```
   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
```


OBJDRIVE

```
;  1          0001  0 %title 'OBJDRIVE - Drive Analysis of Object Files'
;  2          0002  0         module objdrive (
;  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, Object File Analyzer
; 32          0032  1 !
; 33          0033  1 ! Abstract:     This module is the main control for the analysis of object
; 34          0034  1 !               files.
; 35          0035  1 !
; 36          0036  1 !
; 37          0037  1 ! Environment:
; 38          0038  1 !
; 39          0039  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 6 January 1981
; 40          0040  1 !
; 41          0041  1 ! Modified By:
; 42          0042  1 !
; 43          0043  1 !       V03-002 DGB0069         Donald G. Blair         03-Jul-1984
; 44          0044  1 !               Allow the /NOOUTPUT qualifer.
; 45          0045  1 !
; 46          0046  1 !       V03-001 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
; 47          0047  1 !               Change the message prefix to ANLOBJ$ to ensure that
; 48          0048  1 !               message symbols are unique across all ANALYZEs.  This
; 49          0049  1 !               is necessitated by the new merged message files.
; 50          0050  1 !--
```

```
;    52          0051  1 %sbttl 'Module Declarations'
;    53          0052  1 !
;    54          0053  1 ! Libraries and Requires:
;    55          0054  1 !
;    56          0055  1
;    57          0056  1 library 'starlet';
;    58          0057  1 require 'objexereq';
;    59          0493  1
;    60          0494  1 !
;    61          0495  1 ! Table of Contents:
;    62          0496  1 !
;    63          0497  1
;    64          0498  1 forward routine
;    65          0499  1         anl$object: novalue,
;    66          0500  1         anl$object_positionals: novalue,
;    67          0501  1         anl$object2: novalue,
;    68          0502  1         anl$object_record_line: novalue,
;    69          0503  1         anl$object_statistics: novalue;
;    70          0504  1
;    71          0505  1 !
;    72          0506  1 ! External References:
;    73          0507  1 !
;    74          0508  1
;    75          0509  1 external routine
;    76          0510  1         anl$error_count,
;    77          0511  1         anl$format_error,
;    78          0512  1         anl$format_hex,
;    79          0513  1         anl$format_line,
;    80          0514  1         anl$get_object_record,
;    81          0515  1         anl$interact,
;    82          0516  1         anl$object_eom,
;    83          0517  1         anl$object_gsd,
;    84          0518  1         anl$object_hdr,
;    85          0519  1         anl$object_lnk,
;    86          0520  1         anl$object_record_size,
;    87          0521  1         anl$object_tir,
;    88          0522  1         anl$open_next_object_file,
;    89          0523  1         anl$prepare_report_file,
;    90          0524  1         anl$report_line,
;    91          0525  1         anl$report_page,
;    92          0526  1         cli$get_value: addressing_mode(general),
;    93          0527  1         cli$present: addressing_mode(general);
;    94          0528  1
;    95          0529  1 !
;    96          0530  1 ! Global Variables:
;    97          0531  1 !
;    98          0532  1 ! The following variable tells people whether this is an interactive session.
;    99          0533  1
;   100          0534  1 global
;   101          0535  1         anl$gb_interactive: byte;
;   102          0536  1
;   103          0537  1 !
;   104          0538  1 ! Own Variables:
;   105          0539  1 !
;   106          0540  1 ! The following variables contain various positional qualifier values.
;   107          0541  1
;   108          0542  1 own
```

```
;  109          0543  1          type_flag: bitvector[256],
;  110          0544  1          range: vector[2,long];
;  111          0545  1
;  112          0546  1 ! The following flag tells us whether we are currently "within" a module.
;  113          0547  1
;  114          0548  1 own
;  115          0549  1          within_module: byte initial(false);
;  116          0550  1
;  117          0551  1 ! The following two vectors keep track of the count of each object record
;  118          0552  1 ! type and the total bytes for each type.
;  119          0553  1
;  120          0554  1 own
;  121          0555  1          record_count: vector[obj$c_maxrectyp+1,long]
;  122          0556  1                        initial(rep obj$c_maxrectyp+1 of (0)),
;  123          0557  1          byte_total:   vector[obj$c_maxrectyp+1,long]
;  124          0558  1                        initial(rep obj$c_maxrectyp+1 of (0));
```

```
;  126          0559  1 %sbttl 'ANL$OBJECT - Control Analysis of Object Files'
;  127          0560  1 !++
;  128          0561  1 ! Functional Description:
;  129          0562  1 !       This routine is responsible for controlling the analysis of object
;  130          0563  1 !       files.
;  131          0564  1 !
;  132          0565  1 ! Formal Parameters:
;  133          0566  1 !       none
;  134          0567  1 !
;  135          0568  1 ! Implicit Inputs:
;  136          0569  1 !       global data
;  137          0570  1 !
;  138          0571  1 ! Implicit Outputs:
;  139          0572  1 !       global data
;  140          0573  1 !
;  141          0574  1 ! Returned Value:
;  142          0575  1 !       none
;  143          0576  1 !
;  144          0577  1 ! Side Effects:
;  145          0578  1 !
;  146          0579  1 !--
;  147          0580  1
;  148          0581  1
;  149          0582  2 global routine anl$object: novalue = begin
;  150          0583  2
;  151          0584  2 own
;  152          0585  2         own_described_buffer(report_file_spec,nam$c_maxrss);
;  153          0586  2
;  154          0587  2 local
;  155          0588  2         status: long,
;  156          0589  2         type_flag: bitvector[256],
;  157          0590  2         range: vector[2,long];
;  158          0591  2
;  159          0592  2
;  160          0593  2 ! Get the global qualifiers that can be specified for ANALYZE/OBJECT.
;  161          0594  2 ! The first one is the /INTERACTIVE qualifier.
;  162          0595  2
;  163          0596  2 anl$gb_interactive = cli$present(describe('INTERACTIVE'));
;  164          0597  2
;  165          0598  2 ! If the user wants us to generate output, determine the name of the
;  166          0599  2 ! report file.  Note that if this is an interactive session, we always
;  167          0600  2 ! use SYS$OUTPUT.
;  168          0601  2
;  169          0602  2 if cli$present(describe('OUTPUT')) then
;  170          0603  2         if .anl$gb_interactive then
;  171          0604  2                 ch$copy(10,uplit byte ('SYS$OUTPUT'),
;  172          0605  2                         ' ',.report_file_spec[len],.report_file_spec[ptr])
;  173          0606  2         else
;  174          0607  2                 cli$get_value(describe('OUTPUT'),report_file_spec);
;  175          0608  2
;  176          0609  2 ! We go into a loop, once through for each object file or library member.
;  177          0610  2
;  178          0611  3 loop (
;  179          0612  3         local
;  180          0613  3                 local_described_buffer(resultant_file_spec,nam$c_maxrss);
;  181          0614  3
;  182          0615  3         status = anl$open_next_object_file(resultant_file_spec);
```

```
;  183          0616  3
;  184          0617  3 exitif (not .status);
;  185          0618  3
;  186          0619  3         ! Prepare the file to receive the object analysis report.
;  187          0620  3
;  188          0621  3         anl$prepare_report_file(report_file_spec,resultant_file_spec,anlobj$_objheading);
;  189          0622  3
;  190          0623  3         ! Analyze the object file.
