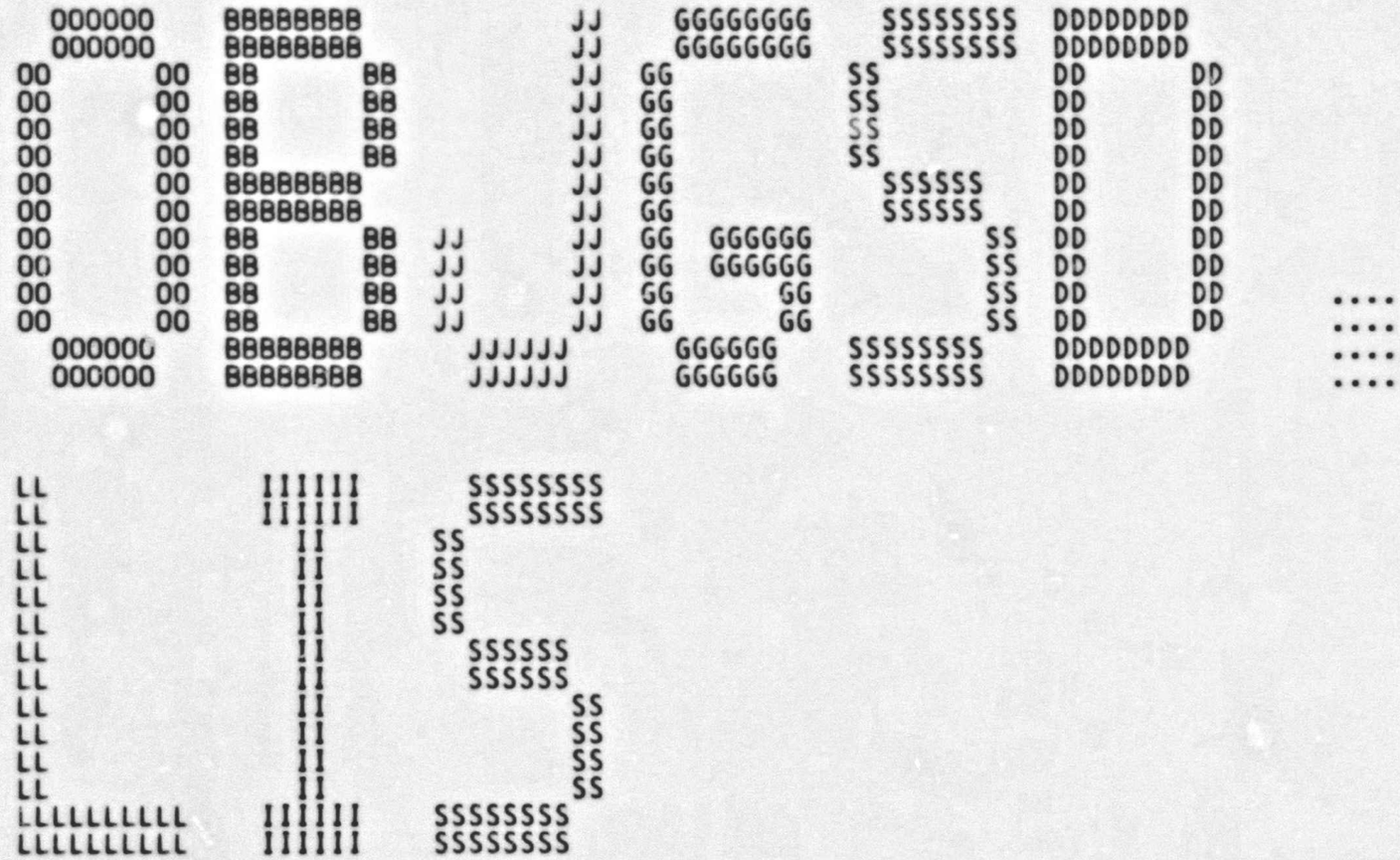
```
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL                YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL                YYY         YYY  ZZZZZZZZZZZZZZZ
   AAAAAAAAA     NNN         NNN     AAAAAAAAA     LLL                YYY         YYY  ZZZZZZZZZZZZZZZ
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
```

FILEID**OBJGSD



```
;  1          0001  0 %title 'OBJGSD - Analyze GSD Records'
;  2          0002  0         module objgsd
;  3          0003  0                 (ident = 'V04-000') =
;  4          0004  1 begin
;  5          0005  1
;  6          0006  1 !
;  7          0007  1 !*****************************************************************************
;  8          0008  1 !*                                                                           *
;  9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                                    *
; 12          0012  1 !*                                                                         *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 18          0018  1 !*  TRANSFERRED.                                                            *
; 19          0019  1 !*                                                                         *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 22          0022  1 !*  CORPORATION.                                                            *
; 23          0023  1 !*                                                                         *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
; 26          0026  1 !*                                                                         *
; 27          0027  1 !*                                                                         *
; 28          0028  1 !*****************************************************************************
; 29          0029  1 !
; 30          0030  1
; 31          0031  1 !++
; 32          0032  1 ! Facility:     VAX/VMS Analyze Facility, Analyze GSD Object Records
; 33          0033  1 !
; 34          0034  1 ! Abstract:     This module handles the analysis of GSD records.
; 35          0035  1 !
; 36          0036  1 !
; 37          0037  1 ! Environment:
; 38          0038  1 !
; 39          0039  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 20 January 1980
; 40          0040  1 !
; 41          0041  1 ! Modified By:
; 42          0042  1 !
; 43          0043  1 !       V03-004 MCN0175         Maria del C. Nasr         9-Jul-1984
; 44          0044  1 !               When processing environment subrecords call
; 45          0045  1 !               ANL$OBJECT_ENV_REF and not ANL$OBJECT_PSECT_REF.
; 46          0046  1 !
; 47          0047  1 !       V03-003 ADE0001         Alan D. Eldridge          6-Jul-1984
; 48          0048  1 !               Add ENV$V_NESTED support.
; 49          0049  1 !
; 50          0050  1 !       V03-002 MCN0158         Maria del C. Nasr        22-Mar-1984
; 51          0051  1 !               Add size parameter to call to ANL$CHECK_SYMBOL.
; 52          0052  1 !
; 53          0053  1 !       V03-001 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
; 54          0054  1 !               Change the message prefix to ANLOBJ$_ to ensure that
; 55          0055  1 !               message symbols are unique across all ANALYZEs.  This
; 56          0056  1 !               is necessitated by the new merged message files.
; 57          0057  1 !--
```

```
;    59          0058  1 %sbttl 'Module Declarations'
;    60          0059  1 !
;    61          0060  1 ! Libraries and Requires:
;    62          0061  1 !
;    63          0062  1
;    64          0063  1 library 'starlet';
;    65          0064  1 require 'objexereq';
;    66          0500  1
;    67          0501  1 !
;    68          0502  1 ! Table of Contents:
;    69          0503  1 !
;    70          0504  1
;    71          0505  1 forward routine
;    72          0506  1         anl$object_gsd: novalue,
;    73          0507  1         anl$object_argument_dsc,
;    74          0508  1         anl$object_psect_ref: novalue,
;    75          0509  1         anl$object_psect_check: novalue,
;    76          0510  1         anl$object_env_ref: novalue,
;    77          0511  1         anl$object_env_check: novalue;
;    78          0512  1
;    79          0513  1 !
;    80          0514  1 ! External References:
;    81          0515  1 !
;    82          0516  1
;    83          0517  1 external routine
;    84          0518  1         anl$check_flags,
;    85          0519  1         anl$check_symbol,
;    86          0520  1         anl$format_data_type,
;    87          0521  1         anl$format_error,
;    88          0522  1         anl$format_flags,
;    89          0523  1         anl$format_hex,
;    90          0524  1         anl$format_line,
;    91          0525  1         anl$format_mask,
;    92          0526  1         anl$format_severity,
;    93          0527  1         anl$object_record_line,
;    94          0528  1         anl$report_line,
;    95          0529  1         lib$free_vm: addressing_mode(general),
;    96          0530  1         lib$get_vm: addressing_mode(general);
;    97          0531  1
;    98          0532  1 !
;    99          0533  1 ! Own Variables:
;   100          0534  1 !
;   101          0535  1
;   102          0536  1 ! The following variables are needed to keep track of psect references and
;   103          0537  1 ! check their validity.
;   104          0538  1
;   105          0539  1 own
;   106          0540  1         highest_def_psect: signed long initial(-1),
;   107          0541  1         highest_ref_psect: signed long initial(-1),
;   108          0542  1         psect_ref_bits: ref bitvector[65536];
;   109          0543  1
;   110          0544  1 ! The following variables perform the same function, but for environments.
;   111          0545  1
;   112          0546  1 own
;   113          0547  1         highest_def_env: signed long initial(-1),
;   114          0548  1         highest_ref_env: signed long initial(-1),
;   115          0549  1         env_ref_bits: ref bitvector[65536];
```

```
;  117          0550  1 %sbttl 'ANL$OBJECT_GSD - Analyze GSD Object Records'
;  118          0551  1 !++
;  119          0552  1 ! Functional Description:
;  120          0553  1 !       This routine is responsible for analyzing the GSD object records.
;  121          0554  1 !
;  122          0555  1 ! Formal Parameters:
;  123          0556  1 !       record_number   The number of this object record.
;  124          0557  1 !       the_record      Address of descriptor of the object record.
;  125          0558  1 !
;  126          0559  1 ! Implicit Inputs:
;  127          0560  1 !       global data
;  128          0561  1 !
;  129          0562  1 ! Implicit Outputs:
;  130          0563  1 !       global data
;  131          0564  1 !
;  132          0565  1 ! Returned Value:
;  133          0566  1 !       none
;  134          0567  1 !
;  135          0568  1 ! Side Effects:
;  136          0569  1 !
;  137          0570  1 !--
;  138          0571  1
;  139          0572  1
;  140          0573  2 global routine anl$object_gsd(record_number,the_record): novalue = begin
;  141          0574  2
;  142          0575  2 bind
;  143          0576  2         record_dsc = .the_record: descriptor;
```

```
;  145          0577  2 ! The following data structures define the various flag bytes and words
;  146          0578  2 ! that are present in GSD records.
;  147          0579  2
;  148          0580  2 ! This defines the flags in a psect definition subrecord:
;  149          0581  2
;  150          0582  2 own
;  151          0583  2         psc_flags_def: vector[11,long] initial(
;  152          0584  2                                         9,
;  153          0585  2                                         uplit byte(%ascic 'GPS$V_PIC'),
;  154          0586  2                                         uplit byte(%ascic 'GPS$V_LIB'),
;  155          0587  2                                         uplit byte(%ascic 'GPS$V_OVL'),
;  156          0588  2                                         uplit byte(%ascic 'GPS$V_REL'),
;  157          0589  2                                         uplit byte(%ascic 'GPS$V_GBL'),
;  158          0590  2                                         uplit byte(%ascic 'GPS$V_SHR'),
;  159          0591  2                                         uplit byte(%ascic 'GPS$V_EXE'),
;  160          0592  2                                         uplit byte(%ascic 'GPS$V_RD'),
;  161          0593  2                                         uplit byte(%ascic 'GPS$V_WRT'),
;  162          0594  2                                         uplit byte(%ascic 'GPS$V_VEC'));
;  163          0595  2
;  164          0596  2 ! This defines the flags in the symbol, entry point, and procedure subrecords.
;  165          0597  2
;  166          0598  2 own
;  167          0599  2         sym_flags_def: vector[5,long] initial(
;  168          0600  2                                         3,
;  169          0601  2                                         uplit byte(%ascic 'GSY$V_WEAK'),
;  170          0602  2                                         uplit byte(%ascic 'GSY$V_DEF'),
;  171          0603  2                                         uplit byte(%ascic 'GSY$V_UNI'),
;  172          0604  2                                         uplit byte(%ascic 'GSY$V_REL'));
;  173          0605  2
;  174          0606  2 ! This defines the flags in the environment subrecord.
;  175          0607  2
;  176          0608  2 own
;  177          0609  2         env_flags_def: vector[3,long] initial(
;  178          0610  2                                         1,
;  179          0611  2                                         uplit byte(%ascic 'ENV$V_DEF'),
;  180          0612  2                                         uplit byte(%ascic 'ENV$V_NESTED'));
;  181          0613  2
;  182          0614  2 ! This defines the flags in the entity check subrecord.
;  183          0615  2
;  184          0616  2 own
;  185          0617  2         entity_flags_def: vector[2,long] initial(
;  186          0618  2                                         0,
;  187          0619  2                                         uplit byte(%ascic 'IDC$V_BINIDENT'));
```

```
;  189          0620  2 own
;  190          0621  2         gsd_subrecord_msg: vector[gsd$c_maxrectyp+1,long] initial(
;  191          0622  2                                                 anlobj$_objgsdpsc,
;  192          0623  2                                                 anlobj$_objgsdsym,
;  193          0624  2                                                 anlobj$_objgsdepm,
;  194          0625  2                                                 anlobj$_objgsdpro,
;  195          0626  2                                                 anlobj$_objgsdsymw,
;  196          0627  2                                                 anlobj$_objgsdepmw,
;  197          0628  2                                                 anlobj$_objgsdprow,
;  198          0629  2                                                 anlobj$_objgsdidc,
;  199          0630  2                                                 anlobj$_objgsdenv,
;  200          0631  2                                                 anlobj$_objgsdlsy,
;  201          0632  2                                                 anlobj$_objgsdlepm,
;  202          0633  2                                                 anlobj$_objgsdlpro,
;  203          0634  2                                                 anlobj$_objgsdspsc);
;  204          0635  2
;  205          0636  2 local
;  206          0637  2         status: long,
;  207          0638  2         gsd_type: byte,
;  208          0639  2         scanp: ref block[,byte],
;  209          0640  2         subrecord_number: long,
;  210          0641  2         fit_ok: byte,
;  211          0642  2         work_dsc: descriptor;
;  212          0643  2
;  213          0644  2
;  214          0645  2 ! We begin by printing a major line for the record.
;  215          0646  2
;  216          0647  2 anl$object_record_line(anlobj$_objgsdrec,.record_number,record_dsc);
;  217          0648  2
;  218          0649  2 ! Now we go into a loop processing the subrecords in the record.
;  219          0650  2 ! SUBRECORD_NUMBER will count them as we go.
;  220          0651  2 ! SCANP will advance along the various subrecords of the record.
;  221          0652  2 ! FIT_OK will remain true unless a field spills off the end of the record.
;  222          0653  2
;  223          0654  2 subrecord_number = 0;
;  224          0655  2 scanp = .record_dsc[ptr] + 1;
;  225          0656  2 fit_ok = true;
;  226          0657  3 while (.scanp lssa .record_dsc[ptr]+.record_dsc[len]) and .fit_ok do (
;  227          0658  3
;  228          0659  3         ! Count the subrecord and prepare to print it nicely.  Then print a
;  229          0660  3         ! minor line for the subrecord.  If the subrecord type is invalid,
;  230          0661  3         ! show the user and forget the record.
