J 4

```
   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
```

```
**FILE**ID**RMSFDL                                    H  6

RRRRRRRR   MM      MM   SSSSSSSS  FFFFFFFFFF  DDDDDDDD   LL
RRRRRRRR   MM      MM   SSSSSSSS  FFFFFFFFFF  DDDDDDDD   LL
RR      RR MMMM  MMMM SS          FF          DD      DD LL
RR      RR MMMM  MMMM SS          FF          DD      DD LL
RR      RR MM  MM  MM SS          FF          DD      DD LL
RR      RR MM  MM  MM SS          FF          DD      DD LL
RRRRRRRR   MM      MM   SSSSSS    FFFFFFFF    DD      DD LL
RRRRRRRR   MM      MM   SSSSSS    FFFFFFFF    DD      DD LL
RR  RR     MM      MM         SS  FF          DD      DD LL
RR  RR     MM      MM         SS  FF          DD      DD LL
RR    RR   MM      MM         SS  FF          DD      DD LL          ....
RR    RR   MM      MM         SS  FF          DD      DD LL          ....
RR      RR MM      MM SSSSSSSS    FF          DDDDDDDD   LLLLLLLLLL  ....
RR      RR MM      MM SSSSSSSS    FF          DDDDDDDD   LLLLLLLLLL  ....

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

```
0 %title 'RMSFDL - Generate FDL for a File'
0         module rmsfdl    (
1                          ident='V04-000') = begin
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! Facility:     VAX/VMS Analyze Facility, Generate FDL for a File
1 !
1 ! Abstract:     This module is responsible for generating the File Definition
1 !               Language (FDL) for an extant file.  The user can then create
1 !               additional similar files, or modify the FDL and create
1 !               different sorts of file.
1 !               See "Functional Specification for FDL - VAX-11 RMS File
1 !               Definition Language" by Ken Henderson.
1 !
1 !
1 ! Environment:
1 !
1 ! Author: Paul C. Anagnostopoulos, Creation Date: 14 July 1981
1 !
1 ! Modified By:
1 !
1 !       V03-006 DGB0049         Donald G. Blair         08-May-1984
1 !               Fix condition handling so ANALYZRMS returns the correct
1 !               error status at image exit.  Change condition handler
1 !               from ANL$CONDITION_HANDLER to ANL$UNWIND_HANDLER.
1 !
1 !       V03-005 PCA1012         Paul C. Anagnostopoulos  6-Apr-1983
1 !               Add code to support the new total area allocation field
1 !               in the area descriptor.
1 !
1 !       V03-004 PCA1011         Paul C. Anagnostopoulos  1-Apr-1983
1 !               Change the message prefix to ANLRMS$_ to ensure that
```

```
 1 !                  message symbols are unique across all ANALYZEs.  This
 1 !                  is necessitated by the new merged message files.
 1 !
 1 !          V03-003 PCA1002         Paul C. Anagnostopoulos 25-Oct-1982
 1 !                  Change the way that FDL lines with quoted strings are
 1 !                  produced so that they use the new ANL$PREPARE_QUOTED_STRING
 1 !                  routine.  Remove all FDL pertaining to area allocation.
 1 !                  Add the new quadword key data types.
 1 !
 1 !          V03-001 PCA0008         Paul Anagnostopoulos    16-Mar-1982
 1 !                  Put out an allocation in the area primary of an FDL spec.
 1 !                  Even though it might not be the entire allocation,
 1 !                  something is better than nothing.
 1 !
 1 !          V03-002 PCA0007         Paul Anagnostopoulos    16-Mar-1982
 1 !                  Don't put out the compression secondaries in a prologue 2
 1 !                  FDL spec.
 1 !--
```

```
;    77         0076  1 %sbttl 'Module Declarations'
;    78         0077  1 !
;    79         0078  1 ! Libraries and Requires:
;    80         0079  1 !
;    81         0080  1
;    82         0081  1 library 'lib';
;    83         0082  1 require 'rmsreq';
;    84         0591  1
;    85         0592  1 !
;    86         0593  1 ! Table of Contents:
;    87         0594  1 !
;    88         0595  1
;    89         0596  1 forward routine
;    90         0597  1         anl$fdl_mode: novalue,
;    91         0598  1         anl$fdl_record: novalue,
;    92         0599  1         anl$fdl_areas: novalue,
;    93         0600  1         anl$fdl_keys: novalue,
;    94         0601  1         anl$analyze_areas: novalue,
;    95         0602  1         anl$analyze_keys: novalue;
;    96         0603  1
;    97         0604  1 !
;    98         0605  1 ! External References:
;    99         0606  1 !
;   100         0607  1
;   101         0608  1 external routine
;   102         0609  1         anl$area_descriptor,
;   103         0610  1         anl$bucket,
;   104         0611  1         anl$fdl_analysis_of_area,
;   105         0612  1         anl$fdl_analysis_of_key,
;   106         0613  1         anl$fdl_file,
;   107         0614  1         anl$format_line,
;   108         0615  1         anl$format_skip,
;   109         0616  1         anl$idx_check_key_stuff,
;   110         0617  1         anl$key_descriptor,
;   111         0618  1         anl$open_next_rms_file,
;   112         0619  1         anl$prepare_quoted_string,
;   113         0620  1         anl$prepare_report_file,
;   114         0621  1         anl$unwind_handler,
;   115         0622  1         anl$3reclaimed_bucket_header,
;   116         0623  1         cli$get_value: addressing_mode(general),
;   117         0624  1         lib$establish: addressing_mode(general);
;   118         0625  1
;   119         0626  1 external
;   120         0627  1         anl$gl_fat: ref block[,byte],
;   121         0628  1         anl$gw_prolog: word;
;   122         0629  1
;   123         0630  1 !
;   124         0631  1 ! Own Variables:
;   125         0632  1 !
;   126         0633  1 ! The following little table is for putting out boolean items.
;   127         0634  1
;   128         0635  1 own
;   129         0636  1         yes_no: vector[2,long] initial(
;   130         0637  1                         uplit byte (%ascic 'no'),
;   131         0638  1                         uplit byte (%ascic 'yes')
;   132         0639  1                         );
```

```
1 %sbttl 'ANL$FDL_MODE - Drive the Generation of an FDL'
1 !++
1 ! Functional Description:
1 !         This routine is responsible for driving the generation of an
1 !         FDL spec for a file.  We open the file and call various routines
1 !         to generate parts of the FDL.
1 !
1 ! Formal Parameters:
1 !         none
1 !
1 ! Implicit Inputs:
1 !         global data
1 !
1 ! Implicit Outputs:
1 !         global data
1 !
1 ! Returned Value:
1 !         none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_mode: novalue = begin
2
2 local
2         status: long;
2 local
2         local_described_buffer(resultant_file_spec,nam$c_maxrss);
2
2
2 ! Establish the condition handler for drastic structure errors.
2
2 lib$establish(anl$unwind_handler);
2
2 ! Begin by opening the file to be analyzed.  If the user blew it, just quit.
2
2 if not anl$open_next_rms_file(resultant_file_spec) then
2         return;
2
2 ! Now we can prepare the output file to receive the FDL specification.
2 ! We don't want any page headings in the file.
2
2 anl$prepare_report_file(0,resultant_file_spec);
2
2 ! Begin the spec with an IDENT that identifies who produced it.
2
2 anl$format_line(0,0,anlrms$_fdlident,0);
2
2 ! Now put out the system primary with the source.
2
2 anl$format_skip(0);
2 anl$format_line(0,0,anlrms$_fdlsystem);
2 anl$format_line(0,1,anlrms$_fdlsource);
2
2 ! Now call routines to put out the file and record primaries.
```

