```
   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
```

```
EEEEEEEEEE  XX      XX  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  UU      UU  FFFFFFFFFF  FFFFFFFFFF
EEEEEEEEEE  XX      XX  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT  UU      UU  FFFFFFFFFF  FFFFFFFFFF
EE          XX      XX  EE          SS              TT      UU      UU  FF          FF
EE          XX      XX  EE          SS              TT      UU      UU  FF          FF
EE            XX  XX    EE          SS              TT      UU      UU  FF          FF
EE            XX  XX    EE          SS              TT      UU      UU  FF          FF
EEEEEEEE        XX      EEEEEEEE      SSSSSS        TT      UU      UU  FFFFFFFF    FFFFFFFF
EEEEEEEE        XX      EEEEEEEE      SSSSSS        TT      UU      UU  FFFFFFFF    FFFFFFFF
EE            XX  XX    EE                  SS      TT      UU      UU  FF          FF
EE            XX  XX    EE                  SS      TT      UU      UU  FF          FF
EE          XX      XX  EE                  SS      TT      UU      UU  FF          FF          ....
EE          XX      XX  EE                  SS      TT      UU      UU  FF          FF          ....
EEEEEEEEEE  XX      XX  EEEEEEEEEE  SSSSSSSS        TT      UUUUUUUUUU  FF          FF          ....
EEEEEEEEEE  XX      XX  EEEEEEEEEE  SSSSSSSS        TT      UUUUUUUUUU  FF          FF          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;    1          0001  0 %title 'EXESTUFF - Analyze Various Parts of an Image'
;    2          0002  0         module exestuff (
;    3          0003  1                         ident='V04-001') = 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, Analyze Parts of an Image
;   32          0032  1 !
;   33          0033  1 ! Abstract:     This module is responsible for analyzing various parts of
;   34          0034  1 !               an image, including the header, patch text, and global
;   35          0035  1 !               symbol table.
;   36          0036  1 !
;   37          0037  1 !
;   38          0038  1 ! Environment:
;   39          0039  1 !
;   40          0040  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 31 March 1981
;   41          0041  1 !
;   42          0042  1 ! Modified By:
;   43          0043  1 !
;   44          0044  1 !       V04-001 MSH0074         Michael S. Harvey       7-Sep-1984
;   45          0045  1 !               Recognize global demand zero ISDs when validating
;   46          0046  1 !               the ISD's length.
;   47          0047  1 !
;   48          0048  1 !       V03-008 ROP0022         Robert Posniak          14-JUL-1984
;   49          0049  1 !               Shift proper field for ISD base virtual
;   50          0050  1 !               address output.
;   51          0051  1 !
;   52          0052  1 !       V03-007 ROP0008         Robert Posniak          14-JUN-1984
;   53          0053  1 !               Change allocation of local_described_buffers from
;   54          0054  1 !               80 to 512.
;   55          0055  1 !
;   56          0056  1 !       V03-006 MCN0168         Maria del C. Nasr       08-May-1984
;   57          0057  1 !               If the image being analyzed was created by V3 or earlier,
```

```
;    58          0058  1 !               then use old offsets to get image name and identification
;    59          0059  1 !               information.
;    60          0060  1 !
;    61          0061  1 !       V03-005 MCN0158         Maria del C. Nasr       22-Mar-1984
;    62          0062  1 !               Use SHL$C_MAXNAMLNG as the image name length to pass
;    63          0063  1 !               to ANL$CHECK_SYMBOL.  Also, eliminate declaration of
;    64          0064  1 !               local loop counter.
;    65          0065  1 !
;    66          0066  1 !       V03-004 LJA0115         Laurie J. Anderson      2-Mar-1984
;    67          0067  1 !               Move the variable 'alias' from local (stack) storage to
;    68          0068  1 !               own storage.  This masks the problem that if you say:
;    69          0069  1 !               anal/image image1,image2 the second image gets the error
;    70          0070  1 !               "not a VAX/VMS image".  Do not know why, except has to
;    71          0071  1 !               to do with the stack.
;    72          0072  1 !
;    73          0073  1 !       V03-003 LJA0106         Laurie J. Anderson      26-Jan-1984
;    74          0074  1 !               1) Change the calls to ANL$GET_IMAGE_BLOCK to the new image
;    75          0075  1 !                  decode routines.
;    76          0076  1 !               2) Check for header block count of 0. Return error if so.
;    77          0077  1 !               3) Also, print out any indirect message filenames when
;    78          0078  1 !                  processing the ISD's.
;    79          0079  1 !               4) Plus in answer to SPR 11-62167, the maximum number of
;    80          0080  1 !                  characters in the patch text is increased from 80 to
;    81          0081  1 !                  something more reasonable, 255.
;    82          0082  1 !
;    83          0083  1 !       V03-002 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
;    84          0084  1 !               Change the message prefix to ANLOBJ$_ to ensure that
;    85          0085  1 !               message symbols are unique across all ANALYZEs.  This
;    86          0086  1 !               is necessitated by the new merged message files.
;    87          0087  1 !
;    88          0088  1 !       V03-001 JWT0075         Jim Teague              14-Dec-1982
;    89          0089  1 !               Update to accomodate changes in image header: 1)CLI images,
;    90          0090  1 !               2)IHD$V_DBGDMT bit, 3)IHS$L_DMTVBN, 4)IHS$L_DMTBYTES.
;    91          0091  1 !
;    92          0092  1 !--
```

```
;    94          0093  1 %sbttl 'Module Declarations'
;    95          0094  1 !
;    96          0095  1 ! Libraries and Requires:
;    97          0096  1 !
;    98          0097  1
;    99          0098  1 library 'lib';
;   100          0099  1 require 'imgmsgdef';
;   101          0185  1 require 'objexereq';
;   102          0621  1
;   103          0622  1 !
;   104          0623  1 ! Table of Contents:
;   105          0624  1 !
;   106          0625  1
;   107          0626  1 forward routine
;   108          0627  1         anl$image_header,
;   109          0628  1         anl$image_isd: novalue,
;   110          0629  1         anl$image_patch_text,
;   111          0630  1         anl$image_gst;
;   112          0631  1
;   113          0632  1 !
;   114          0633  1 ! External References:
;   115          0634  1 !
;   116          0635  1
;   117          0636  1 external routine
;   118          0637  1         anl$check_flags,
;   119          0638  1         anl$check_symbol,
;   120          0639  1         anl$format_error,
;   121          0640  1         anl$format_flags,
;   122          0641  1         anl$format_hex,
;   123          0642  1         anl$format_line,
;   124          0643  1         anl$get_image_block,
;   125          0644  1         anl$object_eom,
;   126          0645  1         anl$object_gsd,
;   127          0646  1         anl$object_hdr,
;   128          0647  1         anl$interact,
;   129          0648  1         anl$object_record_size,
;   130          0649  1         anl$report_line,
;   131          0650  1         anl$report_page,
;   132          0651  1         anl$get_image_header,
;   133          0652  1         anl$get_isd;
;   134          0653  1
;   135          0654  1 external
;   136          0655  1         anl$gb_interactive: byte;
;   137          0656  1
;   138          0657  1 !
;   139          0658  1 ! Own Variables:
;   140          0659  1 !
;   141          0660  1 ! The following table defines the match control values used throughout.
;   142          0661  1
;   143          0662  1 own
;   144          0663  1         match_control: vector[8,long] initial(
;   145          0664  1                                 uplit byte(%ascic 'ISD$K_MATALL'),
;   146          0665  1                                 uplit byte(%ascic 'ISD$K_MATEQU'),
;   147          0666  1                                 uplit byte(%ascic 'ISD$K_MATLEQ'),
;   148          0667  1                                 uplit byte(%ascic 'ISD$K_MATNEV'));
```

```
;   150          0668  1 %sbttl 'ANL$IMAGE_HEADER - Analyze Image Header'
;   151          0669  1 !++
;   152          0670  1 ! Functional Description:
;   153          0671  1 !       This routine is responsible for analyzing an image header.  THis
;   154          0672  1 !       includes formatting it in the report and checking its contents.
;   155          0673  1 !
;   156          0674  1 ! Formal Parameters:
;   157          0675  1 !       image_base      Return starting address of image here.
;   158          0676  1 !       fixup_size      If a fixup section exists, return size here,
;   159          0677  1 !       fixup_vbn       and VBN here.
;   160          0678  1 !
;   161          0679  1 ! Implicit Inputs:
;   162          0680  1 !       global data
;   163          0681  1 !
;   164          0682  1 ! Implicit Outputs:
;   165          0683  1 !       global data
;   166          0684  1 !
;   167          0685  1 ! Returned Value:
;   168          0686  1 !       If interactive session: true if we are to continue, false if not.
;   169          0687  1 !
;   170          0688  1 ! Side Effects:
;   171          0689  1 !
;   172          0690  1 !--
;   173          0691  1
;   174          0692  1
;   175          0693  2 global routine anl$image_header(image_base,fixup_size,fixup_vbn) = begin
;   176          0694  2
;   177          0695  2 own
;   178          0696  2         link_flags_def: vector[7,long] initial(
;   179          0697  2                                         5,
;   180          0698  2                                         uplit byte(%ascic 'IHD$V_LNKDEBUG'),
;   181          0699  2                                         uplit byte(%ascic 'IHD$V_LNKNOTFR'),
;   182          0700  2                                         uplit byte(%ascic 'IHD$V_NOPOBUFS'),
;   183          0701  2                                         uplit byte(%ascic 'IHD$V_PICIMG'),
;   184          0702  2                                         uplit byte(%ascic 'IHD$V_POIMAGE'),
;   185          0703  2                                         uplit byte(%ascic 'IHD$V_DBGDMT')),
;   186          0704  2         alias : word;
;   187          0705  2
;   188          0706  2 local
;   189          0707  2         status: long,
;   190          0708  2         hp: ref block[,byte],
;   191          0709  2         sp: ref block[,byte],
;   192          0710  2         vbn: long,
;   193          0711  2         fixup_address: long;
;   194          0712  2
;   195          0713  2 ! Offsets to image name and identification information in images created by
;   196          0714  2 ! VMS V3.x or earlier.
;   197          0715  2 !
;   198          0716  2 macro
;   199          0717  2         IHI$_IMGNAM = 0,0,0,0 %,
;   200          0718  2         IHI$_IMGID = 16,0,0,0 %,
;   201          0719  2         IHI$_LINKTIME = 32,0,0,0 %,
;   202          0720  2         IHI$_LINKID = 40,0,0,0 %;
;   203          0721  2
;   204          0722  2 bind
;   205          0723  2         v3_majorid = uplit (%ascii'02'),        ! linker major id in V3
;   206          0724  2         v3_minorid = uplit (%ascii'04');        ! linker minor id in V3
```

```
;   207          0725  2
;   208          0726  2 ! We are going to analyze the image header.  Get it.
;   209          0727  2
;   210          0728  2 anl$format_line(0,0,anlobj$_exehdr);
;   211          0729  2 anl$report_line(-1);
;   212          0730  2
;   213          0731  2 status = anl$get_image_header(hp,alias);
;   214          0732  2
;   215          0733  2 ! If we couldn't get the first header block, or if it doesn't end with
;   216          0734  2 ! a %x'ffff' or %x'0003' or %x'0002', then this can't be a native image.
;   217          0735  2 !       -1 = produced by the VAX-11 Linker
;   218          0736  2 !       0  = RSX compatibility mode
;   219          0737  2 !       1  = Activate BPA
;   220          0738  2 !       2  = Name of image to activate is in image header
;   221          0739  2 !       3  = It's a CLI
;   222          0740  2 !
;   223          0741  2 if not .status or
;   224          0742  3    (.alias nequ %x'ffff' and .alias nequ %x'0003' and .alias nequ %x'0002')
;   225          0743  3 then (anl$format_error(anlobj$_exenotnative);
;   226          0744  2     return false;);
;   227          0745  2
;   228          0746  2 ! Begin with the fixed fields at the beginning of the header.
;   229          0747  2
;   230          0748  2 anl$format_line(3,1,anlobj$_exehdrfixed);
;   231          0749  2 anl$report_line(-1);
;   232          0750  2
;   233          0751  2 ! Analyze the image identification info.
;   234          0752  2
;   235          0753  2 anl$format_line(0,2,anlobj$_exehdrimageid,2,hp[ihd$w_majorid],2,hp[ihd$w_minorid]);
;   236          0754  2
;   237          0755  2 ! Analyze the header block count.  If the count is zero, this is a bad
;   238          0756  2 ! image.  The image activator will not activate it.
;   239          0757  2
;   240          0758  2 if .hp[ihd$b_hdrblkcnt] eqlu 0
;   241          0759  2 then
;   242          0760  2     anl$format_error(anlobj$_badhdrblkcount,.hp[ihd$b_hdrblkcnt])
;   243          0761  2 else
;   244          0762  2     anl$format_line(0,2,anlobj$_exehdrblkcount,.hp[ihd$b_hdrblkcnt]);
;   245          0763  2
;   246          0764  2 ! Analyze the image type code.  If shared, print the global section IDs and
;   247          0765  2 ! the match control.
;   248          0766  2
;   249          0767  2 selectoneu .hp[ihd$b_imgtype] of set
;   250          0768  2 [ihd$k_exe]:    anl$format_line(0,2,anlobj$_exehdrtypeexe);
;   251          0769  2
;   252          0770  3 [ihd$k_lim]:    (anl$format_line(2,2,anlobj$_exehdrtypelim);
;   253          0771  3                 anl$format_line(0,3,anlobj$_exehdrgblident,.hp[ihd$l_ident]);
;   254          0772  3                 selectoneu .hp[ihd$v_matchctl] of set
;   255          0773  3                 [isd$k_matall,
;   256          0774  3                  isd$k_matequ,
;   257          0775  3                  isd$k_matleq,
;   258          0776  3                  isd$k_matnev]: anl$format_line(0,3,anlobj$_exehdrmatch,
;   259          0777  3                                                 .match_control[.hp[ihd$v_matchctl]]);
;   260          0778  3                 [otherwise]:    anl$format_error(anlobj$_exebadmatch,.hp[ihd$v_matchctl]);
;   261          0779  2                 tes;);
;   262          0780  2
;   263          0781  2 [otherwise]:    anl$format_error(anlobj$_exebadtype,.hp[ihd$b_imgtype]);
```

```
;  264          0782  2 tes;
;  265          0783  2
;  266          0784  2 ! Analyze the I/O channel count.
;  267          0785  2
;  268          0786  2 if .hp[ihd$w_iochancnt] eqlu 0 then
;  269          0787  2         anl$format_line(0,2,anlobj$_exehdrchandef)
;  270          0788  2 else
;  271          0789  2         anl$format_line(0,2,anlobj$_exehdrchancount,.hp[ihd$w_iochancnt]);
;  272          0790  2
;  273          0791  2 ! Analyze the I/O section page count.
;  274          0792  2
;  275          0793  2 if .hp[ihd$w_imgiocnt] eqlu 0 then
;  276          0794  2         anl$format_line(0,2,anlobj$_exehdrpagedef)
;  277          0795  2 else
;  278          0796  2         anl$format_line(0,2,anlobj$_exehdrpagecount,.hp[ihd$w_imgiocnt]);
;  279          0797  2
;  280          0798  2 ! Analyze the linker-produced flags.  Don't get confused by the match control.
;  281          0799  2
;  282          0800  2 anl$format_flags(2,anlobj$_exehdrflags,.hp[ihd$l_lnkflags] and %x'00ffffff',link_flags_def);
;  283          0801  2 anl$check_flags(.hp[ihd$l_lnkflags] and %x'00ffffff',link_flags_def);
;  284          0802  2
;  285          0803  2 ! Analyze the system version, if specified.
;  286          0804  2
;  287          0805  2 if .hp[ihd$l_sysver] nequ 0 then
;  288          0806  2         anl$format_line(0,2,anlobj$_exehdrsysver,4,hp[ihd$l_sysver]);
;  289          0807  2
;  290          0808  2 ! If the fixed portion is long enough to accomodate a fixup section
;  291          0809  2 ! virtual address (V3A and later), then remember the address.
;  292          0810  2
;  293          0811  2 if .hp+.hp[ihd$w_activoff] gtra hp[ihd$l_iafva] then
;  294          0812  2         fixup_address = .hp[ihd$l_iafva]
;  295          0813  2 else
;  296          0814  2         fixup_address = 0;
;  297          0815  2
;  298          0816  2 ! If this is an interactive session, give the user a chance to quit.
;  299          0817  2
;  300          0818  2 if .anl$gb_interactive then
;  301          0819  2         if not anl$interact() then
;  302          0820  2                 return false;
```

```
;  304          0821  2 ! Now we are going to analyze the information in the activation section.
;  305          0822  2 ! It is always present.
;  306          0823  2
;  307          0824  2 anl$report_line(-1);
;  308          0825  2 anl$format_line(3,1,anlobj$_exehdractive);
;  309          0826  2 anl$report_line(-1);
;  310          0827  2
;  311          0828  2 sp = .hp + .hp[ihd$w_activoff];
;  312          0829  2
;  313          0830  2 ! Analyze the three transfer addresses.
;  314          0831  2
;  315          0832  2 anl$format_line(0,2,anlobj$_exehdrxfer1,.sp[iha$l_tfradr1]);
;  316          0833  2 anl$format_line(0,2,anlobj$_exehdrxfer2,.sp[iha$l_tfradr2]);
;  317          0834  2 anl$format_line(0,2,anlobj$_exehdrxfer3,.sp[iha$l_tfradr3]);
;  318          0835  2
;  319          0836  2 ! Make sure the thing ends with a trailing zero.
;  320          0837  2
;  321          0838  2 if .sp[12,0,32,0] nequ 0 then
;  322          0839  2         anl$format_error(anlobj$_exebadxfer0);
;  323          0840  2
;  324          0841  2 ! If this is an interactive session, give the user a chance to quit.
;  325          0842  2
;  326          0843  2 if .anl$gb_interactive then
;  327          0844  2         if not anl$interact() then
;  328          0845  2                 return false;
```

```
;  330          0846  2 ! Now we are going to analyze the stuff in the symbol table and debug section.
;  331          0847  2 ! It is always present.
;  332          0848  2
;  333          0849  2 anl$report_line(-1);
;  334          0850  2 anl$format_line(3,1,anlobj$_exehdrsymdbg);
;  335          0851  2 anl$report_line(-1);
;  336          0852  2
;  337          0853  2 sp = .hp + .hp[ihd$w_symdbgoff];
;  338          0854  2
;  339          0855  2 ! Analyze the debug symbol table VBN and block count.
;  340          0856  2
;  341          0857  2 anl$format_line(0,2,anlobj$_exehdrdst,.sp[ihs$l_dstvbn],.sp[ihs$w_dstblks]);
;  342          0858  2
;  343          0859  2 ! Analyze the global symbol table VBN and record count.
;  344          0860  2
;  345          0861  2 anl$format_line(0,2,anlobj$_exehdrgst,.sp[ihs$l_gstvbn],.sp[ihs$w_gstrecs]);
;  346          0862  2
;  347          0863  2 ! Analyze the Debugger DMT, if present
;  348          0864  2
;  349          0865  2 if .hp[ihd$v_dbgdmt]
;  350          0866  2 then
;  351          0867  2     anl$format_line(0,2,anlobj$_exehdrdmt,.sp[ihs$l_dmtvbn],.sp[ihs$l_dmtbytes]);
;  352          0868  2
;  353          0869  2 ! If this is an interactive session, give the user a chance to quit.
;  354          0870  2
;  355          0871  2 if .anl$gb_interactive then
;  356          0872  2         if not anl$interact() then
;  357          0873  2             return false;
```

```
;  359          0874  2 ! Now we are going to tackle the image identification section.
;  360          0875  2 ! It is always present.
;  361          0876  2
;  362          0877  2 anl$report_line(-1);
;  363          0878  2 anl$format_line(3,1,anlobj$_exehdrident);
;  364          0879  2 anl$report_line(-1);
;  365          0880  2
;  366          0881  2 sp = .hp + .hp[ihd$w_imgidoff];
;  367          0882  2
;  368          0883  3 begin
;  369          0884  3 local
;  370          0885  3         name_dsc: descriptor;
;  371          0886  3
;  372          0887  3 ! Analyze the image name, image identification, date and time of linking,
;  373          0888  3 ! and linker identification.  If the image was linked with V3 linker, then
;  374          0889  3 ! use old offsets to get information, otherwise use latest values.
;  375          0890  3 !
;  376          0891  3
;  377          0892  3 if .hp[ihd$w_majorid] gtr .v3_majorid
;  378          0893  3   or .hp[ihd$w_minorid] gtr .v3_minorid
;  379          0894  3 then                                          ! after V3 linker
;  380          0895  4         begin
;  381          0896  4         anl$format_line(0,2,anlobj$_exehdrname,sp[ihi$t_imgnam]);
;  382          0897  4         build_descriptor(name_dsc,.sp[0,0,8,0],sp[1,0,8,0]);
;  383          0898  4         anl$check_symbol(name_dsc, shl$c_maxnamlng);
;  384          0899  4         anl$format_line(0,2,anlobj$_exehdrfileid,sp[ihi$t_imgid]);
;  385          0900  4         anl$format_line(0,2,anlobj$_exehdrtime,sp[ihi$q_linktime]);
;  386          0901  4         anl$format_line(0,2,anlobj$_exehdrlinkid,sp[ihi$t_linkid]);
;  387          0902  4         end
;  388          0903  3 else                                          ! V3 or earlier
;  389          0904  4         begin
;  390          0905  4         anl$format_line(0,2,anlobj$_exehdrname,sp[ihi$_imgnam]);
;  391          0906  4         build_descriptor(name_dsc,.sp[0,0,8,0],sp[1,0,8,0]);
;  392          0907  4         anl$check_symbol(name_dsc, shl$c_maxnamlng);
;  393          0908  4         anl$format_line(0,2,anlobj$_exehdrfileid,sp[ihi$_imgid]);