;  231          0662  3
;  232          0663  3         increment (subrecord_number);
;  233          0664  3         anl$report_line(0);
;  234          0665  3
;  235          0666  3         gsd_type = .scanp[0,0,8,0];
;  236          0667  3         if .gsd_type lequ gsd$c_maxrectyp then
;  237          0668  3                 anl$format_line(2,1,.gsd_subrecord_msg[.gsd_type],.subrecord_number)
;  238          0669  4         else (
;  239          0670  4                 anl$format_error(anlobj$_objgsdbadsubtyp,,gsd_type);
;  240          0671  4                 build_descriptor(work_dsc,.record_dsc[len]-(.scanp-.record_dsc[ptr]),.record_dsc[ptr]);
;  241          0672  4                 anl$format_hex(2,work_dsc);
;  242          0673  4                 return;
;  243          0674  3         );
;  244          0675  3
;  245          0676  3         ! Now we can select on the subrecord type and analyze the subrecord.
```

```
;  246          0677  3
;  247          0678  3          case .gsd_type from 0 to gsd$c_maxrectyp of set
```

```
;  249     0679  3          [gsd$c_psc,
;  250     0680  3           gsd$c_spsc]:
;  251     0681  3
;  252     0682  3                   ! We have a psect definition subrecord.  The first field
;  253     0683  3                   ! contains the psect alignment.  Print it and check it.
;  254     0684  3                   ! Also include the psect number for this guy.
;  255     0685  3
;  256     0686  4                   (ensure_field_fit(gps$b_align,record_dsc);
;  257     0687  5                   if .fit_ok then (
;  258     0688  5                           increment (highest_def_psect);
;  259     0689  5                           anl$format_line(0,2,anlobj$_objgsdpscalign,1^.scanp[gps$b_align],
;  260     0690  5                                   .highest_def_psect);
;  261     0691  5                           if .scanp[gps$b_align] gtru obj$c_pscalilim then
;  262     0692  5                                   anl$format_error(anlobj$_objgsdbadalign,obj$c_pscalilim);
;  263     0693  4                   );
;  264     0694  4
;  265     0695  4                   ! The next field is the flags byte.  Print it and check it.
;  266     0696  4
;  267     0697  4                   ensure_field_fit(gps$w_flags,record_dsc);
;  268     0698  5                   if .fit_ok then (
;  269     0699  5                           anl$format_flags(2,anlobj$_objgsdpscflags,.scanp[gps$w_flags],psc_flags_def);
;  270     0700  5                           anl$check_flags(.scanp[gps$w_flags],psc_flags_def);
;  271     0701  4                   );
;  272     0702  4
;  273     0703  4                   ! The next field is the allocation size.  Print it and check.
;  274     0704  4
;  275     0705  4                   ensure_field_fit(gps$l_alloc,record_dsc);
;  276     0706  5                   if .fit_ok then (
;  277     0707  5                           anl$format_line(0,2,anlobj$_objgsdpscalloc,.scanp[gps$l_alloc]);
;  278     0708  5                           if .scanp[gps$l_alloc] gtru %x'3fffffff' then
;  279     0709  5                                   anl$format_error(anlobj$_objp0space);
;  280     0710  5                           if not .scanp[gps$v_rel] and .scanp[gps$l_alloc] nequ 0 then
;  281     0711  5                                   anl$format_error(anlobj$_objpscabslen);
;  282     0712  4                   );
;  283     0713  4
;  284     0714  4                   ! The next field is only present in shareable image psect
;  285     0715  4                   ! entries.  It contain the base address of the psect in
;  286     0716  4                   ! the shareable image.  Print it and check.  We also have
;  287     0717  4                   ! to set up SCANP for the next field, since it can be at
;  288     0718  4                   ! two different offset.
;  289     0719  4
;  290     0720  5                   if .gsd_type eqlu gsd$c_spsc then (
;  291     0721  5                           ensure_field_fit(sgps$l_base,record_dsc);
;  292     0722  5                           if .fit_ok then
;  293     0723  5                                   anl$format_line(0,2,anlobj$_objgsdpscbase,.scanp[sgps$l_base]);
;  294     0724  5                           scanp = scanp[sgps$b_namlng];
;  295     0725  4                   ) else
;  296     0726  4                           scanp = scanp[gps$b_namlng];
;  297     0727  4
;  298     0728  4                   ! The final field is the psect name.  Print it and check it.
;  299     0729  4
;  300     0730  4                   ensure_ascic_fit(0,0,8,0,record_dsc,work_dsc);
;  301     0731  5                   if .fit_ok then (
;  302     0732  5                           anl$format_line(0,2,anlobj$_objsymbol,.work_dsc[len],.work_dsc[ptr]);
;  303     0733  5                           anl$check_symbol(work_dsc, obj$c_symsiz);
;  304     0734  4                   );
;  305     0735  4
```

```
;  306          0736  4                    ! Finally, advance the scan pointer past this subrecord.
;  307          0737  4
;  308          0738  4                    scanp = .work_dsc[ptr] + .work_dsc[len];
;  309          0739  3                    );
```

```
;  311          0740  3          [gsd$c_sym,
;  312          0741  3           gsd$c_epm,
;  313          0742  3           gsd$c_pro,
;  314          0743  3           gsd$c_symw,
;  315          0744  3           gsd$c_epmw,
;  316          0745  3           gsd$c_prow,
;  317          0746  3           gsd$c_lsy,
;  318          0747  3           gsd$c_lepm,
;  319          0748  3           gsd$c_lpro]:
;  320          0749  3
;  321          0750  3                   ! We have a symbol specification, entry point definition,
;  322          0751  3                   ! or procedure definition.  As we proceed, we will need to
;  323          0752  3                   ! know if it is a symbol specification and what kind.
;  324          0753  3
;  325          0754  4                   (local
;  326          0755  4                           symbol_spec: byte,
;  327          0756  4                           symbol_def: byte;
;  328          0757  4
;  329          0758  4
;  330          0759  4                   ! All of these records begin with a data type, so let's print
;  331          0760  4                   ! it in the report.
;  332          0761  4
;  333          0762  4                   ensure_field_fit(gsy$b_datyp,record_dsc);
;  334          0763  5                   if .fit_ok then (
;  335          0764  5                           anl$format_data_type(2,.scanp[gsy$b_datyp]);
;  336          0765  4                   );
;  337          0766  4
;  338          0767  4                   ! All the records also contain a byte of flags.  Let's print
;  339          0768  4                   ! them and check.
;  340          0769  4                   ! There was a BUG in the V2 linker that sometimes caused
;  341          0770  4                   ! flag 11 to be set.  To avoid a flood of SPRs, we will
;  342          0771  4                   ! force that flag off so we won't produce an error.
;  343          0772  4
;  344          0773  4                   ensure_field_fit(gsy$w_flags,record_dsc);
;  345          0774  5                   if .fit_ok then (
;  346          0775  5                           anl$format_flags(2,anlobj$_objsymflags,.scanp[gsy$w_flags],sym_flags_def);
;  347          0776  5                           anl$check_flags(.scanp[gsy$w_flags] and %x'f7ff',sym_flags_def);
;  348          0777  5
;  349          0778  5                           ! Now let's figure out if this is a symbol specification.
;  350          0779  5                           ! Also record whether it is a reference or definition.
;  351          0780  5
;  352          0781  5                           symbol_spec = (.gsd_type eqlu gsd$c_sym) or
;  353          0782  5                                         (.gsd_type eqlu gsd$c_symw) or
;  354          0783  5                                         (.gsd_type eqlu gsd$c_lsy);
;  355          0784  5                           symbol_def  = .scanp[gsy$v_def];
;  356          0785  5
;  357          0786  4                   );
;  358          0787  4
;  359          0788  4                   ! From now on it becomes hard to keep track of where we
;  360          0789  4                   ! are, since different subrecords have different formats.
;  361          0790  4                   ! We will use SCANP to point at successive fields.
;  362          0791  4
;  363          0792  4                   scanp = scanp[gsy$w_flags] + 2;
;  364          0793  4
;  365          0794  4                   ! At this point we have an environment index if this is a
;  366          0795  4                   ! local symbol subrecord.
;  367          0796  4
```

```
;  368     0797  4                    if .gsd_type eqlu gsd$c_lsy or
;  369     0798  4                       .gsd_type eqlu gsd$c_lepm or
;  370     0799  5                       .gsd_type eqlu gsd$c_lpro then (
;  371     0800  5                            ensure_field_fit(0,0,16,0,record_dsc);
;  372     0801  6                            if .fit_ok then (
;  373     0802  6                                 anl$format_line(0,2,anlobj$_objenv,.scanp[0,0,16,0]);
;  374     0803  6                                 anl$object_env_ref(.scanp[0,0,16,0]);
;  375     0804  6                                 scanp = .scanp + 2;
;  376     0805  5                            );
;  377     0806  4                    );
;  378     0807  4
;  379     0808  4                    ! At this point we have some fields that are present in all
;  380     0809  4                    ! records except symbol references.
;  381     0810  4
;  382     0811  5                    if not (.symbol_spec and not .symbol_def) then (
;  383     0812  5
;  384     0813  5                            ! OK, since it's not a symbol reference, then the
;  385     0814  5                            ! next thing is a psect number.  It may be a byte
;  386     0815  5                            ! or a word.  Print it and record the reference.
;  387     0816  5
;  388     0817  5                            if (.gsd_type eqlu gsd$c_sym) or
;  389     0818  5                               (.gsd_type eqlu gsd$c_epm) or
;  390     0819  6                               (.gsd_type eqlu gsd$c_pro)   then (
;  391     0820  6                                    ensure_field_fit(0,0,8,0,record_dsc);
;  392     0821  7                                    if .fit_ok then (
;  393     0822  7                                         anl$format_line(0,2,anlobj$_objpsect,.scanp[0,0,8,0]);
;  394     0823  7                                         anl$object_psect_ref(.scanp[0,0,8,0]);
;  395     0824  7                                         increment (scanp);
;  396     0825  6                                    );
;  397     0826  6
;  398     0827  6                            ) else (
;  399     0828  6
;  400     0829  6                                    ensure_field_fit(0,0,16,0,record_dsc);
;  401     0830  7                                    if .fit_ok then (
;  402     0831  7                                         anl$format_line(0,2,anlobj$_objpsect,.scanp[0,0,16,0]);
;  403     0832  7                                         anl$object_psect_ref(.scanp[0,0,16,0]);
;  404     0833  7                                         scanp = .scanp + 2;
;  405     0834  6                                    );
;  406     0835  5                    );
;  407     0836  5
;  408     0837  5                    ! Continuing on, these records contain a longword
;  409     0838  5                    ! value.  Print it and check it.
;  410     0839  5
;  411     0840  5                    ensure_field_fit(0,0,32,0,record_dsc);
;  412     0841  6                    if .fit_ok then (
;  413     0842  6                            anl$format_line(0,2,anlobj$_objvalue,.scanp[0,0,32,0]);
;  414     0843  6                            scanp = .scanp + 4;
;  415     0844  5                    );
;  416     0845  5
;  417     0846  5                    ! Whew.  OK, now we have the entry mask, but not if
;  418     0847  5                    ! it's a symbol definition (or reference, of course).
;  419     0848  5                    ! Print it and check it.
;  420     0849  5
;  421     0850  6                    if not (.symbol_spec and .symbol_def) then (
;  422     0851  6                            ensure_field_fit(0,0,16,0,record_dsc);
;  423     0852  7                            if .fit_ok then (
;  424     0853  7                                 anl$format_mask(2,.scanp[0,0,16,0]);
```

```
;  425     0854  7                                        scanp = .scanp + 2;
;  426     0855  6                                    );
;  427     0856  5                                );
;  428     0857  4                            );
;  429     0858  4
;  430     0859  4                            ! OK, now all cases join together.  We have the name of the
;  431     0860  4                            ! symbol or entry point.  Print it and check it.
;  432     0861  4
;  433     0862  4                            ensure_ascic_fit(0,0,8,0,record_dsc,work_dsc);
;  434     0863  5                            if .fit_ok then (
;  435     0864  5                                    anl$format_line(0,2,anlobj$_objsymbol,.work_dsc[len],.work_dsc[ptr]);
;  436     0865  5                                    anl$check_symbol(work_dsc, obj$c_symsiz);
;  437     0866  5                                    scanp = .work_dsc[ptr] + .work_dsc[len];
;  438     0867  4                            );
;  439     0868  4
;  440     0869  4                            ! Well, we're done unless it's a procedure definition.  If so,
;  441     0870  4                            ! we have the argument counts and formal descriptors.
;  442     0871  4
;  443     0872  4                            if .gsd_type eqlu gsd$c_pro or
;  444     0873  4                               .gsd_type eqlu gsd$c_prow or
;  445     0874  5                               .gsd_type eqlu gsd$c_lpro then (
;  446     0875  5
;  447     0876  5                                    local
;  448     0877  5                                            max_args: long;
;  449     0878  5
;  450     0879  5                                    ! First we have two bytes containing the minimum and
;  451     0880  5                                    ! maximum argument counts.  Print them and check.
;  452     0881  5
;  453     0882  5                                    ensure_field_fit(0,0,16,0,record_dsc);
;  454     0883  6                                    if .fit_ok then (
;  455     0884  6                                            anl$format_line(0,2,anlobj$_objproargcount,.scanp[0,0,8,0],.scanp[1,0,8,0]);
;  456     0885  6                                            if .scanp[0,0,8,0] gtru .scanp[1,0,8,0] then
;  457     0886  6                                                    anl$format_error(anlobj$_objprominmax);
;  458     0887  6                                            max_args = .scanp[1,0,8,0];
;  459     0888  6                                            scanp = .scanp + 2;
;  460     0889  6
;  461     0890  6                                            ! Now we have the formal argument descriptors,
;  462     0891  6                                            ! one for each argument.
;  463     0892  6
;  464     0893  7                                            incru i from 1 to .max_args do (
;  465     0894  7                                                    anl$format_line(0,2,anlobj$_objproargnum,.i);
;  466     0895  7                                                    fit_ok = anl$object_argument_dsc(3,scanp,record_dsc);
;  467     0896  6                                            );
;  468     0897  5                                    );
;  469     0898  4                            );
;  470     0899  3                    );
```

```
;  472          0900  3          [gsd$c_idc]:
;  473          0901  3
;  474          0902  3                  ! We have an entity identity consistency check subrecord
;  475          0903  3                  ! (groan).  The first field is flags, although it contains
;  476          0904  3                  ! some other stuff we must ignore.