;  191          0624  3
;  192          0625  3         anl$object2();
;  193          0626  2 );
;  194          0627  2
;  195          0628  2 return;
;  196          0629  2
;  197          0630  1 end;


;                                                                       .TITLE  OBJDRIVE OBJDRIVE - Drive Analysis of Object Fi
                                                                                les
                                                                       .IDENT  \V04-000\

                                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                45  56  49  54  43  41  52  45  54  4E  49  00000 P.AAB:  .ASCII  \INTERACTIVE\                                ;
                                                        0000B          .BLKB   1
                                                    0000000B    0000C P.AAA:  .LONG   11                                       ;
                                                    00000000'   00010         .ADDRESS P.AAB                               ;
                                    54  55  50  54  55  4F      00014 P.AAD:  .ASCII  \OUTPUT\                               ;
                                                        0001A          .BLKB   2
                                                    00000006    0001C P.AAC:  .LONG   6                                      ;
                                                    00000000'   00020         .ADDRESS P.AAD                               ;
        54  55  50  54  55  4F  24  53  59  53      00024 P.AAE:  .ASCII  \SYS$OUTPUT\                                     ;
                                    54  55  50  54  55  4F      0002E P.AAG:  .ASCII  \OUTPUT\                               ;
                                                    00000006    00034 P.AAF:  .LONG   6                                      ;
                                                    00000000'   00038         .ADDRESS P.AAG                               ;

                                                                       .PSECT  $OWN$,NOEXE,2

                                                            00000 TYPE_FLAG:
                                                                       .BLKB   32
                                                            00020 RANGE:  .BLKB   8
                                                        00  00028 WITHIN_MODULE:
                                                                       .BYTE   0                                            ;
                                                            00029      .BLKB   3
                                                    00000000# 0002C RECORD_COUNT:
                                                                       .LONG   0[8]                                         ;
                                                    00000000# 0004C BYTE_TOTAL:
                                                                       .LONG   0[8]                                         ;
                                                    000000FF  0006C REPORT_FILE_SPEC:
                                                                       .LONG   255                                          ;
                                                    00000000' 00070    .ADDRESS REPORT_FILE_SPEC+8                          ;
                                                              00074    .BLKB   255

                                                                       .PSECT  $GLOBAL$,NOEXE,2

                                                            00000 ANL$GB_INTERACTIVE::
                                                                       .BLKB   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
.EXTRN  ANLOBJ$_EXEHDRPAGEDEF
.EXTRN  ANLOBJ$_EXEHDRPATCH
.EXTRN  ANLOBJ$_EXEHDRPATCHDATE
.EXTRN  ANLOBJ$_EXEHDRPRIV
.EXTRN  ANLOBJ$_EXEHDRROPATCH
```

```
.EXTRN  ANLOBJ$_EXEHDRRWPATCH
.EXTRN  ANLOBJ$_EXEHDRSYMDBG
.EXTRN  ANLOBJ$_EXEHDRSYSVER
.EXTRN  ANLOBJ$_EXEHDRTEXTVBN
.EXTRN  ANLOBJ$_EXEHDRTIME
.EXTRN  ANLOBJ$_EXEHDRTYPEEXE
.EXTRN  ANLOBJ$_EXEHDRTYPELIM
.EXTRN  ANLOBJ$_EXEHDRUSERECO
.EXTRN  ANLOBJ$_EXEHDRXFER1
.EXTRN  ANLOBJ$_EXEHDRXFER2
.EXTRN  ANLOBJ$_EXEHDRXFER3
.EXTRN  ANLOBJ$_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$_EXEBADISDS1
.EXTRN  ANLOBJ$_EXEBADISDTYPE
.EXTRN  ANLOBJ$_EXEBADMATCH
.EXTRN  ANLOBJ$_EXEBADPATCHLEN
```

```
                                        .EXTRN  ANLOBJ$_EXEBADOBJ
                                        .EXTRN  ANLOBJ$_EXEBADTYPE
                                        .EXTRN  ANLOBJ$_EXEBADXFERO
                                        .EXTRN  ANLOBJ$_EXEHDRISDLONG
                                        .EXTRN  ANLOBJ$_EXEHDRLONG
                                        .EXTRN  ANLOBJ$_EXEISDLENDZRO
                                        .EXTRN  ANLOBJ$_EXEISDLENGBL
                                        .EXTRN  ANLOBJ$_EXEISDLENPRIV
                                        .EXTRN  ANLOBJ$_EXENOTNATIVE
                                        .EXTRN  ANLOBJ$_EXTRABYTES
                                        .EXTRN  ANLOBJ$_FIELDFIT
                                        .EXTRN  ANLOBJ$_FLAGERROR
                                        .EXTRN  ANLOBJ$_NOTOK, ANLOBJ$_OBJBADIDCMATCH
                                        .EXTRN  ANLOBJ$_OBJBADNUM
                                        .EXTRN  ANLOBJ$_OBJBADPOP
                                        .EXTRN  ANLOBJ$_OBJBADPUSH
                                        .EXTRN  ANLOBJ$_OBJBADTYPE
                                        .EXTRN  ANLOBJ$_OBJBADVIELD
                                        .EXTRN  ANLOBJ$_OBJEOMBADSEV
                                        .EXTRN  ANLOBJ$_OBJEOMMISSING
                                        .EXTRN  ANLOBJ$_OBJFADBADAVC
                                        .EXTRN  ANLOBJ$_OBJFADBADRBC
                                        .EXTRN  ANLOBJ$_OBJGSDBADALIGN
                                        .EXTRN  ANLOBJ$_OBJGSDBADSUBTYP
                                        .EXTRN  ANLOBJ$_OBJHDRRES
                                        .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$_FACILITY
                                        .EXTRN  ANL$ERROR_COUNT
                                        .EXTRN  ANL$FORMAT_ERROR
                                        .EXTRN  ANL$FORMAT_HEX, ANL$FORMAT_LINE
                                        .EXTRN  ANL$GET_OBJECT_RECORD
                                        .EXTRN  ANL$INTERACT, ANL$OBJECT_EOM
                                        .EXTRN  ANL$OBJECT_GSD, ANL$OBJECT_HDR
                                        .EXTRN  ANL$OBJECT_LNK, ANL$OBJECT_RECORD_SIZE
                                        .EXTRN  ANL$OBJECT_TIR, ANL$OPEN_NEXT_OBJECT_FILE
                                        .EXTRN  ANL$PREPARE_REPORT_FILE
                                        .EXTRN  ANL$REPORT_LINE
                                        .EXTRN  ANL$REPORT_PAGE
                                        .EXTRN  CLI$GET_VALUE, CLI$PRESENT

                                        .PSECT  $CODE$,NOWRT,2

                            00FC 00000  .ENTRY  ANL$OBJECT, Save R2,R3,R4,R5,R6,R7          ; 0582
57      0000'  CF   9E 00002            MOVAB   REPORT_FILE_SPEC, R7                        ;
```

```
                                56 00000000G   00   9E 00007         MOVAB   CLI$PRESENT, R6                                                ;
                                5E     FED0    CE   9E 0000E         MOVAB   -304(SP), SP                                               ;
                                       0000'   CF   9F 00013         PUSHAB  P.AAA                                                      ; 0596
                                66             01   FB 00017         CALLS   #1, CLI$PRESENT                                            ;
                           0000' CF            50   90 0001A         MOVB    R0, ANL$GB_INTERACTIVE                                     ;
                                       0000'   CF   9F 0001F         PUSHAB  P.AAC                                                      ; 0602
                                66             01   FB 00023         CALLS   #1, CLI$PRESENT                                            ;
                                1D             50   E9 00026         BLBC    R0, 2$                                                     ;
                                0B     0000'   CF   E9 00029         BLBC    ANL$GB_INTERACTIVE, 1$                                     ; 0603
         67          20        0000'   CF         0A   2C 0002E      MOVC5   #10, P.AAE, #32, REPORT_FILE_SPEC, -                       ; 0605
                                                04   B7    00035                 @REPORT_FILE_SPEC+4                                      ;
                                                     0D   11 00037      BRB     2$                                                     ; 0604