;  394          0909  4         anl$format_line(0,2,anlobj$_exehdrtime,sp[ihi$_linktime]);
;  395          0910  4         anl$format_line(0,2,anlobj$_exehdrlinkid,sp[ihi$_linkid]);
;  396          0911  3         end;
;  397          0912  2 end;                        ! of local "name_dsc"
;  398          0913  2
;  399          0914  2
;  400          0915  2 ! If this is an interactive session, give the user a chance to quit.
;  401          0916  2
;  402          0917  2 if .anl$gb_interactive then
;  403          0918  2         if not anl$interact() then
;  404          0919  2                 return false;
```

```
;  406          0920  2 ! Now we are going to analyze the patch section.
;  407          0921  2 ! It may not necessarily exist.
;  408          0922  2
;  409          0923  2 anl$report_line(-1);
;  410          0924  2 anl$format_line(3,1,anlobj$_exehdrpatch);
;  411          0925  2 anl$report_line(-1);
;  412          0926  2
;  413          0927  3 if .hp[ihd$w_patchoff] nequ 0 then (
;  414          0928  3         sp = .hp + .hp[ihd$w_patchoff];
;  415          0929  3
;  416          0930  3         ! Begin with the Digital ECO bits.
;  417          0931  3
;  418          0932  3         anl$format_line(0,2,anlobj$_exehdrdececo,.sp[ihp$l_eco1],.sp[ihp$l_eco2],.sp[ihp$l_eco3]);
;  419          0933  3
;  420          0934  3         ! And the user ECO bits.
;  421          0935  3
;  422          0936  3         anl$format_line(0,2,anlobj$_exehdrusereco,.sp[ihp$l_eco4]);
;  423          0937  3
;  424          0938  3         ! Analyze the read/write and read-only patch area info.
;  425          0939  3
;  426          0940  3         anl$format_line(0,2,anlobj$_exehdrrwpatch,.sp[ihp$l_rw_patadr],.sp[ihp$l_rw_patsiz]);
;  427          0941  3         anl$format_line(0,2,anlobj$_exehdrropatch,.sp[ihp$l_ro_patadr],.sp[ihp$l_ro_patsiz]);
;  428          0942  3
;  429          0943  3         ! Now the VBN of the patch command text.
;  430          0944  3
;  431          0945  3         anl$format_line(0,2,anlobj$_exehdrtextvbn,.sp[ihp$l_patcomtxt]);
;  432          0946  3
;  433          0947  3         ! And the date of most recent patch.
;  434          0948  3
;  435          0949  3         anl$format_line(0,2,anlobj$_exehdrpatchdate,sp[ihp$q_patdate]);
;  436          0950  3
;  437          0951  3         ! If this is an interactive session, give the user a chance to quit.
;  438          0952  3
;  439          0953  3         if .anl$gb_interactive then
;  440          0954  3                 if not anl$interact() then
;  441          0955  3                         return false;
;  442          0956  3 ) else (
;  443          0957  3
;  444          0958  3         ! There is no patch section now.
;  445          0959  3
;  446          0960  3         anl$format_line(0,2,anlobj$_exehdrnopatch);
;  447          0961  2 );
```

```
;  449          0962  2 ! Analyze the image section descriptors.  These begin after all the above
;  450          0963  2 ! sections and can go on for multiple blocks.
;  451          0964  2 ! We also use this loop to search for the fixup section.  If we don't find
;  452          0965  2 ! one, we will inform the caller with zero fixup parameters.
;  453          0966  2
;  454          0967  2 .fixup_size = .fixup_vbn = 0;
;  455          0968  2
;  456          0969  2 anl$report_line(-1);
;  457          0970  2 anl$format_line(3,1,anlobj$_exehdrisd);
;  458          0971  2
;  459          0972  2 vbn = 1;
;  460          0973  3 incru isd from 1 do (
;  461          0974  3
;  462          0975  3         ! First we see if we have run out of ISDs in this block.  If so,
;  463          0976  3         ! we advance to the next block.  This routine keeps track of how
;  464          0977  3         ! many ISD's we've looked at so far.
;  465          0978  3
;  466          0979  3         status = anl$get_isd(hp);
;  467          0980  3
;  468          0981  3         ! Now we see if we are all done with the ISDs.  The return status
;  469          0982  3         ! is IMG$_ENDOFHDR
;  470          0983  3
;  471          0984  3 exitif (.status eqlu img$_endofhdr);
;  472          0985  3
;  473          0986  3         increment (vbn);
;  474          0987  4         if not .status then (
;  475          0988  4                 anl$format_error(.status);
;  476          0989  4 exitloop;
;  477          0990  3         );
;  478          0991  3         sp = .hp;
;  479          0992  3
;  480          0993  3
;  481          0994  3         ! Seems we have an ISD to analyze.  Make sure it fits completely
;  482          0995  3         ! within the block.
;  483          0996  3
;  484          0997  4         if .sp[isd$w_size] gtru .hp+512-.sp then (
;  485          0998  4                 anl$format_error(anlobj$_exehdrisdlong);
;  486          0999  4 exitloop;
;  487          1000  3         );
;  488          1001  3
;  489          1002  3         ! Format and analyze the ISD.
;  490          1003  3
;  491          1004  3         anl$image_isd(.sp,.isd);
;  492          1005  3
;  493          1006  3         ! If this is the first ISD, then we want to return its base address,
;  494          1007  3         ! which is the starting address of the entire image.
;  495          1008  3
;  496          1009  3         if .isd eqlu 1 then
;  497          1010  3                 .image_base = .sp[isd$v_vpn]^9;
;  498          1011  3
;  499          1012  3         ! If we have a fixup section, let's see if this is it.  If so,
;  500          1013  3         ! return its size and VBN.  If they are bad, tell the user.
;  501          1014  3
;  502          1015  3         if .fixup_address nega 0 then
;  503          1016  3                 if .fixup_address eqla .sp[isd$v_vpg]^9 then
;  504          1017  3                         if .sp[isd$w_pagcnt] eqlu 0 or .sp[isd$l_vbn] eqlu 0 then
;  505          1018  3                                 anl$format_error(anlobj$_exebadfixupisd)
```

```
;  506          1019  4                          else (
;  507          1020  4                                  .fixup_size = .sp[isd$w_pagcnt];
;  508          1021  4                                  .fixup_vbn  = .sp[isd$l_vbn];
;  509          1022  3                          );
;  510          1023  3
;  511          1024  3                  ! If this is an interactive session, give the user a chance to quit.
;  512          1025  3
;  513          1026  3                  if .anl$gb_interactive then
;  514          1027  3                          if not anl$interact() then
;  515          1028  3                                  return false;
;  516          1029  3
;  517          1030  2          );
;  518          1031  2
;  519          1032  2          return true;
;  520          1033  2
;  521          1034  1 end;


;                                                                       .TITLE  EXESTUFF EXESTUFF - Analyze Various Parts of an
                                                                                        Image
                                                                       .IDENT  \V04-001\

                                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

        4C  4C  41  54  41  4D  5F  4B  24  44  53  49  0C  00000 P.AAA:  .ASCII  <12>\ISD$K_MATALL\                          ;
        55  51  45  54  41  4D  5F  4B  24  44  53  49  0C  0000D P.AAB:  .ASCII  <12>\ISD$K_MATEQU\                          ;
        51  45  4C  54  41  4D  5F  4B  24  44  53  49  0C  0001A P.AAC:  .ASCII  <12>\ISD$K_MATLEQ\                          ;
        56  45  4E  54  41  4D  5F  4B  24  44  53  49  0C  00027 P.AAD:  .ASCII  <12>\ISD$K_MATNEV\                          ;
47  55  42  45  44  4B  4E  4C  5F  56  24  44  48  49  0E  00034 P.AAE:  .ASCII  <14>\IHD$V_LNKDEBUG\                        ;
52  46  54  4F  4E  4B  4E  4C  5F  56  24  44  48  49  0E  00043 P.AAF:  .ASCII  <14>\IHD$V_LNKNOTFR\                        ;
53  46  55  42  30  50  4F  4E  5F  56  24  44  48  49  0E  00052 P.AAG:  .ASCII  <14>\IHD$V_NOPOBUFS\                        ;
        47  4D  49  43  49  50  5F  56  24  44  48  49  0C  00061 P.AAH:  .ASCII  <12>\IHD$V_PICIMG\                          ;
    45  47  41  4D  49  30  50  5F  56  24  44  48  49  0D  0006E P.AAI:  .ASCII  <13>\IHD$V_POIMAGE\                         ;
        54  4D  44  47  42  44  5F  56  24  44  48  49  0C  0007C P.AAJ:  .ASCII  <12>\IHD$V_DBGDMT\                          ;
                                                    00089           .BLKB   3
                                                00  00  32  30  0008C P.AAK:  .ASCII  \02\<0><0>                            ;
                                                00  00  34  30  00090 P.AAL:  .ASCII  \04\<0><0>                            ;

                                                                       .PSECT  $OWN$,NOEXE,2

                    00000000' 00000000' 00000000' 00000000' 00000 MATCH_CONTROL:
                                                                       .ADDRESS P.AAA, P.AAB, P.AAC, P.AAD                 ;
                                                            00010      .BLKB   16
                                                    00000005 00020 LINK_FLAGS_DEF:
                                                                       .LONG   5                                           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00024      .ADDRESS P.AAE, P.AAF, P.AAG, P.AAH, P.AAI, P.AAJ   ;
                                                            0003C ALIAS:   .BLKB   2

                                                          V3_MAJORID=           P.AAK
                                                          V3_MINORID=           P.AAL
                                                                       .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$_FACILITY
                                .EXTRN  ANL$CHECK_FLAGS
                                .EXTRN  ANL$CHECK_SYMBOL
                                .EXTRN  ANL$FORMAT_ERROR
                                .EXTRN  ANL$FORMAT_FLAGS
                                .EXTRN  ANL$FORMAT_HEX, ANL$FORMAT_LINE
                                .EXTRN  ANL$GET_IMAGE_BLOCK
                                .EXTRN  ANL$OBJECT_EOM, ANL$OBJECT_GSD
                                .EXTRN  ANL$OBJECT_HDR, ANL$INTERACT
                                .EXTRN  ANL$OBJECT_RECORD_SIZE
                                .EXTRN  ANL$REPORT_LINE
                                .EXTRN  ANL$REPORT_PAGE
                                .EXTRN  ANL$GET_IMAGE_HEADER
                                .EXTRN  ANL$GET_ISD, ANL$GB_INTERACTIVE

                                .PSECT  $CODE$,NOWRT,2

                    0FFC 00000  .ENTRY  ANL$IMAGE_HEADER, Save R2,R3,R4,R5,R6,R7,-  ; 0693
                                        R8,R9,R10,R11
5B      0000G   CF  9E 00002    MOVAB   ANL$GB_INTERACTIVE, R11                     ;
5A      0000G   CF  9E 00007    MOVAB   ANL$FORMAT_ERROR, R10                       ;
59      0000G   CF  9E 0000C    MOVAB   ANL$REPORT_LINE, R9                         ;
58      0000G   CF  9E 00011    MOVAB   ANL$FORMAT_LINE, R8                         ;
5E              0C  C2 00016    SUBL2   #12, SP                                     ;
  00000000G     8F  DD 00019    PUSHL   #ANLOBJ$_EXEHDR                             ; 0728
                7E  7C 0001F    CLRQ    -(SP)                                       ;
```

```
                                 68              03  FB 00021          CALLS   #3, ANL$FORMAT_LINE
                                 7E              01  CE 00024          MNEGL   #1, -(SP)                                    ; 0729
                                 69              01  FB 00027          CALLS   #1, ANL$REPORT_LINE
                                           0000'  CF  9F 0002A          PUSHAB  ALIAS                                       ; 0731
                                             04   AE  9F 0002E          PUSHAB  HP
                          0000G  CF              02  FB 00031          CALLS   #2, ANL$GET_IMAGE_HEADER
                                 57              50  D0 00036          MOVL    R0, STATUS
                                 16              57  E9 00039          BLBC    STATUS, 1$                                  ; 0741
                                 50        0000' CF  3C 0003C          MOVZWL  ALIAS, R0                                   ; 0742
                             FFFF 8F             50  B1 00041          CMPW    R0, #65535
                                                 16  13 00046          BEQL    2$
                                 03              50  B1 00048          CMPW    R0, #3
                                                 11  13 0004B          BEQL    2$
                                 02              50  B1 0004D          CMPW    R0, #2
                                                 0C  13 00050          BEQL    2$
                                     00000000G   8F  DD 00052  1$:      PUSHL   #ANLOBJ$_EXENOTNATIVE                       ; 0743
                                 6A              01  FB 00058          CALLS   #1, ANL$FORMAT_ERROR
                                               0471  31 0005B          BRW     41$                                         ; 0744
                                     00000000G   8F  DD 0005E  2$:      PUSHL   #ANLOBJ$_EXEHDRFIXED                        ; 0748
                                                 01  DD 00064          PUSHL   #1
                                                 03  DD 00066          PUSHL   #3
                                 68              03  FB 00068          CALLS   #3, ANL$FORMAT_LINE
                                 7E              01  CE 0006B          MNEGL   #1, -(SP)                                   ; 0749
                                 69              01  FB 0006E          CALLS   #1, ANL$REPORT_LINE
                                 53              6E  D0 00071          MOVL    HP, R3                                      ; 0753
                                           0E    A3  9F 00074          PUSHAB  14(R3)
                                                 02  DD 00077          PUSHL   #2
                                           0C    A3  9F 00079          PUSHAB  12(R3)
                                                 02  DD 0007C          PUSHL   #2
                                     00000000G   8F  DD 0007E          PUSHL   #ANLOBJ$_EXEHDRIMAGEID
                                                 02  DD 00084          PUSHL   #2
                                                 7E  D4 00086          CLRL    -(SP)
                                 68              07  FB 00088          CALLS   #7, ANL$FORMAT_LINE
                                 50        10    A3  9A 0008B          MOVZBL  16(R3), R0                                  ; 0758
                                                 0D  12 0008F          BNEQ    3$
                                                 50  DD 00091          PUSHL   R0                                          ; 0760
                                     00000000G   8F  DD 00093          PUSHL   #ANLOBJ$_BADHDRBLKCOUNT
                                 6A              02  FB 00099          CALLS   #2, ANL$FORMAT_ERROR
                                                 0F  11 0009C          BRB     4$
                                                 50  DD 0009E  3$:      PUSHL   R0                                          ; 0762
                                     00000000G   8F  DD 000A0          PUSHL   #ANLOBJ$_EXEHDRBLKCOUNT
                                                 02  DD 000A6          PUSHL   #2
                                                 7E  D4 000A8          CLRL    -(SP)
                                 68              04  FB 000AA          CALLS   #4, ANL$FORMAT_LINE
                                 50        11    A3  9A 000AD  4$:      MOVZBL  17(R3), R0                                  ; 0767
                                 01              50  91 000B1          CMPB    R0, #1                                      ; 0768
                                                 0F  12 000B4          BNEQ    5$
                                     00000000G   8F  DD 000B6          PUSHL   #ANLOBJ$_EXEHDRTYPEEXE
                                                 02  DD 000BC          PUSHL   #2
                                                 7E  D4 000BE          CLRL    -(SP)
                                 68              03  FB 000C0          CALLS   #3, ANL$FORMAT_LINE
                                                 56  11 000C3          BRB     9$
                                 02              50  91 000C5  5$:      CMPB    R0, #2                                      ; 0770
                                                 46  12 000C8          BNEQ    7$
                                     00000000G   8F  DD 000CA          PUSHL   #ANLOBJ$_EXEHDRTYPELIM
                                                 02  DD 000D0          PUSHL   #2
                                                 02  DD 000D2          PUSHL   #2
```

```
                                68          03  FB 000D4          CALLS   #3, ANL$FORMAT_LINE                                ;
                                      24    A3  DD 000D7          PUSHL   36(R3)                                             ; 0771
                                00000000G   8F  DD 000DA          PUSHL   #ANLOBJ$_EXEHDRGBLIDENT                            ;
                                            03  DD 000E0          PUSHL   #3                                                 ;
                                            7E  D4 000E2          CLRL    -(SP)                                              ;
                                68          04  FB 000E4          CALLS   #4, ANL$FORMAT_LINE                                ;
                     50         23    A3   03  00  EF 000E7          EXTZV   #0, #3, 35(R3), R0                             ; 0772
                                03          50  D1 000ED          CMPL    R0, #3                                             ; 0773
                                            14  1A 000F0          BGTRU   6$                                                 ;
                                      0000'CF40  DD 000F2          PUSHL   MATCH_CONTROL[R0]                                  ; 0777
                                00000000G   8F  DD 000F7          PUSHL   #ANLOBJ$_EXEHDRMATCH                               ; 0776
                                            03  DD 000FD          PUSHL   #3                                                 ;
                                            7E  D4 000FF          CLRL    -(SP)                                              ;
                                68          04  FB 00101          CALLS   #4, ANL$FORMAT_LINE                                ;
                                            15  11 00104          BRB     9$                                                 ;
                                            50  DD 00106 6$:      PUSHL   R0                                                 ; 0778
                                00000000G   8F  DD 00108          PUSHL   #ANLOBJ$_EXEBADMATCH                               ;
                                            08  11 0010E          BRB     8$                                                 ;
                                            50  DD 00110 7$:      PUSHL   R0                                                 ; 0781
                                00000000G   8F  DD 00112          PUSHL   #ANLOBJ$_EXEBADTYPE                                ;
                                6A          02  FB 00118 8$:      CALLS   #2, ANL$FORMAT_ERROR                               ;
                                      1C    A3  B5 0011B 9$:      TSTW    28(R3)                                             ; 0786
                                            0F  12 0011E          BNEQ    10$                                                ;
                                00000000G   8F  DD 00120          PUSHL   #ANLOBJ$_EXEHDRCHANDEF                             ; 0787
                                            02  DD 00126          PUSHL   #2                                                 ;
                                            7E  D4 00128          CLRL    -(SP)                                              ;
                                68          03  FB 0012A          CALLS   #3, ANL$FORMAT_LINE                                ;
                                            11  11 0012D          BRB     11$                                                ;
                                7E          1C  A3 3C 0012F 10$:  MOVZWL  28(R3), -(SP)                                     ; 0789
                                00000000G   8F  DD 00133          PUSHL   #ANLOBJ$_EXEHDRCHANCOUNT                           ;
                                            02  DD 00139          PUSHL   #2                                                 ;
                                            7E  D4 0013B          CLRL    -(SP)                                              ;
                                68          04  FB 0013D          CALLS   #4, ANL$FORMAT_LINE                                ;
                                      1E    A3  B5 00140 11$:      TSTW    30(R3)                                             ; 0793
                                            0F  12 00143          BNEQ    12$                                                ;
                                00000000G   8F  DD 00145          PUSHL   #ANLOBJ$_EXEHDRPAGEDEF                             ; 0794
                                            02  DD 0014B          PUSHL   #2                                                 ;
                                            7E  D4 0014D          CLRL    -(SP)                                              ;