```
2
2 anl$format_skip(0);
2 anl$fdl_file();
2
2 anl$format_skip(0);
2 anl$fdl_record();
2
2 ! Now if this is an indexed file, call routines to put out the area
2 ! primaries, key primaries, analysis_of_area primaries, and
2 ! analysis_of_key primaries.
2
3 if .anl$gl_fat[fat$v_fileorg] eqlu fat$c_indexed then (
3
3         anl$fdl_areas();
3
3         anl$fdl_keys();
3
3         anl$analyze_areas();
3
3         anl$analyze_keys();
2 );
2
2 return;
2
1 end;


                                                  .TITLE  RMSFDL RMSFDL - Generate FDL for a File
                                                  .IDENT  \V04-000\

                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                         6F  6E  02  00000 P.AAA:  .ASCII  <2>\no\                                    ;
                     73  65  79  03  00003 P.AAB:  .ASCII  <3>\yes\                                   ;

                                                  .PSECT  $OWN$,NOEXE,2

             00000000' 00000000'  00000 YES_NO:  .ADDRESS P.AAA, P.AAB                               ;

                                                  .EXTRN  ANLRMS$_OK, ANLRMS$_ALLOC
                                                  .EXTRN  ANLRMS$_ANYTHING
                                                  .EXTRN  ANLRMS$_BACKUP, ANLRMS$_BKT
                                                  .EXTRN  ANLRMS$_BKTAREA
                                                  .EXTRN  ANLRMS$_BKTCHECK
                                                  .EXTRN  ANLRMS$_BKTFLAGS
                                                  .EXTRN  ANLRMS$_BKTFREE
                                                  .EXTRN  ANLRMS$_BKTKEY, ANLRMS$_BKTLEVEL
                                                  .EXTRN  ANLRMS$_BKTNEXT
                                                  .EXTRN  ANLRMS$_BKTPTRSIZE
                                                  .EXTRN  ANLRMS$_BKTRECID
                                                  .EXTRN  ANLRMS$_BKTRECID3
                                                  .EXTRN  ANLRMS$_BKTSAMPLE
                                                  .EXTRN  ANLRMS$_BKTVBNFREE
                                                  .EXTRN  ANLRMS$_BUCKETSIZE
                                                  .EXTRN  ANLRMS$_CELL, ANLRMS$_CELLDATA
                                                  .EXTRN  ANLRMS$_CELLFLAGS
                                                  .EXTRN  ANLRMS$_CHECKHDG
```

