ;
                         0800  00984          .WORD   2048                                    ;
\4                       0000  00986          .WORD   0                                       ;
                     00000C0'  00988          .ADDRESS OBJECT_BUFFER+8                        ;
                     00000000  0098C          .LONG   0                                       ;
                     00000000  00990          .LONG   0                                       ;
                     00000000  00994          .LONG   0                                       ;
\5                         00  00998          .BYTE   0                                       ;
                           00  00999          .BYTE   0                                       ;
                           00  0099A          .BYTE   0                                       ;
\6                         00  0099B          .BYTE   0                                       ;
\1                   00000000  0099C          .LONG   0                                       ;
                     00000000' 009A0          .ADDRESS OBJECT_FAB                             ;
                     00000000  009A4          .LONG   0                                       ;
                     000000FF  009A8 WILDCARD_SPEC:
                               .LONG   255                                                     ;
                     00000000' 009AC          .ADDRESS WILDCARD_SPEC+8                        ;
                               009B0          .BLKB   255
                               00AAF          .BLKB   1
                     00000000  00AB0 WILDCARD_CONTEXT:
                               .LONG   0                                                       ;
                               00AB4 ALL_MODULES:
                               .BLKB   1
                               00AB5          .BLKB   3
                               00AB8 MODULE_LIST:
                               .BLKB   4
                               00ABC MODULE_LIST_SIZE:
```

```
                                                        .BLKB   4
                                            00AC0 MODULE_LIST_INDEX:
                                                        .BLKB   4
                                  00000001  00AC4 GET_NEW_SPEC:
                                                        .LONG   1                                       ;

                                                        .EXTRN  ANLOBJ$_OK, ANLOBJ$_ANYTHING
                                                        .EXTRN  ANLOBJ$_DATATYPE
                                                        .EXTRN  ANLOBJ$_ERRORCOUNT
                                                        .EXTRN  ANLOBJ$_ERRORNONE
                                                        .EXTRN  ANLOBJ$_ERRORS, ANLOBJ$_EXEFIXA
                                                        .EXTRN  ANLOBJ$_EXEFIXAIMAGE
                                                        .EXTRN  ANLOBJ$_EXEFIXALINE
                                                        .EXTRN  ANLOBJ$_EXEFIXCOUNT
                                                        .EXTRN  ANLOBJ$_EXEFIXEXTRA
                                                        .EXTRN  ANLOBJ$_EXEFIXFIXED
                                                        .EXTRN  ANLOBJ$_EXEFIXFLAGS
                                                        .EXTRN  ANLOBJ$_EXEFIXG
                                                        .EXTRN  ANLOBJ$_EXEFIXGIMAGE
                                                        .EXTRN  ANLOBJ$_EXEFIXGLINE
                                                        .EXTRN  ANLOBJ$_EXEFIXLIST
                                                        .EXTRN  ANLOBJ$_EXEFIXNAME
                                                        .EXTRN  ANLOBJ$_EXEFIXNAMEO
                                                        .EXTRN  ANLOBJ$_EXEFIXP
                                                        .EXTRN  ANLOBJ$_EXEFIXPSECT
                                                        .EXTRN  ANLOBJ$_EXEFIXUP
                                                        .EXTRN  ANLOBJ$_EXEFIXUPNONE
                                                        .EXTRN  ANLOBJ$_EXEGST, ANLOBJ$_EXEHDR
                                                        .EXTRN  ANLOBJ$_EXEHDRACTIVE
                                                        .EXTRN  ANLOBJ$_EXEHDRBLKCOUNT
                                                        .EXTRN  ANLOBJ$_EXEHDRCHANCOUNT
                                                        .EXTRN  ANLOBJ$_EXEHDRCHANDEF
                                                        .EXTRN  ANLOBJ$_EXEHDRDECECO
                                                        .EXTRN  ANLOBJ$_EXEHDRDMT
                                                        .EXTRN  ANLOBJ$_EXEHDRDST
                                                        .EXTRN  ANLOBJ$_EXEHDRFILEID
                                                        .EXTRN  ANLOBJ$_EXEHDRFIXED
                                                        .EXTRN  ANLOBJ$_EXEHDRFLAGS
                                                        .EXTRN  ANLOBJ$_EXEHDRGBLIDENT
                                                        .EXTRN  ANLOBJ$_EXEHDRGST
                                                        .EXTRN  ANLOBJ$_EXEHDRIDENT
                                                        .EXTRN  ANLOBJ$_EXEHDRIMAGEID
                                                        .EXTRN  ANLOBJ$_EXEHDRISD
                                                        .EXTRN  ANLOBJ$_EXEHDRISDBASE
                                                        .EXTRN  ANLOBJ$_EXEHDRISDCOUNT
                                                        .EXTRN  ANLOBJ$_EXEHDRISDFLAGS
                                                        .EXTRN  ANLOBJ$_EXEHDRISDGBLNAM
                                                        .EXTRN  ANLOBJ$_EXEHDRISDNUM
                                                        .EXTRN  ANLOBJ$_EXEHDRISDPFCDEF
                                                        .EXTRN  ANLOBJ$_EXEHDRISDPFCSIZ
                                                        .EXTRN  ANLOBJ$_EXEHDRISDTYPE
                                                        .EXTRN  ANLOBJ$_EXEHDRISDVBN
                                                        .EXTRN  ANLOBJ$_EXEHDRLINKID
                                                        .EXTRN  ANLOBJ$_EXEHDRMATCH
                                                        .EXTRN  ANLOBJ$_EXEHDRNAME
                                                        .EXTRN  ANLOBJ$_EXEHDRNOPATCH
                                                        .EXTRN  ANLOBJ$_EXEHDRPAGECOUNT
```