                                68          03  FB 0014F          CALLS   #3, ANL$FORMAT_LINE                                ;
                                            11  11 00152          BRB     13$                                                ;
                                7E          1E  A3 3C 00154 12$:  MOVZWL  30(R3), -(SP)                                     ; 0796
                                00000000G   8F  DD 00158          PUSHL   #ANLOBJ$_EXEHDRPAGECOUNT                           ;
                                            02  DD 0015E          PUSHL   #2                                                 ;
                                            7E  D4 00160          CLRL    -(SP)                                              ;
                                68          04  FB 00162          CALLS   #4, ANL$FORMAT_LINE                                ;
                                      0000' CF  9F 00165 13$:      PUSHAB  LINK_FLAGS_DEF                                     ; 0800
                     7E         20    A3   18  00  EF 00169          EXTZV   #0, #24, 32(R3), -(SP)                         ;
                                00000000G   8F  DD 0016F          PUSHL   #ANLOBJ$_EXEHDRFLAGS                               ;
                                            02  DD 00175          PUSHL   #2                                                 ;
                                0000G       CF  04  FB 00177          CALLS   #4, ANL$FORMAT_FLAGS                           ;
                                      0000' CF  9F 0017C          PUSHAB  LINK_FLAGS_DEF                                     ; 0801
                     7E         20    A3   18  00  EF 00180          EXTZV   #0, #24, 32(R3), -(SP)                         ;
                                0000G       CF  02  FB 00186          CALLS   #2, ANL$CHECK_FLAGS                            ;
                                      28    A3  D5 0018B          TSTL    40(R3)                                             ; 0805
                                            12  13 0018E          BEQL    14$                                                ;
                                      28    A3  9F 00190          PUSHAB  40(R3)                                             ; 0806
                                            04  DD 00193          PUSHL   #4                                                 ;
```

```
                          00000000G  8F  DD 00195          PUSHL   #ANLOBJ$_EXEHDRSYSVER                  ;
                                     02  DD 0019B          PUSHL   #2                                      ;
                                     7E  D4 0019D          CLRL    -(SP)                                   ;
                                 68      05  FB 0019F      CALLS   #5, ANL$FORMAT_LINE                     ;
                                 52  02  A3  3C 001A2 14$: MOVZWL  2(R3), R2                             ; 0811
                                 52      53  C0 001A6      ADDL2   R3, R2                                  ;
                                 50  2C  A3  9E 001A9      MOVAB   44(R3), R0                              ;
                                 50      52  D1 001AD      CMPL    R2, R0                                  ;
                                         06  1B 001B0      BLEQU   15$                                     ;
                                 56  2C  A3  D0 001B2      MOVL    44(R3), FIXUP_ADDRESS                   ; 0812
                                         02  11 001B6      BRB     16$                                     ;
                                         56  D4 001B8 15$: CLRL    FIXUP_ADDRESS                           ; 0814
                                 08      6B  E9 001BA 16$: BLBC    ANL$GB_INTERACTIVE, 17$                 ; 0818
                          0000G  CF      00  FB 001BD      CALLS   #0, ANL$INTERACT                        ; 0819
                                 5E      50  E9 001C2      BLBC    R0, 19$                                 ;
                                 7E      01  CE 001C5 17$: MNEGL   #1, -(SP)                              ; 0824
                                 69      01  FB 001C8      CALLS   #1, ANL$REPORT_LINE                     ;
                                     00000000G  8F  DD 001CB  PUSHL   #ANLOBJ$_EXEHDRACTIVE              ; 0825
                                                01  DD 001D1  PUSHL   #1                                ;
                                                03  DD 001D3  PUSHL   #3                                ;
                                 68             03  FB 001D5  CALLS   #3, ANL$FORMAT_LINE               ;
                                 7E             01  CE 001D8  MNEGL   #1, -(SP)                       ; 0826
                                 69             01  FB 001DB  CALLS   #1, ANL$REPORT_LINE              ;
                                                62  DD 001DE  PUSHL   (SP)                             ; 0832
                                     00000000G  8F  DD 001E0  PUSHL   #ANLOBJ$_EXEHDRXFER1               ;
                                                02  DD 001E6  PUSHL   #2                                ;
                                                7E  D4 001E8  CLRL    -(SP)                           ;
                                 68             04  FB 001EA  CALLS   #4, ANL$FORMAT_LINE              ;
                                         04     A2  DD 001ED  PUSHL   4(SP)                           ; 0833
                                     00000000G  8F  DD 001F0  PUSHL   #ANLOBJ$_EXEHDRXFER2               ;
                                                02  DD 001F6  PUSHL   #2                                ;
                                                7E  D4 001F8  CLRL    -(SP)                           ;
                                 68             04  FB 001FA  CALLS   #4, ANL$FORMAT_LINE              ;
                                         08     A2  DD 001FD  PUSHL   8(SP)                           ; 0834
                                     00000000G  8F  DD 00200  PUSHL   #ANLOBJ$_EXEHDRXFER3               ;
                                                02  DD 00206  PUSHL   #2                                ;
                                                7E  D4 00208  CLRL    -(SP)                           ;
                                 68             04  FB 0020A  CALLS   #4, ANL$FORMAT_LINE              ;
                                         0C     A2  D5 0020D  TSTL    12(SP)                          ; 0838
                                                09  13 00210  BEQL    18$                             ;
                                     00000000G  8F  DD 00212  PUSHL   #ANLOBJ$_EXEBADXFER0               ; 0839
                                 6A             01  FB 00218  CALLS   #1, ANL$FORMAT_ERROR            ;
                                 08             6B  E9 0021B 18$: BLBC    ANL$GB_INTERACTIVE, 20$       ; 0843
                          0000G  CF             00  FB 0021E  CALLS   #0, ANL$INTERACT                  ; 0844
                                 65             50  E9 00223 19$: BLBC    R0, 22$                       ;
                                 7E             01  CE 00226 20$: MNEGL   #1, -(SP)                    ; 0849
                                 69             01  FB 00229      CALLS   #1, ANL$REPORT_LINE           ;
                                     00000000G  8F  DD 0022C      PUSHL   #ANLOBJ$_EXEHDRSYMDBG           ; 0850
                                                01  DD 00232      PUSHL   #1                            ;
                                                03  DD 00234      PUSHL   #3                            ;
                                 68             03  FB 00236      CALLS   #3, ANL$FORMAT_LINE           ;
                                 7E             01  CE 00239      MNEGL   #1, -(SP)                     ; 0851
                                 69             01  FB 0023C      CALLS   #1, ANL$REPORT_LINE           ;
                                 52      04     A3  3C 0023F      MOVZWL  4(R3), SP                     ; 0853
                                 52             53  C0 00243      ADDL2   R3, SP                        ;
                                 7E      08     A2  3C 00246      MOVZWL  8(SP), -(SP)                  ; 0857
                                                62  DD 0024A      PUSHL   (SP)                          ;
```

```
                                00000000G  8F   DD 0024C          PUSHL   #ANLOBJ$_EXEHDRDST                                  ;
                                           02   DD 00252          PUSHL   #2                                                  ;
                                           7E   D4 00254          CLRL    -(SP)                                               ;
                                        68         05   FB 00256          CALLS   #5, ANL$FORMAT_LINE                         ;
                                        7E     0A  A2   3C 00259          MOVZWL  10(SP), -(SP)                               ; 0861
                                               04  A2   DD 0025D          PUSHL   4(SP)                                       ;
                                00000000G      8F   DD 00260          PUSHL   #ANLOBJ$_EXEHDRGST                          ;
                                           02   DD 00266          PUSHL   #2                                                  ;
                                           7E   D4 00268          CLRL    -(SP)                                               ;
                                        68         05   FB 0026A          CALLS   #5, ANL$FORMAT_LINE                         ;
                                11      20 A3      05   E1 0026D          BBC     #5, 32(R3), 21$                             ; 0865
                                        7E     0C  A2   7D 00272          MOVQ    12(SP), -(SP)                               ; 0867
                                00000000G      8F   DD 00276          PUSHL   #ANLOBJ$_EXEHDRDMT                          ;
                                           02   DD 0027C          PUSHL   #2                                                  ;
                                           7E   D4 0027E          CLRL    -(SP)                                               ;
                                        68         05   FB 00280          CALLS   #5, ANL$FORMAT_LINE                         ;
                                        0B         6B   E9 00283 21$:      BLBC    ANL$GB_INTERACTIVE, 23$                     ; 0871
                                0000G   CF         00   FB 00286          CALLS   #0, AN[$INTERACT                            ; 0872
                                        03         50   E8 0028B 22$:      BLBS    R0, 23$                                     ;
                                                023E 31   0028E          BRW     41$                                         ;
                                        7E         01   CE 00291 23$:      MNEGL   #1, -(SP)                                   ; 0877
                                        69         01   FB 00294          CALLS   #1, ANL$REPORT_LINE                         ;
                                00000000G      8F   DD 00297          PUSHL   #ANLOBJ$_EXEHDRIDENT                        ; 0878
                                           01   DD 0029D          PUSHL   #1                                                  ;
                                           03   DD 0029F          PUSHL   #3                                                  ;
                                        68         03   FB 002A1          CALLS   #3, ANL$FORMAT_LINE                         ;
                                        7E         01   CE 002A4          MNEGL   #1, -(SP)                                   ;
                                        69         01   FB 002A7          CALLS   #1, ANL$REPORT_LINE                         ; 0879
                                        52     06  A3   3C 002AA          MOVZWL  6(R3), SP                                   ; 0881
                                        52         53   C0 002AE          ADDL2   R3, SP                                      ;
                                        54     01  A2   9E 002B1          MOVAB   1(R2), R4                                   ; 0897
                                        55     28  A2   9E 002B5          MOVAB   40(R2), R5                                  ; 0899
0000'  CF       0C  A3                 10         00   ED 002B9          CMPZV   #0, #16, 12(R3), V3_MAJORID                 ; 0892
                                           0A   14   002C1          BGTR    24$                                         ;
0000'  CF       0E  A3                 10         00   ED 002C3          CMPZV   #0, #16, 14(R3), V3_MINORID                 ; 0893
                                           45   15   002CB          BLEQ    25$                                         ;
                                           52   DD 002CD 24$:      PUSHL   SP                                          ; 0896
                                00000000G      8F   DD 002CF          PUSHL   #ANLOBJ$_EXEHDRNAME                         ;
                                           02   DD 002D5          PUSHL   #2                                                  ;
                                           7E   D4 002D7          CLRL    -(SP)                                               ;
                                        68         04   FB 002D9          CALLS   #4, ANL$FORMAT_LINE                         ;
                                    04  AE         62   9A 002DC          MOVZBL  (SP), NAME_DSC                              ; 0897
                                    08  AE         54   D0 002E0          MOVL    R4, NAME_DSC+4                              ;
                                           27   DD 002E4          PUSHL   #39                                                 ; 0898
                                        08     AE   9F   002E6          PUSHAB  NAME_DSC                                    ;
                                0000G   CF         02   FB 002E9          CALLS   #2, ANL$CHECK_SYMBOL                        ;
                                           55   DD 002EE          PUSHL   R5                                                  ; 0899
                                00000000G      8F   DD 002F0          PUSHL   #ANLOBJ$_EXEHDRFILEID                       ;
                                           02   DD 002F6          PUSHL   #2                                                  ;
                                           7E   D4 002F8          CLRL    -(SP)                                               ;
                                        68         04   FB 002FA          CALLS   #4, ANL$FORMAT_LINE                         ;
                                        38     A2   9F   002FD          PUSHAB  56(SP)                                      ; 0900
                                00000000G      8F   DD 00300          PUSHL   #ANLOBJ$_EXEHDRTIME                         ;
                                           02   DD 00306          PUSHL   #2                                                  ;
                                           7E   D4 00308          CLRL    -(SP)                                               ;
                                        68         04   FB 0030A          CALLS   #4, ANL$FORMAT_LINE                         ;
                                        40     A2   9F   0030D          PUSHAB  64(SP)                                      ; 0901
```

```
                                                43  11 00310                BRB     26$                                             ;
                                                52  DD 00312 25$:          PUSHL   SP                                              ; 0905
                                    00000000G   8F  DD 00314              PUSHL   #ANLOBJ$_EXEHDRNAME                             ;
                                                02  DD 0031A              PUSHL   #2                                              ;
                                                7E  D4 0031C              CLRL    -(SP)                                           ;
                                            68      04 FB 0031E           CALLS   #4, ANL$FORMAT_LINE                             ;
                                        04  AE      62 9A 00321           MOVZBL  (SP), NAME_DSC                                  ; 0906
                                        08  AE      54 D0 00325           MOVL    R4, NAME_DSC+4                                  ;
                                                27  DD 00329              PUSHL   #39                                             ; 0907
                                            08  AE  9F 0032B              PUSHAB  NAME_DSC                                        ;
                                    0000G   CF      02 FB 0032E           CALLS   #2, ANL$CHECK_SYMBOL                            ;
                                            10  A2  9F 00333              PUSHAB  16(SP)                                          ; 0908
                                    00000000G   8F  DD 00336              PUSHL   #ANLOBJ$_EXEHDRFILEID                           ;
                                                02  DD 0033C              PUSHL   #2                                              ;
                                                7E  D4 0033E              CLRL    -(SP)                                           ;
                                            68      04 FB 00340           CALLS   #4, ANL$FORMAT_LINE                             ;
                                            20  A2  9F 00343              PUSHAB  32(SP)                                          ; 0909
                                    00000000G   8F  DD 00346              PUSHL   #ANLOBJ$_EXEHDRTIME                             ;
                                                02  DD 0034C              PUSHL   #2                                              ;
                                                7E  D4 0034E              CLRL    -(SP)                                           ;
                                            68      04 FB 00350           CALLS   #4, ANL$FORMAT_LINE                             ;
                                                55  DD 00353              PUSHL   R5                                              ; 0910
                                    00000000G   8F  DD 00355 26$:          PUSHL   #ANLOBJ$_EXEHDRLINKID                           ;
                                                02  DD 0035B              PUSHL   #2                                              ;
                                                7E  D4 0035D              CLRL    -(SP)                                           ;
                                            68      04 FB 0035F           CALLS   #4, ANL$FORMAT_LINE                             ;
                                            0B      6B E9 00362           BLBC    ANL$GB_INTERACTIVE, 27$                         ; 0917
                                    0000G   CF      00 FB 00365           CALLS   #0, AN[$INTERACT                                ; 0918
                                            03      50 E8 0036A           BLBS    R0, 27$                                         ;
                                                015F    31 0036D          BRW     41$                                             ;
                                            7E      01 CE 00370 27$:      MNEGL   #1, -(SP)                                       ; 0923
                                            69      01 FB 00373           CALLS   #1, ANL$REPORT_LINE                             ;
                                        00000000G   8F  DD 00376          PUSHL   #ANLOBJ$_EXEHDRPATCH                            ; 0924
                                                01  DD 0037C              PUSHL   #1                                              ;
                                                03  DD 0037E              PUSHL   #3                                              ;
                                            68      03 FB 00380           CALLS   #3, ANL$FORMAT_LINE                             ;
                                            7E      01 CE 00383           MNEGL   #1, -(SP)                                       ; 0925
                                            69      01 FB 00386           CALLS   #1, ANL$REPORT_LINE                             ;
                                                08  A3  B5 00389          TSTW    8(R3)                                           ; 0927
                                                7E  13 0038C              BEQL    28$                                             ;
                                            52      08 A3 3C 0038E        MOVZWL  8(R3), SP                                       ; 0928
                                            52      53 C0 00392           ADDL2   R3, SP                                          ;
                                            7E      04 A2 7D 00395        MOVQ    4(SP), -(SP)                                    ; 0932
                                                62  DD 00399              PUSHL   (SP)                                            ;
                                        00000000G   8F  DD 0039B          PUSHL   #ANLOBJ$_EXEHDRDECECO                           ;
                                                02  DD 003A1              PUSHL   #2                                              ;
                                                7E  D4 003A3              CLRL    -(SP)                                           ;
                                            68      06 FB 003A5           CALLS   #6, ANL$FORMAT_LINE                             ;