;  477          0905  3
;  478          0906  4                  (local
;  479          0907  4                          binary: byte;
;  480          0908  4
;  481          0909  4                  ensure_field_fit(idc$w_flags,record_dsc);
;  482          0910  5                  if .fit_ok then (
;  483          0911  5                          anl$format_flags(2,anlobj$_objgsdidcflags,.scanp[idc$w_flags],entity_flags_def);
;  484          0912  5                          anl$check_flags(.scanp[idc$w_flags] and %x'ffffffc1',entity_flags_def);
;  485          0913  4                  );
;  486          0914  4
;  487          0915  4                  ! If this is a binary identity, then the flags contain a
;  488          0916  4                  ! match control value.  Print it and check.
;  489          0917  4
;  490          0918  4                  if (binary = .scanp[idc$v_binident]) then
;  491          0919  4                          case .scanp[idc$v_idmatch] from 0 to 3 of set
;  492          0920  4                          [idc$c_leq]:    anl$format_line(0,2,anlobj$_objgsdidcmatch,uplit byte(%ascic 'LEQ'))
;  493          0921  4                          [idc$c_equal]:  anl$format_line(0,2,anlobj$_objgsdidcmatch,uplit byte(%ascic 'EQUAL'
;  494          0922  4                          [inrange]:      anl$format_error(anlobj$_objbadidcmatch,.scanp[idc$v_binident]);
;  495          0923  4                          tes;
;  496          0924  4
;  497          0925  4                  ! There is also a standard error severity in the flags word.
;  498          0926  4
;  499          0927  4                  anl$format_severity(2,.scanp[idc$v_errsev]);
;  500          0928  4
;  501          0929  4                  ! Next we have the entity name.
;  502          0930  4
;  503          0931  4                  ensure_ascic_fit(idc$b_namlng,record_dsc,work_dsc);
;  504          0932  4                  if .fit_ok then
;  505          0933  4                          anl$format_line(0,2,anlobj$_objgsdidcent,.work_dsc[len],.work_dsc[ptr]);
;  506          0934  4                  scanp = .work_dsc[ptr] + .work_dsc[len];
;  507          0935  4
;  508          0936  4                  ! This next field is the identity value.  It is a counted
;  509          0937  4                  ! string, which can be a longword value.
;  510          0938  4
;  511          0939  4                  ensure_ascic_fit(0,0,8,0,record_dsc,work_dsc);
;  512          0940  4                  if .fit_ok then
;  513          0941  4                          if .binary then
;  514          0942  4                                  anl$format_line(0,2,anlobj$_objgsdidcvalb,.scanp[1,0,32,0])
;  515          0943  4                          else
;  516          0944  4                                  anl$format_line(0,2,anlobj$_objgsdidcvala,.work_dsc[len],.work_dsc[ptr]);
;  517          0945  4                  scanp = .work_dsc[ptr] + .work_dsc[len];
;  518          0946  4
;  519          0947  4                  ! Finally, we have the name of the object.
;  520          0948  4
;  521          0949  4                  ensure_ascic_fit(0,0,8,0,record_dsc,work_dsc);
;  522          0950  4                  if .fit_ok then
;  523          0951  4                          anl$format_line(0,2,anlobj$_objgsdidcobj,.work_dsc[len],.work_dsc[ptr]);
;  524          0952  4
;  525          0953  4                  ! Advance on past this subrecord.
;  526          0954  4
;  527          0955  4                  scanp = .work_dsc[ptr] + .work_dsc[len];
;  528          0956  3                  );
```

```
;  530          0957  3          [gsd$c_env]:
;  531          0958  3
;  532          0959  3                  ! We have an environment specification subrecord.  The
;  533          0960  3                  ! first field is flags, which we print and check.
;  534          0961  3
;  535          0962  4                  (ensure_field_fit(env$w_flags,record_dsc);
;  536          0963  5                  if .fit_ok then (
;  537          0964  5                          increment (highest_def_env);
;  538          0965  5                          anl$format_flags(2,anlobj$_objgsdenvflags,.scanp[env$w_flags],env_flags_def);
;  539          0966  5                          anl$check_flags(.scanp[env$w_flags],env_flags_def);
;  540          0967  4                  );
;  541          0968  4
;  542          0969  4                  ! The next field is the parent environment index.  Print
;  543          0970  4                  ! it with this environment's index, and check it.
;  544          0971  4
;  545          0972  4                  ensure_field_fit(env$w_envindx,record_dsc);
;  546          0973  5                  if .fit_ok then (
;  547          0974  5                          anl$format_line(0,2,anlobj$_objgsdenvpar,.scanp[env$w_envindx],.highest_def_env);
;  548          0975  5                          anl$object_env_ref(.scanp[env$w_envindx]);
;  549          0976  4                  );
;  550          0977  4
;  551          0978  4                  ! The final field is the environment name.  Print it and check.
;  552          0979  4
;  553          0980  4                  ensure_ascic_fit(env$b_namlng,record_dsc,work_dsc);
;  554          0981  5                  if .fit_ok then (
;  555          0982  5                          anl$format_line(0,2,anlobj$_objsymbol,.work_dsc[len],.work_dsc[ptr]);
;  556          0983  5                          anl$check_symbol(work_dsc, obj$c_symsiz);
;  557          0984  4                  );
;  558          0985  4
;  559          0986  4                  ! Finally, advance the scan pointer past this record.
;  560          0987  4
;  561          0988  4                  scanp = .work_dsc[ptr] + .work_dsc[len];
;  562          0989  3                  );
;  563          0990  3
;  564          0991  3          tes;
;  565          0992  3
;  566          0993  2 );
;  567          0994  2
;  568          0995  2 return;
;  569          0996  2
;  570          0997  1 end;


                                                                .TITLE  OBJGSD OBJGSD - Analyze GSD Records
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                43  49  50  5F  56  24  53  50  47  09  00000 P.AAA:  .ASCII  <9>\GPS$V_PIC\                                    ;
                42  49  4C  5F  56  24  53  50  47  09  0000A P.AAB:  .ASCII  <9>\GPS$V_LIB\                                    ;
                4C  56  4F  5F  56  24  53  50  47  09  00014 P.AAC:  .ASCII  <9>\GPS$V_OVL\                                    ;
                4C  45  52  5F  56  24  53  50  47  09  0001E P.AAD:  .ASCII  <9>\GPS$V_REL\                                    ;
                4C  42  47  5F  56  24  53  50  47  09  00028 P.AAE:  .ASCII  <9>\GPS$V_GBL\                                    ;
                52  48  53  5F  56  24  53  50  47  09  00032 P.AAF:  .ASCII  <9>\GPS$V_SHR\                                    ;
                45  58  45  5F  56  24  53  50  47  09  0003C P.AAG:  .ASCII  <9>\GPS$V_EXE\                                    ;
                    44  52  5F  56  24  53  50  47  08  00046 P.AAH:  .ASCII  <8>\GPS$V_RD\                                     ;
                54  52  57  5F  56  24  53  50  47  09  0004F P.AAI:  .ASCII  <9>\GPS$V_WRT\                                    ;
```

```
                    43  45  56  5F  56  24  53  50  47  09  00059 P.AAJ:   .ASCII  <9>\GPS$V_VEC\                          ;
                4B  41  45  57  5F  56  24  59  53  47  0A  00063 P.AAK:   .ASCII  <10>\GSY$V_WEAK\                        ;
                    46  45  44  5F  56  24  59  53  47  09  0006E P.AAL:   .ASCII  <9>\GSY$V_DEF\                         ;
                    49  4E  55  5F  56  24  59  53  47  09  00078 P.AAM:   .ASCII  <9>\GSY$V_UNI\                         ;
                    4C  45  52  5F  56  24  59  53  47  09  00082 P.AAN:   .ASCII  <9>\GSY$V_REL\                         ;
                    46  45  44  5F  56  24  56  4E  45  09  0008C P.AAO:   .ASCII  <9>\ENV$V_DEF\                         ;
            44  45  54  53  45  4E  5F  56  24  56  4E  45  0C  00096 P.AAP:   .ASCII  <12>\ENV$V_NESTED\                   ;
54  4E  45  44  49  4E  49  42  5F  56  24  43  44  49  0E  000A3 P.AAQ:   .ASCII  <14>\IDC$V_BINIDENT\                   ;
                                        51  45  4C  03  000B2 P.AAR:   .ASCII  <3>\LEQ\                                 ;
                                4C  41  55  51  45  05  000B6 P.AAS:   .ASCII  <5>\EQUAL\                               ;

                                                                       .PSECT  $OWN$,NOEXE,2

                                                    FFFFFFFF   00000 HIGHEST_DEF_PSECT:
                                                                       .LONG   -1                                          ;
                                                    FFFFFFFF   00004 HIGHEST_REF_PSECT:
                                                                       .LONG   -1                                          ;
                                                               00008 PSECT_REF_BITS:
                                                                       .BLKB   4
                                                    FFFFFFFF   0000C HIGHEST_DEF_ENV:
                                                                       .LONG   -1                                          ;
                                                    FFFFFFFF   00010 HIGHEST_REF_ENV:
                                                                       .LONG   -1                                          ;
                                                               00014 ENV_REF_BITS:
                                                                       .BLKB   4
                                                    00000009   00018 PSC_FLAGS_DEF:
                                                                       .LONG   9                                           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0001C          .ADDRESS P.AAA, P.AAB, P.AAC, P.AAD, P.AAE, -      ;
                    00000000' 00000000' 00000000' 00000000' 00034                P.AAF, P.AAG, P.AAH, P.AAI, P.AAJ          ;
                                                    00000003   00044 SYM_FLAGS_DEF:
                                                                       .LONG   3                                           ;
                    00000000' 00000000' 00000000' 00000000' 00048          .ADDRESS P.AAK, P.AAL, P.AAM, P.AAN                ;
                                                    00000001   00058 ENV_FLAGS_DEF:
                                                                       .LONG   1                                           ;
                                        00000000' 00000000' 0005C          .ADDRESS P.AAO, P.AAP                           ;
                                                    00000000   00064 ENTITY_FLAGS_DEF:
                                                                       .LONG   0                                           ;
                                                    00000000' 00068          .ADDRESS P.AAQ                                 ;
00000000G 00000000G 00000000G 00000000G 00000000G 00000000G 0006C GSD_SUBRECORD_MSG:
                                                                       .LONG   ANLOBJ$_OBJGSDPSC, ANLOBJ$_OBJGSDSYM, -    ;
00000000G 00000000G 00000000G 00000000G 00000000G 00000000G 00084              ANLOBJ$_OBJGSDEPM, ANLOBJ$_OBJGSDPRO, -    ;
                                                    00000000G  0009C           ANLOBJ$_OBJGSDSYMW, ANLOBJ$_OBJGSDEPMW, -  ;
                                                                               ANLOBJ$_OBJGSDPROW, ANLOBJ$_OBJGSDIDC, -   ;
                                                                               ANLOBJ$_OBJGSDENV, ANLOBJ$_OBJGSDLSY, -    ;
                                                                               ANLOBJ$_OBJGSDLEPM, ANLOBJ$_OBJGSDLPRO, -  ;
                                                                               ANLOBJ$_OBJGSDSPSC                         ;

                                                                   .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$FORMAT_DATA_TYPE
                                                .EXTRN  ANL$FORMAT_ERROR
                                                .EXTRN  ANL$FORMAT_FLAGS
                                                .EXTRN  ANL$FORMAT_HEX, ANL$FORMAT_LINE
                                                .EXTRN  ANL$FORMAT_MASK
                                                .EXTRN  ANL$FORMAT_SEVERITY
                                                .EXTRN  ANL$OBJECT_RECORD_LINE
                                                .EXTRN  ANL$REPORT_LINE
                                                .EXTRN  LIB$FREE_VM, LIB$GET_VM

                                                .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000  .ENTRY  ANL$OBJECT_GSD, Save R2,R3,R4,R5,R6,R7,R8,- ; 0573
                                                        R9,R10,R11
                          5B 00000000G  8F  D0 00002    MOVL   #ANLOBJ$_FIELDFIT, R11
                          5E            0C  C2 00009    SUBL2  #12, SP
                          56      08    AC  D0 0000C    MOVL   THE_RECORD, R6                          ; 0576
                                        56  DD 00010    PUSHL  R6                                      ; 0647
                                  04    AC  DD 00012    PUSHL  RECORD_NUMBER
                             00000000G  8F  DD 00015    PUSHL  #ANLOBJ$_OBJGSDREC
                   0000G  CF            03  FB 0001B    CALLS  #3, ANL$OBJECT_RECORD_LINE
                                        59  D4 00020    CLRL   SUBRECORD_NUMBER                        ; 0654
                6E      04  A6          01  C1 00022    ADDL3  #1, 4(R6), SCANP                        ; 0655
                          53            01  90 00027    MOVB   #1, FIT_OK                              ; 0656
                          52            6E  D0 0002A 1$: MOVL   SCANP, R2                               ; 0657
                          54            66  3C 0002D    MOVZWL (R6), R4
```

```
             54        04   A6   C0 00030          ADDL2   4(R6), R4                        ;
             54             52   D1 00034          CMPL    R2, R4                           ;
                            01   1F 00037          BLSSU   2$                               ;
                                 04 00039          RET                                      ;
             01             53   E8 0003A 2$:      BLBS    FIT_OK, 3$                       ;
                                 04 0003D          RET                                      ;
                            59   D6 0003E 3$:      INCL    SUBRECORD_NUMBER                 ; 0663
                            7E   D4 00040          CLRL    -(SP)                            ; 0664
        0000G  CF           01   FB 00042          CALLS   #1, ANL$REPORT_LINE              ;
               5A           62   90 00047          MOVB    (R2), GSD_TYPE                   ; 0666
               55           5A   9A 0004A          MOVZBL  GSD_TYPE, R5                     ; 0667
               0C           55   91 0004D          CMPB    R5, #12                          ;
                            12   1A 00050          BGTRU   4$                               ;
                            59   DD 00052          PUSHL   SUBRECORD_NUMBER                 ; 0668
                    0000'CF45    DD 00054          PUSHL   GSD_SUBRECORD_MSG[R5]            ;
                            01   DD 00059          PUSHL   #1                               ;
                            02   DD 0005B          PUSHL   #2                               ;
        0000G  CF           04   FB 0005D          CALLS   #4, ANL$FORMAT_LINE              ;
                            2A   11 00062          BRB     5$                               ;
                            55   DD 00064 4$:      PUSHL   R5                               ; 0670
                 00000000G  8F   DD 00066          PUSHL   #ANLOBJ$_OBJGSDBADSUBTYP         ;
        0000G  CF           02   FB 0006C          CALLS   #2, ANL$FORMAT_ERROR             ;
          50     04   A6    52   C3 00071          SUBL3   R2, 4(R6), R0                    ; 0671
               51           66   3C 00076          MOVZWL  (R6), R1                         ;
     04   AE        50      51   C1 00079          ADDL3   R1, R0, WORK_DSC                 ;
          08   AE      04   A6   D0 0007E          MOVL    4(R6), WORK_DSC+4                ;
                       04   AE   9F 00083          PUSHAB  WORK_DSC                         ; 0672
                            02   DD 00086          PUSHL   #2                               ;
        0000G  CF           02   FB 00088          CALLS   #2, ANL$FORMAT_HEX               ;
                                 04 0008D          RET                                      ; 0669
          0C        00      55   8F 0008E 5$:      CASEB   R5, #0, #12                      ; 0678
 0165     0165      0165    001A    00092 6$:      .WORD   7$-6$,-                          ;
 0410     0165      0165    0165    0009A                  23$-6$,-                         ;
 0165     0165      0165    05DC    000A2                  23$-6$,-                         ;
                            001A    000AA                  23$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           53$-6$,-                         ;
                                                           72$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           23$-6$,-                         ;
                                                           7$-6$                            ;
             7E             53   E9 000AC 7$:      BLBC    FIT_OK, 11$                      ; 0686
             50        02   A2   9E 000AF          MOVAB   2(R2), R0                        ;
             54             50   D1 000B3          CMPL    R0, R4                           ;
                            09   1B 000B6          BLEQU   8$                               ;
                            5B   DD 000B8          PUSHL   R11                              ;
        0000G  CF           01   FB 000BA          CALLS   #1, ANL$FORMAT_ERROR             ;
                            53   94 000BF          CLRB    FIT_OK                           ;