```
07                      .EXTRN  ANLOBJ$_EXEHDRPAGEDEF
                        .EXTRN  ANLOBJ$_EXEHDRPATCH
                        .EXTRN  ANLOBJ$_EXEHDRPATCHDATE
                        .EXTRN  ANLOBJ$_EXEHDRPRIV
                        .EXTRN  ANLOBJ$_EXEHDRROPATCH
08                      .EXTRN  ANLOBJ$_EXEHDRRWPATCH
                        .EXTRN  ANLOBJ$_EXEHDRSYMDBG
                        .EXTRN  ANLOBJ$_EXEHDRSYSVER
09                      .EXTRN  ANLOBJ$_EXEHDRTEXTVBN
                        .EXTRN  ANLOBJ$_EXEHDRTIME
14                      .EXTRN  ANLOBJ$_EXEHDRTYPEEXE
15                      .EXTRN  ANLOBJ$_EXEHDRTYPELIM
                        .EXTRN  ANLOBJ$_EXEHDRUSERECO
                        .EXTRN  ANLOBJ$_EXEHDRXFER1
                        .EXTRN  ANLOBJ$_EXEHDRXFER2
                        .EXTRN  ANLOBJ$_EXEHDRXFER3
19                      .EXTRN  ANLOBJ$_EXEHEADING
                        .EXTRN  ANLOBJ$_EXEPATCH
                        .EXTRN  ANLOBJ$_FLAG, ANLOBJ$_HEXDATA
                        .EXTRN  ANLOBJ$_HEXHEADING1
                        .EXTRN  ANLOBJ$_HEXHEADING2
                        .EXTRN  ANLOBJ$_INDMSGSEC
                        .EXTRN  ANLOBJ$_INTERACT
                        .EXTRN  ANLOBJ$_MASK, ANLOBJ$_OBJCPRREC
                        .EXTRN  ANLOBJ$_OBJDBGREC
                        .EXTRN  ANLOBJ$_OBJENV, ANLOBJ$_OBJEOMFLAGS
                        .EXTRN  ANLOBJ$_OBJEOMREC
                        .EXTRN  ANLOBJ$_OBJEOMSEVABT
                        .EXTRN  ANLOBJ$_OBJEOMSEVERR
                        .EXTRN  ANLOBJ$_OBJEOMSEVIGN
                        .EXTRN  ANLOBJ$_OBJEOMSEVRES
                        .EXTRN  ANLOBJ$_OBJEOMSEVSUC
                        .EXTRN  ANLOBJ$_OBJEOMSEVWRN
                        .EXTRN  ANLOBJ$_OBJEOMWREC
                        .EXTRN  ANLOBJ$_OBJFADPASSMECH
                        .EXTRN  ANLOBJ$_OBJGSDENV
                        .EXTRN  ANLOBJ$_OBJGSDENVFLAGS
                        .EXTRN  ANLOBJ$_OBJGSDENVPAR
                        .EXTRN  ANLOBJ$_OBJGSDEPM
                        .EXTRN  ANLOBJ$_OBJGSDEPMW
                        .EXTRN  ANLOBJ$_OBJGSDIDC
                        .EXTRN  ANLOBJ$_OBJGSDIDCENT
                        .EXTRN  ANLOBJ$_OBJGSDIDCFLAGS
                        .EXTRN  ANLOBJ$_OBJGSDIDCMATCH
                        .EXTRN  ANLOBJ$_OBJGSDIDCOBJ
                        .EXTRN  ANLOBJ$_OBJGSDIDCVALA
                        .EXTRN  ANLOBJ$_OBJGSDIDCVALB
                        .EXTRN  ANLOBJ$_OBJGSDLEPM
                        .EXTRN  ANLOBJ$_OBJGSDLPRO
                        .EXTRN  ANLOBJ$_OBJGSDLSY
                        .EXTRN  ANLOBJ$_OBJGSDPRO
                        .EXTRN  ANLOBJ$_OBJGSDPROW
                        .EXTRN  ANLOBJ$_OBJGSDPSC
                        .EXTRN  ANLOBJ$_OBJGSDPSCALIGN
                        .EXTRN  ANLOBJ$_OBJGSDPSCALLOC
                        .EXTRN  ANLOBJ$_OBJGSDPSCBASE
                        .EXTRN  ANLOBJ$_OBJGSDPSCFLAGS
```

```
        .EXTRN  ANLOBJ$_OBJGSDREC
        .EXTRN  ANLOBJ$_OBJGSDSPSC
        .EXTRN  ANLOBJ$_OBJGSDSYM
        .EXTRN  ANLOBJ$_OBJGSDSYMW
        .EXTRN  ANLOBJ$_OBJGTXREC
        .EXTRN  ANLOBJ$_OBJHDRIGNREC
        .EXTRN  ANLOBJ$_OBJHEADING
        .EXTRN  ANLOBJ$_OBJLITINDEX
        .EXTRN  ANLOBJ$_OBJLNKREC
        .EXTRN  ANLOBJ$_OBJLNMREC
        .EXTRN  ANLOBJ$_OBJMHDCREATE
        .EXTRN  ANLOBJ$_OBJMHDNAME
        .EXTRN  ANLOBJ$_OBJMHDPATCH
        .EXTRN  ANLOBJ$_OBJMHDREC
        .EXTRN  ANLOBJ$_OBJMHDRECSIZ
        .EXTRN  ANLOBJ$_OBJMHDSTRLVL
        .EXTRN  ANLOBJ$_OBJMHDVERSION
        .EXTRN  ANLOBJ$_OBJMTCCORRECT
        .EXTRN  ANLOBJ$_OBJMTCINPUT
        .EXTRN  ANLOBJ$_OBJMTCNAME
        .EXTRN  ANLOBJ$_OBJMTCREC
        .EXTRN  ANLOBJ$_OBJMTCSEQNUM
        .EXTRN  ANLOBJ$_OBJMTCUIC
        .EXTRN  ANLOBJ$_OBJMTCVERSION
        .EXTRN  ANLOBJ$_OBJMTCWHEN
        .EXTRN  ANLOBJ$_OBJPROARGCOUNT
        .EXTRN  ANLOBJ$_OBJPROARGNUM
        .EXTRN  ANLOBJ$_OBJPSECT
        .EXTRN  ANLOBJ$_OBJSRCREC
        .EXTRN  ANLOBJ$_OBJSTATHEADING1
        .EXTRN  ANLOBJ$_OBJSTATHEADING2
        .EXTRN  ANLOBJ$_OBJSTATLINE
        .EXTRN  ANLOBJ$_OBJSTATTOTAL
        .EXTRN  ANLOBJ$_OBJSYMBOL
        .EXTRN  ANLOBJ$_OBJSYMFLAGS
        .EXTRN  ANLOBJ$_OBJTIRARGINDEX
        .EXTRN  ANLOBJ$_OBJTIRCMD
        .EXTRN  ANLOBJ$_OBJTIRCMDSTK
        .EXTRN  ANLOBJ$_OBJTBTREC
        .EXTRN  ANLOBJ$_OBJTIRREC
        .EXTRN  ANLOBJ$_OBJTIRSTOIM
        .EXTRN  ANLOBJ$_OBJTIRVIELD
        .EXTRN  ANLOBJ$_OBJTTLREC
        .EXTRN  ANLOBJ$_OBJVALUE
        .EXTRN  ANLOBJ$_OBJUVALUE
        .EXTRN  ANLOBJ$_PROTECTION
        .EXTRN  ANLOBJ$_SEVERITY
        .EXTRN  ANLOBJ$_TEXT, ANLOBJ$_TEXTHDR
        .EXTRN  ANLOBJ$_NOSUCHMOD
        .EXTRN  ANLOBJ$_BADDATE
        .EXTRN  ANLOBJ$_BADHDRBLKCOUNT
        .EXTRN  ANLOBJ$_BADSEVERITY
        .EXTRN  ANLOBJ$_BADSYM1ST
        .EXTRN  ANLOBJ$_BADSYMCHAR
        .EXTRN  ANLOBJ$_BADSYMLEN
        .EXTRN  ANLOBJ$_EXEBADFIXUPEND
        .EXTRN  ANLOBJ$_EXEBADFIXUPISD
```