                                                0C  A2  DD 003A8          PUSHL   12(SP)                                          ; 0936
                                        00000000G   8F  DD 003AB          PUSHL   #ANLOBJ$_EXEHDRUSERECO                          ;
                                                02  DD 003B1              PUSHL   #2                                              ;
                                                7E  D4 003B3              CLRL    -(SP)                                           ;
                                            68      04 FB 003B5           CALLS   #4, ANL$FORMAT_LINE                             ;
                                                10  A2  DD 003B8          PUSHL   16(SP)                                          ; 0940
                                                14  A2  DD 003BB          PUSHL   20(SP)                                          ;
                                        00000000G   8F  DD 003BE          PUSHL   #ANLOBJ$_EXEHDRRWPATCH                          ;
                                                02  DD 003C4              PUSHL   #2                                              ;
```

```
                                 7E  D4 003C6           CLRL    -(SP)                                                   ;
                              68 05  FB 003C8           CALLS   #5, ANL$FORMAT_LINE                                     ;
                           18 A2  DD 003CB           PUSHL   24(SP)                                                  ; 0941
                           1C A2  DD 003CE           PUSHL   28(SP)                                                  ;
                     00000000G 8F  DD 003D1           PUSHL   #ANLOBJ$_EXEHDRROPATCH                                  ;
                                 02  DD 003D7           PUSHL   #2                                                      ;
                                 7E  D4 003D9           CLRL    -(SP)                                                   ;
                              68 05  FB 003DB           CALLS   #5, ANL$FORMAT_LINE                                     ;
                           20 A2  DD 003DE           PUSHL   32(SP)                                                  ; 0945
                     00000000G 8F  DD 003E1           PUSHL   #ANLOBJ$_EXEHDRTEXTVBN                                  ;
                                 02  DD 003E7           PUSHL   #2                                                      ;
                                 7E  D4 003E9           CLRL    -(SP)                                                   ;
                              68 04  FB 003EB           CALLS   #4, ANL$FORMAT_LINE                                     ;
                           24 A2  9F 003EE           PUSHAB  36(SP)                                                  ; 0949
                     00000000G 8F  DD 003F1           PUSHL   #ANLOBJ$_EXEHDRPATCHDATE                                ;
                                 02  DD 003F7           PUSHL   #2                                                      ;
                                 7E  D4 003F9           CLRL    -(SP)                                                   ;
                              68 04  FB 003FB           CALLS   #4, ANL$FORMAT_LINE                                     ;
                              18 6B  E9 003FE           BLBC    ANL$GB_INTERACTIVE, 29$                                 ; 0953
                  0000G       CF 00  FB 00401           CALLS   #0, AN[$INTERACT                                        ; 0954
                              10 50  E8 00406           BLBS    R0, 29$                                                 ;
                                 00C3 31  00409           BRW     41$                                                     ; 0955
                     00000000G 8F  DD 0040C 28$:       PUSHL   #ANLOBJ$_EXEHDRNOPATCH                                  ; 0960
                                 02  DD 00412           PUSHL   #2                                                      ;
                                 7E  D4 00414           CLRL    -(SP)                                                   ;
                              68 03  FB 00416           CALLS   #3, ANL$FORMAT_LINE                                     ;
                           0C BC  D4 00419 29$:       CLRL    @FIXUP_VBN                                              ; 0967
                           08 BC  D4 0041C           CLRL    @FIXUP_SIZE                                             ;
                           7E 01  CE 0041F           MNEGL   #1, -(SP)                                               ; 0969
                           69 01  FB 00422           CALLS   #1, ANL$REPORT_LINE                                     ;
                     00000000G 8F  DD 00425           PUSHL   #ANLOBJ$_EXEHDRISD                                      ; 0970
                                 01  DD 0042B           PUSHL   #1                                                      ;
                                 03  DD 0042D           PUSHL   #3                                                      ;
                              68 03  FB 0042F           CALLS   #3, ANL$FORMAT_LINE                                     ;
                              54 01  D0 00432           MOVL    #1, VBN                                                 ; 0972
                              53 01  D0 00435           MOVL    #1, ISD                                                 ; 1015
                                 5E  DD 00438 30$:       PUSHL   SP                                                      ; 0979
                  0000G       CF 01  FB 0043A           CALLS   #1, ANL$GET_ISD                                         ;
                              57 50  D0 0043F           MOVL    R0, STATUS                                              ;
                     084D8640 8F 57  D1 00442           CMPL    STATUS, #139298368                                      ; 0984
                                 25 13  00449           BEQL    33$                                                     ;
                                 54 D6  0044B           INCL    VBN                                                     ; 0986
                              04 57  E8 0044D           BLBS    STATUS, 31$                                             ; 0987
                                 57 DD  00450           PUSHL   STATUS                                                  ; 0988
                                 19 11  00452           BRB     32$                                                     ;
                              52 6E  D0 00454 31$:       MOVL    HP, SP                                                  ; 0991
                        50 6E 52  C3 00457           SUBL3   SP, HP, R0                                              ; 0997
                           50 0200 C0 9E 0045B           MOVAB   512(R0), R0                                             ;
                  50       62 10  00 ED 00460           CMPZV   #0, #16, (SP), R0                                       ;
                                 0B 1B  00465           BLEQU   34$                                                     ;
                     00000000G 8F  DD 00467           PUSHL   #ANLOBJ$_EXEHDRISDLONG                                  ; 0998
                              6A 01  FB 0046D 32$:       CALLS   #1, ANL$FORMAT_ERROR                                    ;
                                 59 11  00470 33$:       BRB     40$                                                     ; 0997
                                 0C BB  00472 34$:       PUSHR   #^M<R2,R3>                                              ; 1004
                  0000V       CF 02  FB 00474           CALLS   #2, ANL$IMAGE_ISD                                       ;
                              01 53  D1 00479           CMPL    ISD, #1                                                 ; 1009
                                 0B 12  0047C           BNEQ    35$                                                     ;
```

```
                50      04   A2            15          00   EF 0047E             EXTZV   #0, #21, 4(SP), R0                      ; 1010
                        04   BC            50          09   78 00484             ASHL    #9, R0, @IMAGE_BASE                     ;
                                                       56   D5 00489 35$:        TSTL    FIXUP_ADDRESS                           ; 1015
                                                       2E   13 0048B             BEQL    38$                                     ;
                50      04   A2            17          00   EF 0048D             EXTZV   #0, #23, 4(SP), R0                      ; 1016
                             50            50          09   78 00493             ASHL    #9, R0, R0                              ;
                             50                        56   D1 00497             CMPL    FIXUP_ADDRESS, R0                       ;
                                                       1F   12 0049A             BNEQ    38$                                     ;
                                                 02    A2   B5 0049C             TSTW    2(SP)                                   ; 1017
                                                       05   13 0049F             BEQL    36$                                     ;
                                                 0C    A2   D5 004A1             TSTL    12(SP)                                  ;
                                                       0B   12 004A4             BNEQ    37$                                     ;
                                          00000000G    8F   DD 004A6 36$:        PUSHL   #ANLOBJ$_EXEBADFIXUPISD                 ; 1018
                                              6A            01   FB 004AC             CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                       0A   11 004AF             BRB     38$                                     ;
                                    08   BC       02    A2   3C 004B1 37$:        MOVZWL  2(SP), @FIXUP_SIZE                      ; 1020
                                    0C   BC       0C    A2   D0 004B6             MOVL    12(SP), @FIXUP_VBN                      ; 1021
                                         08            6B   E9 004BB 38$:        BLBC    ANL$GB_INTERACTIVE, 39$                 ; 1026
                                0000G    CF            00   FB 004BE             CALLS   #0, ANL$INTERACT                        ; 1027
                                         09            50   E9 004C3             BLBC    R0, 41$                                 ;
                                                       53   D6 004C6 39$:        INCL    ISD                                     ; 0973
                                                    FF6D    31 004C8             BRW     30$                                     ;
                                              50            01   D0 004CB 40$:        MOVL    #1, R0                                  ; 1032
                                                            04 004CE             RET                                             ;
                                                       50   D4 004CF 41$:        CLRL    R0                                      ; 1034
                                                            04 004D1             RET                                             ;

; Routine Size:  1234 bytes,    Routine Base:  $CODE$ + 0000
```

```
;   523          1035  1 %sbttl 'ANL$IMAGE_ISD - Analyze ISD Structure'
;   524          1036  1 !++
;   525          1037  1 ! Functional Description:
;   526          1038  1 !       This routine is responsible for formatting and analyzing an
;   527          1039  1 !       Image Section Descriptor.
;   528          1040  1 !
;   529          1041  1 ! Formal Parameters:
;   530          1042  1 !       the_isd         Address of the ISD.
;   531          1043  1 !       isd_number      The sequence number of this ISD.
;   532          1044  1 !
;   533          1045  1 ! Implicit Inputs:
;   534          1046  1 !       global data
;   535          1047  1 !
;   536          1048  1 ! Implicit Outputs:
;   537          1049  1 !       global data
;   538          1050  1 !
;   539          1051  1 ! Returned Value:
;   540          1052  1 !       none
;   541          1053  1 !
;   542          1054  1 ! Side Effects:
;   543          1055  1 !
;   544          1056  1 !--
;   545          1057  1
;   546          1058  1
;   547          1059  2 global routine anl$image_isd(the_isd,isd_number): novalue = begin
;   548          1060  2
;   549          1061  2 bind
;   550          1062  2         sp = the_isd: ref block[,byte];
;   551          1063  2
;   552          1064  2 own
;   553          1065  2         space_names: vector[4,long] initial(
;   554          1066  2                                 uplit byte (%ascic 'P0'),
;   555          1067  2                                 uplit byte (%ascic 'P1'),
;   556          1068  2                                 uplit byte (%ascic 'S0'),
;   557          1069  2                                 uplit byte (%ascic 'S1???')),
;   558          1070  2
;   559          1071  2         isd_flags_def: vector[20,long] initial(
;   560          1072  2                                 18,
;   561          1073  2                                 uplit byte(%ascic 'ISD$V_GBL'),
;   562          1074  2                                 uplit byte(%ascic 'ISD$V_CRF'),
;   563          1075  2                                 uplit byte(%ascic 'ISD$V_DZRO'),
;   564          1076  2                                 uplit byte(%ascic 'ISD$V_WRT'),
;   565          1077  2                                 0,0,0,
;   566          1078  2                                 uplit byte(%ascic 'ISD$V_LASTCLU'),
;   567          1079  2                                 uplit byte(%ascic 'ISD$V_COPYALWAY'),
;   568          1080  2                                 uplit byte(%ascic 'ISD$V_BASED'),
;   569          1081  2                                 uplit byte(%ascic 'ISD$V_FIXUPVEC'),
;   570          1082  2                                 0,0,0,0,0,0,
;   571          1083  2                                 uplit byte(%ascic 'ISD$V_VECTOR'),
;   572          1084  2                                 uplit byte(%ascic 'ISD$V_PROTECT')),
;   573          1085  2
;   574          1086  2         isd_types: vector[5,long] initial(
;   575          1087  2                                 uplit byte (%ascic 'NORMAL'),
;   576          1088  2                                 uplit byte (%ascic 'SHRFXD'),
;   577          1089  2                                 uplit byte (%ascic 'PRVFXD'),
;   578          1090  2                                 uplit byte (%ascic 'SHRPIC'),
;   579          1091  2                                 uplit byte (%ascic 'PRVPIC'));
```

```
;  580          1092  2
;  581          1093  2 local
;  582          1094  2         blk_ptr: ref block[, byte],
;  583          1095  2         status;
;  584          1096  2
;  585          1097  2 literal
;  586          1098  2         section_suffix_size = 4,
;  587          1099  2         long_c  = 4;
;  588          1100  2
;  589          1101  2 macro
;  590          1102  2         long_u = 0, 0, 32, 0 %;
;  591          1103  2
;  592          1104  2 ! It is assumed that the ISD fits in the header block.  We can freely
;  593          1105  2 ! reference the fields.
;  594          1106  2
;  595          1107  2 ! Begin with a heading line for this ISD.
;  596          1108  2
;  597          1109  2 anl$report_line(-1);
;  598          1110  2 anl$format_line(3,2,anlobj$_exehdrisdnum,.isd_number,.sp[isd$w_size]);
;  599          1111  2
;  600          1112  2 ! Analyze the page count.
;  601          1113  2
;  602          1114  2 anl$format_line(0,3,anlobj$_exehdrisdcount,.sp[isd$w_pagcnt]);
;  603          1115  2
;  604          1116  2 ! Analyze the base virtual page number and space bits.
;  605          1117  2
;  606          1118  2 anl$format_line(0,3,anlobj$_exehdrisdbase,.sp[isd$v_vpg]^9,.space_names[.sp[4,21,2,0]]);
;  607          1119  2 if .sp[isd$v_p1] and .sp[isd$v_system] then
;  608          1120  2         anl$format_error(anlobj$_exebadisds1);
;  609          1121  2
;  610          1122  2 ! Analyze the page fault cluster size.
;  611          1123  2
;  612          1124  2 if .sp[isd$b_pfc] eqlu 0 then
;  613          1125  2         anl$format_line(0,3,anlobj$_exehdrisdpfcdef)
;  614          1126  2 else
;  615          1127  2         anl$format_line(0,3,anlobj$_exehdrisdpfcsiz,.sp[isd$b_pfc]);
;  616          1128  2
;  617          1129  2 ! Analyze the ISD flags, ignoring the match control bits.
;  618          1130  2
;  619          1131  2 anl$format_flags(3,anlobj$_exehdrisdflags,.sp[isd$l_flags] and %x'00ffff8f',isd_flags_def);
;  620          1132  2 anl$check_flags(.sp[isd$l_flags] and %x'00ffff8f',isd_flags_def);
;  621          1133  2
;  622          1134  2 ! Analyze the ISD type code.
;  623          1135  2
;  624          1136  2 selectoneu .sp[isd$b_type] of set
;  625          1137  2 [0 to 4]:                anl$format_line(0,3,anlobj$_exehdrisdtype,.isd_types[.sp[isd$b_type]]);
;  626          1138  2
;  627          1139  2 [isd$k_usrstack]:       anl$format_line(0,3,anlobj$_exehdrisdtype,uplit byte (%ascic 'USRSTACK'));
;  628          1140  2
;  629          1141  2 [otherwise]:            anl$format_error(anlobj$_exebadisdtype,.sp[isd$b_type]);
;  630          1142  2 tes;
;  631          1143  2
;  632          1144  2 ! If this is a demand-zero section, we are done.
;  633          1145  2
;  634          1146  3 if .sp[isd$v_dzro] then (
;  635          1147  4         if .sp[isd$w_size] gtru (
;  636          1148  4                 if .sp[isd$v_gbl] then isd$c_maxlenglbl
```

```
;  637          1149  4                   else isd$c_lendzro)
;  638          1150  3           then
;  639          1151  3                   anl$format_error(anlobj$_exeisdlendzro);
;  640          1152  3           return;
;  641          1153  2 );
;  642          1154  2
;  643          1155  2 ! Analyze the base VBN.
;  644          1156  2
;  645          1157  2 anl$format_line(0,3,anlobj$_exehdrisdvbn,.sp[isd$l_vbn]);
;  646          1158  2
;  647          1159  2 ! Before we leave, let's see if this ISD points to an indirect message
;  648          1160  2 ! file.  If so, print out this filename.  To check this, the vector and
;  649          1161  2 ! protect flags must be set, and the page count is 1.  If the page count
;  650          1162  2 ! is greater than 1, this ISD is probably a "direct" message section in
;  651          1163  2 ! which the messages in text have spanned more than one block, so don't
;  652          1164  2 ! bother continuing, we only want indirect.  Then reading the VBN which
;  653          1165  2 ! this ISD points to, the type field will tell if it's a privileged sharable
;  654          1166  2 ! image or a user written system service, or a message section.  Only if it
;  655          1167  2 ! is an indirect message section, is any further information given.
;  656          1168  2
;  657          1169  3 if .sp[isd$v_vector] and .sp[isd$v_protect] and (.sp[isd$w_pagcnt] eqlu 1)
;  658          1170  2 then
;  659          1171  3     begin
;  660          1172  3     status = anl$get_image_block( .sp[isd$l_vbn], blk_ptr );
;  661          1173  3     if not .status
;  662          1174  3     then
;  663          1175  3         return (.status);
;  664          1176  3     if .blk_ptr[plv$l_type ] eqlu plv$c_typ_msg
;  665          1177  3     then
;  666          1178  4         begin
;  667          1179  4         blk_ptr = .blk_ptr + $byteoffset(plv$l_usrundwn);
;  668          1180  4         while .blk_ptr[long_u] nequ 0 do
;  669          1181  5             begin
;  670          1182  5             bind msc_ptr = .blk_ptr + .blk_ptr[long_u] : block[,byte];
;  671          1183  5             if .msc_ptr[ msc$b_type ] eqlu msc$c_ind
;  672          1184  5             then
;  673          1185  5                 anl$format_line(0,3,anlobj$_indmsgsec,msc_ptr[msc$b_indnamlen]);
;  674          1186  5             blk_ptr = .blk_ptr + long_c;        ! Add the size of a longword
;  675          1187  4             end;
;  676          1188  3         end;
;  677          1189  2     end;
;  678          1190  2
;  679          1191  2 ! If this isn't a global section, we're done.
;  680          1192  2
;  681          1193  3 if not .sp[isd$v_gbl] then (
;  682          1194  3         if .sp[isd$w_size] gtru isd$c_lenpriv then
;  683          1195  3                 anl$format_error(anlobj$_exeisdlenpriv);
;  684          1196  3         return;
;  685          1197  2 );
;  686          1198  2
;  687          1199  2 ! Analyze the global section identification.
;  688          1200  2
;  689          1201  2 anl$format_line(0,3,anlobj$_exehdrgblident,.sp[isd$l_ident]);
;  690          1202  2
;  691          1203  2 ! Analyze the match control.
;  692          1204  2
;  693          1205  2 selectoneu .sp[isd$v_matchctl] of set
```

```
2 [isd$k_matall,
2   isd$k_matequ,
2   isd$k_matleq,
2   isd$k_matnev]: anl$format_line(0,3,anlobj$_exehdrmatch,.match_control[.sp[isd$v_matchctl]]);
2
2 [otherwise]:     anl$format_error(anlobj$_exebadmatch,.sp[isd$v_matchctl]);
2 tes;
2
2 ! Analyze the global section name.
2
2 anl$format_line(0,3,anlobj$_exehdrisdgblnam,sp[isd$t_gblnam]);
3 begin
3 local
3         name_dsc: descriptor;
3
3 build_descriptor(name_dsc,.sp[20,0,8,0],sp[21,0,8,0]);
3 anl$check_symbol(name_dsc, shl$c_maxnamlng+section_suffix_size);
2 end;
2
2 ! We are done.
2
2 if .sp[isd$w_size] gtru isd$c_lenglbl then
2         anl$format_error(anlobj$_exeisdlengbl);
2
2 return;
2
1 end;
```