                                                     57   DD 00039 1$:  PUSHL   R7                                                     ; 0607
                                       0000'   CF   9F 0003B         PUSHAB  P.AAF                                                      ;
                           00000000G   00         02   FB 0003F      CALLS   #2, CLI$GET_VALUE                                          ;
                                6E     FF         8F   9A 00046 2$:  MOVZBL  #255, RESULTANT_FILE_SPEC                                  ; 0613
                           04   AE     08   AE   9E   0004A         MOVAB   RESULTANT_FILE_SPEC+8, -                                   ;
                                                                             RESULTANT_FILE_SPEC+4                                      ;
                                                5E   DD 0004F         PUSHL   SP                                                         ; 0615
                           0000G CF            01   FB 00051         CALLS   #1, ANL$OPEN_NEXT_OBJECT_FILE                              ;
                                52             50   D0 00056         MOVL    R0, STATUS                                                 ;
                                17             52   E9 00059         BLBC    STATUS, 3$                                                 ; 0617
                                   00000000G   8F   DD 0005C         PUSHL   #ANLOBJ$_OBJHEADING                                        ; 0621
                                          04   AE   9F 00062         PUSHAB  RESULTANT_FILE_SPEC                                        ;
                                               57   DD 00065         PUSHL   R7                                                         ;
                           0000G CF            03   FB 00067         CALLS   #3, ANL$PREPARE_REPORT_FILE                                ;
                           0000V CF            00   FB 0006C         CALLS   #0, ANL$OBJECT2                                            ; 0625
                                               D3   11 00071         BRB     2$                                                         ; 0607
                                                    04   00073 3$:    RET                                                                ; 0630

; Routine Size:  116 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  199          0631  1 %sbttl 'ANL$OBJECT_POSITIONALS - Process Positional Qualifiers'
;  200          0632  1 !++
;  201          0633  1 ! Functional Description:
;  202          0634  1 !       This routine is called by the OBJINPUT module whenever it scans
;  203          0635  1 !       off the next file spec from the command line.  We need to process
;  204          0636  1 !       positional qualifiers.
;  205          0637  1 !
;  206          0638  1 ! Formal Parameters:
;  207          0639  1 !       none
;  208          0640  1 !
;  209          0641  1 ! Implicit Inputs:
;  210          0642  1 !       global data
;  211          0643  1 !
;  212          0644  1 ! Implicit Outputs:
;  213          0645  1 !       global data
;  214          0646  1 !
;  215          0647  1 ! Returned Value:
;  216          0648  1 !       none
;  217          0649  1 !
;  218          0650  1 ! Side Effects:
;  219          0651  1 !
;  220          0652  1 !--
;  221          0653  1
;  222          0654  1
;  223          0655  2 global routine anl$object_positionals: novalue = begin
;  224          0656  2
;  225          0657  2 local
;  226          0658  2         all_types: byte;
;  227          0659  2
;  228          0660  2
;  229          0661  2 ! First we process the qualifiers that specify which record types are to be
;  230          0662  2 ! analyzed.  If none are specified, we analyze all records.  If any are
;  231          0663  2 ! specified, we analyze only those specified.  NOTE that we always analyze
;  232          0664  2 ! module headers and end-of-module records, as well as any invalid records.
;  233          0665  2
;  234          0666  2 all_types = not cli$present(describe('DBG')) and
;  235          0667  2              not cli$present(describe('EOM')) and
;  236          0668  2              not cli$present(describe('GSD')) and
;  237          0669  2              not cli$present(describe('LNK')) and
;  238          0670  2              not cli$present(describe('MHD')) and
;  239          0671  2              not cli$present(describe('TBT')) and
;  240          0672  2              not cli$present(describe('TIR'));
;  241          0673  2
;  242          0674  2 ch$fill(%x'ff', %allocation(type_flag),type_flag);
;  243          0675  2 type_flag[obj$c_dbg] = .all_types or cli$present(describe('DBG'));
;  244          0676  2 type_flag[obj$c_gsd] = .all_types or cli$present(describe('GSD'));
;  245          0677  2 type_flag[obj$c_lnk] = .all_types or cli$present(describe('LNK'));
;  246          0678  2 type_flag[obj$c_tbt] = .all_types or cli$present(describe('TBT'));
;  247          0679  2 type_flag[obj$c_tir] = .all_types or cli$present(describe('TIR'));
;  248          0680  2
;  249          0681  2 ! There used to be other positional qualifiers, but not any more.
;  250          0682  2
;  251          0683  2 return;
;  252          0684  2
;  253          0685  1 end;
```

```
                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                47  42  44     0003C P.AAI:    .ASCII  \DBG\                           ;
                               0003F           .BLKB   1
                  00000003     00040 P.AAH:    .LONG   3                               ;
                  00000000'    00044           .ADDRESS P.AAI                          ;
                4D  4F  45     00048 P.AAK:    .ASCII  \EOM\                           ;
                               0004B           .BLKB   1
                  00000003     0004C P.AAJ:    .LONG   3                               ;
                  00000000'    00050           .ADDRESS P.AAK                          ;
                44  53  47     00054 P.AAM:    .ASCII  \GSD\                           ;
                               00057           .BLKB   1
                  00000003     00058 P.AAL:    .LONG   3                               ;
                  00000000'    0005C           .ADDRESS P.AAM                          ;
                4B  4E  4C     00060 P.AAO:    .ASCII  \LNK\                           ;
                               00063           .BLKB   1
                  00000003     00064 P.AAN:    .LONG   3                               ;
                  00000000'    00068           .ADDRESS P.AAO                          ;
                44  48  4D     0006C P.AAQ:    .ASCII  \MHD\                           ;
                               0006F           .BLKB   1
                  00000003     00070 P.AAP:    .LONG   3                               ;
                  00000000'    00074           .ADDRESS P.AAQ                          ;
                54  42  54     00078 P.AAS:    .ASCII  \TBT\                           ;
                               0007B           .BLKB   1
                  00000003     0007C P.AAR:    .LONG   3                               ;
                  00000000'    00080           .ADDRESS P.AAS                          ;
                52  49  54     00084 P.AAU:    .ASCII  \TIR\                           ;
                               00087           .BLKB   1
                  00000003     00088 P.AAT:    .LONG   3                               ;
                  00000000'    0008C           .ADDRESS P.AAU                          ;
                47  42  44     00090 P.AAW:    .ASCII  \DBG\                           ;
                               00093           .BLKB   1
                  00000003     00094 P.AAV:    .LONG   3                               ;
                  00000000'    00098           .ADDRESS P.AAW                          ;
                44  53  47     0009C P.AAY:    .ASCII  \GSD\                           ;
                               0009F           .BLKB   1
                  00000003     000A0 P.AAX:    .LONG   3                               ;
                  00000000'    000A4           .ADDRESS P.AAY                          ;
                4B  4E  4C     000A8 P.ABA:    .ASCII  \LNK\                           ;
                               000AB           .BLKB   1
                  00000003     000AC P.AAZ:    .LONG   3                               ;
                  00000000'    000B0           .ADDRESS P.ABA                          ;
                54  42  54     000B4 P.ABC:    .ASCII  \TBT\                           ;
                               000B7           .BLKB   1
                  00000003     000B8 P.ABB:    .LONG   3                               ;
                  00000000'    000BC           .ADDRESS P.ABC                          ;
                52  49  54     000C0 P.ABE:    .ASCII  \TIR\                           ;
                               000C3           .BLKB   1