```
                                .EXTRN  ANLOBJ$_EXEBADFIXUPVBN
                                .EXTRN  ANLOBJ$_EXEBADISDS1
44                              .EXTRN  ANLOBJ$_EXEBADISDTYPE
                                .EXTRN  ANLOBJ$_EXEBADMATCH
                                .EXTRN  ANLOBJ$_EXEBADPATCHLEN
54                              .EXTRN  ANLOBJ$_EXEBADOBJ
                                .EXTRN  ANLOBJ$_EXEBADTYPE
55                              .EXTRN  ANLOBJ$_EXEBADXFERO
                                .EXTRN  ANLOBJ$_EXEHDRISDLONG
56                              .EXTRN  ANLOBJ$_EXEHDRLONG
                                .EXTRN  ANLOBJ$_EXEISDLENDZRO
57                              .EXTRN  ANLOBJ$_EXEISDLENGBL
58                              .EXTRN  ANLOBJ$_EXEISDLENPRIV
                                .EXTRN  ANLOBJ$_EXENOTNATIVE
                                .EXTRN  ANLOBJ$_EXTRABYTES
56                              .EXTRN  ANLOBJ$_FIELDFIT
                                .EXTRN  ANLOBJ$_FLAGERROR
                                .EXTRN  ANLOBJ$_NOTOK, ANLOBJ$_OBJBADIDCMATCH
64                              .EXTRN  ANLOBJ$_OBJBADNUM
65                              .EXTRN  ANLOBJ$_OBJBADPOP
66                              .EXTRN  ANLOBJ$_OBJBADPUSH
                                .EXTRN  ANLOBJ$_OBJBADTYPE
                                .EXTRN  ANLOBJ$_OBJBADVIELD
                                .EXTRN  ANLOBJ$_OBJEOMBADSEV
                                .EXTRN  ANLOBJ$_OBJEOMMISSING
67                              .EXTRN  ANLOBJ$_OBJFADBADAVC
                                .EXTRN  ANLOBJ$_OBJFADBADRBC
                                .EXTRN  ANLOBJ$_OBJGSDBADALIGN
                                .EXTRN  ANLOBJ$_OBJGSDBADSUBTYP
68                              .EXTRN  ANLOBJ$_OBJHDRRES
73                              .EXTRN  ANLOBJ$_OBJMHDBADRECSIZ
                                .EXTRN  ANLOBJ$_OBJMHDBADSTRLVL
                                .EXTRN  ANLOBJ$_OBJMHDMISSING
                                .EXTRN  ANLOBJ$_OBJNONTIRCMD
                                .EXTRN  ANLOBJ$_OBJNOPSC
                                .EXTRN  ANLOBJ$_OBJNULLREC
                                .EXTRN  ANLOBJ$_OBJPOSPACE
                                .EXTRN  ANLOBJ$_OBJPROMINMAX
                                .EXTRN  ANLOBJ$_OBJPSCABSLEN
                                .EXTRN  ANLOBJ$_OBJRECTOOBIG
                                .EXTRN  ANLOBJ$_OBJTIRRES
                                .EXTRN  ANLOBJ$_OBJUNDEFENV
                                .EXTRN  ANLOBJ$_OBJUNDEFLIT
                                .EXTRN  ANLOBJ$_OBJUNDEFPSC
                                .EXTRN  ANALYZE$_FACILITY
                                .EXTRN  ANL$OBJECT_POSITIONALS
                                .EXTRN  CLI$GET_VALUE, LBR$CLOSE
                                .EXTRN  LBR$GET_INDEX, LBR$GET_RECORD
                                .EXTRN  LBR$INI_CONTROL
                                .EXTRN  LBR$LOOKUP_KEY, LBR$OPEN
                                .EXTRN  LIB$FIND_FILE, LIB$FREE_VM
                                .EXTRN  LIB$GET_VM, STR$TRIM

                                .PSECT  $CODE$,NOWRT,2

003C 00000 ADD_MODULE_TO_LIST:
                                .WORD   Save R2,R3,R4,R5                        ; 0603
```

```
                                                 51       04    AC  D0 00002          MOVL    MODULE_NAME, R1                                  ; 0606
                                          50     0000'  CF          1F  C5 00006          MULL3   #31, MODULE_LIST_SIZE, R0                ; 0611
                                    1F    20       04   B1          6'  2C 0000C          MOVC5   (R1), @4(R1), #32, #31, @MODULE_LIST[R0] ;
                                                                0000'DF40      00012                                                           ;
                                                                0000'  CF  D6 00016          INCL    MODULE_LIS'_SIZE                         ; 0615
                                                 50             01  D0 0001A          MOVL    #1, R0                                   ; 0617
                                                                    04 0001D          RET                                              ; 0619

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  189          0620  2 ! If the wildcard context is zero, it means this is the first call, or
;  190          0621  2 ! we finished with a file spec on the previous call.  So we must obtain
;  191          0622  2 ! the next file spec from the command line.
;  192          0623  2
;  193          0624  3 if .get_new_spec then (
;  194          0625  3         wildcard_spec[len] = nam$c_maxrss;
;  195          0626  3         status = cli$get_value(describe('file_specs'),wildcard_spec);
;  196          0627  3
;  197          0628  3         ! If there are no more specs, we are all done.
;  198          0629  3
;  199          0630  3         if not .status then
;  200          0631  3                 return false;
;  201          0632  3         str$trim(wildcard_spec,wildcard_spec,wildcard_spec);
;  202          0633  3
;  203          0634  3         ! Call a routine to process any positional qualifiers for this spec.
;  204          0635  3         ! We don't know how to do that.
;  205          0636  3
;  206          0637  3         anl$object_positionals();
;  207          0638  3
;  208          0639  3         ! Now we have to find out if this new spec has a /INCLUDE qualifier
;  209          0640  3         ! attached to it, meaning that it is an object library.  If it is,
;  210          0641  3         ! then the module list will be filled in with the module names.
;  211          0642  3
;  212          0643  3         status = lib$get_vm(%ref(1000*obj$c_symsiz),module_list);
;  213          0644  3         check (.status, .status);
;  214          0645  3         object_library = anl$object_include(all_modules,module_list_size,.module_list);
;  215          0646  3         module_list_index = -1;
;  216          0647  2 );
;  217          0648  2
;  218          0649  2 ! On the other hand, if the previous call done is true, we may have just
;  219          0650  2 ! finished processing a file.  This is only true if we're processing
;  220          0651  2 ! an individual object file.  Better close it.
;  221          0652  2
;  222          0653  3 if (.object_fab[fab$w_ifi] nequ 0) then (
;  223          0654  3         status = $close(fab=object_fab);
;  224          0655  3         check (.status, anlobj$_closein,1,resultant_spec,.status,.object_fab[fab$l_stv]);
;  225          0656  2 );
```