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

                                          30 50 02  00094 P.AAM:  .ASCII  <2>\P0\                 ;
                                          31 50 02  00097 P.AAN:  .ASCII  <2>\P1\                 ;
                                          30 53 02  0009A P.AAO:  .ASCII  <2>\S0\                 ;
                                 3F 3F 3F 31 53 05  0009D P.AAP:  .ASCII  <5>\S1???\              ;
                  4C 42 47 5F 56 24 44 53 49 09  000A3 P.AAQ:  .ASCII  <9>\ISD$V_GBL\          ;
                  46 52 43 5F 56 24 44 53 49 09  000AD P.AAR:  .ASCII  <9>\ISD$V_CRF\          ;
               4F 52 5A 44 5F 56 24 44 53 49 0A  000B7 P.AAS:  .ASCII  <10>\ISD$V_DZRO\        ;
                  54 52 57 5F 56 24 44 53 49 09  000C2 P.AAT:  .ASCII  <9>\ISD$V_WRT\          ;
      55 4C 43 54 53 41 4C 5F 56 24 44 53 49 0D  000CC P.AAU:  .ASCII  <13>\ISD$V_LASTCLU\     ;
   41 57 4C 41 59 50 4F 43 5F 56 24 44 53 49 0F  000DA P.AAV:  .ASCII  <15>\ISD$V_COPYALWAY\   ;
                                             59  000E9                                        ;
            44 45 53 41 42 5F 56 24 44 53 49 0B  000EA P.AAW:  .ASCII  <11>\ISD$V_BASED\       ;
   43 45 56 50 55 58 49 46 5F 56 24 44 53 49 0E  000F6 P.AAX:  .ASCII  <14>\ISD$V_FIXUPVEC\    ;
         52 4F 54 43 45 56 5F 56 24 44 53 49 0C  00105 P.AAY:  .ASCII  <12>\ISD$V_VECTOR\      ;
      54 43 45 54 4F 52 50 5F 56 24 44 53 49 0D  00112 P.AAZ:  .ASCII  <13>\ISD$V_PROTECT\     ;
                           4C 41 4D 52 4F 4E 06  00120 P.ABA:  .ASCII  <6>\NORMAL\            ;
                           44 58 46 52 48 53 06  00127 P.ABB:  .ASCII  <6>\SHRFXD\            ;
                           44 58 46 56 52 50 06  0012E P.ABC:  .ASCII  <6>\PRVFXD\            ;
                           43 49 50 52 48 53 06  00135 P.ABD:  .ASCII  <6>\SHRPIC\            ;
                           43 49 50 56 52 50 06  0013C P.ABE:  .ASCII  <6>\PRVPIC\            ;
                     4B 43 41 54 53 52 53 55 08  00143 P.ABF:  .ASCII  <8>\USRSTACK\          ;