                  00000003     000C4 P.ABD:    .LONG   3                               ;
                  00000000'    000C8           .ADDRESS P.ABE                          ;


                                                .PSECT  $CODE$,NOWRT,2

                       03FC 00000               .ENTRY  ANL$OBJECT_POSITIONALS, Save R2,R3,R4,R5,-  ; 0655
```

```
                                                                         R6,R7,R8,R9                          :
                        59    0000'  CF   9E 00002          MOVAB   TYPE_FLAG, R9                             :
                        58    0000'  CF   9E 00007          MOVAB   P.AAH, R8                                 :
                        57 00000000G 00   9E 0000C          MOVAB   CLI$PRESENT, R7                           :
                                     58   DD 00013          PUSHL   R8                                        : 0666
                        67           01   FB 00015          CALLS   #1, CLI$PRESENT                           :
                        53           50   D0 00018          MOVL    R0, R3                                    :
                                0C   A8   9F 0001B          PUSHAB  P.AAJ                                     : 0667
                        67           01   FB 0001E          CALLS   #1, CLI$PRESENT                           :
                        52           50   D0 00021          MOVL    R0, R2                                    :
                        52           53   C8 00024          BISL2   R3, R2                                    :
                                18   A8   9F 00027          PUSHAB  P.AAL                                     : 0668
                        67           01   FB 0002A          CALLS   #1, CLI$PRESENT                           :
                        53           50   D0 0002D          MOVL    R0, R3                                    :
                        53           52   C8 00030          BISL2   R2, R3                                    :
                                24   A8   9F 00033          PUSHAB  P.AAN                                     : 0669
                        67           01   FB 00036          CALLS   #1, CLI$PRESENT                           :
                        52           50   D0 00039          MOVL    R0, R2                                    :
                        52           53   C8 0003C          BISL2   R3, R2                                    :
                                30   A8   9F 0003F          PUSHAB  P.AAP                                     : 0670
                        67           01   FB 00042          CALLS   #1, CLI$PRESENT                           :
                        53           50   D0 00045          MOVL    R0, R3                                    :
                        53           52   C8 00048          BISL2   R2, R3                                    :
                                3C   A8   9F 0004B          PUSHAB  P.AAR                                     : 0671
                        67           01   FB 0004E          CALLS   #1, CLI$PRESENT                           :
                        52           50   D0 00051          MOVL    R0, R2                                    :
                        52           53   C8 00054          BISL2   R3, R2                                    :
                                48   A8   9F 00057          PUSHAB  P.AAT                                     : 0672
                        67           01   FB 0005A          CALLS   #1, CLI$PRESENT                           :
                        50           52   C8 0005D          BISL2   R2, R0                                    :
                        56           50   92 00060          MCOMB   R0, ALL_TYPES                             : 0671
     20       FF   8F   6E           00   2C 00063          MOVC5   #0, (SP), #255, #32, TYPE_FLAG            : 0674
                                     69      00069                                                            :
                                54   A8   9F 0006A          PUSHAB  P.AAV                                     : 0675
                        67           01   FB 0006D          CALLS   #1, CLI$PRESENT                           :
                   52   50           56   89 00070          BISB3   ALL_TYPES, R0, R2                         :
     69            01   04           52   F0 00074          INSV    R2, #4, #1, TYPE_FLAG                     :
                                60   A8   9F 00079          PUSHAB  P.AAX                                     : 0676
                        67           01   FB 0007C          CALLS   #1, CLI$PRESENT                           :
                   52   50           56   89 0007F          BISB3   ALL_TYPES, R0, R2                         :
     69            01   01           52   F0 00083          INSV    R2, #1, #1, TYPE_FLAG                     :
                                6C   A8   9F 00088          PUSHAB  P.AAZ                                     : 0677
                        67           01   FB 0008B          CALLS   #1, CLI$PRESENT                           :
                   52   50           56   89 0008E          BISB3   ALL_TYPES, R0, R2                         :
     69            01   06           52   F0 00092          INSV    R2, #6, #1, TYPE_FLAG                     :
                                78   A8   9F 00097          PUSHAB  P.ABB                                     : 0678
                        67           01   FB 0009A          CALLS   #1, CLI$PRESENT                           :
                   52   50           56   89 0009D          BISB3   ALL_TYPES, R0, R2                         :
     69            01   05           52   F0 000A1          INSV    R2, #5, #1, TYPE_FLAG                     :
                              0084   C8   9F 000A6          PUSHAB  P.ABD                                     : 0679
                        67           01   FB 000AA          CALLS   #1, CLI$PRESENT                           :
                   52   50           56   89 000AD          BISB3   ALL_TYPES, R0, R2                         :
     69            01   02           52   F0 000B1          INSV    R2, #2, #1, TYPE_FLAG                     :
                                          04 000B6          RET                                               : 0685

; Routine Size:  183 bytes,    Routine Base:  $CODE$ + 0074
```

```
;  255          0686  1 %sbttl 'ANL$OBJECT2 - Generate Object Analysis Report'
;  256          0687  1 !++
;  257          0688  1 ! Functional Description:
;  258          0689  1 !       This routine is responsible for generating the analysis report
;  259          0690  1 !       for a single object file.  The object file is ready to read, and the
;  260          0691  1 !       report file is ready to write.
;  261          0692  1 !
;  262          0693  1 ! Formal Parameters:
;  263          0694  1 !       none
;  264          0695  1 !
;  265          0696  1 ! Implicit Inputs:
;  266          0697  1 !       global data
;  267          0698  1 !
;  268          0699  1 ! Implicit Outputs:
;  269          0700  1 !       global data
;  270          0701  1 !
;  271          0702  1 ! Returned Value:
;  272          0703  1 !       none
;  273          0704  1 !
;  274          0705  1 ! Side Effects:
;  275          0706  1 !
;  276          0707  1 !--
;  277          0708  1
;  278          0709  1
;  279          0710  2 global routine anl$object2: novalue = begin
;  280          0711  2
;  281          0712  2 local
;  282          0713  2         status: long,
;  283          0714  2         record_number: long,
;  284          0715  2         record_dsc: descriptor,
;  285          0716  2         scanp: ref block[,byte],
;  286          0717  2         type: byte;
;  287          0718  2
;  288          0719  2
;  289          0720  2 ! This is the main record analysis loop.  We go through once for each record
;  290          0721  2 ! in the object file.
;  291          0722  2
;  292          0723  3 incru record_number from 1 do (
;  293          0724  3
;  294          0725  3         ! Get the next object record.  If we reach end-of-file, we're done
;  295          0726  3         ! with this loop.
;  296          0727  3
;  297          0728  3         status = anl$get_object_record(record_dsc);
;  298          0729  3
;  299          0730  3 exitif (not .status);
;  300          0731  3
;  301          0732  3         ! See if there is a type code in the record.  If not, that's an error.
;  302          0733  3
;  303          0734  3         scanp = .record_dsc[ptr];
;  304          0735  4         if .record_dsc[len] gtru 0 then (
;  305          0736  4
;  306          0737  4                 ! Now we know we have a record type, so we can see if the
;  307          0738  4                 ! user wants to analyze it.  If so, select on the type code.
;  308          0739  4                 ! If not, just ignore it.