```
        .EXTRN  ANLRMS$_CONTIG, ANLRMS$_CREATION
        .EXTRN  ANLRMS$_CTLSIZE
        .EXTRN  ANLRMS$_DATAREC
        .EXTRN  ANLRMS$_DATABKTVBN
        .EXTRN  ANLRMS$_DUMPHEADING
        .EXTRN  ANLRMS$_EOF, ANLRMS$_ERRORCOUNT
        .EXTRN  ANLRMS$_ERRORNONE
        .EXTRN  ANLRMS$_ERRORS, ANLRMS$_EXPIRATION
        .EXTRN  ANLRMS$_FILEATTR
        .EXTRN  ANLRMS$_FILEHDR
        .EXTRN  ANLRMS$_FILEID, ANLRMS$_FILEORG
        .EXTRN  ANLRMS$_FILESPEC
        .EXTRN  ANLRMS$_FLAG, ANLRMS$_GLOBALBUFS
        .EXTRN  ANLRMS$_HEXDATA
        .EXTRN  ANLRMS$_HEXHEADING1
        .EXTRN  ANLRMS$_HEXHEADING2
        .EXTRN  ANLRMS$_IDXAREA
        .EXTRN  ANLRMS$_IDXAREAALLOC
        .EXTRN  ANLRMS$_IDXAREABKTSZ
        .EXTRN  ANLRMS$_IDXAREANEXT
        .EXTRN  ANLRMS$_IDXAREANOALLOC
        .EXTRN  ANLRMS$_IDXAREAQTY
        .EXTRN  ANLRMS$_IDXAREARECL
        .EXTRN  ANLRMS$_IDXAREAUSED
        .EXTRN  ANLRMS$_IDXKEY, ANLRMS$_IDXKEYAREAS
        .EXTRN  ANLRMS$_IDXKEYBKTSZ
        .EXTRN  ANLRMS$_IDXKEYBYTES
        .EXTRN  ANLRMS$_IDXKEY1TYPE
        .EXTRN  ANLRMS$_IDXKEYDATAVBN
        .EXTRN  ANLRMS$_IDXKEYFILL
        .EXTRN  ANLRMS$_IDXKEYFLAGS
        .EXTRN  ANLRMS$_IDXKEYKEYSZ
        .EXTRN  ANLRMS$_IDXKEYNAME
        .EXTRN  ANLRMS$_IDXKEYNEXT
        .EXTRN  ANLRMS$_IDXKEYMINREC
        .EXTRN  ANLRMS$_IDXKEYNULL
        .EXTRN  ANLRMS$_IDXKEYPOSS
        .EXTRN  ANLRMS$_IDXKEYROOTLVL
        .EXTRN  ANLRMS$_IDXKEYROOTVBN
        .EXTRN  ANLRMS$_IDXKEYSEGS
        .EXTRN  ANLRMS$_IDXKEYSIZES
        .EXTRN  ANLRMS$_IDXPRIMREC
        .EXTRN  ANLRMS$_IDXPRIMRECFLAGS
        .EXTRN  ANLRMS$_IDXPRIMRECID
        .EXTRN  ANLRMS$_IDXPRIMRECLEN
        .EXTRN  ANLRMS$_IDXPRIMRECRRV
        .EXTRN  ANLRMS$_IDXPROAREAS
        .EXTRN  ANLRMS$_IDXPROLOG
        .EXTRN  ANLRMS$_IDXREC, ANLRMS$_IDXRECPTR
        .EXTRN  ANLRMS$_IDXSIDR
        .EXTRN  ANLRMS$_IDXSIDRDUPCNT
        .EXTRN  ANLRMS$_IDXSIDRFLAGS
        .EXTRN  ANLRMS$_IDXSIDRRECID
        .EXTRN  ANLRMS$_IDXSIDRPTRFLAGS
        .EXTRN  ANLRMS$_IDXSIDRPTRREF
        .EXTRN  ANLRMS$_INTERCOMMAND
        .EXTRN  ANLRMS$_INTERHDG
        .EXTRN  ANLRMS$_LONGREC
        .EXTRN  ANLRMS$_MAXRECSIZE
        .EXTRN  ANLRMS$_NOBACKUP
        .EXTRN  ANLRMS$_NOEXPIRATION
        .EXTRN  ANLRMS$_NOSPANFILLER
        .EXTRN  ANLRMS$_PERFORM
        .EXTRN  ANLRMS$_PROLOGFLAGS
        .EXTRN  ANLRMS$_PROLOGVER
        .EXTRN  ANLRMS$_PROT, ANLRMS$_RECATTR
        .EXTRN  ANLRMS$_RECFMT, ANLRMS$_RECLAIMBKT
        .EXTRN  ANLRMS$_RELBUCKET
        .EXTRN  ANLRMS$_RELEOFVBN
        .EXTRN  ANLRMS$_RELMAXREC
        .EXTRN  ANLRMS$_RELPROLOG
        .EXTRN  ANLRMS$_RELIAB, ANLRMS$_REVISION
        .EXTRN  ANLRMS$_STATHDG
        .EXTRN  ANLRMS$_SUMMARYHDG
        .EXTRN  ANLRMS$_OWNERUIC
        .EXTRN  ANLRMS$_JNL, ANLRMS$_AIJNL
        .EXTRN  ANLRMS$_BIJNL, ANLRMS$_ATJNL
        .EXTRN  ANLRMS$_ATTOP, ANLRMS$_BADCMD
        .EXTRN  ANLRMS$_BADPATH
        .EXTRN  ANLRMS$_BADVBN, ANLRMS$_DOWNHELP
        .EXTRN  ANLRMS$_DOWNPATH
        .EXTRN  ANLRMS$_EMPTYBKT
        .EXTRN  ANLRMS$_NODATA, ANLRMS$_NODOWN
        .EXTRN  ANLRMS$_NONEXT, ANLRMS$_NORECLAIMED
        .EXTRN  ANLRMS$_NORECS, ANLRMS$_NORRV
        .EXTRN  ANLRMS$_RESTDONE
        .EXTRN  ANLRMS$_STACKFULL
        .EXTRN  ANLRMS$_UNINITINDEX
        .EXTRN  ANLRMS$_FDLIDENT
        .EXTRN  ANLRMS$_FDLSYSTEM
        .EXTRN  ANLRMS$_FDLSOURCE
        .EXTRN  ANLRMS$_FDLFILE
        .EXTRN  ANLRMS$_FDLALLOC
        .EXTRN  ANLRMS$_FDLNOALLOC
        .EXTRN  ANLRMS$_FDLBESTTRY
        .EXTRN  ANLRMS$_FDLBUCKETSIZE
        .EXTRN  ANLRMS$_FDLCLUSTERSIZE
        .EXTRN  ANLRMS$_FDLCONTIG
        .EXTRN  ANLRMS$_FDLEXTENSION
        .EXTRN  ANLRMS$_FDLGLOBALBUFS
        .EXTRN  ANLRMS$_FDLMAXRECORD
        .EXTRN  ANLRMS$_FDLFILENAME
        .EXTRN  ANLRMS$_FDLORG, ANLRMS$_FDLOWNER
        .EXTRN  ANLRMS$_FDLPROTECTION
        .EXTRN  ANLRMS$_FDLRECORD
        .EXTRN  ANLRMS$_FDLSPAN
        .EXTRN  ANLRMS$_FDLCC, ANLRMS$_FDLVFCSIZE
        .EXTRN  ANLRMS$_FDLFORMAT
        .EXTRN  ANLRMS$_FDLSIZE
        .EXTRN  ANLRMS$_FDLAREA
        .EXTRN  ANLRMS$_FDLKEY, ANLRMS$_FDLCHANGES
        .EXTRN  ANLRMS$_FDLDATAAREA
        .EXTRN  ANLRMS$_FDLDATAFILL
        .EXTRN  ANLRMS$_FDLDATAKEYCOMPB
        .EXTRN  ANLRMS$_FDLDATARECCOMPB
        .EXTRN  ANLRMS$_FDLDUPS
        .EXTRN  ANLRMS$_FDLINDEXAREA
        .EXTRN  ANLRMS$_FDLINDEXCOMPB
        .EXTRN  ANLRMS$_FDLINDEXFILL
        .EXTRN  ANLRMS$_FDLL1INDEXAREA
        .EXTRN  ANLRMS$_FDLKEYNAME
        .EXTRN  ANLRMS$_FDLNORECS
        .EXTRN  ANLRMS$_FDLNULLKEY
        .EXTRN  ANLRMS$_FDLNULLVALUE
        .EXTRN  ANLRMS$_FDLPROLOG
        .EXTRN  ANLRMS$_FDLSEGLENGTH
        .EXTRN  ANLRMS$_FDLSEGPOS
        .EXTRN  ANLRMS$_FDLSEGTYPE
        .EXTRN  ANLRMS$_FDLANALAREA
        .EXTRN  ANLRMS$_FDLRECL
        .EXTRN  ANLRMS$_FDLANALKEY
        .EXTRN  ANLRMS$_FDLDATAKEYCOMP
        .EXTRN  ANLRMS$_FDLDATARECCOMP
        .EXTRN  ANLRMS$_FDLDATARECS
        .EXTRN  ANLRMS$_FDLDATASPACE
        .EXTRN  ANLRMS$_FDLDEPTH
        .EXTRN  ANLRMS$_FDLDUPSPER
        .EXTRN  ANLRMS$_FDLIDXCOMP
        .EXTRN  ANLRMS$_FDLIDXFILL
        .EXTRN  ANLRMS$_FDLIDXSPACE
        .EXTRN  ANLRMS$_FDLIDXL1RECS
        .EXTRN  ANLRMS$_FDLDATALENMEAN
        .EXTRN  ANLRMS$_FDLIDXLENMEAN
        .EXTRN  ANLRMS$_STATAREA
        .EXTRN  ANLRMS$_STATRECL
        .EXTRN  ANLRMS$_STATKEY
        .EXTRN  ANLRMS$_STATDEPTH
        .EXTRN  ANLRMS$_STATIDXL1RECS
        .EXTRN  ANLRMS$_STATIDXLENMEAN
        .EXTRN  ANLRMS$_STATIDXSPACE
        .EXTRN  ANLRMS$_STATIDXFILL
        .EXTRN  ANLRMS$_STATIDXCOMP
        .EXTRN  ANLRMS$_STATDATARECS
        .EXTRN  ANLRMS$_STATDUPSPER
        .EXTRN  ANLRMS$_STATDATALENMEAN
        .EXTRN  ANLRMS$_STATDATASPACE
        .EXTRN  ANLRMS$_STATDATAFILL
        .EXTRN  ANLRMS$_STATDATAKEYCOMP
        .EXTRN  ANLRMS$_STATDATARECCOMP
        .EXTRN  ANLRMS$_STATEFFICIENCY
        .EXTRN  ANLRMS$_BADAREA1ST2
        .EXTRN  ANLRMS$_BADAREABKTSIZE
        .EXTRN  ANLRMS$_BADAREAFIT
        .EXTRN  ANLRMS$_BADAREAID
        .EXTRN  ANLRMS$_BADAREANEXT
        .EXTRN  ANLRMS$_BADAREAROOT
        .EXTRN  ANLRMS$_BADAREAUSED
        .EXTRN  ANLRMS$_BADBKTAREAID
        .EXTRN  ANLRMS$_BADBKTCHECK
        .EXTRN  ANLRMS$_BADBKTFREE
        .EXTRN  ANLRMS$_BADBKTKEYID
        .EXTRN  ANLRMS$_BADBKTLEVEL
        .EXTRN  ANLRMS$_BADBKTROOTBIT
        .EXTRN  ANLRMS$_BADBKTSAMPLE
        .EXTRN  ANLRMS$_BADCELLFIT
        .EXTRN  ANLRMS$_BADCHECKSUM
        .EXTRN  ANLRMS$_BADDATARECBITS
        .EXTRN  ANLRMS$_BADDATARECFIT
        .EXTRN  ANLRMS$_BADDATARECPS
        .EXTRN  ANLRMS$_BAD3IDXKEYFIT
        .EXTRN  ANLRMS$_BADIDXLASTKEY
        .EXTRN  ANLRMS$_BADIDXORDER
        .EXTRN  ANLRMS$_BADIDXRECBITS
        .EXTRN  ANLRMS$_BADIDXRECFIT
        .EXTRN  ANLRMS$_BADIDXRECPS
        .EXTRN  ANLRMS$_BADKEYAREAID
        .EXTRN  ANLRMS$_BADKEYDATABKT
        .EXTRN  ANLRMS$_BADKEYDATAFIT
        .EXTRN  ANLRMS$_BADKEYDATATYPE
        .EXTRN  ANLRMS$_BADKEYIDXBKT
        .EXTRN  ANLRMS$_BADKEYFILL
        .EXTRN  ANLRMS$_BADKEYFIT
        .EXTRN  ANLRMS$_BADKEYREFID
        .EXTRN  ANLRMS$_BADKEYROOTLEVEL
        .EXTRN  ANLRMS$_BADKEYSEGCOUNT
        .EXTRN  ANLRMS$_BADKEYSEGVEC
        .EXTRN  ANLRMS$_BADKEYSUMMARY
        .EXTRN  ANLRMS$_BADREADNOPAR
        .EXTRN  ANLRMS$_BADREADPAR
        .EXTRN  ANLRMS$_BADSIDRDUPCT
        .EXTRN  ANLRMS$_BADSIDRPTRFIT
        .EXTRN  ANLRMS$_BADSIDRPTRSZ
        .EXTRN  ANLRMS$_BADSIDRSIZE
        .EXTRN  ANLRMS$_BADSTREAMEOF
        .EXTRN  ANLRMS$_BADVBNFREE
        .EXTRN  ANLRMS$_BKTLOOP
        .EXTRN  ANLRMS$_EXTENDERR
        .EXTRN  ANLRMS$_FLAGERROR
        .EXTRN  ANLRMS$_MISSINGBKT
        .EXTRN  ANLRMS$_NOTOK, ANLRMS$_SPANERROR
        .EXTRN  ANLRMS$_TOOMANYRECS
        .EXTRN  ANLRMS$_UNWIND, ANLRMS$_VFCTOOSHORT
        .EXTRN  ANLRMS$_CACHEFULL
        .EXTRN  ANLRMS$_CACHERELFAIL
        .EXTRN  ANLRMS$_FACILITY
        .EXTRN  ANL$AREA_DESCRIPTOR
        .EXTRN  ANL$BUCKET, ANL$FDL_ANALYSIS_OF_AREA
        .EXTRN  ANL$FDL_ANALYSIS_OF_KEY
        .EXTRN  ANL$FDL_FILE, ANL$FORMAT_LINE
        .EXTRN  ANL$FORMAT_SKIP
        .EXTRN  ANL$IDX_CHECK_KEY_STUFF
        .EXTRN  ANL$KEY_DESCRIPTOR
        .EXTRN  ANL$OPEN_NEXT_RMS_FILE
        .EXTRN  ANL$PREPARE_QUOTED_STRING
        .EXTRN  ANL$PREPARE_REPORT_FILE
        .EXTRN  ANL$UNWIND_HANDLER
        .EXTRN  ANL$3RECLAIMED_BUCKET_HEADER
        .EXTRN  CLI$GET_VALUE, LIB$ESTABLISH
```