```
;  227          0657  2 ! We have obtained a wildcard spec from the file parameter, and any associated
;  228          0658  2 ! positional qualifiers.
;  229          0659  2
;  230          0660  2 ! Now we have to see if we need to find the next file that matches the
;  231          0661  2 ! current file spec.  This is always the case if it's not an object library.
;  232          0662  2 ! If it is, then we must search if this is the first time or we finished
;  233          0663  2 ! with the previous library.
;  234          0664  2
;  235          0665  2 if not .object_library or
;  236          0666  3    (.object_library and .module_list_index eqlu -1) then (
;  237          0667  3         resultant_spec[len] = nam$c_maxrss;
;  238          0668  3         status = lib$find_file(wildcard_spec,resultant_spec,wildcard_context,
;  239          0669  3                                (if .object_library then describe('.OLB') else describe('.OBJ')),
;  240          0670  3                                0, stv, %ref(2) );
;  241          0671  3         str$trim(resultant_spec,resultant_spec,resultant_spec);
;  242          0672  3
;  243          0673  3         ! If we failed to find a file, then free up the module list, reset
;  244          0674  3         ! the wildcard context, and call ourselves recursively to do the
;  245          0675  3         ! next file spec.  Also give an error, unless we just plain ran
;  246          0676  3         ! out of files.
;  247          0677  3
;  248          0678  4         if not .status then (
;  249          0679  4                 if .status nequ rms$_nmf then
;  250          0680  4                         signal (anlobj$_openin,1,resultant_spec,.stv);
;  251          0681  4                 status = lib$free_vm(%ref(1000*obj$c_symsiz),module_list);
;  252          0682  4                 check (.status, .status);
;  253          0683  4                 get_new_spec = true;
;  254          0684  4                 return anl$open_next_object_file(opened_spec_dsc);
;  255          0685  3         );
;  256          0686  3
;  257          0687  3         ! Hey, we got a file spec.  Open the library or file, as appropriate.
;  258          0688  3
;  259          0689  3         get_new_spec = false;
;  260          0690  4         if .object_library then (
;  261          0691  4                 status = lbr$ini_control(library_index,%ref(lbr$c_read),%ref(lbr$c_typ_obj)),
;  262          0692  4                 check (.status, .status);
;  263          0693  4                 status = lbr$open(library_index,resultant_spec ;
;  264          0694  4                 check (.status, anlobj$_openin,1,resultant_spec,.status);
;  265          0695  4         ) else (
;  266          0696  4                 object_fab[fab$b_fns] = .resultant_spec[len];
;  267          0697  4                 object_fab[fab$l_fna] = .resultant_spec[ptr];
;  268          0698  4                 status = $open(fab=object_fab);
;  269          0699  4                 check (.status, anlobj$_openin,1,resultant_spec,.status,.object_fab[fab$l_stv]);
;  270          0700  5                 if .status then (
;  271          0701  5                         status = $connect(rab=object_rab);
;  272          0702  5                         check (.status, anlobj$_openin,1,resultant_spec,.status,.object_rab[rab$l_stv]);
;  273          0703  4                 );
;  274          0704  3         );
;  275          0705  3
;  276          0706  3         ! If the open failed, then we need to recurse to try the next file.
;  277          0707  3
;  278          0708  3         if not .status then
;  279          0709  3                 return anl$open_next_object_file(opened_spec_dsc);
;  280          0710  3
;  281          0711  3         ! If this is an object library, it may be the case that our call
;  282          0712  3         ! to ANL$OBJECT_INCLUDE told us that the user wanted all modules.
;  283          0713  3         ! If so, let's ask the librarian for them; it will call ADD_MODULE_
```

```
;  284          0714  3          ! TO_LIST for each one.
;  285          0715  3
;  286          0716  4          if .object_library and .all_modules then (
;  287          0717  4                  module_list_size = 0;
;  288          0718  4                  status = lbr$get_index(library_index,%ref(1),add_module_to_list);
;  289          0719  4                  check (.status, .status);
;  290          0720  3          );
;  291          0721  2 );
```

```
;  293          0722  2 ! We know that we have a good file opened.
;  294          0723  2
;  295          0724  2 ! OK, now we may be processing an object library.  If so, we have to
;  296          0725  2 ! get the next module name out of the module list and prepare to read it.
;  297          0726  2
;  298          0727  3 if .object_library then (
;  299          0728  3         local
;  300          0729  3                 module_name_dsc: descriptor,
;  301          0730  3                 module_text_rfa: block[8,byte];
;  302          0731  3
;  303          0732  3         increment (module_list_index);
;  304          0733  3
;  305          0734  3         ! If we are at the end of the list, then call ourselves recursively
;  306          0735  3         ! to process the next file matching the current spec.  But first we
;  307          0736  3         ! must close the library we just finished.
;  308          0737  3
;  309          0738  4         if .module_list_index eqlu .module_list_size then (
;  310          0739  4                 status = lbr$close(library_index);
;  311          0740  4                 check (.status,anlobj$_closein,1,resultant_spec,.status);
;  312          0741  4                 module_list_index = -1;
;  313          0742  4                 return anl$open_next_object_file(opened_spec_dsc);
;  314          0743  3         );
;  315          0744  3
;  316          0745  3         ! Prepare to read the next module.  If it isn't in the library,
;  317          0746  3         ! recurse to try the next one.
;  318          0747  3
;  319          0748  3         module_name_dsc[0,0,32,0] = obj$c_symsiz;
;  320          0749  3         module_name_dsc[ptr] = module_list[.module_list_index,0,0,0,0];
;  321          0750  3         str$trim(module_name_dsc,module_name_dsc,module_name_dsc);
;  322          0751  3         status = lbr$lookup_key(library_index,module_name_dsc,module_text_rfa);
;  323          0752  4         if not .status then (
;  324          0753  4                 check (.status, anlobj$_nosuchmod,2,resultant_spec,module_name_dsc);
;  325          0754  4                 return anl$open_next_object_file(opened_spec_dsc);
;  326          0755  3         );
;  327          0756  2 );
;  328          0757  2
;  329          0758  2 ! Finally, we have to build the real resultant file spec.  If we're not
;  330          0759  2 ! doing an object library, then we already have it.  If we are, we have
;  331          0760  2 ! to append the module name.  In all cases, trim trailing blanks.
;  332          0761  2
;  333          0762  2 str$trim(opened_spec_dsc,resultant_spec,opened_spec_dsc);
;  334          0763  3 if .object_library then (
;  335          0764  3         ch$copy(1,uplit byte(' '),obj$c_symsiz,module_list[.module_list_index,0,0,0,0],
;  336          0765  3                 ' ',obj$c_symsiz+1,.opened_spec_dsc[ptr]+.opened_spec_dsc[len]);
;  337          0766  3         opened_spec_dsc[len] = .opened_spec_dsc[len] + obj$c_symsiz+1;
;  338          0767  3         str$trim(opened_spec_dsc,opened_spec_dsc,opened_spec_dsc);
;  339          0768  2 );
;  340          0769  2
;  341          0770  2 return true;
;  342          0771  2
;  343          0772  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                73  63  65  70  73  5F  65  6C  69  66  00000 P.AAB:  .ASCII  \file_specs\
```

```
                                                0000A                   .BLKB   2
                                     0000000A   0000C P.AAA:           .LONG   10                                              ;
                                     00000000'  00010                   .ADDRESS P.AAB                                          ;
                                  42 4C 4F 2E   00014 P.AAD:           .ASCII  \.OLB\                                          ;
                                     00000004   00018 P.AAC:           .LONG   4                                               ;
                                     00000000'  0001C                   .ADDRESS P.AAD                                          ;
                                  4A 42 4F 2E   00020 P.AAF:           .ASCII  \.OBJ\                                          ;
                                     00000004   00024 P.AAE:           .LONG   4                                               ;
                                     00000000'  00028                   .ADDRESS P.AAF                                          ;
                                           20   0002C P.AAG:           .ASCII  \ \                                             ;