;  309          0740  4
;  310          0741  5                 if .type_flag[.scanp[obj$b_rectyp]] then (
;  311          0742  5
```

```
;   312          0743  5                          selectoneu .scanp[obj$b_rectyp] of set
;   313          0744  5                          [obj$c_hdr]:    anl$object_hdr(.record_number,record_dsc);
;   314          0745  5
;   315          0746  5                          [obj$c_gsd]:    anl$object_gsd(.record_number,record_dsc);
;   316          0747  5
;   317          0748  5                          [obj$c_tir,
;   318          0749  5                           obj$c_dbg,
;   319          0750  5                           obj$c_tbt]:    anl$object_tir(.record_number,record_dsc);
;   320          0751  5
;   321          0752  5                          [obj$c_eom,
;   322          0753  5                           obj$c_eomw]:   anl$object_eom(.record_number,record_dsc);
;   323          0754  5
;   324          0755  5                          [obj$c_lnk]:    anl$object_lnk(.record_number,record_dsc);
;   325          0756  5
;   326          0757  6                          [otherwise]:    (anl$format_error(anlobj$_objbadtype,.record_number,.scanp[obj$b_rec
;   327          0758  5                                          anl$format_hex(1,record_dsc););
;   328          0759  5                          tes;
;   329          0760  5
;   330          0761  5                          ! Make sure that this record isn't longer than the
;   331          0762  5                          ! maximum specified in the module header.
;   332          0763  5
;   333          0764  5                          anl$object_record_size(.record_dsc[len]);
;   334          0765  5
;   335          0766  5                          ! Skip a couple of lines to make it look nice.
;   336          0767  5
;   337          0768  5                          anl$report_line(-1);
;   338          0769  5                          anl$report_line(-1);
;   339          0770  5
;   340          0771  5                          ! If this is an interactive session, let's find out if
;   341          0772  5                          ! the user wants to continue or quit.
;   342          0773  5
;   343          0774  5                          if .anl$gb_interactive then
;   344          0775  5 exitif (not anl$interact());
;   345          0776  5
;   346          0777  4                      );
;   347          0778  4
;   348          0779  4              ) else (
;   349          0780  4
;   350          0781  4                      ! There was no record type.  Tell the user.
;   351          0782  4
;   352          0783  4                      anl$format_error(anlobj$_objnullrec,.record_number);
;   353          0784  4                      anl$report_line(-1);
;   354          0785  4                      anl$report_line(-1);
;   355          0786  3              );
;   356          0787  3
;   357          0788  2 );
```

```
;  359          0789  2 ! We have read all the records of the object file.  If we are still within
;  360          0790  2 ! a module, then an end-of-module record is missing.  We have to force one
;  361          0791  2 ! so that various error checks can be made.
;  362          0792  2
;  363          0793  3 if .within_module then (
;  364          0794  3         anl$format_error(anlobj$_objeommissing);
;  365          0795  3         anl$object_eom();
;  366          0796  3         within_module = false;
;  367          0797  2 );
;  368          0798  2
;  369          0799  2 ! Now we can print the summary statistics on a new page.
;  370          0800  2
;  371          0801  2 anl$report_page();
;  372          0802  2 anl$object_statistics();
;  373          0803  2
;  374          0804  2 ! Tell the user how many errors were uncovered.
;  375          0805  2
;  376          0806  2 anl$error_count();
;  377          0807  2
;  378          0808  2 ! Finally, print the command line that was used to generate the report.
;  379          0809  2
;  380          0810  3 begin
;  381          0811  3 local
;  382          0812  3         local_described_buffer(command_line,80);
;  383          0813  3
;  384          0814  3 cli$get_value(describe('$LINE'),command_line);
;  385          0815  3 anl$format_line(0,0,anlobj$_anything,command_line);
;  386          0816  2 end;
;  387          0817  2
;  388          0818  2 return;
;  389          0819  2
;  390          0820  1 end;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                45  4E  49  4C  24  000CC P.ABG:  .ASCII  \$LINE\                          ;
                                      000D1     .BLKB   3
                        00000005      000D4 P.ABF:  .LONG   5                             ;
                        00000000'     000D8     .ADDRESS P.ABG                          ;


                                                .PSECT  $CODE$,NOWRT,2

                              007C 00000        .ENTRY  ANL$OBJECT2, Save R2,R3,R4,R5,R6  ; 0710
                56      0000G   CF   9E 00002   MOVAB   ANL$FORMAT_ERROR, R6              ;
                55      0000G   CF   9E 00007   MOVAB   ANL$REPORT_LINE, R5               ;
                5E        A0    AE   9E 0000C   MOVAB   -96(SP), SP                     ;
                53              01   D0 00010   MOVL    #1, RECORD_NUMBER               ; 0723
                          58    AE   9F 00013 1$:  PUSHAB  RECORD_DSC                   ; 0728
        0000G   CF              01   FB 00016   CALLS   #1, ANL$GET_OBJECT_RECORD       ;
                54              50   D0 0001B   MOVL    R0, STATUS                      ;
                03              54   E8 0001E   BLBS    STATUS, 2$                      ; 0730
                            00D1   31 00021     BRW     15$                             ;
                52        5C    AE   D0 00024 2$:  MOVL    RECORD_DSC+4, SCANP         ; 0734
```

```
                                                58  AE  B5 00028          TSTW    RECORD_DSC                                      ; 0735
                                                    03  12 0002B          BNEQ    3$
                                                00A9  31 0002D          BRW     13$
                                            50      62  9A 00030 3$:      MOVZBL  (SCANP), R0                                     ; 0741
                                        03  0000'  CF  50  E0 00033          BBS     R0, TYPE_FLAG, 4$
                                                00B4  31 00039          BRW     14$
                                                    0C  12 0003C 4$:      BNEQ    5$                                              ; 0744
                                                58  AE  9F 0003E          PUSHAB  RECORD_DSC
                                                    53  DD 00041          PUSHL   RECORD_NUMBER
                                            0000G   CF  02  FB 00043          CALLS   #2, AN[$OBJECT_HDR
                                                    6B  11 00048          BRB     12$
                                            01      62  91 0004A 5$:      CMPB    (SCANP), #1                                     ; 0746
                                                    0C  12 0004D          BNEQ    6$
                                                58  AE  9F 0004F          PUSHAB  RECORD_DSC
                                                    53  DD 00052          PUSHL   RECORD_NUMBER
                                            0000G   CF  02  FB 00054          CALLS   #2, AN[$OBJECT_GSD
                                                    5A  11 00059          BRB     12$
                                            02      62  91 0005B 6$:      CMPB    (SCANP), #2                                     ; 0748
                                                    0A  13 0005E          BEQL    7$
                                            04      62  91 00060          CMPB    (SCANP), #4
                                                    11  1F 00063          BLSSU   8$
                                            05      62  91 00065          CMPB    (SCANP), #5
                                                    0C  1A 00068          BGTRU   8$
                                                58  AE  9F 0006A 7$:      PUSHAB  RECORD_DSC                                      ; 0750
                                                    53  DD 0006D          PUSHL   RECORD_NUMBER
                                            0000G   CF  02  FB 0006F          CALLS   #2, AN[$OBJECT_TIR
                                                    3F  11 00074          BRB     12$
                                            03      62  91 00076 8$:      CMPB    (SCANP), #3                                     ; 0752
                                                    05  13 00079          BEQL    9$
                                            07      62  91 0007B          CMPB    (SCANP), #7
                                                    0C  12 0007E          BNEQ    10$
                                                58  AE  9F 00080 9$:      PUSHAB  RECORD_DSC                                      ; 0753
                                                    53  DD 00083          PUSHL   RECORD_NUMBER