```
                                                              .EXTRN  ANL$GL_FAT, ANL$GW_PROLOG

                                                              .PSECT  $CODE$,NOWRT,2

                                           000C 00000         .ENTRY  ANL$FDL_MODE, Save R2,R3                    ; 0664
                         53     0000G   CF   9E 00002         MOVAB   ANL$FORMAT_SKIP, R3                         ;
                         52     0000G   CF   9E 00007         MOVAB   ANL$FORMAT_LINE, R2                         ;
                         5E      FEFC   CE   9E 0000C         MOVAB   -260(SP), SP                                ;
                         7E        FF   8F   9A 00011         MOVZBL  #255, RESULTANT_FILE_SPEC                   ; 0669
                   04    AE        08   AE   9E 00015         MOVAB   RESULTANT_FILE_SPEC+8, -                    ;
                                                                      RESULTANT_FILE_SPEC+4                       ;
                                0000G   CF   9F 0001A         PUSHAB  ANL$UNWIND_HANDLER                          ; 0674
            00000000G    00             01   FB 0001E         CALLS   #1, LIB$ESTABLISH                           ;
                                        5E   DD 00025         PUSHL   SP                                          ; 0678
                0000G    CF             01   FB 00027         CALLS   #1, ANL$OPEN_NEXT_RMS_FILE                  ;
                         64             50   E9 0002C         BLBC    R0, 1$                                      ;
                                        5E   DD 0002F         PUSHL   SP                                          ; 0684
                                        7E   D4 00031         CLRL    -(SP)                                       ;
                0000G    CF             02   FB 00033         CALLS   #2, ANL$PREPARE_REPORT_FILE                 ;
                                        7E   D4 00038         CLRL    -(SP)                                       ; 0688
                            00000000G   8F   DD 0003A         PUSHL   #ANLRMS$_FDLIDENT                           ;
                                        7E   7C 00040         CLRQ    -(SP)                                       ;
                         62             04   FB 00042         CALLS   #4, ANL$FORMAT_LINE                         ;
                                        7E   D4 00045         CLRL    -(SP)                                       ; 0692
                         63             01   FB 00047         CALLS   #1, ANL$FORMAT_SKIP                         ;
                            00000000G   8F   DD 0004A         PUSHL   #ANLRMS$_FDLSYSTEM                          ; 0693
                                        7E   7C 00050         CLRQ    -(SP)                                       ;
                         62             03   FB 00052         CALLS   #3, ANL$FORMAT_LINE                         ;
                            00000000G   8F   DD 00055         PUSHL   #ANLRMS$_FDLSOURCE                          ; 0694
                                        01   DD 0005B         PUSHL   #1                                          ;
                                        7E   D4 0005D         CLRL    -(SP)                                       ;
                         62             03   FB 0005F         CALLS   #3, ANL$FORMAT_LINE                         ;
                                        7E   D4 00062         CLRL    -(SP)                                       ; 0698
                         63             01   FB 00064         CALLS   #1, ANL$FORMAT_SKIP                         ;
                0000G    CF             00   FB 00067         CALLS   #0, ANL$FDL_FILE                            ; 0699
                                        7E   D4 0006C         CLRL    -(SP)                                       ; 0701
                         63             01   FB 0006E         CALLS   #1, ANL$FORMAT_SKIP                         ;
                0000V    CF             00   FB 00071         CALLS   #0, ANL$FDL_RECORD                          ; 0702
      02      0000G  DF                 04   04   ED 00076    CMPZV   #4, #4, @ANL$GL_FAT, #2                     ; 0708
                                        14   12 0007D         BNEQ    1$                                          ;
                0000V    CF             00   FB 0007F         CALLS   #0, ANL$FDL_AREAS                           ; 0710
                0000V    CF             00   FB 00084         CALLS   #0, ANL$FDL_KEYS                            ; 0712
                0000V    CF             00   FB 00089         CALLS   #0, ANL$ANALYZE_AREAS                       ; 0714
                0000V    CF             00   FB 0008E         CALLS   #0, ANL$ANALYZE_KEYS                        ; 0716
                                             04 00093 1$:     RET                                                 ; 0721