                                                                       .EXTRN  SYS$CLOSE, SYS$OPEN
                                                                       .EXTRN  SYS$CONNECT

                                                                       .PSECT  $CODE$,NOWRT,2

                                         0FFC 00000                    .ENTRY  ANL$OPEN_NEXT_OBJECT_FILE, Save R2,R3,R4,-      ; 0582
                                                                               R5,R6,R7,R8,R9,R10,R11                          ;
                                5B 00000000G   00   9E 00002           MOVAB   STR$TRIM, R11                                   ;
                                5A 00000000G   00   9E 00009           MOVAB   LIB$SIGNAL, R10                                 ;
                                59      0000'  CF   9E 00010           MOVAB   RESULTANT_SPEC, R9                              ;
                                5E             1C   C2 00015           SUBL2   #28, SP                                         ;
                                57             04   AC   D0 00018       MOVL    OPENED_SPEC, R7                                 ; 0585
                                6C          0AC4    C9   E9 0001C       BLBC    GET_NEW_SPEC, 3$                                ; 0624
                          09A8  C9             FF   8F   9B 00021       MOVZBW  #255, WILDCARD_SPEC                             ; 0625
                                             09A8   C9   9F 00027       PUSHAB  WILDCARD_SPEC                                   ; 0626
                                             0000'  CF   9F 0002B       PUSHAB  P.AAA                                           ;
                   00000000G  00                    02   FB 0002F       CALLS   #2, CLI$GET_VALUE                               ;
                              52                    50   D0 00036       MOVL    R0, STATUS                                      ;
                              03                    52   E8 00039       BLBS    STATUS, 1$                                      ; 0630
                                                 02A8   31 0003C       BRW     24$                                             ;
                                          09A8   C9   9F 0003F 1$:     PUSHAB  WILDCARD_SPEC                                   ; 0632
                                          09A8   C9   9F 00043         PUSHAB  WILDCARD_SPEC                                   ;
                                          09A8   C9   9F 00047         PUSHAB  WILDCARD_SPEC                                   ;
                                6B                    03   FB 0004B       CALLS   #3, STR$TRIM                                    ;
                            0000G  CF                 00   FB 0004E       CALLS   #0, ANL$OBJECT_POSITIONALS                      ; 0637
                                          0AB8   C9   9F 00053         PUSHAB  MODULE_LIST                                     ; 0643
                                08     AE    7918   8F   3C 00057       MOVZWL  #31000, 8(SP)                                   ;
                                         08     AE   9F   0005D         PUSHAB  8(SP)                                           ;
                   00000000G  00                    02   FB 00060       CALLS   #2, LIB$GET_VM                                  ;
                              52                    50   D0 00067       MOVL    R0, STATUS                                      ;
                              05                    52   E8 0006A       BLBS    STATUS, 2$                                      ; 0644
                                                    52   DD 0006D       PUSHL   STATUS                                          ;
                                6A                  01   FB 0006F       CALLS   #1, LIB$SIGNAL                                  ;
                                          0AB8   C9   DD 00072 2$:     PUSHL   MODULE_LIST                                     ; 0645
                                          0ABC   C9   9F 00076         PUSHAB  MODULE_LIST_SIZE                                ;
                                          0AB4   C9   9F 0007A         PUSHAB  ALL_MODULES                                     ;
                            0000V  CF                 03   FB 0007E       CALLS   #3, ANL$OBJECT_INCLUDE                          ;
                             0107  C9                 50   90 00083       MOVB    R0, OBJECT_LIBRARY                              ;
                             0AC0  C9                 01   CE 00088       MNEGL   #1, MODULE_LIST_INDEX                           ; 0646
                                          010E   C9   B5 0008D 3$:     TSTW    OBJECT_FAB+2                                    ; 0653
                                                    24   13 00091       BEQL    4$                                              ;
                                          010C   C9   9F 00093         PUSHAB  OBJECT_FAB                                      ; 0654
                   00000000G  00                    01   FB 00097       CALLS   #1, SYS$CLOSE                                   ;
                              52                    50   D0 0009E       MOVL    R0, STATUS                                      ;
                              13                    52   E8 000A1       BLBS    STATUS, 4$                                      ; 0655
                                          0118   C9   DD 000A4         PUSHL   OBJECT_FAB+12                                   ;
```

```
                                    52  DD 000A8          PUSHL   STATUS                                          ;
                                    59  DD 000AA          PUSHL   R9                                              ;
                                    01  DD 000AC          PUSHL   #1                                              ;
                          00B11052  8F  DD 000AE          PUSHL   #11604050                                       ;
                         6A         05  FB 000B4          CALLS   #5, LIB$SIGNAL                                  ;
                         50   0107  C9  9A 000B7 4$:      MOVZBL  OBJECT_LIBRARY, R0                              ; 0665
                         0E         50  E9 000BC          BLBC    R0, 5$                                          ;
                FFFFFFFF 8F   0AC0  C9  D1 000BF          CMPL    MODULE_LIST_INDEX, #-1                          ; 0666
                                    03  13 000C8          BEQL    5$                                              ;
                                015A  31 000CA          BRW     16$                                             ;
                         69     FF  8F  9B 000CD 5$:      MOVZBW  #255, RESULTANT_SPEC                            ; 0667
                      04 AE         02  D0 000D1          MOVL    #2, 4(SP)                                       ; 0670
                            04  AE  9F  000D5          PUSHAB  4(SP)                                           ;
                            0C  AE  9F  000D8          PUSHAB  STV                                             ; 0668
                                7E  D4  000DB          CLRL    -(SP)                                           ;
                         07         50  E9 000DD          BLBC    R0, 6$                                          ; 0669
                         50   0000' CF  9E 000E0          MOVAB   P.AAC, R0                                       ;
                                    05  11 000E5          BRB     7$                                              ;
                         50   0000' CF  9E 000E7 6$:      MOVAB   P.AAE, R0                                       ;
                                    50  DD 000EC 7$:      PUSHL   R0                                              ;
                              0AB0  C9  9F 000EE          PUSHAB  WILDCARD_CONTEXT                                ; 0668
                                    59  DD 000F2          PUSHL   R9                                              ;
                              09A8  C9  9F 000F4          PUSHAB  WILDCARD_SPEC                                   ;
                00000000G     00 ..     07  FB 000F8          CALLS   #7, LIB$FIND_FILE                               ;
                         52         50  D0 000FF          MOVL    R0, STATUS                                      ;
                                    59  DD 00102          PUSHL   R9                                              ; 0671
                                    59  DD 00104          PUSHL   R9                                              ;
                                    59  DD 00106          PUSHL   R9                                              ;
                         6B         03  FB 00108          CALLS   #3, STR$TRIM                                    ;
                         40         52  E8 0010B          BLBS    STATUS, 10$                                     ; 0678
                000182CA 8F         52  D1 0010E          CMPL    STATUS, #99018                                  ; 0679
                                    10  13 00115          BEQL    8$                                              ;
                              08  AE  DD 00117          PUSHL   STV                                             ; 0680
                                    59  DD 0011A          PUSHL   R9                                              ;
                                    01  DD 0011C          PUSHL   #1                                              ;
                          00B1109A  8F  DD 0011E          PUSHL   #11604122                                       ;
                         6A         04  FB 00124          CALLS   #4, LIB$SIGNAL                                  ;
                              0AB8  C9  9F 00127 8$:      PUSHAB  MODULE_LIST                                     ; 0681
                      08 AE   7918  8F  3C 0012B          MOVZWL  #31000, 8(SP)                                   ;
                            08  AE  9F  00131          PUSHAB  8(SP)                                           ;
                00000000G     00        02  FB 00134          CALLS   #2, LIB$FREE_VM                                 ;
                         52         50  D0 0013B          MOVL    R0, STATUS                                      ;
                         05         52  E8 0013E          BLBS    STATUS, 9$                                      ; 0682
                                    52  DD 00141          PUSHL   STATUS                                          ;
                         6A         01  FB 00143          CALLS   #1, LIB$SIGNAL                                  ;
                      0AC4 C9       01  D0 00146 9$:      MOVL    #1, GET_NEW_SPEC                                ; 0683
                                015A  31 00148          BRW     21$                                             ; 0684
                              0AC4  C9  D4 0014E 10$:     CLRL    GET_NEW_SPEC                                    ; 0689
                         48   0107  C9  E9 00152          BLBC    OBJECT_LIBRARY, 12$                             ; 0690
                      04 AE         01  D0 00157          MOVL    #1, 4(SP)                                       ; 0691
                            04  AE  9F  0015B          PUSHAB  4(SP)                                           ;
                      04 AE         01  D0 0015E          MOVL    #1, 4(SP)                                       ;
                            04  AE  9F  00162          PUSHAB  4(SP)                                           ;
                              0108  C9  9F 00165          PUSHAB  LIBRARY_INDEX                                   ;
                00000000G     00        03  FB 00169          CALLS   #3, LBR$INI_CONTROL                             ;
                         52         50  D0 00170          MOVL    R0, STATUS                                      ;
                         05         52  E8 00173          BLBS    STATUS, 11$                                     ;
```