             7E             53   E9 000C1 8$:      BLBC    FIT_OK, 12$                      ; 0687
                    0000'   CF   D6 000C4          INCL    HIGHEST_DEF_PSECT                ; 0688
                    0000'   CF   DD 000C8          PUSHL   HIGHEST_DEF_PSECT                ; 0690
          7E        01      01   A2 78 000CC       ASHL    1(R2), #1, -(SP)                 ; 0689
                 00000000G  8F   DD 000D1          PUSHL   #ANLOBJ$_OBJGSDPSCALIGN          ;
                            02   DD 000D7          PUSHL   #2                               ;
```

```
                                     7E  D4 000D9         CLRL    -(SP)                                 ;
                 0000G   CF          05  FB 000DB         CALLS   #5, ANL$FORMAT_LINE                   ;
                         09     01   A2  91 000E0         CMPB    1(R2), #9                             ; 0691
                                     0D  1B 000E4         BLEQU   9$                                    ;
                                     09  DD 000E6         PUSHL   #9                                    ; 0692
                         00000000G   8F  DD 000E8         PUSHL   #ANLOBJ$_OBJGSDBADALIGN               ;
                 0000G   CF          02  FB 000EE         CALLS   #2, ANL$FORMAT_ERROR                  ;
                         4C          53  E9 000F3   9$:   BLBC    FIT_OK, 12$                           ; 0697
                         50     04   A2  9E 000F6         MOVAB   4(R2), R0                             ;
                         54          50  D1 000FA         CMPL    R0, R4                                ;
                                     09  1B 000FD         BLEQU   10$                                   ;
                                     5B  DD 000FF         PUSHL   R11                                   ;
                 0000G   CF          01  FB 00101         CALLS   #1, ANL$FORMAT_ERROR                  ;
                                     53  94 00106         CLRB    FIT_OK                                ;
                         76          53  E9 00108  10$:   BLBC    FIT_OK, 14$                           ; 0698
                             0000'   CF  9F 0010B         PUSHAB  PSC_FLAGS_DEF                         ; 0699
                         7E     02   A2  3C 0010F         MOVZWL  2(R2), -(SP)                          ;
                         00000000G   8F  DD 00113         PUSHL   #ANLOBJ$_OBJGSDPSCFLAGS               ;
                                     02  DD 00119         PUSHL   #2                                    ;
                 0000G   CF          04  FB 0011B         CALLS   #4, ANL$FORMAT_FLAGS                  ;
                             0000'   CF  9F 00120         PUSHAB  PSC_FLAGS_DEF                         ; 0700
                         7E     02   A2  3C 00124         MOVZWL  2(R2), -(SP)                          ;
                 0000G   CF          02  FB 00128         CALLS   #2, ANL$CHECK_FLAGS                   ;
                         51          53  E9 0012D  11$:   BLBC    FIT_OK, 14$                           ; 0705
                         50     08   A2  9E 00130         MOVAB   8(R2), R0                             ;
                         54          50  D1 00134         CMPL    R0, R4                                ;
                                     09  1B 00137         BLEQU   12$                                   ;
                                     5B  DD 00139         PUSHL   R11                                   ;
                 0000G   CF          01  FB 0013B         CALLS   #1, ANL$FORMAT_ERROR                  ;
                                     53  94 00140         CLRB    FIT_OK                                ;
                         3C          53  E9 00142  12$:   BLBC    FIT_OK, 14$                           ; 0706
                                04   A2  DD 00145         PUSHL   4(R2)                                 ; 0707
                         00000000G   8F  DD 00148         PUSHL   #ANLOBJ$_OBJGSDPSCALLOC               ;
                                     02  DD 0014E         PUSHL   #2                                    ;
                                     7E  D4 00150         CLRL    -(SP)                                 ;
                 0000G   CF          04  FB 00152         CALLS   #4, ANL$FORMAT_LINE                   ;
        3FFFFFFF   8F           04   A2  D1 00157         CMPL    4(R2), #1073741823                    ; 0708
                                     0B  1B 0015F         BLEQU   13$                                   ;
                         00000000G   8F  DD 00161         PUSHL   #ANLOBJ$_OBJPOSPACE                   ; 0709
                 0000G   CF          01  FB 00167         CALLS   #1, ANL$FORMAT_ERROR                  ;
            10       02   A2         03  E0 0016C  13$:   BBS     #3, 2(R2), 14$                        ; 0710
                                04   A2  D5 00171         TSTL    4(R2)                                 ;
                                     0B  13 00174         BEQL    14$                                   ;
                         00000000G   8F  DD 00176         PUSHL   #ANLOBJ$_OBJPSCABSLEN                 ; 0711
                 0000G   CF          01  FB 0017C         CALLS   #1, ANL$FORMAT_ERROR                  ;
                         0C          55  91 00181  14$:   CMPB    R5, #12                               ; 0720
                                     2F  12 00184         BNEQ    17$                                   ;
                         27          53  E9 00186         BLBC    FIT_OK, 16$                           ; 0721
                         50     0C   A2  9E 00189         MOVAB   12(R2), R0                            ;
                         54          50  D1 0018D         CMPL    R0, R4                                ;
                                     09  1B 00190         BLEQU   15$                                   ;
                                     5B  DD 00192         PUSHL   R11                                   ;
                 0000G   CF          01  FB 00194         CALLS   #1, ANL$FORMAT_ERROR                  ;
                                     53  94 00199         CLRB    FIT_OK                                ;
                         12          53  E9 0019B  15$:   BLBC    FIT_OK, 16$                           ; 0722
                                08   A2  DD 0019E         PUSHL   8(R2)                                 ; 0723
                         00000000G   8F  DD 001A1         PUSHL   #ANLOBJ$_OBJGSDPSCBASE                ;
```

```
                                02  DD 001A7          PUSHL   #2                                    ;
                                7E  D4 001A9          CLRL    -(SP)                                 ;
                 0000G  CF      04  FB 001AB          CALLS   #4, ANL$FORMAT_LINE                   ;
                        6E      0C  C0 001B0  16$:    ADDL2   #12, SCANP                            ; 0724
                                03  11 001B3          BRB     18$                                   ; 0720
                        6E      08  C0 001B5  17$:    ADDL2   #8, SCANP                             ; 0726
                        1F      53  E9 001B8  18$:    BLBC    FIT_OK, 20$                           ; 0730
          50            6E      01  C1 001BB          ADDL3   #1, SCANP, R0                         ;
                        54      50  D1 001BF          CMPL    R0, R4                                ;
                                09  1B 001C2          BLEQU   19$                                   ;
                                5B  DD 001C4          PUSHL   R11                                   ;
                 0000G  CF      01  FB 001C6          CALLS   #1, ANL$FORMAT_ERROR                  ;
                                53  94 001CB          CLRB    FIT_OK                                ;
                        0A      53  E9 001CD  19$:    BLBC    FIT_OK, 20$                           ;
                    04  AE   00 BE  9A 001D0          MOVZBL  @SCANP, WORK_DSC                      ;
       08   AE          6E      01  C1 001D5          ADDL3   #1, SCANP, WORK_DSC+4                 ;
                        03      53  E8 001DA  20$:    BLBS    FIT_OK, 21$                           ;
                              0566  31 001DD          BRW     78$                                   ;
                        50   04 AE  3C 001E0  21$:    MOVZWL  WORK_DSC, R0                          ;
                        50      08  C6 001E4          DIVL2   #8, R0                                ;
                        50      6E  C0 001E7          ADDL2   SCANP, R0                             ;
                                50  D6 001EA          INCL    R0                                    ;
                        54      50  D1 001EC          CMPL    R0, R4                                ;
                                03  1A 001EF          BGTRU   22$                                   ;
                              052F  31 001F1          BRW     77$                                   ;
                              0523  31 001F4  22$:    BRW     76$                                   ;
                        35      53  E9 001F7  23$:    BLBC    FIT_OK, 25$                           ; 0762
                        50   02 A2  9E 001FA          MOVAB   2(R2), R0                             ;
                        54      50  D1 001FE          CMPL    R0, R4                                ;
                                09  1B 00201          BLEQU   24$                                   ;
                                5B  DD 00203          PUSHL   R11                                   ;
                 0000G  CF      01  FB 00205          CALLS   #1, ANL$FORMAT_ERROR                  ;
                                53  94 0020A          CLRB    FIT_OK                                ;
                        78      53  E9 0020C  24$:    BLBC    FIT_OK, 29$                           ; 0763
                        7E   01 A2  9A 0020F          MOVZBL  1(R2), -(SP)                          ; 0764
                                02  DD 00213          PUSHL   #2                                    ;
                 0000G  CF      02  FB 00215          CALLS   #2, ANL$FORMAT_DATA_TYPE              ;
                        6A      53  E9 0021A          BLBC    FIT_OK, 29$                           ; 0773
                        50   04 A2  9E 0021D          MOVAB   4(R2), R0                             ;
                        54      50  D1 00221          CMPL    R0, R4                                ;
                                09  1B 00224          BLEQU   25$                                   ;
                                5B  DD 00226          PUSHL   R11                                   ;
                 0000G  CF      01  FB 00228          CALLS   #1, ANL$FORMAT_ERROR                  ;
                                53  94 0022D          CLRB    FIT_OK                                ;
                        55      53  E9 0022F  25$:    BLBC    FIT_OK, 29$                           ; 0774
                          0000' CF  9F 00232          PUSHAB  SYM_FLAGS_DEF                         ; 0775
                        7E   02 A2  3C 00236          MOVZWL  2(R2), -(SP)                          ;
                      00000000G 8F  DD 0023A          PUSHL   #ANLOBJ$_OBJSYMFLAGS                  ;
                                02  DD 00240          PUSHL   #2                                    ;
                 0000G  CF      04  FB 00242          CALLS   #4, ANL$FORMAT_FLAGS                  ;
                          0000' CF  9F 00247          PUSHAB  SYM_FLAGS_DEF                         ; 0776
                        50   02 A2  3C 0024B          MOVZWL  2(R2), R0                             ;
          7E            50 FFFF0800 8F  CB 0024F      BICL3   #-63488, R0, -(SP)                    ;
                 0000G  CF      02  FB 00257          CALLS   #2, ANL$CHECK_FLAGS                   ;
                                50  D4 0025C          CLRL    R0                                    ; 0781
                        01      55  91 0025E          CMPB    R5, #1                                ;
                                02  12 00261          BNEQ    26$                                   ;
```

```
                                    50  D6 00263        INCL    R0                              ;
                                    51  D4 00265 26$:   CLRL    R1                              ; 0782
                        04          55  91 00267        CMPB    R5, #4                          ;
                                    02  12 0026A        BNEQ    27$                             ;
                                    51  D6 0026C        INCL    R1                              ;
                        51          50  C8 0026E 27$:   BISL2   R0, R1                          ;
                                    50  D4 00271        CLRL    R0                              ; 0783
                        09          55  91 00273        CMPB    R5, #9                          ;
                                    02  12 00276        BNEQ    28$                             ;
                                    50  D6 00278        INCL    R0                              ;
              58        50          51  89 0027A 28$:   BISB3   R1, R0, SYMBOL_SPEC             ;
   50     02  A2        01          01  EF 0027E        EXTZV   #1, #1, 2(R2), R0               ; 0784
                        57          50  90 00284        MOVB    R0, SYMBOL_DEF                  ;
                        6E          04  C0 00287 29$:   ADDL2   #4, SCANP                       ; 0792
                        09          55  91 0028A        CMPB    R5, #9                          ; 0797
                                    0A  13 0028D        BEQL    30$                             ;
                        0A          55  91 0028F        CMPB    R5, #10                         ; 0798
                                    05  13 00292        BEQL    30$                             ;
                        0B          55  91 00294        CMPB    R5, #11                         ; 0799
                                    37  12 00297        BNEQ    32$                             ;
                        34          53  E9 00299 30$:   BLBC    FIT_OK, 32$                     ; 0800
              50        6E          02  C1 0029C        ADDL3   #2, SCANP, R0                   ;
                        54          50  D1 002A0        CMPL    R0, R4                          ;
                                    09  1B 002A3        BLEQU   31$                             ;
                                    5B  DD 002A5        PUSHL   R11                             ;
                 0000G  CF          01  FB 002A7        CALLS   #1, ANL$FORMAT_ERROR            ;
                                    53  94 002AC        CLRB    FIT_OK                          ;
                        1F          53  E9 002AE 31$:   BLBC    FIT_OK, 32$                     ; 0801
                        7E     00   BE  3C 002B1        MOVZWL  @SCANP, -(SP)                   ; 0802
                        00000000G   8F  DD 002B5        PUSHL   #ANLOBJ$_OBJENV                 ;
                                    02  DD 002BB        PUSHL   #2                              ;
                                    7E  D4 002BD        CLRL    -(SP)                           ;
                 0000G  CF          04  FB 002BF        CALLS   #4, ANL$FORMAT_LINE             ;
                        7E     00   BE  3C 002C4        MOVZWL  @SCANP, -(SP)                   ; 0803
                 0000V  CF          01  FB 002C8        CALLS   #1, ANL$OBJECT_ENV_REF          ;
                        6E          02  C0 002CD        ADDL2   #2, SCANP                       ; 0804
                        06          58  E9 002D0 32$:   BLBC    SYMBOL_SPEC, 33$                ; 0811
                        03          57  E8 002D3        BLBS    SYMBOL_DEF, 33$                 ;
                                  00D7  31 002D6        BRW     43$                             ;
                        01          55  91 002D9 33$:   CMPB    R5, #1                          ; 0817
                                    0A  13 002DC        BEQL    34$                             ;
                        02          55  91 002DE        CMPB    R5, #2                          ; 0818
                                    05  13 002E1        BEQL    34$                             ;
                        03          55  91 002E3        CMPB    R5, #3                          ; 0819
                                    38  12 002E6        BNEQ    36$                             ;
                        12          53  E9 002E8 34$:   BLBC    FIT_OK, 35$                     ; 0820
              50        6E          01  C1 002EB        ADDL3   #1, SCANP, R0                   ;
                        54          50  D1 002EF        CMPL    R0, R4                          ;
                                    09  1B 002F2        BLEQU   35$                             ;
                                    5B  DD 002F4        PUSHL   R11                             ;
                 0000G  CF          01  FB 002F6        CALLS   #1, ANL$FORMAT_ERROR            ;
                                    53  94 002FB        CLRB    FIT_OK                          ;
                        6C          53  E9 002FD 35$:   BLBC    FIT_OK, 39$                     ; 0821
                        7E     00   BE  9A 00300        MOVZBL  @SCANP, -(SP)                   ; 0822
                        00000000G   8F  DD 00304        PUSHL   #ANLOBJ$_OBJPSECT               ;
                                    02  DD 0030A        PUSHL   #2                              ;
                                    7E  D4 0030C        CLRL    -(SP)                           ;
```

```
                 0000G  CF              04  FB 0030E             CALLS   #4, ANL$FORMAT_LINE
                        7E        00    BE  9A 00313             MOVZBL  @SCANP, -(SP)                   ;0823
                 0000V  CF              01  FB 00317             CALLS   #1, ANL$OBJECT_PSECT_REF
                        6E              6E  D6 0031C             INCL    SCANP                           ;0824
                                        37  11 0031E             BRB     38$                             ;0817