                                            0000G   CF  02  FB 00085          CALLS   #2, AN[$OBJECT_EOM
                                                    29  11 0008A          BRB     12$
                                            06      62  91 0008C 10$:      CMPB    (SCANP), #6                                     ; 0755
                                                    0C  12 0008F          BNEQ    11$
                                                58  AE  9F 00091          PUSHAB  RECORD_DSC
                                                    53  DD 00094          PUSHL   RECORD_NUMBER
                                            0000G   CF  02  FB 00096          CALLS   #2, AN[$OBJECT_LNK
                                                    18  11 0009B          BRB     12$
                                            7E      62  9A 0009D 11$:      MOVZBL  (SCANP), -(SP)                                  ; 0757
                                                    53  DD 000A0          PUSHL   RECORD_NUMBER
                                        00000000G   8F  DD 000A2          PUSHL   #ANLOBJ$_OBJBADTYPE
                                            66      03  FB 000A8          CALLS   #3, ANL$FORMAT_ERROR
                                                58  AE  9F 000AB          PUSHAB  RECORD_DSC                                      ; 0758
                                                    01  DD 000AE          PUSHL   #1
                                            0000G   CF  02  FB 000B0          CALLS   #2, ANL$FORMAT_HEX
                                            7E      58  AE  3C 000B5 12$:      MOVZWL  RECORD_DSC, -(SP)                           ; 0764
                                            0000G   CF  01  FB 000B9          CALLS   #1, AN[$OBJECT_RECORD_SIZE
                                            7E          01  CE 000BE          MNEGL   #1, -(SP)                                   ; 0768
                                            65          01  FB 000C1          CALLS   #1, ANL$REPORT_LINE
                                            7E          01  CE 000C4          MNEGL   #1, -(SP)                                   ; 0769
                                            65          01  FB 000C7          CALLS   #1, ANL$REPORT_LINE
                                            21      0000'   CF  E9 000CA          BLBC    ANL$GB_INTERACTIVE, 14$                 ; 0774
                                            0000G   CF      00  FB 000CF          CALLS   #0, AN[$INTERACT                        ; 0775
                                            19              50  E8 000D4          BLBS    R0, 14$
```

```
                                    1C  11 000D7                 BRB     15$                                                     ;
                                    53  DD 000D9 13$:           PUSHL   RECORD_NUMBER                                           ; 0783
                          00000000G 8F  DD 000DB                PUSHL   #ANLOBJ$_OBJNULLREC                                     ;
                         66         02  FB 000E1                CALLS   #2, ANL$FORMAT_ERROR                                    ;
                         7E         01  CE 000E4                MNEGL   #1, -(SP)                                               ; 0784
                         65         01  FB 000E7                CALLS   #1, ANL$REPORT_LINE                                     ;
                         7E         01  CE 000EA                MNEGL   #1, -(SP)                                               ; 0785
                         65         01  FB 000ED                CALLS   #1, ANL$REPORT_LINE                                     ;
                                    53  D6 000F0 14$:           INCL    RECORD_NUMBER                                           ; 0723
                                FF1E    31 000F2                BRW     1$                                                      ;
                         12   0000'  CF  E9 000F5 15$:           BLBC    WITHIN_MODULE, 16$                                      ; 0793
                          00000000G 8F  DD 000FA                PUSHL   #ANLOBJ$_OBJEOMMISSING                                  ; 0794
                         66         01  FB 00100                CALLS   #1, ANL$FORMAT_ERROR                                    ;
                   0000G CF         00  FB 00103                CALLS   #0, ANL$OBJECT_EOM                                      ; 0795
                         0000'      CF  94 00108                CLRB    WITHIN_MODULE                                           ; 0796
                   0000G CF         00  FB 0010C 16$:           CALLS   #0, ANL$REPORT_PAGE                                     ; 0801
                   0000V CF         00  FB 00111                CALLS   #0, ANL$OBJECT_STATISTICS                               ; 0802
                   0000G CF         00  FB 00116                CALLS   #0, ANL$ERROR_COUNT                                     ; 0806
                         6E   50    8F  9A 0011B                MOVZBL  #80, COMMAND_LINE                                       ; 0812
                      04 AE   08    AE  9E 0011F                MOVAB   COMMAND_LINE+8, COMMAND_LINE+4                          ;
                                    5E  DD 00124                PUSHL   SP                                                      ; 0814
                         0000'      CF  9F 00126                PUSHAB  P.ABF                                                   ;
                00000000G 00        02  FB 0012A                CALLS   #2, CLI$GET_VALUE                                       ;
                                    5E  DD 00131                PUSHL   SP                                                      ; 0815
                          00000000G 8F  DD 00133                PUSHL   #ANLOBJ$_ANYTHING                                       ;
                                    7E  7C 00139                CLRQ    -(SP)                                                   ;
                   0000G CF         04  FB 0013B                CALLS   #4, ANL$FORMAT_LINE                                     ;
                                        04 00140                RET                                                             ; 0820

; Routine Size:  321 bytes,    Routine Base:  $CODE$ + 012B
```

```
;  392          0821  1 %sbttl 'ANL$OBJECT_RECORD_LINE - Print Record Lines'
;  393          0822  1 !++
;  394          0823  1 ! Functional Description:
;  395          0824  1 !       This routine is responsible for printing the major record line
;  396          0825  1 !       for each object record to be analyzed.  Because of this, the
;  397          0826  1 !       routine gets a good summary of the object file.  Thus it is also
;  398          0827  1 !       responsible for some simple checking of the order of records and
;  399          0828  1 !       some overall statistics.
;  400          0829  1 !
;  401          0830  1 ! Formal Parameters:
;  402          0831  1 !       record_msg      The message code for the major record line.
;  403          0832  1 !       record_number   The number of this record.
;  404          0833  1 !       the_record      Address of descriptor of the record.
;  405          0834  1 !
;  406          0835  1 ! Implicit Inputs:
;  407          0836  1 !       global data
;  408          0837  1 !
;  409          0838  1 ! Implicit Outputs:
;  410          0839  1 !       global data
;  411          0840  1 !
;  412          0841  1 ! Returned Value:
;  413          0842  1 !       none
;  414          0843  1 !
;  415          0844  1 ! Side Effects:
;  416          0845  1 !
;  417          0846  1 !--
;  418          0847  1
;  419          0848  1
;  420          0849  2 global routine anl$object_record_line(record_msg,record_number,the_record): novalue = begin
;  421          0850  2
;  422          0851  2 bind
;  423          0852  2         record_dsc = .the_record: descriptor;
;  424          0853  2
;  425          0854  2 local
;  426          0855  2         scanp: ref block[,byte],
;  427          0856  2         module_header: byte;
;  428          0857  2
;  429          0858  2
;  430          0859  2 ! The record is guaranteed to be at least one byte long.  Therefore we can
;  431          0860  2 ! always print the record line.
;  432          0861  2
;  433          0862  2 anl$format_line(4,0,.record_msg,.record_number,.record_dsc[len]);
;  434          0863  2
;  435          0864  2 ! Now we want to check a little bit of the overall structure of the module
;  436          0865  2 ! Split up depending upon whether we are within a module or not.
;  437          0866  2
;  438          0867  2 scanp = .record_dsc[ptr];
;  439          0868  2 module_header = false;
;  440          0869  2 if .scanp[obj$b_rectyp] eqlu obj$c_hdr then
;  441          0870  2         if .record_dsc[len] gequ 2 then
;  442          0871  2                 if .scanp[obj$b_subtyp] eqlu obj$c_hdr_mhd then
;  443          0872  2                         module_header = true;
;  444          0873  2
;  445          0874  3 if not .within_module then (
;  446          0875  3
;  447          0876  3         ! We are not within a module, so this had better be a module header.