```
                                52  DD 00176          PUSHL   STATUS
                            6A      01  FB 00178          CALLS   #1, LIB$SIGNAL
                                59  DD 0017B 11$:      PUSHL   R9                                      ;0693
                          0108  C9  9F 0017D          PUSHAB  LIBPARY_INDEX
                00000000G   00      02  FB 00181          CALLS   #2, LBR$OPEN
                            52      50  D0 00188          MOVL    R0, STATUS
                            6A      52  E8 0018B          BLBS    STATUS, 15$                             ;0694
                                52  DD 0018E          PUSHL   STATUS
                                59  DD 00190          PUSHL   R9
                                01  DD 00192          PUSHL   #1
                        00B1109A    8F  DD 00194          PUSHL   #11604122
                            6A      04  FB 0019A          CALLS   #4, LIB$SIGNAL
                                56  11  0019D          BRB     14$                                     ;0690
                    0140    C9      69  90 0019F 12$:     MOVB    RESULTANT_SPEC, OBJECT_FAB+52           ;0696
                    0138    C9  04      A9  D0 001A4          MOVL    RESULTANT_SPEC+4, OBJECT_FAB+44         ;0697
                          010C  C9  9F 001AA          PUSHAB  OBJECT_FAB                              ;0698
                00000000G   00      01  FB 001AE          CALLS   #1, SYS$OPEN
                            52      50  D0 001B5          MOVL    R0, STATUS
                            16      52  E8 001B8          BLBS    STATUS, 13$                             ;0699
                          0118  C9  DD 001BB          PUSHL   OBJECT_FAB+12
                                52  DD 001BF          PUSHL   STATUS
                                59  DD 001C1          PUSHL   R9
                                01  DD 001C3          PUSHL   #1
                        00B1109A    8F  DD 001C5          PUSHL   #11604122
                            6A      05  FB 001CB          CALLS   #5, LIB$SIGNAL
                            24      52  E9 001CE          BLBC    STATUS, 14$                             ;0700
                          0964  C9  9F 001D1 13$:     PUSHAB  OBJECT_RAB                              ;0701
                00000000G   00      01  FB 001D5          CALLS   #1, SYS$CONNECT
                            52      50  D0 001DC          MOVL    R0, STATUS
                            16      52  E8 001DF          BLBS    STATUS, 15$                             ;0702
                          0970  C9  DD 001E2          PUSHL   OBJECT_RAB+12
                                52  DD 001E6          PUSHL   STATUS
                                59  DD 001E8          PUSHL   R9
                                01  DD 001EA          PUSHL   #1
                        00B1109A    8F  DD 001EC          PUSHL   #11604122
                            6A      05  FB 001F2          CALLS   #5, LIB$SIGNAL
                            69      52  E9 001F5 14$:     BLBC    STATUS, 19$                             ;0708
                            2A  0107    C9  E9 001F8 15$:     BLBC    OBJECT_LIBRARY, 16$                     ;0716
                            25  0AB4    C9  E9 001FD          BLBC    ALL_MODULES, 16$
                              0ABC  C9  D4 00202          CLRL    MODULE_LIST_SIZE                        ;0717
                              FDD8  CF  9F 00206          PUSHAB  ADD_MODULE_TO_LIST                      ;0718
                        08    AE      01  D0 0020A          MOVL    #1, 8(SP)
                                08  AE  9F 0020E          PUSHAB  8(SP)
                          0108  C9  9F 00211          PUSHAB  LIBRARY_INDEX
                00000000G   00      03  FB 00215          CALLS   #3, LBR$GET_INDEX
                            52      50  D0 0021C          MOVL    R0, STATUS
                            05      52  E8 0021F          BLBS    STATUS, 16$                             ;0719
                                52  DD 00222          PUSHL   STATUS
                            6A      01  FB 00224          CALLS   #1, LIB$SIGNAL
                            03  0107    C9  E8 00227 16$:     BLBS    OBJECT_LIBRARY, 17$                     ;0727
                                0080 31  0022C          BRW     22$
                              0AC0  C9  D6 0022F 17$:     INCL    MODULE_LIST_INDEX                       ;0732
                        0ABC    C9  0AC0  C9  D1 00233          CMPL    MODULE_LIST_INDEX, MODULE_LIST_SIZE     ;0738
                                27  12  0023A          BNEQ    20$
                          0108  C9  9F 0023C          PUSHAB  LIBRARY_INDEX                           ;0739
                00000000G   00      01  FB 00240          CALLS   #1, LBR$CLOSE
                            52      50  D0 00247          MOVL    R0, STATUS
```

```
                                0F              52  E8 0024A          BLBS    STATUS, 18$                                     ; 0740
                                                52  DD 0024D          PUSHL   STATUS                                          ;
                                                59  DD 0024F          PUSHL   R9                                              ;
                                                01  DD 00251          PUSHL   #1                                              ;
                                    00B11052    8F  DD 00253          PUSHL   #11604050                                       ;
                                6A              04  FB 00259          CALLS   #4, LIB$SIGNAL                                  ;
                          0AC0  C9              01  CE 0025C 18$:      MNEGL   #1, MODULE_LIST_INDEX                           ; 0741
                                                44  11 00261 19$:      BRB     21$                                             ; 0742
                            14  AE              1F  D0 00263 20$:      MOVL    #31, MODULE_NAME_DSC                            ; 0748
                    50      0AC0  C9            1F  C5 00267          MULL3   #31, MODULE_LIST_INDEX, R0                      ; 0749
                            18  AE      0AB8 D940  9E 0026D          MOVAB   @MODULE_LIST[R0], MODULE_NAME_DSC+4             ;
                                                14  AF  9F 00274      PUSHAB  MODULE_NAME_DSC                                 ; 0750
                                                 8  AE  9F 00277      PUSHAB  MODULE_NAME_DSC                                 ;
                                                1C  AE  9F 0027A      PUSHAB  MODULE_NAME_DSC                                 ;
                                6B                  03  FB 0027D      CALLS   #3, STR$TRIM                                    ;
                                                0C  AE  9F 00280      PUSHAB  MODULE_TEXT_RFA                                 ; 0751
                                                18  AE  9F 00283      PUSHAB  MODULE_NAME_DSC                                 ;
                                              0108  C9  9F 00286      PUSHAB  LIBRARY_INDEX                                   ;
                        00000000G   00              03  FB 0028A      CALLS   #3, LBR$LOOKUP_KEY                              ;
                                52                  50  D0 00291      MOVL    R0, STATUS                                      ;
                                18                  52  E8 00294      BLBS    STATUS, 22$                                     ; 0752
                                                14  AE  9F 00297      PUSHAB  MODULE_NAME_DSC                                 ; 0753
                                                59  DD 0029A          PUSHL   R9                                              ;
                                                02  DD 0029C          PUSHL   #2                                              ;
                                    00000000G   8F  DD 0029E          PUSHL   #ANLOBJ$_NOSUCHMOD                              ;
                                6A              04  FB 002A4          CALLS   #4, LIB$SIGNAL                                  ;
                                                57  DD 002A7 21$:      PUSHL   R7                                              ; 0754
                          FD52  CF              01  FB 002A9          CALLS   #1, ANL$OPEN_NEXT_OBJECT_FILE                   ;
                                                    04 002AE          RET                                                     ;
                                                57  DD 002AF 22$:      PUSHL   R7                                              ; 0762
                                              0280  8F  BB 002B1      PUSHR   #^M<R7,R9>                                      ;
                                6B                  03  FB 002B5      CALLS   #3, STR$TRIM                                    ;
                                26            0107  C9  E9 002B8      BLBC    OBJECT_LIBRARY, 23$                             ; 0763
                    56      0AC0  C9              1F  C5 002BD      MULL3   #31, MODULE_LIST_INDEX, R6                      ; 0764
                                58                  67  3C 002C3      MOVZWL  (R7), R8                                        ; 0765
                                58            04    A7  C0 002C6      ADDL2   4(R7), R8                                       ;
                    68      0000'  CF               01  28 002CA      MOVC3   #1, P.AAG, (R8)                                 ;
                    63      0AB8 D946               1F  28 002D0      MOVC3   #31, @MODULE_LIST[R6], (R3)                     ;
                                67                  20  A0 002D7      ADDW2   #32, (R7)                                       ; 0766
                                                    57  DD 002DA      PUSHL   R7                                              ; 0767
                                                    57  DD 002DC      PUSHL   R7                                              ;
                                                    57  DD 002DE      PUSHL   R7                                              ;
                                6B                  03  FB 002E0      CALLS   #3, STR$TRIM                                    ;
                                50                  01  D0 002E3 23$:      MOVL    #1, R0                                      ; 0770
                                                    04 002E6          RET                                                     ;
                                                    50  D4 002E7 24$:      CLRL    R0                                          ; 0772
                                                    04 002E9          RET                                                     ;