                        61              53  E9 00320 36$:       BLBC    FIT_OK, 40$                     ;0829
           50           6E              02  C1 00323             ADDL3   #2, SCANP, R0
                        54              50  D1 00327             CMPL    R0, R4
                                        09  1B 0032A             BLEQU   37$
                                        5B  DD 0032C             PUSHL   R11
                 0000G  CF              01  FB 0032E             CALLS   #1, ANL$FORMAT_ERROR
                                        53  94 00333             CLRB    FIT_OK
                        4C              53  E9 00335 37$:       BLBC    FIT_OK, 40$                     ;0830
                        7E        00    BE  3C 00338             MOVZWL  @SCANP, -(SP)                   ;0831
                            00000000G   8F  DD 0033C             PUSHL   #ANLOBJ$_OBJPSECT
                                        02  DD 00342             PUSHL   #2
                                        7E  D4 00344             CLRL    -(SP)
                 0000G  CF              04  FB 00346             CALLS   #4, ANL$FORMAT_LINE
                        7E        00    BE  3C 0034B             MOVZWL  @SCANP, -(SP)                   ;0832
                 0000V  CF              01  FB 0034F             CALLS   #1, ANL$OBJECT_PSECT_REF
                        6E              02  C0 00354             ADDL2   #2, SCANP                       ;0833
                        2A              53  E9 00357 38$:       BLBC    FIT_OK, 40$                     ;0840
           50           6E              04  C1 0035A             ADDL3   #4, SCANP, R0
                        54              50  D1 0035E             CMPL    R0, R4
                                        09  1B 00361             BLEQU   39$
                                        5B  DD 00363             PUSHL   R11
                 0000G  CF              01  FB 00365             CALLS   #1, ANL$FORMAT_ERROR
                                        53  94 0036A             CLRB    FIT_OK
                        15              53  E9 0036C 39$:       BLBC    FIT_OK, 40$                     ;0841
                              00        BE  DD 0036F             PUSHL   @SCANP                          ;0842
                            00000000G   8F  DD 00372             PUSHL   #ANLOBJ$_OBJVALUE
                                        02  DD 00378             PUSHL   #2
                                        7E  D4 0037A             CLRL    -(SP)
                 0000G  CF              04  FB 0037C             CALLS   #4, ANL$FORMAT_LINE
                        6E              04  C0 00381             ADDL2   #4, SCANP                       ;0843
                        03              58  E9 00384 40$:       BLBC    SYMBOL_SPEC, 41$                ;0850
                        26              57  E8 00387             BLBS    SYMBOL_DEF, 43$
                        62              53  E9 0038A 41$:       BLBC    FIT_OK, 45$                     ;0851
           50           6E              02  C1 0038D             ADDL3   #2, SCANP, R0
                        54              50  D1 00391             CMPL    R0, R4
                                        09  1B 00394             BLEQU   42$
                                        5B  DD 00396             PUSHL   R11
                 0000G  CF              01  FB 00398             CALLS   #1, ANL$FORMAT_ERROR
                                        53  94 0039D             CLRB    FIT_OK
                        79              53  E9 0039F 42$:       BLBC    FIT_OK, 46$                     ;0852
                        7E        00    BE  3C 003A2             MOVZWL  @SCANP, -(SP)                   ;0853
                                        02  DD 003A6             PUSHL   #2
                 0000G  CF              02  FB 003A8             CALLS   #2, ANL$FORMAT_MASK
                        6E              02  C0 003AD             ADDL2   #2, SCANP                       ;0854
                        68              53  E9 003B0 43$:       BLBC    FIT_OK, 46$                     ;0862
           50           6E              01  C1 003B3             ADDL3   #1, SCANP, R0
                        54              50  D1 003B7             CMPL    R0, R4
                                        09  1B 003BA             BLEQU   44$
                                        5B  DD 003BC             PUSHL   R11
                 0000G  CF              01  FB 003BE             CALLS   #1, ANL$FORMAT_ERROR
                                        53  94 003C3             CLRB    FIT_OK
                        53              53  E9 003C5 44$:       BLBC    FIT_OK, 46$
```

```
                     04   AE        00   BE   9A 003C8          MOVZBL  @SCANP, WORK_DSC                                    ;
                08   AE             6E   01   C1 003CD          ADDL3   #1, SCANP, WORK_DSC+4                              ;
                                    46   53   E9 003D2          BLBC    FIT_OK, 46$                                        ;
                                    50        04   AE 3C 003D5  MOVZWL  WORK_DSC, R0                                       ;
                                    50        08   C6 003D9     DIVL2   #8, R0                                             ;
                                    50             6E   C0 003DC  ADDL2   SCANP, R0                                        ;
                                                   50   D6 003DF  INCL    R0                                             ;
                                    54             50   D1 003E1  CMPL    R0, R4                                         ;
                                                   09   1B 003E4  BLEQU   45$                                            ;
                                                   5B   DD 003E6  PUSHL   R11                                            ;
                              0000G CF             01   FB 003E8  CALLS   #1, ANL$FORMAT_ERROR                           ;
                                                   53   94 003ED  CLRB    FIT_OK                                         ;
                                    29             53   E9 003EF  45$:    BLBC    FIT_OK, 46$                            ; 0863
                                         08   AE   DD 003F2          PUSHL   WORK_DSC+4                                     ; 0864
                                    7E   08   AE   3C 003F5          MOVZWL  WORK_DSC, -(SP)                                ;
                                    00000000G  8F   DD 003F9          PUSHL   #ANLOBJ$_OBJSYMBOL                             ;
                                               02   DD 003FF          PUSHL   #2                                             ;
                                               7E   D4 00401          CLRL    -(SP)                                          ;
                              0000G CF             05   FB 00403     CALLS   #5, ANL$FORMAT_LINE                            ;
                                                   1F   DD 00408     PUSHL   #31                                            ; 0865
                                         08   AE   9F   0040A       PUSHAB  WORK_DSC                                       ;
                              0000G CF             02   FB 0040D     CALLS   #2, ANL$CHECK_SYMBOL                           ;
                                    50        04   AE 3C 00412      MOVZWL  WORK_DSC, R0                                   ; 0866
                                    6E        08 BE40   9E 00416     MOVAB   @WORK_DSC+4[R0], SCANP                         ;
                                    03             55   91 0041B  46$:    CMPB    R5, #3                                 ; 0872
                                                   0A   13 0041E     BEQL    47$                                            ;
                                    06             55   91 00420     CMPB    R5, #6                                 ; 0873
                                                   05   13 00423     BEQL    47$                                            ;
                                    0B             55   91 00425     CMPB    R5, #11                                ; 0874
                                                   75   12 00428     BNEQ    52$                                            ;
                                    72             53   E9 0042A  47$:    BLBC    FIT_OK, 52$                            ; 0882
                          50        6E             02   C1 0042D     ADDL3   #2, SCANP, R0                                  ;
                                    54             50   D1 00431     CMPL    R0, R4                                         ;
                                                   09   1B 00434     BLEQU   48$                                            ;
                                                   5B   DD 00436     PUSHL   R11                                            ;
                              0000G CF             01   FB 00438     CALLS   #1, ANL$FORMAT_ERROR                           ;
                                                   53   94 0043D     CLRB    FIT_OK                                         ;
                                    5D             53   E9 0043F  48$:    BLBC    FIT_OK, 52$                            ; 0883
                                    54             6E   D0 00442     MOVL    SCANP, R4                                      ; 0884
                                    7E        01   A4   9A 00445     MOVZBL  1(R4), -(SP)                                   ;
                                    7E             64   9A 00449     MOVZBL  (R4), -(SP)                                    ;
                                         00000000G  8F   DD 0044C     PUSHL   #ANLOBJ$_OBJPROARGCOUNT                        ;
                                                    02   DD 00452     PUSHL   #2                                             ;
                                                    7E   D4 00454     CLRL    -(SP)                                          ;
                              0000G CF              05   FB 00456     CALLS   #5, ANL$FORMAT_LINE                            ;
                                01  A4              64   91 0045B     CMPB    (R4), 1(R4)                                    ; 0885
                                                    0B   1B 0045F     BLEQU   49$                                            ;
                                         00000000G  8F   DD 00461     PUSHL   #ANLOBJ$_OBJPROMINMAX                          ; 0886
                              0000G CF              01   FB 00467     CALLS   #1, ANL$FORMAT_ERROR                           ;
                                    55        01   A4   9A 0046C  49$:    MOVZBL  1(R4), MAX_ARGS                        ; 0887
                                    6E             02   C0 00470     ADDL2   #2, SCANP                                      ; 0888
                                    54             01   D0 00473     MOVL    #1, I                                  ; 0893
                                                   22   11 00476     BRB     51$                                            ;
                                                   54   DD 00478  50$:    PUSHL   I                                      ; 0894
                                         00000000G  8F   DD 0047A     PUSHL   #ANLOBJ$_OBJPROARGNUM                          ;
                                                    02   DD 00480     PUSHL   #2                                             ;
                                                    7E   D4 00482     CLRL    -(SP)                                          ;
```

```
                        0000G  CF           04  FB 00484       CALLS    #4, ANL$FORMAT_LINE                     ;
                                            56  DD 00489       PUSHL    R6                                      ; 0895
                                     04     AE  9F 0048B       PUSHAB   SCANP                                   ;
                                            03  DD 0048E       PUSHL    #3                                      ;
                        0000V  CF           03  FB 00490       CALLS    #3, ANL$OBJECT_ARGUMENT_DSC             ;
                               53           50  90 00495       MOVB     R0, FIT_OK                              ;
                                            54  D6 00498       INCL     I                                       ; 0893
                               55           54  D1 0049A 51$:  CMPL     I, MAX_ARGS                             ;
                                            D9  1B 0049D       BLEQU    50$                                     ;
                                          FB88  31 0049F 52$:  BRW      1$                                      ; 0678
                               3B           53  E9 004A2 53$:  BLBC     FIT_OK, 55$                             ; 0909
                               50    03     A2  9E 004A5       MOVAB    3(R2), R0                               ;
                               54           50  D1 004A9       CMPL     R0, R4                                  ;
                                            09  1B 004AC       BLEQU    54$                                     ;
                                            5B  DD 004AE       PUSHL    R11                                     ;
                        0000G  CF           01  FB 004B0       CALLS    #1, ANL$FORMAT_ERROR                    ;
                                            53  94 004B5       CLRB     FIT_OK                                  ;
                               26           53  E9 004B7 54$:  BLBC     FIT_OK, 55$                             ; 0910
                                  0000'     CF  9F 004BA       PUSHAB   ENTITY_FLAGS_DEF                        ; 0911
                               7E    01     A2  3C 004BE       MOVZWL   1(R2), -(SP)                            ;
                              00000000G     8F  DD 004C2       PUSHL    #ANLOBJ$_OBJGSDIDCFLAGS                 ;
                                            02  DD 004C8       PUSHL    #2                                      ;
                        0000G  CF           04  FB 004CA       CALLS    #4, ANL$FORMAT_FLAGS                    ;
                                  0000'     CF  9F 004CF       PUSHAB   ENTITY_FLAGS_DEF                        ; 0912
                               50    01     A2  3C 004D3       MOVZWL   1(R2), R0                               ;
                       7E      50           3E  CB 004D7       BICL3    #62, R0, -(SP)                          ;
                        0000G  CF           02  FB 004DB       CALLS    #2, ANL$CHECK_FLAGS                     ;
   50        01        A2      01           00  EF 004E0 55$:  EXTZV    #0, #1, 1(R2), R0                       ; 0918
                               55           50  90 004E6       MOVB     R0, BINARY                              ;
                               3E           50  E9 004E9       BLBC     R0, 61$                                 ;
   57        01        A2      02           01  EF 004EC       EXTZV    #1, #2, 1(R2), R7                       ; 0919
                       03      00           57  CF 004F2       CASEL    R7, #0, #3                              ;
 0023                0023    000E         0008     004F6 56$:  .WORD    57$-56$,-                               ;
                                                                        58$-56$,-                               ;
                                                                        60$-56$,-                               ;
                                                                        60$-56$                                 ;
                                  0000'     CF  9F 004FE 57$:  PUSHAB   P.AAR                                   ; 0920
                                            04  11 00502       BRB      59$                                     ;
                                  0000'     CF  9F 00504 58$:  PUSHAB   P.AAS                                   ; 0921
                              00000000G     8F  DD 00508 59$:  PUSHL    #ANLOBJ$_OBJGSDIDCMATCH                 ;
                                            02  DD 0050E       PUSHL    #2                                      ;
                                            7E  D4 00510       CLRL     -(SP)                                   ;
                        0000G  CF           04  FB 00512       CALLS    #4, ANL$FORMAT_LINE                     ;
                                            11  11 00517       BRB      61$                                     ;
   7E        01        A2      01           00  EF 00519 60$:  EXTZV    #0, #1, 1(R2), -(SP)                    ; 0922
                              00000000G     8F  DD 0051F       PUSHL    #ANLOBJ$_OBJBADIDCMATCH                 ;
                        0000G  CF           02  FB 00525       CALLS    #2, ANL$FORMAT_ERROR                    ;
   7E        01        A2      03           03  EF 0052A 61$:  EXTZV    #3, #3, 1(R2), -(SP)                    ; 0927
                                            02  DD 00530       PUSHL    #2                                      ;
                        0000G  CF           02  FB 00532       CALLS    #2, ANL$FORMAT_SEVERITY                 ;
                               55           53  E9 00537       BLBC     FIT_OK, 64$                             ; 0931
                               50    04     A2  9E 0053A       MOVAB    4(R2), R0                               ;
                               54           50  D1 0053E       CMPL     R0, R4                                  ;
                                            09  1B 00541       BLEQU    62$                                     ;
                                            5B  DD 00543       PUSHL    R11                                     ;
                        0000G  CF           01  FB 00545       CALLS    #1, ANL$FORMAT_ERROR                    ;
                                            53  94 0054A       CLRB     FIT_OK                                  ;
```

```
                                40              53    E9  0054C 62$:     BLBC    FIT_OK, 64$                             ;
                             04 AE          03  A2    9A  0054F         MOVZBL  3(R2), WORK_DSC                         ;