; Routine Size:  148 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %sbttl 'ANL$FDL_RECORD - Generate RECORD primary for FDL'
1 !++
1 ! Functional Description:
1 !       This routine is responsible for generating the RECORD primary in an
1 !       FDL spec.  This primary describes things about the record format
1 !       of the file.
1 !
1 ! Formal Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_record: novalue = begin
2
2
2 ! We just format a line for each item in the record primary.
2
2 anl$format_line(0,0,anlrms$_fdlrecord);
2 anl$format_line(0,1,anlrms$_fdlspan,.yes_no[not .anl$gl_fat[fat$v_nospan] and 1]);
2 anl$format_line(0,1,anlrms$_fdlcc,
3                 (if .anl$gl_fat[fat$v_impliedcc] then uplit byte (%ascic 'carriage_return')
3             else if .anl$gl_fat[fat$v_fortrancc] then uplit byte (%ascic 'fortran')
3             else if .anl$gl_fat[fat$v_printcc]   then uplit byte (%ascic 'print')
2             else                                      uplit byte (%ascic 'none')));
2 if .anl$gl_fat[fat$v_rtype] eqlu fat$c_vfc then
2       anl$format_line(0,1,anlrms$_fdlvfcsize,.anl$gl_fat[fat$b_vfcsize]);
2 anl$format_line(0,1,anlrms$_fdlformat,
3                 (selectoneu .anl$gl_fat[fat$v_rtype] of set
3                 [fat$c_undefined]:      uplit byte (%ascic 'undefined');
3                 [fat$c_fixed]:          uplit byte (%ascic 'fixed');
3                 [fat$c_variable]:       uplit byte (%ascic 'variable');
3                 [fat$c_vfc]:            uplit byte (%ascic 'vfc');
3                 [fat$c_stream]:         uplit byte (%ascic 'stream');
3                 [fat$c_streamlf]:       uplit byte (%ascic 'stream_lf');
3                 [fat$c_streamcr]:       uplit byte (%ascic 'stream_cr');
2                 tes));
2 anl$format_line(0,1,anlrms$_fdlsize,.anl$gl_fat[fat$w_maxrec]);
2
2 return;
2
1 end;