                                                                    .PSECT  $OWN$,NOEXE,2

                                                 0003E            .BLKB   2
```

```
                  00000000' 00000000' 00000000' 00000000' 00040 SPACE_NAMES:
                                                                        .ADDRESS P.AAM, P.AAN, P.AAO, P.AAP                             ;
                                                00000012  00050 ISD_FLAGS_DEF:
                                                                        .LONG   18                                                      ;
                  00000000' 00000000' 00000000' 00000000' 00054         .ADDRESS P.AAQ, P.AAR, P.AAS, P.AAT                             ;
                            00000000  00000000  00000000  00064         .LONG   0, 0, 0                                                 ;
                  00000000' 00000000' 00000000' 00000000' 00070         .ADDRESS P.AAU, P.AAV, P.AAW, P.AAX                             ;
00000000  00000000  00000000  00000000  00000000  00000000  00080         .LONG   0, 0, 0, 0, 0, 0                                        ;
                                      00000000' 00000000' 00098         .ADDRESS P.AAY, P.AAZ                                           ;
          00000000' 00000000' 00000000' 00000000' 00000000' 000A0 ISD_TYPES:
                                                                        .ADDRESS P.ABA, P.ABB, P.ABC, P.ABD, P.ABE                      ;


                                                                        .PSECT  $CODE$,NOWRT,2

                                                  00FC 00000            .ENTRY  ANL$IMAGE_ISD, Save R2,R3,R4,R5,R6,R7                   ; 1059
                                        57      0000G CF   9E 00002     MOVAB   ANL$FORMAT_ERROR, R7                                    ;
                                        56      0000' CF   9E 00007     MOVAB   ISD_FLAGS_DEF, R6                                       ;
                                        55      0000G CF   9E 0000C     MOVAB   ANL$FORMAT_LINE, R5                                     ;
                                        5E            0C   C2 00011     SUBL2   #12, SP                                                 ;
                                        7E            01   CE 00014     MNEGL   #1, -(SP)                                               ; 1109
                                  0000G CF            01   FB 00017     CALLS   #1, ANL$REPORT_LINE                                     ;
                                        52      04    AC   D0 0001C     MOVL    SP, R2                                                  ; 1110
                                        7E            62   3C 00020     MOVZWL  (R2), -(SP)                                             ;
                                                08    AC   DD 00023     PUSHL   ISD_NUMBER                                              ;
                                           00000000G  8F   DD 00026     PUSHL   #ANLOBJ$_EXEHDRISDNUM                                   ;
                                                      02   DD 0002C     PUSHL   #2                                                      ;
                                                      03   DD 0002E     PUSHL   #3                                                      ;
                                        65            05   FB 00030     CALLS   #5, ANL$FORMAT_LINE                                     ;
                                        7E      02    A2   3C 00033     MOVZWL  2(R2), -(SP)                                            ; 1114
                                           00000000G  8F   DD 00037     PUSHL   #ANLOBJ$_EXEHDRISDCOUNT                                 ;
                                                      03   DD 0003D     PUSHL   #3                                                      ;
                                                      7E   D4 0003F     CLRL    -(SP)                                                   ;
                                        65            04   FB 00041     CALLS   #4, ANL$FORMAT_LINE                                     ;
                  50    06    A2                    02   EF 00044     EXTZV   #5, #2, 6(R2), R0                                       ; 1118
                                              F0 A640   DD 0004A     PUSHL   SPACE_NAMES[R0]                                         ;
                  50    04    A2                    00   EF 0004E     EXTZV   #0, #23, 4(R2), R0                                      ;
                        7E                    50      09   78 00054     ASHL    #9, R0, -(SP)                                           ;
                                           00000000G  8F   DD 00058     PUSHL   #ANLOBJ$_EXEHDRISDBASE                                  ;
                                                      03   DD 0005E     PUSHL   #3                                                      ;
                                                      7E   D4 00060     CLRL    -(SP)                                                   ;
                                        65            05   FB 00062     CALLS   #5, ANL$FORMAT_LINE                                     ;
                        0E    06    A2                05   E1 00065     BBC     #5, 6(R2), 1$                                           ; 1119
                        09    06    A2                06   E1 0006A     BBC     #6, 6(R2), 1$                                           ; 1120
                                           00000000G  8F   DD 0006F     PUSHL   #ANLOBJ$_EXEBADISDS1                                    ;
                                        67            01   FB 00075     CALLS   #1, ANL$FORMAT_ERROR                                    ;
                                              07      A2   95 00078 1$:  TSTB    7(R2)                                                   ; 1124
                                                      0F   12 0007B     BNEQ    2$                                                      ;
                                           00000000G  8F   DD 0007D     PUSHL   #ANLOBJ$_EXEHDRISDPFCDEF                                ; 1125
                                                      03   DD 00083     PUSHL   #3                                                      ;
                                                      7E   D4 00085     CLRL    -(SP)                                                   ;
                                        65            03   FB 00087     CALLS   #3, ANL$FORMAT_LINE                                     ;
                                                      11   11 0008A     BRB     3$                                                      ;
                                        7E      07    A2   9A 0008C 2$:  MOVZBL  7(R2), -(SP)                                           ; 1127
                                           00000000G  8F   DD 00090     PUSHL   #ANLOBJ$_EXEHDRISDPFCSIZ                                ;
                                                      03   DD 00096     PUSHL   #3                                                      ;
```

```
                                                7E  D4 00098          CLRL    -(SP)
                                             65 04  FB 0009A          CALLS   #4, ANL$FORMAT_LINE
                                                56  DD 0009D 3$:     PUSHL   R6                                  ;1131
                                             54 08  A2 0009F          MOVAB   8(R2), R4
                                          53 64 FF000070  8F CB 000A3 BICL3   #-16777104, (R4), R3
                                                53  DD 000AB          PUSHL   R3
                                       00000000G  8F  DD 000AD        PUSHL   #ANLOBJ$_EXEHDRISDFLAGS
                                                03  DD 000B3          PUSHL   #3
                                       0000G    CF  04  FB 000B5      CALLS   #4, ANL$FORMAT_FLAGS
                                                0048  8F BB 000BA     PUSHR   #^M<R3,R6>                         ;1132
                                       0000G    CF  02  FB 000BE      CALLS   #2, ANL$CHECK_FLAGS
                                                50 0B  A2 9A 000C3    MOVZBL  11(R2), R0                         ;1136
                                                04  50  91 000C7      CMPB    R0, #4                             ;1137
                                                    06  1A 000CA      BGTRU   4$
                                                50 A640  DD 000CC     PUSHL   ISD_TYPES[R0]
                                                    0A  11 000D0      BRB     5$
                                             FD 8F      50  91 000D2 4$:  CMPB   R0, #253                         ;1139
                                                    13  12 000D6      BNEQ    6$
                                                0000' CF  9F 000D8    PUSHAB  P.ABF
                                       00000000G  8F  DD 000DC 5$:    PUSHL   #ANLOBJ$_EXEHDRISDTYPE
                                                03  DD 000E2          PUSHL   #3
                                                7E  D4 000E4          CLRL    -(SP)
                                             65 04  FB 000E6          CALLS   #4, ANL$FORMAT_LINE
                                                0B  11 000E9          BRB     7$
                                                50  DD 000EB 6$:     PUSHL   R0                                 ;1141
                                       00000000G  8F  DD 000ED        PUSHL   #ANLOBJ$_EXEBADISDTYPE
                                             67 02  FB 000F3          CALLS   #2, ANL$FORMAT_ERROR
                                          1B 64     02  E1 000F6 7$:  BBC     #2, (R4), 10$                      ;1146
                                             06  64  E9 000FA        BLBC    (R4), 8$                           ;1148
                                             50 40  8F  9A 000FD      MOVZBL  #64, R0
                                                03  11 00101         BRB     9$
                                             50  0C  D0 00103 8$:     MOVL    #12, R0
                                          50 62 10  00  ED 00106 9$:  CMPZV   #0, #16, (R2), R0                  ;1147
                                                66  1B 0010B          BLEQU   15$
                                       00000000G  8F  DD 0010D        PUSHL   #ANLOBJ$_EXEISDLENDZRO             ;1151
                                                66  11 00113         BRB     16$
                                             0C A2  DD 00115 10$:     PUSHL   12(R2)                             ;1157
                                       00000000G  8F  DD 00118        PUSHL   #ANLOBJ$_EXEHDRISDVBN
                                                03  DD 0011E         PUSHL   #3
                                                7E  D4 00120          CLRL    -(SP)
                                             65 04  FB 00122          CALLS   #4, ANL$FORMAT_LINE
                                          44 64     11  E1 00125      BBC     #17, (R4), 14$                     ;1169
                                          40 64     12  E1 00129      BBC     #18, (R4), 14$
                                             01 02  A2  B1 0012D      CMPW    2(R2), #1
                                                3A  12 00131         BNEQ    14$
                                                5E  DD 00133         PUSHL   SP                                 ;1172
                                             0C A2  DD 00135         PUSHL   12(R2)
                                       0000G    CF  02  FB 00138      CALLS   #2, ANL$GET_IMAGE_BLOCK
                                             01 50  E8 0013D         BLBS    STATUS, 11$                        ;1173
                                                04  00140          RET
                                             02 00  BE  D1 00141 11$: CMPL   @BLK_PTR, #2                       ;1176
                                                26  12 00145         BNEQ    14$
                                             6E 10  C0  00147         ADDL2   #16, BLK_PTR                       ;1179
                                             00 BE  D5  0014A 12$:    TSTL    @BLK_PTR                           ;1180
                                                1E  13 0014D         BEQL    14$
                                          50 7E     9E  C1 0014F      ADDL3   @BLK_PTR, BLK_PTR, R0              ;1182
                                             01 60  91  00153         CMPB    (R0), #1                           ;1183
```

```
                                      10  12 00156          BNEQ    13$                                  ;
                             08       A0  9F 00158          PUSHAB  8(R0)                                ; 1185
                      00000000G       8F  DD 0015B          PUSHL   #ANLOBJ$_INDMSGSEC                   ;
                                      03  DD 00161          PUSHL   #3                                   ;
                                      7E  D4 00163          CLRL    -(SP)                                ;
                  65                  04  FB 00165          CALLS   #4, ANL$FORMAT_LINE                  ;
                  6E                  04  C0 00168 13$:     ADDL2   #4, BLK_PTR                          ; 1186
                                      DD  11 0016B          BRB     12$                                  ; 1180
                  0D                  64  E8 0016D 14$:     BLBS    (R4), 17$                            ; 1193
                  10                  62  B1 00170          CMPW    (R2), #16                            ; 1194
                                      72  1B 00173 15$:     BLEQU   21$                                  ;
                      00000000G       8F  DD 00175          PUSHL   #ANLOBJ$_EXEISDLENPRIV               ; 1195
                                      67  11 0017B 16$:     BRB     20$                                  ;
                             10       A2  DD 0017D 17$:     PUSHL   16(R2)                               ; 1201
                      00000000G       8F  DD 00180          PUSHL   #ANLOBJ$_EXEHDRGBLIDENT              ;
                                      03  DD 00186          PUSHL   #3                                   ;
                                      7E  D4 00188          CLRL    -(SP)                                ;
                  65                  04  FB 0018A          CALLS   #4, ANL$FORMAT_LINE                  ;
     50         64        03          04  EF 0018D          EXTZV   #4, #3, (R4), R0                     ; 1205
                  03                  50  D1 00192          CMPL    R0, #3                               ; 1206
                                      13  1A 00195          BGTRU   18$                                  ;
                          B0 A640         DD 00197          PUSHL   MATCH_CONTROL[R0]                    ; 1209
                      00000000G       8F  DD 0019B          PUSHL   #ANLOBJ$_EXEHDRMATCH                 ;
                                      03  DD 001A1          PUSHL   #3                                   ;
                                      7E  D4 001A3          CLRL    -(SP)                                ;
                  65                  04  FB 001A5          CALLS   #4, ANL$FORMAT_LINE                  ;
                                      0B  11 001A8          BRB     19$                                  ;
                                      50  DD 001AA 18$:     PUSHL   R0                                   ; 1211
                      00000000G       8F  DD 001AC          PUSHL   #ANLOBJ$_EXEBADMATCH                 ;
                  67                  02  FB 001B2          CALLS   #2, ANL$FORMAT_ERROR                 ;
                             14       A2  9F 001B5 19$:     PUSHAB  20(R2)                               ; 1216
                      00000000G       8F  DD 001B8          PUSHL   #ANLOBJ$_EXEHDRISDGBLNAM             ;
                                      03  DD 001BE          PUSHL   #3                                   ;
                                      7E  D4 001C0          CLRL    -(SP)                                ;
                  65                  04  FB 001C2          CALLS   #4, ANL$FORMAT_LINE                  ;
            04    AE         14       A2  9A 001C5          MOVZBL  20(R2), NAME_DSC                     ; 1221
            08    AE         15       A2  9E 001CA          MOVAB   21(R2), NAME_DSC+4                   ;
                                      2B  DD 001CF          PUSHL   #43                                  ; 1222
                             08       AE  9F 001D1          PUSHAB  NAME_DSC                             ;
         0000G    CF                  02  FB 001D4          CALLS   #2, ANL$CHECK_SYMBOL                 ;
                  24                  62  B1 001D9          CMPW    (R2), #36                            ; 1227
                                      09  1B 001DC          BLEQU   21$                                  ;
                      00000000G       8F  DD 001DE          PUSHL   #ANLOBJ$_EXEISDLENGBL                ; 1228
                  67                  01  FB 001E4 20$:     CALLS   #1, ANL$FORMAT_ERROR                 ;
                                          04 001E7 21$:     RET                                          ; 1232