                             08 AE          04  A2    9E  00554         MOVAB   4(R2), WORK_DSC+4                       ;
                                33              53    E9  00559         BLBC    FIT_OK, 64$                             ;
                                50          04  AE    3C  0055C         MOVZWL  WORK_DSC, R0                            ;
                                50              08    C6  00560         DIVL2   #8, R0                                  ;
                                50          01 A042   9E  00563         MOVAB   1(R0)[R2], R0                           ;
                                54              50    D1  00568         CMPL    R0, R4                                  ;
                                                09    1B  0056B         BLEQU   63$                                     ;
                                                5B    DD  0056D         PUSHL   R11                                     ;
                         0000G  CF                    01  FB  0056F         CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                      53  94  00574         CLRB    FIT_OK                                  ;
                                16                    53  E9  00576 63$:     BLBC    FIT_OK, 64$                             ; 0932
                                                08    AE  DD  00579         PUSHL   WORK_DSC+4                              ; 0933
                                7E              08    AE  3C  0057C         MOVZWL  WORK_DSC, -(SP)                         ;
                                    00000000G   8F        DD  00580         PUSHL   #ANLOBJ$_OBJGSDIDCENT                   ;
                                                02        DD  00586         PUSHL   #2                                      ;
                                                7E        D4  00588         CLRL    -(SP)                                   ;
                         0000G  CF                        05  FB  0058A         CALLS   #5, ANL$FORMAT_LINE                     ;
                                50          04  AE        3C  0058F 64$:     MOVZWL  WORK_DSC, R0                            ; 0934
                                6E          08 BE40       9E  00593         MOVAB   @WORK_DSC+4[R0], SCANP                  ;
                                6F                        53  E9  00598         BLBC    FIT_OK, 68$                             ; 0939
                     50                     6E            01  C1  0059B         ADDL3   #1, SCANP, R0                           ;
                                54                        50  D1  0059F         CMPL    R0, R4                                  ;
                                                          09  1B  005A2         BLEQU   65$                                     ;
                                                          5B  DD  005A4         PUSHL   R11                                     ;
                         0000G  CF                        01  FB  005A6         CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                          53  94  005AB         CLRB    FIT_OK                                  ;
                                5A                        53  E9  005AD 65$:     BLBC    FIT_OK, 68$                             ;
                             04 AE              00        BE  9A  005B0         MOVZBL  @SCANP, WORK_DSC                        ;
                08 AE                           6E        01  C1  005B5         ADDL3   #1, SCANP, WORK_DSC+4                   ;
                                                4D        53  E9  005BA         BLBC    FIT_OK, 68$                             ;
                                50          04  AE        3C  005BD         MOVZWL  WORK_DSC, R0                            ;
                                50              08        C6  005C1         DIVL2   #8, R0                                  ;
                                50              6E        C0  005C4         ADDL2   SCANP, R0                               ;
                                                50        D6  005C7         INCL    R0                                      ;
                                54              50        D1  005C9         CMPL    R0, R4                                  ;
                                                09        1B  005CC         BLEQU   66$                                     ;
                                                5B        DD  005CE         PUSHL   R11                                     ;
                         0000G  CF                        01  FB  005D0         CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                          53  94  005D5         CLRB    FIT_OK                                  ;
                                30                        53  E9  005D7 66$:     BLBC    FIT_OK, 68$                             ; 0940
                                17                        55  E9  005DA         BLBC    BINARY, 67$                             ; 0941
                                50                        6E  D0  005DD         MOVL    SCANP, R0                               ; 0942
                                                01        A0  DD  005E0         PUSHL   1(R0)                                   ;
                                    00000000G   8F        DD  005E3         PUSHL   #ANLOBJ$_OBJGSDIDCVALB                  ;
                                                02        DD  005E9         PUSHL   #2                                      ;
                                                7E        D4  005EB         CLRL    -(SP)                                   ;
                         0000G  CF                        04  FB  005ED         CALLS   #4, ANL$FORMAT_LINE                     ;
                                                          16  11  005F2         BRB     68$                                     ;
                                                08        AE  DD  005F4 67$:     PUSHL   WORK_DSC+4                              ; 0944
                                7E              08        AE  3C  005F7         MOVZWL  WORK_DSC, -(SP)                         ;
                                    00000000G   8F        DD  005FB         PUSHL   #ANLOBJ$_OBJGSDIDCVALA                  ;
                                                02        DD  00601         PUSHL   #2                                      ;
                                                7E        D4  00603         CLRL    -(SP)                                   ;
                         0000G  CF                        05  FB  00605         CALLS   #5, ANL$FORMAT_LINE                     ;
                                50          04  AE        3C  0060A 68$:     MOVZWL  WORK_DSC, R0                            ; 0945
```

```
                                 6E         08 BE40   9E 0060E          MOVAB   @WORK_DSC+4[R0], SCANP                ;
                                 55               53   E9 00613          BLBC    FIT_OK, 71$                          ; 0949
                          50     6E               01   C1 00616          ADDL3   #1, SCANP, R0                        ;
                                 54               50   D1 0061A          CMPL    R0, R4                               ;
                                                  09   1B 0061D          BLEQU   69$                                  ;
                                                  5B   DD 0061F          PUSHL   R11                                  ;
                             0000G CF             01   FB 00621          CALLS   #1, ANL$FORMAT_ERROR                 ;
                                                  53   94 00626          CLRB    FIT_OK                               ;
                                 40               53   E9 00628 69$:     BLBC    FIT_OK, 71$                          ;
                                   04     AE      00   BE 9A 0062B       MOVZBL  @SCANP, WORK_DSC                     ;
                          08     AE       6E         01   C1 00630       ADDL3   #1, SCANP, WORK_DSC+4                ;
                                 33               53   E9 00635          BLBC    FIT_OK, 71$                          ;
                                 50         04    AE   3C 00638          MOVZWL  WORK_DSC, R0                         ;
                                 50               08   C6 0063C          DIVL2   #8, R0                               ;
                                 50               6E   C0 0063F          ADDL2   SCANP, R0                            ;
                                                  50   D6 00642          INCL    R0                                   ;
                                 54               50   D1 00644          CMPL    R0, R4                               ;
                                                  09   1B 00647          BLEQU   70$                                  ;
                                                  5B   DD 00649          PUSHL   R11                                  ;
                             0000G CF             01   FB 0064B          CALLS   #1, ANL$FORMAT_ERROR                 ;
                                                  53   94 00650          CLRB    FIT_OK                               ;
                                 16               53   E9 00652 70$:     BLBC    FIT_OK, 71$                          ; 0950
                                           08     AE   DD 00655          PUSHL   WORK_DSC+4                           ; 0951
                                 7E        08     AE   3C 00658          MOVZWL  WORK_DSC, -(SP)                      ;
                                     00000000G    8F   DD 0065C          PUSHL   #ANLOBJ$_OBJGSDIDCOBJ                ;
                                                  02   DD 00662          PUSHL   #2                                   ;
                                                  7E   D4 00664          CLRL    -(SP)                                ;
                             0000G CF             05   FB 00666          CALLS   #5, ANL$FORMAT_LINE                  ;
                                                00D8   31 0066B 71$:     BRW     78$                                  ; 0955
                                 FA               53   E9 0066E 72$:     BLBC    FIT_OK, 71$                          ; 0962
                                 50         03    A2   9E 00671          MOVAB   3(R2), R0                            ;
                                 54               50   D1 00675          CMPL    R0, R4                               ;
                                                  09   1B 00678          BLEQU   73$                                  ;
                                                  5B   DD 0067A          PUSHL   R11                                  ;
                             0000G CF             01   FB 0067C          CALLS   #1, ANL$FORMAT_ERROR                 ;
                                                  53   94 00681          CLRB    FIT_OK                               ;
                                 E5               53   E9 00683 73$:     BLBC    FIT_OK, 71$                          ; 0963
                                          0000'  CF   D6 00686          INCL    HIGHEST_DEF_ENV                      ; 0964
                                          0000'  CF   9F 0068A          PUSHAB  ENV_FLAGS_DEF                        ; 0965
                                 7E         01    A2   3C 0068E          MOVZWL  1(R2), -(SP)                         ;
                                     00000000G    8F   DD 00692          PUSHL   #ANLOBJ$_OBJGSDENVFLAGS              ;
                                                  02   DD 00698          PUSHL   #2                                   ;
                             0000G CF             04   FB 0069A          CALLS   #4, ANL$FORMAT_FLAGS                 ;
                                          0000'  CF   9F 0069F          PUSHAB  ENV_FLAGS_DEF                        ; 0966
                                 7E         01    A2   3C 006A3          MOVZWL  1(R2), -(SP)                         ;
                             0000G CF             02   FB 006A7          CALLS   #2, ANL$CHECK_FLAGS                  ;
                                 BC              53   E9 006AC          BLBC    FIT_OK, 71$                          ; 0972
                                 50         05    A2   9E 006AF          MOVAB   5(R2), R0                            ;
                                 54               50   D1 006B3          CMPL    R0, R4                               ;
                                                  09   1B 006B6          BLEQU   74$                                  ;
                                                  5B   DD 006B8          PUSHL   R11                                  ;
                             0000G CF             01   FB 006BA          CALLS   #1, ANL$FORMAT_ERROR                 ;
                                                  53   94 006BF          CLRB    FIT_OK                               ;
                                 A7               53   E9 006C1 74$:     BLBC    FIT_OK, 71$                          ; 0973
                                          0000'  CF   DD 006C4          PUSHL   HIGHEST_DEF_ENV                      ; 0974
                                 7E         03    A2   3C 006C8          MOVZWL  3(R2), -(SP)                         ;
                                     00000000G    8F   DD 006CC          PUSHL   #ANLOBJ$_OBJGSDENVPAR                ;
```

```
                                 02  DD 006D2          PUSHL   #2                                                    ;
                                 7E  D4 006D4          CLRL    -(SP)                                                 ;
                0000G  CF        05  FB 006D6          CALLS   #5, ANL$FORMAT_LINE                                   ;
                       7E    03  A2  3C 006DB          MOVZWL  3(R2), -(SP)                                          ; 0975
                0000V  CF        01  FB 006DF          CALLS   #1, ANL$OBJECT_ENV_REF                                ;
                       5F        53  E9 006E4          BLBC    FIT_OK, 78$                                           ; 0980
                       50    06  A2  9E 006E7          MOVAB   6(R2), R0                                             ;
                       54        50  D1 006EB          CMPL    R0, R4                                                ;
                                 09  1B 006EE          BLEQU   75$                                                   ;
                                 5B  DD 006F0          PUSHL   R11                                                   ;
                0000G  CF        01  FB 006F2          CALLS   #1, ANL$FORMAT_ERROR                                  ;
                                 53  94 006F7          CLRB    FIT_OK                                                ;
                       4A        53  E9 006F9 75$:     BLBC    FIT_OK, 78$                                           ;
                  04   AE    05  A2  9A 006FC          MOVZBL  5(R2), WORK_DSC                                       ;
                  08   AE    06  A2  9E 00701          MOVAB   6(R2), WORK_DSC+4                                     ;
                       3D        53  E9 00706          BLBC    FIT_OK, 78$                                           ;
                       50    04  AE  3C 00709          MOVZWL  WORK_DSC, R0                                          ;
                       50        08  C6 0070D          DIVL2   #8, R0                                                ;
                       50    01  A042 9E 00710          MOVAB   1(R0)[R2], R0                                         ;
                       54        50  D1 00715          CMPL    R0, R4                                                ;
                                 09  1B 00718          BLEQU   77$                                                   ;
                                 5B  DD 0071A 76$:     PUSHL   R11                                                   ;
                0000G  CF        01  FB 0071C          CALLS   #1, ANL$FORMAT_ERROR                                  ;
                                 53  94 00721          CLRB    FIT_OK                                                ;
                       20        53  E9 00723 77$:     BLBC    FIT_OK, 78$                                           ; 0981
                          08     AE  DD 00726          PUSHL   WORK_DSC+4                                            ; 0982
                       7E    08  AE  3C 00729          MOVZWL  WORK_DSC, -(SP)                                       ;
                          00000000G 8F  DD 0072D       PUSHL   #ANLOBJ$_OBJSYMBOL                                    ;
                                 02  DD 00733          PUSHL   #2                                                    ;
                                 7E  D4 00735          CLRL    -(SP)                                                 ;
                0000G  CF        05  FB 00737          CALLS   #5, ANL$FORMAT_LINE                                   ;
                                 1F  DD 0073C          PUSHL   #31                                                   ; 0983
                          08     AE  9F 0073E          PUSHAB  WORK_DSC                                              ;
                0000G  CF        02  FB 00741          CALLS   #2, ANL$CHECK_SYMBOL                                  ;
                       50    04  AE  3C 00746 78$:     MOVZWL  WORK_DSC, R0                                          ; 0988
                       6E    08 BE40 9E 0074A          MOVAB   @WORK_DSC+4[R0], SCANP                                ;
                          F8D8     31 0074F          BRW     1$                                                    ; 0657
                                   04 00752          RET                                                           ; 0997

; Routine Size:  1875 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  572          0998  1 %sbttl 'ANL$OBJECT_ARGUMENT_DSC - Analyze Argument Descriptors'
;  573          0999  1 !++
;  574          1000  1 ! Functional Description:
;  575          1001  1 !       This routine analyzes argument descriptors, which appear in GSD
;  576          1002  1 !       records and in TIR commands.