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

```
72  75  74  65  72  5F  65  67  61  69  72  72  61  63  0F   00007 P.AAC:    .ASCII   <15>\carriage_return\                     ;
                                                            6E   00016                                                          ;
                        6E  61  72  74  72  6F  66  07   00017 P.AAD:    .ASCII   <7>\fortran\                              ;
                                    74  6E  69  72  70  05   0001F P.AAE:    .ASCII   <5>\print\                                ;
                                        65  6E  6F  6E  04   00025 P.AAF:    .ASCII   <4>\none\                                 ;
                64  65  6E  69  66  65  64  6E  75  09   0002A P.AAG:    .ASCII   <9>\undefined\                            ;
                                    64  65  78  69  66  05   00034 P.AAH:    .ASCII   <5>\fixed\                                ;
                    65  6C  62  61  69  72  61  76  08   0003A P.AAI:    .ASCII   <8>\variable\                             ;
                                            63  66  76  03   00043 P.AAJ:    .ASCII   <3>\vfc\                                  ;
                            6D  61  65  72  74  73  06   00047 P.AAK:    .ASCII   <6>\stream\                               ;
                66  6C  5F  6D  61  65  72  74  73  09   0004E P.AAL:    .ASCII   <9>\stream_lf\                            ;
                72  63  5F  6D  61  65  72  74  73  09   00058 P.AAM:    .ASCII   <9>\stream_cr\                            ;


                                                                          .PSECT   $CODE$,NOWRT,2

                                                     001C  00000            .ENTRY   ANL$FDL_RECORD, Save R2,R3,R4             ; 0746
                                  54        0000G  CF  9E  00002            MOVAB    ANL$GL_FAT, R4                            ;
                                  53        0000G  CF  9E  00007            MOVAB    ANL$FORMAT_LINE, R3                       ;
                                  52        0000'  CF  9E  0000C            MOVAB    P.AAC, R2                                 ;
                               00000000G           8F  DD  00011            PUSHL    #ANLRMS$_FDLRECORD                        ; 0751
                                                   7E  7C  00017            CLRQ     -(SP)                                     ;
                                  63               03  FB  00019            CALLS    #3, ANL$FORMAT_LINE                       ;
                                  50               64  D0  0001C            MOVL     ANL$GL_FAT, R0                            ; 0752
        50          01      A0    01               03  EF  0001F            EXTZV    #3, #1, 1(R0), R0                         ;
                            50    01               50  CB  00025            BICL3    R0, #1, R0                                ;
                                        0000'CF40      DD  00029            PUSHL    YES_NO[R0]                                ;
                               00000000G           8F  DD  0002E            PUSHL    #ANLRMS$_FDLSPAN                          ;
                                                   01  DD  00034            PUSHL    #1                                        ;
                                                   7E  D4  00036            CLRL     -(SP)                                     ;
                                  63               04  FB  00038            CALLS    #4, ANL$FORMAT_LINE                       ;
                                  50               64  D0  0003B            MOVL     ANL$GL_FAT, R0                            ; 0754
                    07          01  A0               01  E1  0003E            BBC      #1, 1(R0), 1$                             ;
                                  51               62  9E  00043            MOVAB    P.AAC, R1                                 ;
                                                   51  DD  00046            PUSHL    R1                                        ;
                                                   1E  11  00048            BRB      5$                                        ;
                                  09        01     A0  E9  0004A 1$:        BLBC     1(R0), 2$                                 ; 0755
                                  51        10     A2  9E  0004E            MOVAB    P.AAD, R1                                 ;
                                  50               51  D0  00052            MOVL     R1, R0                                    ;
                                                   0F  11  00055            BRB      4$                                        ;
                    06          01  A0               02  E1  00057 2$:        BBC      #2, 1(R0), 3$                             ; 0756
                                  50        18     A2  9E  0005C            MOVAB    P.AAE, R0                                 ;
                                                   04  11  00060            BRB      4$                                        ;
                                  50        1E     A2  9E  00062 3$:        MOVAB    P.AAF, R0                                 ; 0757
                                                   50  DD  00066 4$:        PUSHL    R0                                        ; 0755
                               00000000G           8F  DD  00068 5$:        PUSHL    #ANLRMS$_FDLCC                            ; 0753
                                                   01  DD  0006E            PUSHL    #1                                        ;
                                                   7E  D4  00070            CLRL     -(SP)                                     ;
                                  63               04  FB  00072            CALLS    #4, ANL$FORMAT_LINE                       ;
                                  50               64  D0  00075            MOVL     ANL$GL_FAT, R0                            ; 0758
        03                  60    04               00  ED  00078            CMPZV    #0, #4, (R0), #3                          ;
                                                   11  12  0007D            BNEQ     6$                                        ;
                                  7E        0F     A0  9A  0007F            MOVZBL   15(R0), -(SP)                             ; 0759
                               00000000G           8F  DD  00083            PUSHL    #ANLRMS$_FDLVFCSIZE                       ;
                                                   01  DD  00089            PUSHL    #1                                        ;
```

```
                                                  7E  D4 0008B        CLRL    -(SP)                                         :
                                       63         04  FB 0008D        CALLS   #4, ANL$FORMAT_LINE                           :
        51       00   B4               04         00  EF 00090  6$:   EXTZV   #0, #4, @ANL$GL_FAT, R1                       : 0761
                                                  06  12 00096        BNEQ    7$                                            : 0762
                                       50    23   A2  9E 00098        MOVAB   P.AAG, R0                                     :
                                                  45  11 0009C        BRB     14$                                           :
                                       01         51  D1 0009E  7$:   CMPL    R1, #1                                        : 0763
                                                  06  12 000A1        BNEQ    8$                                            :
                                       50    2D   A2  9E 000A3        MOVAB   P.AAH, R0                                     :
                                                  3A  11 000A7        BRB     14$                                           :
                                       02         51  D1 000A9  8$:   CMPL    R1, #2                                        : 0764
                                                  06  12 000AC        BNEQ    9$                                            :
                                       50    33   A2  9E 000AE        MOVAB   P.AAI, R0                                     :
                                                  2F  11 000B2        BRB     14$                                           :
                                       03         51  D1 000B4  9$:   CMPL    R1, #3                                        : 0765
                                                  06  12 000B7        BNEQ    10$                                           :
                                       50    3C   A2  9E 000B9        MOVAB   P.AAJ, R0                                     :
                                                  24  11 000BD        BRB     14$                                           :
                                       04         51  D1 000BF  10$:  CMPL    R1, #4                                        : 0766
                                                  06  12 000C2        BNEQ    11$                                           :
                                       50    40   A2  9E 000C4        MOVAB   P.AAK, R0                                     :
                                                  19  11 000C8        BRB     14$                                           :
                                       05         51  D1 000CA  11$:  CMPL    R1, #5                                        : 0767
                                                  06  12 000CD        BNEQ    12$                                           :
                                       50    47   A2  9E 000CF        MOVAB   P.AAL, R0                                     :
                                                  0E  11 000D3        BRB     14$                                           :
                                       06         51  D1 000D5  12$:  CMPL    R1, #6                                        : 0768
                                                  05  13 000D8        BEQL    13$                                           :
                                       7E         01  CE 000DA        MNEGL   #1, -(SP)                                     :
                                                  06  11 000DD        BRB     15$                                           :
                                       50    51   A2  9E 000DF  13$:  MOVAB   P.AAM, R0                                     :
                                                  50  DD 000E3  14$:  PUSHL   R0                                            :
                                         00000000G 8F  DD 000E5  15$:  PUSHL   #ANLRMS$_FDLFORMAT                            : 0760
                                                  01  DD 000EB        PUSHL   #1                                            :
                                                  7E  D4 000ED        CLRL    -(SP)                                         :