; Routine Size:  488 bytes,    Routine Base:  $CODE$ + 04D2
```

```
;  722          1233  1 %sbttl 'ANL$IMAGE_PATCH_TEXT - Print Image Patch Text'
;  723          1234  1 !++
;  724          1235  1 ! Functional Description:
;  725          1236  1 !       This routine is responsible for printing the patch text in the
;  726          1237  1 !       analysis report.
;  727          1238  1 !
;  728          1239  1 ! Formal Parameters:
;  729          1240  1 !       none
;  730          1241  1 !
;  731          1242  1 ! Implicit Inputs:
;  732          1243  1 !       global data
;  733          1244  1 !
;  734          1245  1 ! Implicit Outputs:
;  735          1246  1 !       global data
;  736          1247  1 !
;  737          1248  1 ! Returned Value:
;  738          1249  1 !       If interactive session: true if we are to continue, false otherwise.
;  739          1250  1 !
;  740          1251  1 ! Side Effects:
;  741          1252  1 !
;  742          1253  1 !--
;  743          1254  1
;  744          1255  1
;  745          1256  2 global routine anl$image_patch_text = begin
;  746          1257  2
;  747          1258  2 local
;  748          1259  2         bp: ref block[,byte],
;  749          1260  2         sp: ref block[,byte],
;  750          1261  2         patch_vbn: long,
;  751          1262  2         length: signed long,
;  752          1263  2         take: long,
;  753          1264  2         alias,
;  754          1265  2         local_described_buffer(out_record_dsc,512);
;  755          1266  2
;  756          1267  2
;  757          1268  2 ! The image header patch section has already been checked.  If this image
;  758          1269  2 ! doesn't have any patches, then we can leave.
;  759          1270  2
;  760          1271  2 anl$get_image_header(bp,alias);
;  761          1272  2 if .bp[ihd$w_patchoff] eqlu 0 then
;  762          1273  2         return true;
;  763          1274  2 sp = .bp + .bp[ihd$w_patchoff];
;  764          1275  2 if .sp[ihp$l_patcomtxt] eqlu 0 then
;  765          1276  2         return true;
;  766          1277  2
;  767          1278  2 ! We seem to have patch text.  Let's eject the page and start with a heading.
;  768          1279  2
;  769          1280  2 anl$report_page();
;  770          1281  2 anl$format_line(0,0,anlobj$_exepatch);
;  771          1282  2 anl$report_line(0);
;  772          1283  2 anl$report_line(0);
;  773          1284  2
;  774          1285  2 ! We need the VBN of the patch text.  Get the first block.
;  775          1286  2
;  776          1287  2 patch_vbn = .sp[ihp$l_patcomtxt];
;  777          1288  2 anl$get_image_block(.patch_vbn,bp);
;  778          1289  2 sp = .bp;
```

```
;  779          1290  2
;  780          1291  2 ! OK, now we are going to loop through the patch records in the patch
;  781          1292  2 ! text area.  We construct each record from the blocks of the image and
;  782          1293  2 ! print them.
;  783          1294  2
;  784          1295  3 loop (
;  785          1296  3
;  786          1297  3         ! Sit in a loop and build the next patch record.  PATCH_VBN is the
;  787          1298  3         ! block number we are at.  SP points to the beginning of the record,
;  788          1299  3         ! which is a length.  If not positive, that's the end of the
;  789          1300  3         ! patch text.
;  790          1301  3
;  791          1302  3         length = .sp[0,0,16,1];
;  792          1303  3 exitif (.length leq 0);
;  793          1304  3
;  794          1305  4         if .length gtru 255 then (
;  795          1306  4                 anl$format_error(anlobj$_exebadpatchlen,255);
;  796          1307  4 exitloop;
;  797          1308  3         );
;  798          1309  3         sp = .sp + 2;
;  799          1310  3
;  800          1311  3         out_record_dsc[len] = 0;
;  801          1312  4         loop (
;  802          1313  4
;  803          1314  4                 ! If we have run off the end of this block, let's get another.
;  804          1315  4
;  805          1316  5                 if .sp geqa .bp+512 then (
;  806          1317  5                         increment (patch_vbn);
;  807          1318  5                         anl$get_image_block(.patch_vbn, bp);
;  808          1319  5                         sp = .bp;
;  809          1320  4                 );
;  810          1321  4
;  811          1322  4                 ! If we have built the entire record, drop out.
;  812          1323  4
;  813          1324  4         exitif (.length eql 0);
;  814          1325  4
;  815          1326  4                 ! Take as many bytes as we can from this block to build
;  816          1327  4                 ! the record.  Adjust things.
;  817          1328  4
;  818          1329  4                 take = minu(.length, .bp+512-.sp);
;  819          1330  4                 ch$move(.take,.sp, .out_record_dsc[ptr]+.out_record_dsc[len]);
;  820          1331  4                 out_record_dsc[len] = .out_record_dsc[len] + .take;
;  821          1332  4                 sp = .sp + .take + .take mod 2;
;  822          1333  4                 length = .length - .take;
;  823          1334  3         );
;  824          1335  3
;  825          1336  3         ! Now we print the record.
;  826          1337  3
;  827          1338  3         anl$format_line(0,1,anlobj$_anything,out_record_dsc);
;  828          1339  2 );
;  829          1340  2
;  830          1341  2 ! If this is an interactive session, let's find out if the user wants to
;  831          1342  2 ! continue or quit.
;  832          1343  2
;  833          1344  2 if .anl$gb_interactive then
;  834          1345  2         return anl$interact()
;  835          1346  2 else
```

```
;  836          1347  2          return true;
;  837          1348  2
;  838          1349  1 end;


                                                07FC 00000          .ENTRY  ANL$IMAGE_PATCH_TEXT, Save R2,R3,R4,R5,R6,- ; 1256
                                                                            R7,R8,R9,R10
                                     5E      FDF0   CE   9E 00002           MOVAB   -528(SP), SP
                                08   AE      0200   8F   3C 00007           MOVZWL  #512, OUT_RECORD_DSC                  ; 1265
                                0C   AE        10   AE   9E 0000D           MOVAB   OUT_RECORD_DSC+8, OUT_RECORD_DSC+4
                                                    5E   DD 00012           PUSHL   SP                                    ; 1271
                                                 08 AE   9F 00014           PUSHAB  BP
                                0000G   CF          02   FB 00017           CALLS   #2, ANL$GET_IMAGE_HEADER
                                        50       04 AE   D0 0001C           MOVL    BP, R0                                ; 1272
                                                 08 A0   B5 00020           TSTW    8(R0)
                                                    0A   13 00023           BEQL    1$
                                        57       08 A0   3C 00025           MOVZWL  8(R0), SP                             ; 1274
                                        57          50   C0 00029           ADDL2   R0, SP
                                                 20 A7   D5 0002C           TSTL    32(SP)                                ; 1275
                                                    03   12 0002F 1$:       BNEQ    2$
                                                 00DB   31 00031           BRW     11$
                                0000G   CF          00   FB 00034 2$:       CALLS   #0, ANL$REPORT_PAGE                   ; 1280
                                     00000000G   8F      DD 00039           PUSHL   #ANLOBJ$_EXEPATCH                     ; 1281
                                                    7E   7C 0003F           CLRQ    -(SP)
                                0000G   CF          03   FB 00041           CALLS   #3, ANL$FORMAT_LINE
                                                    7E   D4 00046           CLRL    -(SP)                                 ; 1282
                                0000G   CF          01   FB 00048           CALLS   #1, ANL$REPORT_LINE
                                                    7E   D4 0004D           CLRL    -(SP)                                 ; 1283
                                0000G   CF          01   FB 0004F           CALLS   #1, ANL$REPORT_LINE
                                        5A       20 A7   D0 00054           MOVL    32(SP), PATCH_VBN                     ; 1287
                                                 04 AE   9F 00058           PUSHAB  BP                                    ; 1288
                                                    5A   DD 0005B           PUSHL   PATCH_VBN
                                0000G   CF          02   FB 0005D           CALLS   #2, ANL$GET_IMAGE_BLOCK
                                        57       04 AE   D0 00062           MOVL    BP, SP                                ; 1289
                                        56          67   32 00066 3$:       CVTWL   (SP), LENGTH                          ; 1302
                                                    18   15 00069           BLEQ    4$                                    ; 1303
                             000000FF   8F          56   D1 0006B           CMPL    LENGTH, #255                          ; 1305
                                                    11   1B 00072           BLEQU   5$
                                        7E       FF 8F   9A 00074           MOVZBL  #255, -(SP)                           ; 1306
                                             00000000G   DD 00078           PUSHL   #ANLOBJ$_EXEBADPATCHLEN
                                0000G   CF          02   FB 0007E           CALLS   #2, ANL$FORMAT_ERROR
                                                    7F   11 00083 4$:       BRB     10$                                   ; 1305
                                        57          02   C0 00085 5$:       ADDL2   #2, SP                                ; 1309
                                                 08 AE   B4 00088           CLRW    OUT_RECORD_DSC                        ; 1311
                             58      04   AE 00000200   8F   C1 0008B       ADDL3   #512, BP, R8                          ; 1316
                                        58          57   D1 00094 6$:       CMPL    SP, R8
                                                    10   1F 00097           BLSSU   7$
                                                 5A D6   00099              INCL    PATCH_VBN                             ; 1317
                                                 04 AE   9F 0009B           PUSHAB  BP                                    ; 1318
                                                 5A DD   0009E              PUSHL   PATCH_VBN
                                0000G   CF          02   FB 000A0           CALLS   #2, ANL$GET_IMAGE_BLOCK
                                        57       04 AE   D0 000A5           MOVL    BP, SP                                ; 1319
                                                    56   D5 000A9 7$:       TSTL    LENGTH                                ; 1324
                                                    42   13 000AB           BEQL    9$
```

```
                58        04    AE 00000200    8F  C1 000AD          ADDL3   #512, BP, R8                                    ; 1329
                51              58             57  C3 000B6          SUBL3   SP, R8, R1                                      ;
                                50             56  D0 000BA          MOVL    LENGTH, R0                                      ;
                                51             50  D1 000BD          CMPL    R0, R1                                          ;
                                               03  1B 000C0          BLEQU   8$                                              ;
                                50             51  D0 000C2          MOVL    R1, R0                                          ;
                                59             50  D0 000C5 8$:      MOVL    R0, TAKE                                        ;
                                50       08    AE  3C 000C8          MOVZWL  OUT_RECORD_DSC, R0                              ; 1330
                                50       0C    AE  C0 000CC          ADDL2   OUT_RECORD_DSC+4, R0                            ;
                60              67             59  28 000D0          MOVC3   TAKE, (SP), (R0)                                ;
                          08    AE             59  A0 000D4          ADDW2   TAKE, OUT_RECORD_DSC                            ; 1331
                51              57             59  C1 000D8          ADDL3   TAKE, SP, R1                                    ; 1332
          7E    00              59             01  7A 000DC          EMUL    #1, TAKE, #0, -(SP)                             ;
          50    50              8E             02  7B 000E1          EDIV    #2, (SP)+, R0, R0                               ;
                57              51             50  C1 000E6          ADDL3   R0, R1, SP                                      ;
                                56             59  C2 000EA          SUBL2   TAKE, LENGTH                                    ;
                                               A5  11 000ED          BRB     6$                                              ; 1333
                                         08    AE  9F 000EF 9$:      PUSHAB  OUT_RECORD_DSC                                  ; 1311
                                00000000G      8F  DD 000F2          PUSHL   #ANL$OBJ$_ANYTHING                              ; 1338
                                               01  DD 000F8          PUSHL   #1                                              ;
                                               7E  D4 000FA          CLRL    -(SP)                                           ;
                          0000G CF             04  FB 000FC          CALLS   #4, ANL$FORMAT_LINE                             ;
                                            FF62 31  00101          BRW     3$                                              ; 1289
                                06      0000G CF  E9  00104 10$:     BLBC    ANL$GB_INTERACTIVE, 11$                         ; 1344
                          0000G CF             00  FB 00109          CALLS   #0, ANL$INTERACT                                ; 1345
                                               04  0010E          RET                                                     ; 1347
                                50             01  D0 0010F 11$:     MOVL    #1, R0                                          ;
                                               04  00112          RET                                                     ; 1349

; Routine Size:  275 bytes,    Routine Base:  $CODE$ + 06BA
```

```
;  840     1350  1 %sbttl 'ANL$IMAGE_GST - Analyze Global Symbol Table'
;  841     1351  1 !++
;  842     1352  1 ! Functional Description:
;  843     1353  1 !       This routine is responsible for analyzing the global symbol table
;  844     1354  1 !       of a shareable image.  We format the information in the report and
;  845     1355  1 !       check its validity.
;  846     1356  1 !
;  847     1357  1 ! Formal Parameters:
;  848     1358  1 !       none
;  849     1359  1 !
;  850     1360  1 ! Implicit Inputs:
;  851     1361  1 !       global data
;  852     1362  1 !
;  853     1363  1 ! Implicit Outputs:
;  854     1364  1 !       global data
;  855     1365  1 !
;  856     1366  1 ! Returned Value:
;  857     1367  1 !       If interactive session:  true if we are to continue, false if not.
;  858     1368  1 !
;  859     1369  1 ! Side Effects:
;  860     1370  1 !
;  861     1371  1 !--
;  862     1372  1
;  863     1373  1
;  864     1374  2 global routine anl$image_gst = begin
;  865     1375  2
;  866     1376  2 local
;  867     1377  2         bp: ref block[,byte],
;  868     1378  2         sp: ref block[,byte],
;  869     1379  2         gst_vbn: long,
;  870     1380  2         gst_record_count: long,
;  871     1381  2         length: long,
;  872     1382  2         take: long,
;  873     1383  2         alias,
;  874     1384  2         local_described_buffer(record_dsc,512);
;  875     1385  2
;  876     1386  2
;  877     1387  2 ! The global symbol table origin information has already been checked.
;  878     1388  2 ! If this isn't a shareable image or the information is missing, forget it.
;  879     1389  2
;  880     1390  2 anl$get_image_header(bp,alias);
;  881     1391  2 if .bp[ihd$b_imgtype] nequ ihd$k_lim or .bp[ihd$w_symdbgoff] eqlu 0 then
;  882     1392  2         return true;
;  883     1393  2 sp = .bp + .bp[ihd$w_symdbgoff];
;  884     1394  2 if .sp[ihs$l_gstvbn] eqlu 0 then
;  885     1395  2         return true;
;  886     1396  2
;  887     1397  2 ! We seem to have a GST.  Let's eject the page and start with a heading.
;  888     1398  2
;  889     1399  2 anl$report_page();
;  890     1400  2 anl$format_line(0,0,anlobj$_exegst);
;  891     1401  2 anl$report_line(0);
;  892     1402  2 anl$report_line(0);
;  893     1403  2
;  894     1404  2 ! We need the VBN of the global symbol table and its record count.  Get
;  895     1405  2 ! the first block of the table.
;  896     1406  2
```