;  577          1003  1 !
;  578          1004  1 ! Formal Parameters:
;  579          1005  1 !       indent_level    Level at which to indent lines.
;  580          1006  1 !       scanp_address   Address of argument descriptor block pointer.  We
;  581          1007  1 !                       update it to point past the block.
;  582          1008  1 !       the_record      Address of descriptor of record containing block.
;  583          1009  1 !
;  584          1010  1 ! Implicit Inputs:
;  585          1011  1 !       global data
;  586          1012  1 !
;  587          1013  1 ! Implicit Outputs:
;  588          1014  1 !       global data
;  589          1015  1 !
;  590          1016  1 ! Returned Value:
;  591          1017  1 !       True if descriptor fit in record; false otherwise.
;  592          1018  1 !
;  593          1019  1 ! Side Effects:
;  594          1020  1 !
;  595          1021  1 !--
;  596          1022  1
;  597          1023  1
;  598          1024  2 global routine anl$object_argument_dsc(indent_level,scanp_address,the_record) = begin
;  599          1025  2
;  600          1026  2 bind
;  601          1027  2         scanp = .scanp_address: ref block[,byte],
;  602          1028  2         record_dsc = .the_record: descriptor;
;  603          1029  2
;  604          1030  2 own
;  605          1031  2         passing_mechanism_table: vector[4,long] initial(
;  606          1032  2                                 uplit byte(%ascic 'UNKNOWN'),
;  607          1033  2                                 uplit byte(%ascic 'VALUE'),
;  608          1034  2                                 uplit byte(%ascic 'REF'),
;  609          1035  2                                 uplit byte(%ascic 'DESC'));
;  610          1036  2
;  611          1037  2 local
;  612          1038  2         fit_ok: byte,
;  613          1039  2         work_dsc: descriptor;
;  614          1040  2
;  615          1041  2
;  616          1042  2 ! The argument descriptor begins with a validation control byte containing
;  617          1043  2 ! the passing mechanism.  Print it and check it.
;  618          1044  2
;  619          1045  2 fit_ok = true;
;  620          1046  2
;  621          1047  2 ensure_field_fit(0,0,8,0,record_dsc);
;  622          1048  3 if .fit_ok then (
;  623          1049  3         anl$format_line(0,.indent_level,anlobj$_objfadpassmech,.passing_mechanism_table[.scanp[0,0,2,0]]);
;  624          1050  3         if .scanp[0,2,6,0] nequ 0 then
;  625          1051  3                 anl$format_error(anlobj$_objfadbadavc);
;  626          1052  3         increment (scanp);
;  627          1053  2 );
;  628          1054  2
```

```
;  629          1055  2 ensure_ascic_fit(0,0,8,0,record_dsc,work_dsc);
;  630          1056  3 if .fit_ok then (
;  631          1057  3         if .work_dsc[len] nequ 0 then
;  632          1058  3                 anl$format_error(anlobj$_objfadbadrbc);
;  633          1059  3         scanp = .work_dsc[ptr] + .work_dsc[len];
;  634          1060  2 );
;  635          1061  2
;  636          1062  2 return .fit_ok;
;  637          1063  2
;  638          1064  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                     4E  57  4F  4E  4B  4E  55  07  000BC P.AAT:  .ASCII  <7>\UNKNOWN\                          ;
                             45  55  4C  41  56  05  000C4 P.AAU:  .ASCII  <5>\VALUE\                            ;
                                 46  45  52  03  000CA P.AAV:  .ASCII  <3>\REF\                              ;
                                 43  53  45  44  04  000CE P.AAW:  .ASCII  <4>\DESC\                             ;

                                                                .PSECT  $OWN$,NOEXE,2

                00000000' 00000000' 00000000' 00000000' 000A0 PASSING_MECHANISM_TABLE:
                                                                .ADDRESS P.AAT, P.AAU, P.AAV, P.AAW             ;


                                                                .PSECT  $CODE$,NOWRT,2

                                    00FC 00000                  .ENTRY  ANL$OBJECT_ARGUMENT_DSC, Save R2,R3,R4,R5,- ; 1024
                                                                        R6,R7                                   ;
                            57      0000G   CF  9E  00002       MOVAB   ANL$FORMAT_ERROR, R7                    ;
                            56  00000000G   8F  D0  00007       MOVL    #ANLOBJ$_FIELDFIT, R6                   ;
                            5E              08  C2  0000E       SUBL2   #8, SP                                  ;
                            53        08    AC  D0  00011       MOVL    SCANP_ADDRESS, R3                       ; 1027
                            52        0C    AC  D0  00015       MOVL    THE_RECORD, R2                          ; 1028
                            55              01  90  00019       MOVB    #1, FIT_OK                              ; 1045
                            6E              55  E9  0001C       BLBC    FIT_OK, 4$                              ; 1047
                     51     63              01  C1  0001F       ADDL3   #1, (R3), R1                            ;
                            50              62  3C  00023       MOVZWL  (R2), R0                                ;
                            50        04    A2  C0  00026       ADDL2   4(R2), R0                               ;
                            50              51  D1  0002A       CMPL    R1, R0                                  ;
                                            07  1B  0002D       BLEQU   1$                                      ;
                                            56  DD  0002F       PUSHL   R6                                      ;
                            67              01  FB  00031       CALLS   #1, ANL$FORMAT_ERROR                    ;
                                            55  94  00034       CLRB    FIT_OK                                  ;
                            77              55  E9  00036 1$:   BLBC    FIT_OK, 5$                              ; 1048
                            54              63  D0  00039       MOVL    (R3), R4                                ; 1049
                50          64      02      00  EF  0003C       EXTZV   #0, #2, (R4), R0                        ;
                                  0000'CF   40  DD  00041       PUSHL   PASSING_MECHANISM_TABLE[R0]             ;
                                00000000G   8F  DD  00046       PUSHL   #ANLOBJ$_OBJFADPASSMECH                 ;
                                      04    AC  DD  0004C       PUSHL   INDENT_LEVEL                            ;
                                            7E  D4  0004F       CLRL    -(SP)                                   ;
                                0000G   CF      04  FB  00051   CALLS   #4, ANL$FORMAT_LINE                     ;
                                  FC    8F      64  93  00056   BITB    (R4), #252                              ; 1050
                                            09  13  0005A       BEQL    2$                                      ;
                                00000000G   8F  DD  0005C       PUSHL   #ANLOBJ$_OBJFADBADAVC                   ; 1051
```

```
                                67               01 FB 00062          CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                 63 D6 00065 2$:      INCL    (R3)                                    ; 1052
                                5E               55 E9 00067          BLBC    FIT_OK, 7$                              ; 1055
                          51    63               01 C1 0006A          ADDL3   #1, (R3), R1                            ;
                                50               62 3C 0006E          MOVZWL  (R2), R0                                ;
                                50         04    A2 C0 00071          ADDL2   4(R2), R0                               ;
                                50               51 D1 00075          CMPL    R1, R0                                  ;
                                                 07 1B 00078          BLEQU   3$                                      ;
                                                 56 DD 0007A          PUSHL   R6                                      ;
                                67               01 FB 0007C          CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                 55 94 0007F          CLRB    FIT_OK                                  ;
                                44               55 E9 00081 3$:      BLBC    FIT_OK, 7$                              ;
                                6E         00    B3 9A 00084          MOVZBL  @0(R3), WORK_DSC                        ;
                          04    AE               63 C1 00088          ADDL3   #1, (R3), WORK_DSC+4                    ;
                                38               55 E9 0008D 4$:      BLBC    FIT_OK, 7$                              ;
                                50               6E 3C 00090          MOVZWL  WORK_DSC, R0                            ;
                                50               08 C6 00093          DIVL2   #8, R0                                  ;
                                50               63 C0 00096          ADDL2   (R3), R0                                ;
                                51         01    A0 9E 00099          MOVAB   1(R0), R1                               ;
                                50               62 3C 0009D          MOVZWL  (R2), R0                                ;
                                50         04    A2 C0 000A0          ADDL2   4(R2), R0                               ;
                                50               51 D1 000A4          CMPL    R1, R0                                  ;
                                                 07 1B 000A7          BLEQU   5$                                      ;
                                                 56 DD 000A9          PUSHL   R6                                      ;
                                67               01 FB 000AB          CALLS   #1, ANL$FORMAT_ERROR                    ;
                                                 55 94 000AE          CLRB    FIT_OK                                  ;
                                15               55 E9 000B0 5$:      BLBC    FIT_OK, 7$                              ; 1056
                                                 6E B5 000B3          TSTW    WORK_DSC                                ; 1057
                                                 09 13 000B5          BEQL    6$                                      ;
                          00000000G              8F DD 000B7          PUSHL   #ANLOBJ$_OBJFADBADRBC                   ; 1058
                                67               01 FB 000BD          CALLS   #1, ANL$FORMAT_ERROR                    ;
                                50               6E 3C 000C0 6$:      MOVZWL  WORK_DSC, R0                            ; 1059
                                63         04 BE40 9E 000C3          MOVAB   @WORK_DSC+4[R0], (R3)                   ;
                                50               55 9A 000C8 7$:      MOVZBL  FIT_OK, R0                              ; 1062
                                                    04 000CB          RET                                             ; 1064

; Routine Size:  204 bytes,    Routine Base:  $CODE$ + 0753
```

```
;  640          1065  1 %sbttl 'ANL$OBJECT_PSECT_REF - Mark Psect Reference'
;  641          1066  1 !++
;  642          1067  1 ! Functional Description:
;  643          1068  1 !       This routine is called to mark a psect reference in the psect
;  644          1069  1 !       reference bitvector.  By remembering every psect that is referenced
;  645          1070  1 !       we can check whether undefined psects are ever referenced.
;  646          1071  1 !
;  647          1072  1 ! Formal Parameters:
;  648          1073  1 !       psect_number    The number of the psect that was referenced.
;  649          1074  1 !
;  650          1075  1 ! Implicit Inputs:
;  651          1076  1 !       global data
;  652          1077  1 !
;  653          1078  1 ! Implicit Outputs:
;  654          1079  1 !       global data
;  655          1080  1 !
;  656          1081  1 ! Returned Value:
;  657          1082  1 !       none
;  658          1083  1 !
;  659          1084  1 ! Side Effects:
;  660          1085  1 !
;  661          1086  1 !--
;  662          1087  1
;  663          1088  1
;  664          1089  2 global routine anl$object_psect_ref(psect_number): novalue = begin
;  665          1090  2
;  666          1091  2 local
;  667          1092  2         status: long;
;  668          1093  2
;  669          1094  2
;  670          1095  2 ! We begin by checking to see whether or not a psect reference bitvector
;  671          1096  2 ! has been allocated.  If not, we allocate it and clear it.
;  672          1097  2
;  673          1098  3 if .psect_ref_bits eqla 0 then (
;  674          1099  3         status = lib$get_vm(%ref(65536/8),psect_ref_bits);
;  675          1100  3         check (.status,.status);
;  676          1101  3         ch$fill(%x'00', 65536/8,.psect_ref_bits);
;  677          1102  2 );
;  678          1103  2
;  679          1104  2 ! Now we can set the psect bit and remember the highest referenced psect.
;  680          1105  2
;  681          1106  2 psect_ref_bits[.psect_number] = true;
;  682          1107  2 highest_ref_psect = max(.psect_number,.highest_ref_psect);
;  683          1108  2
;  684          1109  2 return;
;  685          1110  2
;  686          1111  1 end;


                                        007C 00000          .ENTRY  ANL$OBJECT_PSECT_REF, Save R2,R3,R4,R5,R6   ; 1089
                                5E      CF   9E 00002       MOVAB   PSECT_REF_BITS, R6                          ;
                                5E           04  C2 00007   SUBL2   #4, SP                                      ;
                                             66  D5 0000A   TSTL    PSECT_REF_BITS                              ; 1098
                                             27  12 0000C   BNEQ    2$                                          ;
```

```
                                   56 DD 0000E          PUSHL   R6                                        ; 1099
                  04  AE    2000   8F 3C 00010          MOVZWL  #8192, 4(SP)                              ;
                              04   AE 9F 00016          PUSHAB  4(SP)                                     ;
             00000000G  00         02 FB 00019          CALLS   #2, LIB$GET_VM                            ;
                        09         50 E8 00020          BLBS    STATUS, 1$                                ; 1100
                                   50 DD 00023          PUSHL   STATUS                                    ;
             00000000G  00         01 FB 00025          CALLS   #1, LIB$SIGNAL                            ;
   2000  8F          00         6E 00 2C 0002C 1$:     MOVC5   #0, (SP), #0, #8192, @PSECT_REF_BITS      ; 1101
                              00   B6    00033                                                           ;
             00      00  B6    04   AC E2 00035 2$:     BBSS    PSECT_NUMBER, @PSECT_REF_BITS, 3$         ; 1106
                         50    04   AC D0 0003B 3$:     MOVL    PSECT_NUMBER, R0                          ; 1107
                     FC  A6         50 D1 0003F          CMPL    R0, HIGHEST_REF_PSECT                     ;
                                   04 18 00043          BGEQ    4$                                        ;
                         50    FC   A6 D0 00045          MOVL    HIGHEST_REF_PSECT, R0                     ;
                     FC  A6         50 D0 00049 4$:     MOVL    R0, HIGHEST_REF_PSECT                     ;
                                      04 0004D          RET                                               ; 1111

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 081F
```

```
;  688          1112  1 %sbttl 'ANL$OBJECT_PSECT_CHECK - Check Psect References'
;  689          1113  1 !++
;  690          1114  1 ! Functional Description:
;  691          1115  1 !       This routine is called at the end of an object module to check the
;  692          1116  1 !       psect references.  We need to make sure that no undefined psects
;  693          1117  1 !       were referenced.
;  694          1118  1 !
;  695          1119  1 ! Formal Parameters:
;  696          1120  1 !       none
;  697          1121  1 !
;  698          1122  1 ! Implicit Inputs:
;  699          1123  1 !       global data
;  700          1124  1 !
;  701          1125  1 ! Implicit Outputs:
;  702          1126  1 !       global data
;  703          1127  1 !
;  704          1128  1 ! Returned Value:
;  705          1129  1 !       none
;  706          1130  1 !
;  707          1131  1 ! Side Effects:
;  708          1132  1 !
;  709          1133  1 !--
;  710          1134  1
;  711          1135  1
;  712          1136  2 global routine anl$object_psect_check: novalue = begin
;  713          1137  2
;  714          1138  2 local
;  715          1139  2         status: long;
;  716          1140  2
;  717          1141  2
;  718          1142  2 ! First let's make sure that at least one psect was defined.  An object
;  719          1143  2 ! module must define at least one.