; Routine Size:  746 bytes,    Routine Base:  $CODE$ + 001E
```

```
;  345          0773  1 %sbttl 'ANL$OBJECT_INCLUDE - Process a /INCLUDE Qualifier'
;  346          0774  1 !++
;  347          0775  1 ! Functional Description:
;  348          0776  1 !       This routine is called to process the /INCLUDE qualifier that
;  349          0777  1 !       might be attached to a file spec.  We need to return a list of
;  350          0778  1 !       the module names in the qualifier.
;  351          0779  1 !
;  352          0780  1 ! Formal Parameters:
;  353          0781  1 !       all             Address of a byte to set if the user wants all
;  354          0782  1 !                       modules analyzed.
;  355          0783  1 !       list_size       Address of a longword in which we return the size
;  356          0784  1 !                       of the module list.
;  357          0785  1 !       list            Address of a vector of blocks in which we
;  358          0786  1 !                       place the list.
;  359          0787  1 !
;  360          0788  1 ! Implicit Inputs:
;  361          0789  1 !       global data
;  362          0790  1 !
;  363          0791  1 ! Implicit Outputs:
;  364          0792  1 !       global data
;  365          0793  1 !
;  366          0794  1 ! Returned Value:
;  367          0795  1 !       True if there is a /INCLUDE qualifier, false if not.
;  368          0796  1 !
;  369          0797  1 ! Side Effects:
;  370          0798  1 !
;  371          0799  1 !--
;  372          0800  1
;  373          0801  1
;  374          0802  2 global routine anl$object_include(all,list_size,list) = begin
;  375          0803  2
;  376          0804  2 bind
;  377          0805  2         all_modules = .all: byte,
;  378          0806  2         module_list_size = .list_size: long,
;  379          0807  2         module_list = .list: blockvector[,obj$c_symsiz,byte];
;  380          0808  2
;  381          0809  2 local
;  382          0810  2         status: long;
;  383          0811  2 local
;  384          0812  2         local_described_buffer(module_name,obj$c_symsiz);
;  385          0813  2
;  386          0814  2
;  387          0815  2 ! Try to get the first module name.  If there is no qualifier, then
;  388          0816  2 ! just return false.
;  389          0817  2
;  390          0818  2 status = cli$get_value(describe('include'),module_name);
;  391          0819  2 if not .status then
;  392          0820  2         return false;
;  393          0821  2
;  394          0822  2 ! If the first name is an asterisk, then the user wants all modules.
;  395          0823  2
;  396          0824  3 if ch$eql(.module_name[len],.module_name[ptr], 1,uplit byte('*'),' ') then (
;  397          0825  3         all_modules = true;
;  398          0826  3         return true;
;  399          0827  2 ) else
;  400          0828  2         all_modules = false;
;  401          0829  2
```

```
;  402          0830  2 ! Now we loop for each module name and add it to the list.
;  403          0831  2
;  404          0832  2 module_list_size = 0;
;  405          0833  3 do (
;  406          0834  3         ch$move(obj$c_symsiz,.module_name[ptr], module_list[.module_list_  ~*,0,0,0,0]);
;  407          0835  3         increment (module_list_size);
;  408          0836  3         status = cli$get_value(describe('include'),module_name);
;  409          0837  2 ) until not .status;
;  410          0838  2
;  411          0839  2 return true;
;  412          0840  2
;  413          0841  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                65  64  75  6C  63  6E  69  0002D P.AAI:  .ASCII  \include\                                    ;
                                                  00000007  00034 P.AAH:  .LONG   7                                          ;
                                                  00000000' 00038         .ADDRESS P.AAI                                     ;
                                                        2A  0003C P.AAJ:  .ASCII  \*\                                        ;
                                65  64  75  6C  63  6E  69  0003D P.AAL:  .ASCII  \include\                                   ;
                                                  00000007  00044 P.AAK:  .LONG   7                                          ;
                                                  00000000' 00048         .ADDRESS P.AAL                                     ;


                                                                .PSECT  $CODE$,NOWRT,2
                                                    00FC  00000         .ENTRY  ANL$OBJECT_INCLUDE, Save R2,R3,R4,R5,R6,R7  ; 0802
                                    57 00000000G    00  9E  00002       MOVAB   CLI$GET_VALUE, R7                          ;
                                    5E              24  C2  00009       SUBL2   #36, SP                                    ;
                                                    1F  DD  0000C       PUSHL   #31                                        ; 0812
                                04  AE      08      AE  9E  0000E       MOVAB   MODULE_NAME+8, MODULE_NAME+4               ;
                                                    5E  DD  00013       PUSHL   SP                                         ; 0818
                                        0000'       CF  9F  00015       PUSHAB  P.AAH                                      ;
                                    67              02  FB  00019       CALLS   #2, CLI$GET_VALUE                          ;
                                    56              50  D0  0001C       MOVL    R0, STATUS                                 ;
                                    39              56  E9  0001F       BLBC    STATUS, 4$                                 ; 0819
01                  20  04  BE                      6E  2D  00022       CMPC5   MODULE_NAME, @MODULE_NAME+4, #32, #1, P.AAJ ; 0824
                                        0000'       CF      00028       ;
                                                    06  12  0002B       BNEQ    1$                                         ;
                                04  BC              01  90  0002D       MOVB    #1, @ALL                                   ; 0825
                                                    24  11  00031       BRB     3$                                         ; 0826
                                      04  BC        94  00033 1$:       CLRB    @ALL                                       ; 0828
                                      08  BC        D4  00036           CLRL    @LIST_SIZE                                 ; 0832
                                50    08  BC        1F  C5  00039 2$:    MULL3   #31, @LIST_SIZE, R0                        ; 0834
                             0C BC40  04  BE        1F  28  0003E       MOVC3   #31, @MODULE_NAME+4, @LIST[R0]             ;
                                      08  BC        D6  00045           INCL    @LIST_SIZE                                 ; 0835
                                                    5E  DD  00048       PUSHL   SP                                         ; 0836
                                        0000'       CF  9F  0004A       PUSHAB  P.AAK                                      ;
                                    67              02  FB  0004E       CALLS   #2, CLI$GET_VALUE                          ;
                                    56              50  D0  00051       MOVL    R0, STATUS                                 ;
                                    E2              56  E8  00054       BLBS    STATUS, 2$                                 ; 0837
                                    50              01  D0  00057 3$:    MOVL    #1, R0                                     ; 0839
                                                    04  0005A           RET                                                ;
                                                    50  D4  0005B 4$:    CLRL    R0                                         ; 0841
```