                                       63         04  FB 000EF        CALLS   #4, ANL$FORMAT_LINE                           :
                                       50         64  D0 000F2        MOVL    ANL$GL_FAT, R0                                : 0770
                                       7E    10   A0  3C 000F5        MOVZWL  16(R0), -(SP)                                 :
                                         00000000G 8F  DD 000F9        PUSHL   #ANLRMS$_FDLSIZE                              :
                                                  01  DD 000FF        PUSHL   #1                                            :
                                                  7E  D4 00101        CLRL    -(SP)                                         :
                                       63         04  FB 00103        CALLS   #4, ANL$FORMAT_LINE                           :
                                                      04 00106        RET                                                   : 0774

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0094
```

```
1 %sbttl 'ANL$FDL_AREAS - Generate AREA Primaries for FDL'
1 !++
1 ! Functional Description:
1 !       This routine is responsible for generating the area primaries in
1 !       an FDL spec.  This is needed for defining indexed files.
1 !
1 ! Formal Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_areas: novalue = begin
2
2 local
2               p: bsd,
2               sp: ref block[,byte],
2               area_count: long,
2               id: long;
2
2
2 ! We begin by setting up a BSD for the prolog and reading it in.
2
2 init_bsd(p);
2 p[bsd$w_size] = 1;
2 p[bsd$l_vbn] = 1;
2 anl$bucket(p,0);
2
2 ! Now we will scan all of the area descriptors.  Read in the first one.
2
2 sp = .p[bsd$l_bufptr];
2 area_count = .sp[plg$b_amax];
2
2 p[bsd$l_vbn] = .sp[plg$b_avbn];
2 p[bsd$l_offset] = 0;
2 anl$bucket(p,0);
2
2 ! Loop through the descriptors one by one.
2
3 incru id from 0 to .area_count-1 do (
3
3               ! Generate the FDL for this descriptor.
3
3               sp = .p[bsd$l_bufptr] + .p[bsd$l_offset];
3
3               anl$format_skip(0);
;   328         0832  3             anl$format_line(0,0,anlrms$_fdlarea,.id);
;   329         0833  3
;   330         0834  3             ! If an extent has been allocated but the total allocation is zero,
;   331         0835  3             ! then this file was created before the total allocation field
;   332         0836  3             ! existed.  Just put out a zero allocation with a comment.
;   333         0837  3             ! Otherwise, we can put out the total area allocation.
;   334         0838  3
;   335         0839  3             if .sp[area$l_cvbn] nequ 0 and .sp[area$l_total_alloc] eqlu 0 then
;   336         0840  3                     anl$format_line(0,1,anlrms$_fdlnoalloc)
;   337         0841  3             else
;   338         0842  3                     anl$format_line(0,1,anlrms$_fdlalloc,.sp[area$l_total_alloc]);
;   339         0843  3
;   340         0844  3             anl$format_line(0,1,anlrms$_fdlbucketsize,.sp[area$b_arbktsz]);
;   341         0845  3             anl$format_line(0,1,anlrms$_fdlextension,.sp[area$w_deq]);
;   342         0846  3
;   343         0847  3             ! Now we can advance on to the next descriptor.  In the process,
;   344         0848  3             ! we will check it for validity.
;   345         0849  3
;   346         0850  3             anl$area_descriptor(p,.id,false);
;   347         0851  2 );
;   348         0852  2
;   349         0853  2 anl$bucket(p,-1);
;   350         0854  2 return;
;   351         0855  2
;   352         0856  1 end;



                                            00FC 00000           .ENTRY  ANL$FDL_AREAS, Save R2,R3,R4,R5,R6,R7      ; 0798
                       57       0000G  CF   9E 00002           MOVAB   ANL$BUCKET, R7                             ;
                       56       0000G  CF   9E 00007           MOVAB   ANL$FORMAT_LINE, R6                        ;
                       5E              18   C2 0000C           SUBL2   #24, SP                                    ;
     18          00    6E              00   2C 0000F           MOVC5   #0, (SP), #0, #24, P                       ; 0809
                                       6E      00014                                                              ;
                  02   AE              01   B0 00015           MOVW    #1, P+2                                    ; 0810
                  04   AE              01   D0 00019           MOVL    #1, P+4                                    ; 0811
                                       7E   D4 0001D           CLRL    -(SP)                                      ; 0812
                               04      AE   9F 0001F           PUSHAB  P                                          ;
                       67              02   FB 00022           CALLS   #2, ANL$BUCKET                             ;
                       53      0C      AE   D0 00025           MOVL    P+12, SP                                   ; 0816
                       52      67      A3   9A 00029           MOVZBL  103(SP), AREA_COUNT                        ; 0817
                  04   AE      66      A3   9A 0002D           MOVZBL  102(SP), P+4                               ; 0819
                               08      AE   D4 00032           CLRL    P+8                                        ; 0820
                                       7E   D4 00035           CLRL    -(SP)                                      ; 0821
                               04      AE   9F 00037           PUSHAB  P                                          ;
                       67              02   FB 0003A           CALLS   #2, ANL$BUCKET                             ;
                                       52   D7 0003D           DECL    R2                                         ; 0825
                                       54   D4 0003F           CLRL    ID                                         ;
                                       73   11 00041           BRB     4$                                         ;
            53    0C   AE      08      AE   C1 00043  1$:      ADDL3   P+8, P+12, SP                              ; 0829
                                       7E   D4 00049           CLRL    -(SP)                                      ; 0831
               0000G   CF              01   FB 0004B           CALLS   #1, ANL$FORMAT_SKIP                        ;
                                       54   DD 00050           PUSHL   ID                                         ; 0832
                         00000000G     8F   DD 00052           PUSHL   #ANLRMS$_FDLAREA                           ;
                                       7E   7C 00058           CLRQ    -(SP)                                      ;
```

```
                66            04  FB 0005A         CALLS   #4, ANL$FORMAT_LINE                      :
                       0C     A3  D5 0005D         TSTL    12(SP)                                   : 0839
                              14  13 00060         BEQL    2$                                       :
                       32     A3  D5 00062         TSTL    50(SP)                                   :
                              0F  12 00065         BNEQ    2$                                       :
            00000000G         8F  DD 00067         PUSHL   #ANLRMS$_FDLNOALLOC                      : 0840
                              01  DD 0006D         PUSHL   #1                                       :
                              7E  D4 0006F         CLRL    -(SP)                                    :
                66            03  FB 00071         CALLS   #3, ANL$FORMAT_LINE                      :
                              10  11 00074         BRB     3$                                       :
                       32     A3  DD 00076 2$:     PUSHL   50(SP)                                   : 0842
            00000000G         8F  DD 00079         PUSHL   #ANLRMS$_FDLALLOC                        :
                              01  DD 0007F         PUSHL   #1                                       :
                              7E  D4 00081         CLRL    -(SP)                                    :
                66            04  FB 00083         CALLS   #4, ANL$FORMAT_LINE                      :
                7E     03     A3  9A 00086 3$:     MOVZBL  3(SP), -(SP)                             : 0844
            00000000G         8F  DD 0008A         PUSHL   #ANLRMS$_FDLBUCKETSIZE                   :
                              01  DD 00090         PUSHL   #1                                       :
                              7E  D4 00092         CLRL    -(SP)                                    :
                66            04  FB 00094         CALLS   #4, ANL$FORMAT_LINE                      :
                7E     24     A3  3C 00097         MOVZWL  36(SP), -(SP)                            : 0845
            00000000G         8F  DD 0009B         PUSHL   #ANLRMS$_FDLEXTENSION                    :
                              01  DD 000A1         PUSHL   #1                                       :
                              7E  D4 000A3         CLRL    -(SP)                                    :
                66            04  FB 000A5         CALLS   #4, ANL$FORMAT_LINE                      :
                              7E  D4 000A8         CLRL    -(SP)                                    : 0850
                              54  DD 000AA         PUSHL   ID                                       :
                       08     AE  9F 000AC         PUSHAB  P                                        :
          0000G  CF           03  FB 000AF         CALLS   #3, ANL$AREA_DESCRIPTOR                  :
                              54  D6 000B4         INCL    ID                                       : 0825
                52            54  D1 000B6 4$:     CMPL    ID, R2                                   :
                              88  1B 000B9         BLEQU   1$                                       :
                7E            01  CE 000BB         MNEGL   #1, -(SP)                                : 0853
                       04     AE  9F 000BE         PUSHAB  P                                        :
                67            02  FB 000C1         CALLS   #2, ANL$BUCKET                           :
                              04  000C4            RET                                              : 0856

; Routine Size:  197 bytes,    Routine Base:  $CODE$ + 019B
```

```
1 %sbttl 'ANL$FDL_KEYS - Generate KEY Primaries for FDL'
1 !++
1 ! Functional Description:
1 !        This routine is responsible for generating the key primaries in an
1 !        FDL spec.  These are needed for indexed files.
1 !
1 ! Formal Parameters:
1 !        none
1 !
1 ! Implicit Inputs:
1 !        global data
1 !
1 ! Implicit Outputs:
1 !        global data
1 !
1 ! Returned Value:
1 !        none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_keys: novalue = begin
2
2 own