;  720          1144  2
;  721          1145  2 if .highest_def_psect lss 0 then
;  722          1146  2         anl$format_error(anlobj$_objnopsc);
;  723          1147  2
;  724          1148  2 ! OK, now we are going to make sure that all referenced psects were defined.
;  725          1149  2 ! We do this by looping through any psect referenced bits whose number is
;  726          1150  2 ! higher than the highest defined psect.
;  727          1151  2
;  728          1152  3 if .highest_ref_psect gtr .highest_def_psect then (
;  729          1153  3         anl$format_error(anlobj$_objundefpsc);
;  730          1154  4         incru i from .highest_def_psect+1 to .highest_ref_psect do (
;  731          1155  4                 if .psect_ref_bits[.i] then
;  732          1156  4                         anl$format_error(anlobj$_objbadnum,.i);
;  733          1157  3         );
;  734          1158  2 );
;  735          1159  2
;  736          1160  2 ! Now we can reset everything for the next module.
;  737          1161  2
;  738          1162  2 highest_def_psect = highest_ref_psect = -1;
;  739          1163  3 if .psect_ref_bits neqa 0 then (
;  740          1164  3         status = lib$free_vm(%ref(65536/8),psect_ref_bits);
;  741          1165  3         check (.status,.status);
;  742          1166  3         psect_ref_bits = 0;
;  743          1167  2 );
;  744          1168  2
```

```
;  745          1169  2 return;
;  746          1170  2
;  747          1171  1 end;


                                                 003C 00000           .ENTRY  ANL$OBJECT_PSECT_CHECK, Save R2,R3,R4,R5    ; 1136
                                        55     0000G CF   9E 00002    MOVAB   ANL$FORMAT_ERROR, R5                        ;
                                        54     0000' CF   9E 00007    MOVAB   HIGHEST_DEF_PSECT, R4                       ;
                                        5E           04   C2 0000C    SUBL2   #4, SP                                      ;
                                                     64   D5 0000F    TSTL    HIGHEST_DEF_PSECT                           ; 1145
                                                     09   18 00011    BGEQ    1$                                          ;
                                    00000000G        8F   DD 00013    PUSHL   #ANLOBJ$_OBJNOPSC                           ; 1146
                                        65           01   FB 00019    CALLS   #1, ANL$FORMAT_ERROR                        ;
                                        64     04    A4   D1 0001C 1$:  CMPL    HIGHEST_REF_PSECT, HIGHEST_DEF_PSECT        ; 1152
                                                     25   15 00020    BLEQ    4$                                          ;
                                    00000000G        8F   DD 00022    PUSHL   #ANLOBJ$_OBJUNDEFPSC                        ; 1153
                                        65           01   FB 00028    CALLS   #1, ANL$FORMAT_ERROR                        ;
                                        52           64   7D 0002B    MOVQ    HIGHEST_DEF_PSECT, I                        ; 1154
                                                     10   11 0002E    BRB     3$                                          ;
                                OB      08   B4      52   E1 00030 2$:  BBC     I, @PSECT_REF_BITS, 3$                      ; 1155
                                                     52   DD 00035    PUSHL   I                                           ; 1156
                                    00000000G        8F   DD 00037    PUSHL   #ANLOBJ$_OBJBADNUM                          ;
                                        65           02   FB 0003D    CALLS   #2, ANL$FORMAT_ERROR                        ;
                                                     52   D6 00040 3$:  INCL    I                                           ; 1154
                                        53           52   D1 00042    CMPL    I, R3                                       ;
                                                     E9   1B 00045    BLEQU   2$                                          ;
                                  04    A4           01   CE 00047 4$:  MNEGL   #1, HIGHEST_REF_PSECT                      ; 1162
                                        64           01   CE 0004B    MNEGL   #1, HIGHEST_DEF_PSECT                      ;
                                              08     A4   D5 0004E    TSTL    PSECT_REF_BITS                              ; 1163
                                                     22   13 00051    BEQL    6$                                          ;
                                              08     A4   9F 00053    PUSHAB  PSECT_REF_BITS                              ; 1164
                                        04    AE     2000 8F   3C 00056    MOVZWL  #8192, 4(SP)                                ;
                                              04     AE   9F 0005C    PUSHAB  4(SP)                                       ;
                                00000000G     00         02   FB 0005F    CALLS   #2, LIB$FREE_VM                             ;
                                              09         50   E8 00066    BLBS    STATUS, 5$                                  ; 1165
                                                         50   DD 00069    PUSHL   STATUS                                      ;
                                00000000G     00         01   FB 0006B    CALLS   #1, LIB$SIGNAL                              ;
                                              08         A4   D4 00072 5$:  CLRL    PSECT_REF_BITS                              ; 1166
                                                         04   00075 6$:  RET                                                 ; 1171

; Routine Size:  118 bytes,    Routine Base:  $CODE$ + 086D
```

```
;  749          1172  1 %sbttl 'ANL$OBJECT_ENV_REF - Mark Environment Reference'
;  750          1173  1 !++
;  751          1174  1 ! Functional Description:
;  752          1175  1 !       This routine is called to mark a environment reference in the environ-
;  753          1176  1 !       ment reference bitvector.  By remembering every environment that is
;  754          1177  1 !       referenced we can check whether undefined environments are ever
;  755          1178  1 !       referenced.
;  756          1179  1 !
;  757          1180  1 ! Formal Parameters:
;  758          1181  1 !       env_number      The number of the environment that was referenced.
;  759          1182  1 !
;  760          1183  1 ! Implicit Inputs:
;  761          1184  1 !       global data
;  762          1185  1 !
;  763          1186  1 ! Implicit Outputs:
;  764          1187  1 !       global data
;  765          1188  1 !
;  766          1189  1 ! Returned Value:
;  767          1190  1 !       none
;  768          1191  1 !
;  769          1192  1 ! Side Effects:
;  770          1193  1 !
;  771          1194  1 !--
;  772          1195  1
;  773          1196  1
;  774          1197  2 global routine anl$object_env_ref(env_number): novalue = begin
;  775          1198  2
;  776          1199  2 local
;  777          1200  2         status: long;
;  778          1201  2
;  779          1202  2
;  780          1203  2 ! We begin by checking to see whether or not an environment reference bitvector
;  781          1204  2 ! has been allocated.  If not, we allocate it and clear it.
;  782          1205  2
;  783          1206  3 if .env_ref_bits eqla 0 then (
;  784          1207  3         status = lib$get_vm(%ref(65536/8),env_ref_bits);
;  785          1208  3         check (.status,.status);
;  786          1209  3         ch$fill(%x'00', 65536/8,.env_ref_bits);
;  787          1210  2 );
;  788          1211  2
;  789          1212  2 ! Now we can set the environment bit and remember the highest referenced one.
;  790          1213  2
;  791          1214  2 env_ref_bits[.env_number] = true;
;  792          1215  2 highest_ref_env = max(.env_number,.highest_ref_env);
;  793          1216  2
;  794          1217  2 return;
;  795          1218  2
;  796          1219  1 end;


                                007C 00000          .ENTRY  ANL$OBJECT_ENV_REF, Save R2,R3,R4,R5,R6     ; 1197
                          56    CF  9E 00002          MOVAB   ENV_REF_BITS, R6                            ;
                          5E    04  C2 00007          SUBL2   #4, SP                                      ;
                                66  D5 0000A          TSTL    ENV_REF_BITS                                ; 1206
```

```
                                                                27  12 0000C          BNEQ    2$                                                      ;
                                                                56  DD 0000E          PUSHL   R6                                                      ; 1207
                                        04    AE      2000      8F  3C 00010          MOVZWL  #8192, 4(SP)                                            ;
                                                        04      AE  9F 00016          PUSHAB  4(SP)                                                   ;
                                00000000G   00                  02  FB 00019          CALLS   #2, LIB$GET_VM                                          ;
                                            09                  50  E8 00020          BLBS    STATUS, 1$                                              ; 1208
                                                                50  DD 00023          PUSHL   STATUS                                                  ;
                                00000000G   00                  01  FB 00025          CALLS   #1, LIB$SIGNAL                                          ;
    2000    8F                      00      6E                  00  2C 0002C 1$:      MOVC5   #0, (SP), #0, #8192, @ENV_REF_BITS                      ; 1209
                                                        00      B6  00033                                                                           ;
                                    00      00      B6      04  AC  E2 00035 2$:      BBSS    ENV_NUMBER, @ENV_REF_BITS, 3$                           ; 1214
                                            50      04      AC  D0 0003B 3$:      MOVL    ENV_NUMBER, R0                                          ; 1215
                                        FC  A6                  50  D1 0003F          CMPL    R0, HIGHEST_REF_ENV                                     ;
                                                                04  18 00043          BGEQ    4$                                                      ;
                                            50          FC      A6  D0 00045          MOVL    HIGHEST_REF_ENV, R0                                     ;
                                        FC  A6                  50  D0 00049 4$:      MOVL    R0, HIGHEST_REF_ENV                                     ;
                                                                    04 0004D          RET                                                             ; 1219

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 08E3
```

```
;  798          1220  1 %sbttl 'ANL$OBJECT_ENV_CHECK - Check Environment References'
;  799          1221  1 !++
;  800          1222  1 ! Functional Description:
;  801          1223  1 !       This routine is called at the end of an object module to check the
;  802          1224  1 !       environment references.  We need to make sure that no undefined
;  803          1225  1 !       environments were referenced.
;  804          1226  1 !
;  805          1227  1 ! Formal Parameters:
;  806          1228  1 !       none
;  807          1229  1 !
;  808          1230  1 ! Implicit Inputs:
;  809          1231  1 !       global data
;  810          1232  1 !
;  811          1233  1 ! Implicit Outputs:
;  812          1234  1 !       global data
;  813          1235  1 !
;  814          1236  1 ! Returned Value:
;  815          1237  1 !       none
;  816          1238  1 !
;  817          1239  1 ! Side Effects:
;  818          1240  1 !
;  819          1241  1 !--
;  820          1242  1
;  821          1243  1
;  822          1244  2 global routine anl$object_env_check: novalue = begin
;  823          1245  2
;  824          1246  2 local
;  825          1247  2         status: long;
;  826          1248  2
;  827          1249  2
;  828          1250  2 ! We are going to make sure that all referenced environments were defined.
;  829          1251  2 ! We do this by looping through any environment reference bits whose number is
;  830          1252  2 ! higher than the highest defined environment.
;  831          1253  2
;  832          1254  3 if .highest_ref_env gtr .highest_def_env then (
;  833          1255  3         anl$format_error(anlobj$_objundefenv);
;  834          1256  4         incru i from .highest_def_env+1 to .highest_ref_env do (
;  835          1257  4                 if .env_ref_bits[.i] then
;  836          1258  4                         anl$format_error(anlobj$_objbadnum,.i);
;  837          1259  3         );
;  838          1260  2 );
;  839          1261  2
;  840          1262  2 ! Now we can reset everything for the next module.
;  841          1263  2
;  842          1264  2 highest_def_env = highest_ref_env = -1;
;  843          1265  2 if .env_ref_bits neqa 0 then (
;  844          1266  3         status = lib$free_vm(%ref(65536/8),env_ref_bits);
;  845          1267  3         check (.status,.status);
;  846          1268  3         env_ref_bits = 0;
;  847          1269  2 );
;  848          1270  2
;  849          1271  2 return;
;  850          1272  2
;  851          1273  1 end;
```

```
                                                 001C 00000          .ENTRY  ANL$OBJECT_ENV_CHECK, Save R2,R3,R4                 ; 1244
                          54      0000'  CF   9E 00002          MOVAB   ENV_REF_BITS, R4                                    ;
                          5E             04   C2 00007          SUBL2   #4, SP                                              ;
                       F8 A4        FC   A4   D1 0000A          CMPL    HIGHEST_REF_ENV, HIGHEST_DEF_ENV                    ; 1254
                                         2A   15 0000F          BLEQ    3$                                                  ;
                              00000000G  8F   DD 00011          PUSHL   #ANLOBJ$_OBJUNDEFENV                                ; 1255
                            0000G CF     01   FB 00017          CALLS   #1, ANL$FORMAT_ERROR                                ;
                                   52        F8   A4   7D 0001C          MOVQ    HIGHEST_DEF_ENV, I                          ; 1256
                                             12   11 00020          BRB     2$                                                  ;
                    0D      00 B4           52   E1 00022 1$:       BBC     I, @ENV_REF_BITS, 2$                        ; 1257
                                             52   DD 00027          PUSHL   I                                                   ; 1258
                              00000000G  8F   DD 00029          PUSHL   #ANLOBJ$_OBJBADNUM                                  ;
                            0000G CF     02   FB 0002F          CALLS   #2, ANL$FORMAT_ERROR                                ;
                                             52   D6 00034 2$:       INCL    I                                                   ; 1256
                                   53        52   D1 00036          CMPL    I, R3                                               ;
                                             E7   1B 00039          BLEQU   1$                                                  ;
                       FC A4                 01   CE 0003B 3$:       MNEGL   #1, HIGHEST_REF_ENV                                 ; 1264
                       F8 A4                 01   CE 0003F          MNEGL   #1, HIGHEST_DEF_ENV                                 ;
                                             64   D5 00043          TSTL    ENV_REF_BITS                                        ; 1265
                                             20   13 00045          BEQL    5$                                                  ;
                                             54   DD 00047          PUSHL   R4                                                  ; 1266
                       04 AE        2000     8F   3C 00049          MOVZWL  #8192, 4(SP)                                        ;
                                      04 AE   9F   0004F          PUSHAB  4(SP)                                               ;
                    00000000G  00            02   FB 00052          CALLS   #2, LIB$FREE_VM                                     ;
                                   09        50   E8 00059          BLBS    STATUS, 4$                                          ; 1267
                                             50   DD 0005C          PUSHL   STATUS                                              ;
                    00000000G  00            01   FB 0005E          CALLS   #1, LIB$SIGNAL                                      ;
                                             64   D4 00065 4$:       CLRL    ENV_REF_BITS                                        ; 1268
                                             04   00067 5$:       RET                                                         ; 1273

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 0931

;  852          1274  1
;  853          1275  0 end eludom



                                                                    .EXTRN  LIB$SIGNAL

;                                    PSECT SUMMARY
;
;      Name                      Bytes                          Attributes
;
;  $OWN$                           176  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          211  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         2457  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;
;                                 -------- Symbols --------      Pages       Processing
```

```
;       File                                    Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1             9776        46         0         581       00:01.0




;                                       COMMAND QUALIFIERS

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

; Size:          2457 code + 387 data bytes
; Run Time:        00:42.8
; Elapsed Time:    02:26.0
; Lines/CPU Min:    1789
; Lexemes/CPU-Min: 16888
; Memory Used:  666 pages
; Compilation Complete
```