```
                                    04 0005D        RET                                                     ;

; Routine Size:  94 bytes,    Routine Base:  $CODE$ + 0308
```

```
;  415          0842  1 %sbttl 'ANL$GET_OBJECT_RECORD - Read Record from Object File'
;  416          0843  1 !++
;  417          0844  1 ! Functional Description:
;  418          0845  1 !       This routine is called to read the next record from the current
;  419          0846  1 !       object file, which is assumed to be open.
;  420          0847  1 !
;  421          0848  1 ! Formal Parameters:
;  422          0849  1 !       buffer          Address of a descriptor to fill in.
;  423          0850  1 !
;  424          0851  1 ! Implicit Inputs:
;  425          0852  1 !       global data
;  426          0853  1 !
;  427          0854  1 ! Implicit Outputs:
;  428          0855  1 !       global data
;  429          0856  1 !
;  430          0857  1 ! Returned Value:
;  431          0858  1 !       True if there is another record, false if not.
;  432          0859  1 !
;  433          0860  1 ! Side Effects:
;  434          0861  1 !
;  435          0862  1 !--
;  436          0863  1
;  437          0864  1
;  438          0865  2 global routine anl$get_object_record(buffer) = begin
;  439          0866  2
;  440          0867  2 bind
;  441          0868  2         buffer_dsc = .buffer: descriptor;
;  442          0869  2
;  443          0870  2 local
;  444          0871  2         status: long;
;  445          0872  2
;  446          0873  2 ! We split up depending upon whether it's an object library.
;  447          0874  2
;  448          0875  3 if .object_library then (
;  449          0876  3         status = lbr$get_record(library_index,object_buffer,buffer_dsc);
;  450          0877  3         if .status eqlu rms$_eof then
;  451          0878  3                 return false;
;  452          0879  3         check (.status, anlobj$_readerr,1,resultant_spec,.status);
;  453          0880  3
;  454          0881  3 ) else (
;  455          0882  3
;  456          0883  3         status = $get(rab=object_rab);
;  457          0884  3         if .status eqlu rms$_eof then
;  458          0885  3                 return false;
;  459          0886  3         check (.status, anlobj$_readerr,1,resultant_spec,.status,.object_rab[rab$l_stv]);
;  460          0887  3         build_descriptor(buffer_dsc,.object_rab[rab$w_rsz],.object_rab[rab$l_rbf]);
;  461          0888  2 );
;  462          0889  2
;  463          0890  2 return true;
;  464          0891  2
;  465          0892  1 end;


                                                        .EXTRN  SYS$GET

03                                      000C 00000      .ENTRY  ANL$GET_OBJECT_RECORD, Save R2,R3           ; 086
```

```
06                                      53 00000000G   00  9E 00002          MOVAB   LIB$SIGNAL, R3                                  ;
11                                      52       04    AC  D0 00009          MOVL    BUFFER, R2                                      ; 0868
                                        30     0000'   CF  E9 0000D          BLBC    OBJECT_LIBRARY, 1$                              ; 0875
                                                       52  DD 00012          PUSHL   R2                                              ; 0876
15                                             0000'   CF  9F 00014          PUSHAB  OBJECT_BUFFER                                   ;
17                                             0000'   CF  9F 00018          PUSHAB  LIBRARY_INDEX                                   ;
19                                 00000000G   00          03  FB 0001C      CALLS   #3, LBR$GET_RECORD                              ;
                                   0001827A    8F          50  D1 00023      CMPL    STATUS, #98938                                  ; 0877
                                               51  13 0002A                  BEQL    4$                                              ;
                                           4A              50  E8 0002C      BLBS    STATUS, 3$                                      ; 0879
                                                           50  DD 0002F      PUSHL   STATUS                                          ;
                                               0000'   CF  9F 00031          PUSHAB  RESULTANT_SPEC                                  ;
                                                           01  DD 00035      PUSHL   #1                                              ;
                                             00B110B2      8F  DD 00037      PUSHL   #11604146                                       ;
                                           63              04  FB 0003D      CALLS   #4, LIB$SIGNAL                                  ;
                                                           37  11 00040      BRB     3$                                              ; 0875
                                               0000'   CF  9F 00042  1$:     PUSHAB  OBJECT_RAB                                      ; 0883
                                   00000000G   00          01  FB 00046      CALLS   #1, SYS$GET                                     ;
                                   0001827A    8F          50  D1 0004D      CMPL    STATUS, #98938                                  ; 0884
                                               27  13 00054                  BEQL    4$                                              ;
                                           15              50  E8 00056      BLBS    STATUS, 2$                                      ; 0886
                                               0000'   CF  DD 00059          PUSHL   OBJECT_RAB+12                                   ;
                                                           50  DD 0005D      PUSHL   STATUS                                          ;
                                               0000'   CF  9F 0005F          PUSHAB  RESULTANT_SPEC                                  ;
                                                           01  DD 00063      PUSHL   #1                                              ;
                                             00B110B2      8F  DD 00065      PUSHL   #11604146                                       ;
                                           63              05  FB 00068      CALLS   #5, LIB$SIGNAL                                  ;
                                           62      0000'   CF  3C 0006E  2$:  MOVZWL  OBJECT_RAB+34, (R2)                             ; 0887
                                     04    A2      0000'   CF  D0 00073      MOVL    OBJECT_RAB+40, 4(R2)                            ;
                                           50              01  D0 00079  3$:  MOVL    #1, R0                                          ; 0890
                                                           04  0007C        RET                                                     ;
                                                       50  D4 0007D  4$:     CLRL    R0                                              ; 0892
                                                           04  0007F        RET                                                     ;

; Routine Size:  128 bytes,    Routine Base:  $CODE$ + 0366

;  466          0893  1
;  467          0894  0 end eludom


                                                                               .EXTRN  LIB$SIGNAL

;                                       PSECT SUMMARY
;
;      Name                             Bytes                          Attributes
;
;  $OWN$                                 2760  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                                 998  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                                  76  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                   Library Statistics
```

```
;
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776       54         0       581         00:01.0
```

```
;                                    COMMAND QUALIFIERS

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

; Size:          998 code + 2836 data bytes
; Run Time:        00:19.8
; Elapsed Time:    00:57.9
; Lines/CPU Min:    2706
; Lexemes/CPU-Min: 20240
; Memory Used:  268 pages
; Compilation Complete
```

OB MISC
LIS

RMS21DX
LIS

RMS31DX
LIS

RMS
LIS

OBJUTIR
LIS