```
;  897          1407  2 gst_vbn = .sp[ihs$l_gstvbn];
;  898          1408  2 gst_record_count = .sp[ihs$w_gstrecs];
;  899          1409  2 anl$get_image_block(.gst_vbn,bp);
;  900          1410  2 sp = .bp;
;  901          1411  2
;  902          1412  2 ! OK, now we are going to loop through the object records in the global
;  903          1413  2 ! symbol table.  We construct each record from the blocks of the image and
;  904          1414  2 ! analyze them using the object file analysis routines.
;  905          1415  2
;  906          1416  3 incru record_number from 1 to .gst_record_count do (
;  907          1417  3
;  908          1418  3         ! Sit in a loop and build the next object record.  GST_VBN is the
;  909          1419  3         ! block number we are at.  SP points to the beginning of the record,
;  910          1420  3         ! which is a length.
;  911          1421  3
;  912          1422  3         length = .sp[0,0,16,0];
;  913          1423  3         sp = .sp + 2;
;  914          1424  3         record_dsc[len] = 0;
;  915          1425  3
;  916          1426  4         loop (
;  917          1427  4
;  918          1428  4                 ! If we have run off the end of this block, let's get another.
;  919          1429  4
;  920          1430  5                 if .sp gequa .bp+512 then (
;  921          1431  5                         increment (gst_vbn);
;  922          1432  5                         anl$get_image_block(.gst_vbn, bp);
;  923          1433  5                         sp = .bp;
;  924          1434  4                 );
;  925          1435  4
;  926          1436  4                 ! If we have built the entire record, drop out.
;  927          1437  4
;  928          1438  4         exitif (.length eqlu 0);
;  929          1439  4
;  930          1440  4                 ! Take as many bytes as we can from this block to build
;  931          1441  4                 ! the record.  Adjust things.
;  932          1442  4
;  933          1443  4                 take = minu(.length, .bp+512-.sp);
;  934          1444  4                 ch$move(.take,.sp, .record_dsc[ptr]+.record_dsc[len]);
;  935          1445  4                 record_dsc[len] = .record_dsc[len] + .take;
;  936          1446  4                 sp = .sp + .take + .take mod 2;
;  937          1447  4                 length = .length - .take;
;  938          1448  3         );
;  939          1449  3
;  940          1450  3         ! Now we can analyze the record, assuming it is a least one byte
;  941          1451  3         ! in length.  Select on its type.
;  942          1452  3
;  943          1453  4         if .record_dsc[len] gequ 1 then (
;  944          1454  4
;  945          1455  4                 selectoneu ch$rchar(.record_dsc[ptr]) of set
;  946          1456  4                 [obj$c_hdr]:    anl$object_hdr(.record_number,record_dsc);
;  947          1457  4
;  948          1458  4                 [obj$c_gsd]:    anl$object_gsd(.record_number,record_dsc);
;  949          1459  4
;  950          1460  4                 [obj$c_eom]:    anl$object_eom(.record_number,record_dsc);
;  951          1461  4
;  952          1462  5                 [otherwise]:    (anl$format_error(anlobj$_exebadobj,.record_number,ch$rchar(.record_dsc[ptr]
;  953          1463  4                                 anl$format_hex(1,record_dsc););
```

```
;  954          1464  4                    tes;
;  955          1465  4
;  956          1466  4                    ! Make sure that this record isn't longer than the maximum size
;  957          1467  4                    ! specified in the module header.
;  958          1468  4
;  959          1469  4                    anl$object_record_size(.record_dsc[len]);
;  960          1470  4
;  961          1471  4                    ! Skip a couple of lines to make it look nice.
;  962          1472  4
;  963          1473  4                    anl$report_line(-1);
;  964          1474  4                    anl$report_line(-1);
;  965          1475  4
;  966          1476  4                    ! If this is an interactive session, let's find out if the
;  967          1477  4                    ! user wants to continue or quit.
;  968          1478  4
;  969          1479  4                    if .anl$gb_interactive then
;  970          1480  4                            if not anl$interact() then
;  971          1481  4                                    return false;
;  972          1482  4
;  973          1483  4            ) else (
;  974          1484  4
;  975          1485  4                    ! There was no record type.  Tell the user.
;  976          1486  4
;  977          1487  4                    anl$format_error(anlobj$_objnullrec,,.record_number);
;  978          1488  4                    anl$report_line(-1);
;  979          1489  4                    anl$report_line(-1);
;  980          1490  3            );
;  981          1491  2 );
;  982          1492  2
;  983          1493  2 return true;
;  984          1494  2
;  985          1495  1 end;

                                 0FFC 00000          .ENTRY  ANL$IMAGE_GST, Save R2,R3,R4,R5,R6,R7,R8,-      ; 1374
                                                             R9,R10,R11
            5E      FDEC    CE   9E 00002           MOVAB   -532(SP), SP                                    ;
      0C    AE      0200    8F   3C 00007           MOVZWL  #512, RECORD_DSC                                ; 1384
      10    AE        14    AE   9E 0000D           MOVAB   RECORD_DSC+8, RECORD_DSC+4                      ;
                      04    AE   9F 00012           PUSHAB  ALIAS                                           ; 1390
                      0C    AE   9F 00015           PUSHAB  BP                                              ;
0000G CF                    02   FB 00018           CALLS   #2, ANL$GET_IMAGE_HEADER                        ;
            50        08    AE   D0 0001D           MOVL    BP, R0                                          ; 1391
            02        11    A0   91 00021           CMPB    17(R0), #2                                      ;
                            03   13 00025           BEQL    2$                                              ;
                      0158  31   00027 1$:           BRW     15$                                             ;
                      04    A0   B5 0002A 2$:       TSTW    4(R0)                                           ;
                            F8   13 0002D           BEQL    1$                                              ;
            57        04    A0   3C 0002F           MOVZWL  4(R0), SP                                       ; 1393
            57              50   C0 00033           ADDL2   R0, SP                                          ;
                      04    A7   D5 00036           TSTL    4(SP)                                           ; 1394
                            EC   13 00039           BEQL    1$                                              ;
0000G CF                    00   FB 0003B           CALLS   #0, ANL$REPORT_PAGE                             ; 1399
                00000000G   8F   DD 00040           PUSHL   #ANLOBJ$_EXEGST                                 ; 1400
```

```
                                            7E    7C 00046         CLRQ     -(SP)                              ;
                      0000G   CF            03    FB 00048         CALLS    #3, ANL$FORMAT_LINE                ;
                                            7E    D4 0004D         CLRL     -(SP)                              ; 1401
                      0000G   CF            01    FB 0004F         CALLS    #1, ANL$REPORT_LINE                ;
                                            7E    D4 00054         CLRL     -(SP)                              ; 1402
                      0000G   CF            01    FB 00056         CALLS    #1, ANL$REPORT_LINE                ;
                              5B       04   A7    D0 0005B         MOVL     4(SP), GST_VBN                     ; 1407
                              6E       0A   A7    3C 0005F         MOVZWL   10(SP), GST_RECORD_COUNT           ; 1408
                                       08   AE    9F 00063         PUSHAB   BP                                 ; 1409
                                            5B    DD 00066         PUSHL    GST_VBN                            ;
                      0000G   CF            02    FB 00068         CALLS    #2, ANL$GET_IMAGE_BLOCK            ;
                              57       08   AE    D0 0006D         MOVL     BP, SP                             ; 1410
                              58            01    D0 00071         MOVL     #1, RECORD_NUMBER                  ; 1416
                                          0103    31 00074         BRW      14$                                ;
                              56            87    3C 00077  3$:    MOVZWL   (SP)+, LENGTH                      ; 1422
                                       0C   AE    B4 0007A         CLRW     RECORD_DSC                         ; 1424
            59        08      AE  00000200  8F    C1 0007D         ADDL3    #512, BP, R9                       ; 1430
                              59            57    D1 00086  4$:    CMPL     SP, R9                             ;
                                            10    1F 00089         BLSSU    5$                                 ;
                                            5B    D6 0008B         INCL     GST_VBN                            ; 1431
                                       08   AE    9F 0008D         PUSHAB   BP                                 ; 1432
                                            5B    DD 00090         PUSHL    GST_VBN                            ;
                      0000G   CF            02    FB 00092         CALLS    #2, ANL$GET_IMAGE_BLOCK            ;
                              57       08   AE    D0 00097         MOVL     BP, SP                             ; 1433
                                            56    D5 0009B  5$:    TSTL     LENGTH                             ; 1438
                                            42    13 0009D         BEQL     7$                                 ;
            59        08      AE  00000200  8F    C1 0009F         ADDL3    #512, BP, R9                       ; 1443
            51                59            57    C3 000A8         SUBL3    SP, R9, R1                         ;
                              50            56    D0 000AC         MOVL     LENGTH, R0                         ;
                              51            50    D1 000AF         CMPL     R0, R1                             ;
                                            03    1B 000B2         BLEQU    6$                                 ;
                              50            51    D0 000B4         MOVL     R1, R0                             ;
                              5A            50    D0 000B7  6$:    MOVL     R0, TAKE                           ;
                              50       0C   AE    3C 000BA         MOVZWL   RECORD_DSC, R0                     ; 1444
                              50       10   AE    C0 000BE         ADDL2    RECORD_DSC+4, R0                   ;
            60                67            5A    28 000C2         MOVC3    TAKE, (SP), (R0)                   ;
                      0C      AE            5A    A0 000C6         ADDW2    TAKE, RECORD_DSC                   ; 1445
            51                57            5A    C1 000CA         ADDL3    TAKE, SP, R1                       ; 1446
   7E       00                5A            01    7A 000CE         EMUL     #1, TAKE, #0, -(SP)                ;
   50       50                8E            02    7B 000D3         EDIV     #2, (SP)+, R0, R0                  ;
            57                51            50    C1 000D8         ADDL3    R0, R1, SP                         ;
                              56            5A    C2 000DC         SUBL2    TAKE, LENGTH                       ; 1447
                                            A5    11 000DF         BRB      4$                                 ; 1424
                                       0C   AE    B5 000E1  7$:    TSTW     RECORD_DSC                         ; 1453
                                            75    13 000E4         BEQL     12$                                ;
                              52       10   BE    9A 000E6         MOVZBL   @RECORD_DSC+4, R2                  ; 1455
                                            0C    12 000EA         BNEQ     8$                                 ; 1456
                                       0C   AE    9F 000EC         PUSHAB   RECORD_DSC                         ;
                                            58    DD 000EF         PUSHL    RECORD_NUMBER                      ;
                      0000G   CF            02    FB 000F1         CALLS    #2, ANL$OBJECT_HDR                 ;
                                            3B    11 000F6         BRB      11$                                ;
                              01            52    91 000F8  8$:    CMPB     R2, #1                             ; 1458
                                            0C    12 000FB         BNEQ     9$                                 ;
                                       0C   AE    9F 000FD         PUSHAB   RECORD_DSC                         ;
                                            58    DD 00100         PUSHL    RECORD_NUMBER                      ;
                      0000G   CF            02    FB 00102         CALLS    #2, ANL$OBJECT_GSD                 ;
                                            2A    11 00107         BRB      11$                                ;
```

```
                         03              52  91 00109 9$:     CMPB    R2, #3                                  ; 1460
                                         0C  12 0010C         BNEQ    10$
                               0C        AE  9F 0010E         PUSHAB  RECORD_DSC
                                         58  DD 00111         PUSHL   RECORD_NUMBER
                     0000G     CF              02  FB 00113         CALLS   #2, AN[$OBJECT_EOM
                                               19  11 00118         BRB     11$
                                         52  DD 0011A 10$:     PUSHL   R2                                      ; 1462
                                         58  DD 0011C         PUSHL   RECORD_NUMBER
                           00000000G     8F  DD 0011E         PUSHL   #ANLOBJ$_EXEBADOBJ
                     0000G     CF              03  FB 00124         CALLS   #3, ANL$FORMAT_ERROR
                               0C        AE  9F 00129         PUSHAB  RECORD_DSC                              ; 1463
                                         01  DD 0012C         PUSHL   #1
                     0000G     CF              02  FB 0012E         CALLS   #2, ANL$FORMAT_HEX
                               7E  0C    AE  3C 00133 11$:     MOVZWL  RECORD_DSC, -(SP)                       ; 1469
                     0000G     CF              01  FB 00137         CALLS   #1, AN[$OBJECT_RECORD_SIZE
                               7E              01  CE 0013C         MNEGL   #1, -(SP)                       ; 1473
                     0000G     CF              01  FB 0013F         CALLS   #1, ANL$REPORT_LINE
                               7E              01  CE 00144         MNEGL   #1, -(SP)                       ; 1474
                     0000G     CF              01  FB 00147         CALLS   #1, ANL$REPORT_LINE
                         27        0000G     CF  E9 0014C         BLBC    ANL$GB_INTERACTIVE, 13$         ; 1479
                     0000G     CF              00  FB 00151         CALLS   #0, AN[$INTERACT                ; 1480
                         1F                  50  E8 00156         BLBS    R0, 13$
                                             2B  11 00159         BRB     16$                             ; 1481
                                         58  DD 0015B 12$:     PUSHL   RECORD_NUMBER                   ; 1487
                           00000000G     8F  DD 0015D         PUSHL   #ANLOBJ$_OBJNULLREC
                     0000G     CF              02  FB 00163         CALLS   #2, ANL$FORMAT_ERROR
                               7E              01  CE 00168         MNEGL   #1, -(SP)                       ; 1488
                     0000G     CF              01  FB 0016B         CALLS   #1, ANL$REPORT_LINE
                               7E              01  CE 00170         MNEGL   #1, -(SP)                       ; 1489
                     0000G     CF              01  FB 00173         CALLS   #1, ANL$REPORT_LINE
                                         58  D6 00178 13$:     INCL    RECORD_NUMBER                   ; 1416
                         6E              58  D1 0017A 14$:     CMPL    RECORD_NUMBER, GST_RECORD_COUNT
                                         03  1A 0017D         BGTRU   15$
                                       FEF5  31 0017F         BRW     3$
                                 50      01  D0 00182 15$:     MOVL    #1, R0                          ; 1493
                                             04 00185         RET
                                         50  D4 00186 16$:     CLRL    R0                              ; 1495
                                             04 00188         RET

; Routine Size:  393 bytes,    Routine Base:  $CODE$ + 07CD

;  986          1496  1
;  987          1497  0 end eludom



;                                        PSECT SUMMARY

;       Name                            Bytes                           Attributes

;  $PLIT$                                332  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                                 180  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                               2390  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]LIB.L32;1         18619         88          0        1000        00:01.8
;
;
;
;                                       COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:EXESTUFF/OBJ=OBJ$:EXESTUFF MSRC$:EXESTUFF/UPDATE=(ENH$:EXESTUFF)
;
; Size:         2390 code + 512 data bytes
; Run Time:       00:40.8
; Elapsed Time:   01:58.6
; Lines/CPU Min:   2202
; Lexemes/CPU-Min: 15132
; Memory Used:  392 pages
; Compilation Complete
```

SETSHOAL
CLS

OBJEXEREQ
REQ

EXEFIXUP
CLS

ANALYZRMS
MAP

SHOWACL
CLS

EXESTUFF
CLS

ANALYZ

EXEINPUT
CLS

ANALYZOBJ
MAP

EXEDRIVE
CLS

RMSREQ
REQ