;  448          0877  3         ! If not tell the user.
```

```
;  449          0878  3
;  450          0879  3          if not .module_header then
;  451          0880  3                  anl$format_error(anlobj$_objmhdmissing);
;  452          0881  3          within_module = true;
;  453          0882  3
;  454          0883  3      ) else (
;  455          0884  3
;  456          0885  3          ! We are within a module, so this had better not be a module header.
;  457          0886  3          ! If it is, tell the user.  If it is an end of module record, then
;  458          0887  3          ! we are done with the module.
;  459          0888  3
;  460          0889  3          if .module_header then
;  461          0890  3                  anl$format_error(anlobj$_objeommissing)
;  462          0891  3          else
;  463          0892  4                  within_module = not (.scanp[obj$b_rectyp] eqlu obj$c_eom or
;  464          0893  3                                       .scanp[obj$b_rectyp] eqlu obj$c_eomw);
;  465          0894  2      );
;  466          0895  2
;  467          0896  2      ! Now we can collect some statistics.  For each record type, we will count
;  468          0897  2      ! the number of such records and add up the number of bytes.
;  469          0898  2
;  470          0899  2      increment (record_count[.scanp[obj$b_rectyp]]);
;  471          0900  2      byte_total[.scanp[obj$b_rectyp]] = .byte_total[.scanp[obj$b_rectyp]] + .record_dsc[len];
;  472          0901  2
;  473          0902  2      return;
;  474          0903  2
;  475          0904  1      end;


                                001C 00000          .ENTRY  ANL$OBJECT_RECORD_LINE, Save R2,R3,R4       ; 0849
                          54    0000' CF   9E 00002       MOVAB   WITHIN_MODULE, R4                           ;
                          53      0C AC   D0 00007       MOVL    THE_RECORD, R3                              ; 0852
                          7E         63   3C 0000B       MOVZWL  (R3)+, -(SP)                                ; 0862
                          7E      04 AC   7D 0000E       MOVQ    RECORD_MSG, -(SP)                           ;
                          7E         04   7D 00012       MOVQ    #4, -(SP)                                   ;
                  0000G   CF         05   FB 00015       CALLS   #5, ANL$FORMAT_LINE                         ;
                          50      04 A3   D0 0001A       MOVL    4(R3), SCANP                                ; 0867
                                  51   94 0001E       CLRB    MODULE_HEADER                               ; 0868
                          52         60   9A 00020       MOVZBL  (SCANP)+, R2                                ; 0869
                                  0D   12 00023       BNEQ    1$                                          ;
                          02         63   B1 00025       CMPW    (R3), #2                                    ; 0870
                                  08   1F 00028       BLSSU   1$                                          ;
                                  01 A0   95 0002A       TSTB    1(SCANP)                                    ; 0871
                                  03   12 0002D       BNEQ    1$                                          ;
                          51         01   90 0002F       MOVB    #1, MODULE_HEADER                           ; 0872
                          13         64   E8 00032 1$:    BLBS    WITHIN_MODULE, 3$                           ; 0874
                          0B         51   E8 00035       BLBS    MODULE_HEADER, 2$                           ; 0879
                              00000000G   8F   DD 00038       PUSHL   #ANLOBJ$_OBJMHDMISSING                      ; 0880
                  0000G   CF         01   FB 0003E       CALLS   #1, ANL$FORMAT_ERROR                        ;
                          64         01   90 00043 2$:    MOVB    #1, WITHIN_MODULE                           ; 0881
                                  28   11 00046       BRB     7$                                          ; 0874
                          0D         51   E9 00048 3$:    BLBC    MODULE_HEADER, 4$                           ; 0889
                              00000000G   8F   DD 0004B       PUSHL   #ANLOBJ$_OBJEOMMISSING                      ; 0890
                  0000G   CF         01   FB 00051       CALLS   #1, ANL$FORMAT_ERROR                        ;
```

```
                                        18  11 00056           BRB     7$                                                      ;
                                        51  D4 00058 4$:      CLRL    R1                                                      ; 0892
                                    03  52  91 0005A           CMPB    R2, #3                                                  ;
                                        02  12 0005D           BNEQ    5$                                                      ;
                                        51  D6 0005F           INCL    R1                                                      ;
                                        50  D4 00061 5$:      CLRL    R0                                                      ; 0893
                                    07  52  91 00063           CMPB    R2, #7                                                  ;
                                        02  12 00066           BNEQ    6$                                                      ;
                                        50  D6 00068           INCL    R0                                                      ;
                                    50  51  C8 0006A 6$:      BISL2   R1, R0                                                  ;
                                    64  50  92 0006D           MCOMB   R0, WITHIN_MODULE                                       ; 0892
                                 04 A442  D6 00070 7$:      INCL    RECORD_COUNT[R2]                                        ; 0899
                                    50  63  3C 00074           MOVZWL  (R3), R0                                                ; 0900
                                24 A442  50  C0 00077           ADDL2   R0, BYTE_TOTAL[R2]                                      ;
                                            04 0007C           RET                                                             ; 0904

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 026C
```

```
;  477          0905  1 %sbttl 'ANL$OBJECT_STATISTICS - Print Summary Statistics'
;  478          0906  1 !++
;  479          0907  1 ! Functional Description:
;  480          0908  1 !       This routine is called to print the summary statistics with
;  481          0909  1 !       record counts and byte totals.
;  482          0910  1 !
;  483          0911  1 ! Formal Parameters:
;  484          0912  1 !       none
;  485          0913  1 !
;  486          0914  1 ! Implicit Inputs:
;  487          0915  1 !       global data
;  488          0916  1 !
;  489          0917  1 ! Implicit Outputs:
;  490          0918  1 !       global data
;  491          0919  1 !
;  492          0920  1 ! Returned Value:
;  493          0921  1 !       none
;  494          0922  1 !
;  495          0923  1 ! Side Effects:
;  496          0924  1 !
;  497          0925  1 !--
;  498          0926  1
;  499          0927  1
;  500          0928  2 global routine anl$object_statistics: novalue = begin
;  501          0929  2
;  502          0930  2 own
;  503          0931  2         type_msg: vector[obj$c_maxrectyp+1,long] initial(
;  504          0932  2                                 uplit byte(%ascic 'HDR'),
;  505          0933  2                                 uplit byte(%ascic 'GSD'),
;  506          0934  2                                 uplit byte(%ascic 'TIR'),
;  507          0935  2                                 uplit byte(%ascic 'EOM'),
;  508          0936  2                                 uplit byte(%ascic 'DBG'),
;  509          0937  2                                 uplit byte(%ascic 'TBT'),
;  510          0938  2                                 uplit byte(%ascic 'LNK'),
;  511          0939  2                                 uplit byte(%ascic 'EOMW'));
;  512          0940  2
;  513          0941  2 local
;  514          0942  2         i: long,
;  515          0943  2         total_record_count: long,
;  516          0944  2         total_byte_total: long;
;  517          0945  2
;  518          0946  2
;  519          0947  2 !  First we print some heading lines.
;  520          0948  2
;  521          0949  2 anl$format_line(0,0,anlobj$_objstatheading1);
;  522          0950  2 anl$report_line(0);
;  523          0951  2 anl$format_line(0,0,anlobj$_objstatheading2);
;  524          0952  2 anl$report_line(0);
;  525          0953  2
;  526          0954  2 ! Now we loop through the statistics vectors and print a line for each one.
;  527          0955  2 ! We also total the record count and byte total.
;  528          0956  2
;  529          0957  2 total_record_count = total_byte_total = 0;
;  530          0958  3 incru i from 0 to obj$c_maxrectyp do (
;  531          0959  3         anl$format_line(0,0,anlobj$_objstatline,.type_msg[.i],.record_count[.i],.byte_total[.i]);
;  532          0960  3         total_record_count = .total_record_count + .record_count[.i];
;  533          0961  3         total_byte_total   = .total_byte_total   + .byte_total[.i];
```

```
;  534          0962  2 );
;  535          0963  2
;  536          0964  2 ! Now we can print the totals.
;  537          0965  2
;  538          0966  2 anl$report_line(0);
;  539          0967  2 anl$format_line(0,0,anlobj$_objstattotal,.total_record_count,.total_byte_total);
;  540          0968  2 anl$report_line(0);
;  541          0969  2 anl$report_line(0);
;  542          0970  2
;  543          0971  2 ! Finally, clear the statistics vectors for the next module.
;  544          0972  2
;  545          0973  2 ch$fill(%x'00', %allocation(record_count),record_count);
;  546          0974  2 ch$fill(%x'00', %allocation(byte_total),  byte_total);
;  547          0975  2
;  548          0976  2 return;
;  549          0977  2
;  550          0978  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    52  44  48  03  000DC P.ABH:  .ASCII  <3>\HDR\                                    ;
                                    44  53  47  03  000E0 P.ABI:  .ASCII  <3>\GSD\                                    ;
                                    52  49  54  03  000E4 P.ABJ:  .ASCII  <3>\TIR\                                    ;
                                    4D  4F  45  03  000E8 P.ABK:  .ASCII  <3>\EOM\                                    ;
                                    47  42  44  03  000EC P.ABL:  .ASCII  <3>\DBG\                                    ;
                                    54  42  54  03  000F0 P.ABM:  .ASCII  <3>\TBT\                                    ;
                                    4B  4E  4C  03  000F4 P.ABN:  .ASCII  <3>\LNK\                                    ;
                                57  4D  4F  45  04  000F8 P.ABO:  .ASCII  <4>\EOMW\                                   ;

                                                        .PSECT  $OWN$,NOEXE,2

                                                00173           .BLKB   1
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00174 TYPE_MSG:
                                                        .ADDRESS P.ABH, P.ABI, P.ABJ, P.ABK, P.ABL, -           ;
                                00000000' 00000000' 0018C               P.ABM, P.ABN, P.ABO                     ;


                                                        .PSECT  $CODE$,NOWRT,2

                                                01FC 00000      .ENTRY  ANL$OBJECT_STATISTICS, Save R2,R3,R4,R5,R6,-; 0928
                                                                        R7,R8                                       ;
                                    58      0000G   CF  9E 00002        MOVAB   ANL$FORMAT_LINE, R8                 ;
                                    57      0000G   CF  9E 00007        MOVAB   ANL$REPORT_LINE, R7                 ;
                                    56      0000'   CF  9E 0000C        MOVAB   BYTE_TOTAL, R6                      ;
                                        00000000G   8F  DD 00011        PUSHL   #ANLOBJ$_OBJSTATHEADING1            ; 0949
                                                    7E  7C 00017        CLRQ    -(SP)                               ;
                                    68              03  FB 00019        CALLS   #3, ANL$FORMAT_LINE                 ;
                                                    7E  D4 0001C        CLRL    -(SP)                               ; 0950
                                    67              01  FB 0001E        CALLS   #1, ANL$REPORT_LINE                 ;
                                        00000000G   8F  DD 00021        PUSHL   #ANLOBJ$_OBJSTATHEADING2            ; 0951
                                                    7E  7C 00027        CLRQ    -(SP)                               ;
                                    68              03  FB 00029        CALLS   #3, ANL$FORMAT_LINE                 ;
                                                    7E  D4 0002C        CLRL    -(SP)                               ; 0952
                                    67              01  FB 0002E        CALLS   #1, ANL$REPORT_LINE                 ;
```

```
                                                53  7C 00031          CLRQ    TOTAL_BYTE_TOTAL                                    ; 0957
                                                52  D4 00033          CLRL    I                                                   ; 0958
                                              6642  DD 00035 1$:       PUSHL   BYTE_TOTAL[I]                                       ; 0959
                                            E0 A642  DD 00038          PUSHL   RECORD_COUNT[I]                                     ;
                                         0128 C642  DD 0003C          PUSHL   TYPE_MSG[I]                                         ;
                                    00000000G   8F  DD 00041          PUSHL   #ANLOBJ$_OBJSTATLINE                                ;
                                                7E  7C 00047          CLRQ    -(SP)                                               ;
                                          68        06  FB 00049          CALLS   #6, ANL$FORMAT_LINE                             ;
                                          54      E0 A642  C0 0004C          ADDL2   RECORD_COUNT[I], TOTAL_RECORD_COUNT          ; 0960
                                          53      6642  C0 00051          ADDL2   BYTE_TOTAL[I], TOTAL_BYTE_TOTAL             ; 0961
                                                52  D6 00055          INCL    I                                                   ; 0958
                                          07        52  D1 00057          CMPL    I, #7                                           ;
                                                D9  1B 0005A          BLEQU   1$                                                  ;
                                                7E  D4 0005C          CLRL    -(SP)                                               ; 0966
                                          67        01  FB 0005E          CALLS   #1, ANL$REPORT_LINE                             ;
                                                53  DD 00061          PUSHL   TOTAL_BYTE_TOTAL                                    ; 0967
                                                54  DD 00063          PUSHL   TOTAL_RECORD_COUNT                                  ;
                                    00000000G   8F  DD 00065          PUSHL   #ANLOBJ$_OBJSTATTOTAL                               ;
                                                7E  7C 0006B          CLRQ    -(SP)                                               ;
                                          68        05  FB 0006D          CALLS   #5, ANL$FORMAT_LINE                             ;
                                                7E  D4 00070          CLRL    -(SP)                                               ; 0968
                                          67        01  FB 00072          CALLS   #1, ANL$REPORT_LINE                             ;
                                                7E  D4 00075          CLRL    -(SP)                                               ; 0969
                                          67        01  FB 00077          CALLS   #1, ANL$REPORT_LINE                             ;
          20              00          6E        00  2C 0007A          MOVC5   #0, (SP), #0, #32, RECORD_COUNT                     ; 0973
                                            E0    A6        0007F                                                               ;
          20              00          6E        00  2C 00081          MOVC5   #0, (SP), #0, #32, BYTE_TOTAL                       ; 0974
                                                66      00086                                                                   ;
                                                    04  00087          RET                                                     ; 0978

; Routine Size:  136 bytes,    Routine Base:  $CODE$ + 02E9

;  551          0979  1
;  552          0980  0 end eludom



;                                       PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
; $GLOBAL$                        1  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
; $OWN$                         404  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
; $PLIT$                        253  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
; $CODE$                        881  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                   Library Statistics
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
```

```
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1          9776        23         0       581          00:01.0
;
;
;
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OBJDRIVE/OBJ=OBJ$:OBJDRIVE MSRC$:OBJDRIVE/UPDATE=(ENH$:OBJDRIVE)
;
; Size:          881 code + 658 data bytes
; Run Time:        00:18.2
; Elapsed Time:    00:46.4
; Lines/CPU Min:    3228
; Lexemes/CPU-Min: 15337
; Memory Used:  177 pages
; Compilation Complete
```