2         types: vector[8,long] initial(
2                         uplit byte (%ascic 'string'),
2                         uplit byte (%ascic 'int2'),
2                         uplit byte (%ascic 'bin2'),
2                         uplit byte (%ascic 'int4'),
2                         uplit byte (%ascic 'bin4'),
2                         uplit byte (%ascic 'decimal'),
2                         uplit byte (%ascic 'int8'),
2                         uplit byte (%ascic 'bin8')
2                         );
2 local
2         p: bsd,
2         id: long,
2         sp: ref block[,byte],
2         i: long;
2
2
2 ! We will be looking at all of the key descriptors.  Set up a BSD for the
2 ! first one.
2
2 init_bsd(p);
2 p[bsd$w_size] = 1;
2 p[bsd$l_vbn] = 1;
2 p[bsd$l_offset] = 0;
2 anl$bucket(p,0);
2
2 ! Now we can loop through the key descriptors.
2
3 incru id from 0 do (
3
3         ! Now we can format the FDL for the key.
```

```
3
3        sp = .p[bsd$l_bufptr] + .p[bsd$l_offset];
3
3        anl$format_skip(0);
3        anl$format_line(0,0,anlrms$_fdlkey,.id);
3        anl$format_line(0,1,anlrms$_fdlchanges,.yes_no[.sp[key$v_chgkeys] and 1]);
3
3        ! The data key and record compression flags are meaningful only for
3        ! a prologue 3 file.  Furthermore, the data record compression flag
3        ! only makes sense on the primary key.
3
4        if .anl$gw_prolog eqlu plg$c_ver_3 then (
4                anl$format_line(0,1,anlrms$_fdldatakeycompb,.yes_no[.sp[key$v_key_compr] and 1]);
4                if .id eqlu 0 then
4                        anl$format_line(0,1,anlrms$_fdldatareccompb,
4                                        .yes_no[.sp[key$v_rec_compr] and 1]);
3        );
3
3        anl$format_line(0,1,anlrms$_fdldataarea,.sp[key$b_danum]);
3        anl$format_line(0,1,anlrms$_fdldatafill,(.sp[key$w_datfill] * 100) /
3                                                (.sp[key$b_datbktsz]*512));
3        anl$format_line(0,1,anlrms$_fdldups,.yes_no[.sp[key$v_dupkeys] and 1]);
3        anl$format_line(0,1,anlrms$_fdlindexarea,.sp[key$b_ianum]);
3
3        ! The index compression flag is only used for prologue 3 files.
3
3        if .anl$gw_prolog eqlu plg$c_ver_3 then
3                anl$format_line(0,1,anlrms$_fdlindexcompb,.yes_no[.sp[key$v_idx_compr] and 1]);
3
3        anl$format_line(0,1,anlrms$_fdlindexfill,(.sp[key$w_idxfill] * 100) /
3                                                 (.sp[key$b_idxbktsz]*512));
3        anl$format_line(0,1,anlrms$_fdll1indexarea,.sp[key$b_lanum]);
3
3        ! For the key name, we have to produce a quoted string containing
3        ! the name.  This goes in the output line along with the NAME keyword.
3
4        begin
4        local
4                name_dsc: descriptor,
4                local_described_buffer(string_buf,key$s_keynam*2+2);
4
4        build_descriptor(name_dsc, key$s_keynam,sp[key$t_keynam]);
4        anl$prepare_quoted_string(name_dsc,string_buf);
4        anl$format_line(0,1,anlrms$_fdlkeyname,string_buf);
3        end;
3
3        anl$format_line(0,1,anlrms$_fdlnullkey,.yes_no[.sp[key$v_nulkeys] and 1]);
3        if .sp[key$v_nulkeys] then
3                anl$format_line(0,1,anlrms$_fdlnullvalue,.sp[key$b_nullchar]);
3
3        ! The prolog version only appears in the primary key.
3
3        if .id eqlu 0 then
3                anl$format_line(0,1,anlrms$_fdlprolog,.anl$gw_prolog);
3
3        ! To put out the segment sizes and positions, we have to loop
3        ! through the segment arrays.
```

```
;  468          0971 3
;  469          0972 4              begin
;  470          0973 4              bind
;  471          0974 4                      size_vector = sp[key$b_size0]: vector[,byte],
;  472          0975 4                      pos_vector = sp[key$w_position0]: vector[,word];
;  473          0976 4
;  474          0977 5              incru i from 0 to .sp[key$b_segments]-1 do (
;  475          0978 5                      anl$format_line(0,1,anlrms$_fdlseglength,.i,.size_vector[.i]);
;  476          0979 5                      anl$format_line(0,1,anlrms$_fdlsegpos,.i,.pos_vector[.i]);
;  477          0980 4              );
;  478          0981 3              end;
;  479          0982 3
;  480          0983 3              ! Now we can put out the key data type.
;  481          0984 3
;  482          0985 3              anl$format_line(0,1,anlrms$_fdlsegtype,.types[.sp[key$b_datatype]]);
;  483          0986 3
;  484          0987 3              ! Now we can go on to the next descriptor, if there is one.
;  485          0988 3              ! This will also check the descriptor's validity.
;  486          0989 3
;  487          0990 3      exitif (not anl$key_descriptor(p,.id,0,false));
;  488          0991 2      );
;  489          0992 2
;  490          0993 2      anl$bucket(p,-1);
;  491          0994 2      return;
;  492          0995 2
;  493          0996 1      end;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                        67  6E  69  72  74  73  06  00062 P.AAN:        .ASCII  <6>\string\                                    ;
                                32  74  6E  69  04  00069 P.AAO:        .ASCII  <4>\int2\                                      ;
                                32  6E  69  62  04  0006E P.AAP:        .ASCII  <4>\bin2\                                      ;
                                34  74  6E  69  04  00073 P.AAQ:        .ASCII  <4>\int4\                                      ;
                                34  6E  69  62  04  00078 P.AAR:        .ASCII  <4>\bin4\                                      ;
                    6C  61  6D  69  63  65  64  07  0007D P.AAS:        .ASCII  <7>\decimal\                                   ;
                                38  74  6E  69  04  00085 P.AAT:        .ASCII  <4>\int8\                                      ;
                                38  6E  69  62  04  0008A P.AAU:        .ASCII  <4>\bin8\                                      ;

                                                                        .PSECT  $OWN$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00008 TYPES:  .ADDRESS P.AAN, P.AAO, P.AAP, P.AAQ, P.AAR, -           ;
                                        00000000' 00000000' 00020                  P.AAS, P.AAT, P.AAU                          ;


                                                                        .PSECT  $CODE$,NOWRT,2

                                             01FC 00000                 .ENTRY  ANL$FDL_KEYS, Save R2,R3,R4,R5,R6,R7,R8         ; 0880
                          58        0000G  CF   9E 00002                MOVAB   ANL$GW_PROLOG, R8                                ;
                          57        0000'  CF   9E 00007                MOVAB   YES_NO, R7                                       ;
                          56        0000G  CF   9E 0000C                MOVAB   ANL$FORMAT_LINE, R6                              ;
                          5E           94  AE   9E 00011                MOVAB   -108(SP), SP                                     ;
          18          00  6E               00   2C 00015                MOVC5   #0, (SP), #0, #24, P                             ; 0903
                                       54  AE      0001A                                                                        ;
                  56  AE                   01   B0 0001C                MOVW    #1, P+2                                          ; 0904
```

```
                                                            B  8
RMSFDL          RMSFDL - Generate FDL for a File            16-Sep-1984 00:02:41    VAX-11 Bliss-32 V4.0-742          Page  20
V04-000         ANL$FDL_KEYS - Generate KEY Primaries for FDL   14-Sep-1984 11:53:00    [ANALYZ.SRC]RMSFDL.B32;1               (6)

                        58    AE                   01  7D 00020         MOVQ     #1, P+4                          ; 0905
                                                   7E  D4 00024         CLRL     -(SP)                            ; 0907
                                           58      AE  9F 00026         PUSHAB   P                                ;
                     0000G    CF                   02  FB 00029         CALLS    #2, ANL$BUCKET                   ;
                                                   55  D4 0002E         CLRL     ID                               ; 0911
              52      60      AE           5C      AE  C1 00030  1$:    ADDL3    P+8, P+12, SP                    ; 0915
                                                   7E  D4 00036         CLRL     -(SP)                            ; 0917
                     0000G    CF                   01  FB 00038         CALLS    #1, ANL$FORMAT_SKIP              ;
                                                   55  DD 0003D         PUSHL    ID                               ; 0918
                                    00000000G      8F  DD 0003F         PUSHL    #ANLRMS$_FDLKEY                  ;
                                                   7E  7C 00045         CLRQ     -(SP)                            ;
                              66                   04  FB 00047         CALLS    #4, ANL$FORMAT_LINE              ;
                              53           10      A2  9E 0004A         MOVAB    16(SP), R3                       ; 0919
    50        63              01                   01  EF 0004E         EXTZV    #1, #1, (R3), R0                 ;
                                                 6740  DD 00053         PUSHL    YES_NO[R0]                       ;
                                    00000000G      8F  DD 00056         PUSHL    #AN[RMS$_FDLCHANGES              ;
                                                   01  DD 0005C         PUSHL    #1                               ;
                                                   7E  D4 0005E         CLRL     -(SP)                            ;
                              66                   04  FB 00060         CALLS    #4, ANL$FORMAT_LINE              ;
                              03                   68  B1 00063         CMPW     ANL$GW_PROLOG, #3                ; 0925
                                                   2E  12 00066         BNEQ     2$                               ;
    50        63              01                   06  EF 00068         EXTZV    #6, #1, (R3), R0                 ; 0926
                                                 6740  DD 0006D         PUSHL    YES_NO[R0]                       ;
                                    00000000G      8F  DD 00070         PUSHL    #AN[RMS$_FDLDATAKEYCOMPB         ;
                                                   01  DD 00076         PUSHL    #1                               ;
                                                   7E  D4 00078         CLRL     -(SP)                            ;
                              66                   04  FB 0007A         CALLS    #4, ANL$FORMAT_LINE              ;
                                                   55  D5 0007D         TSTL     ID                               ; 0927
                                                   15  12 0007F         BNEQ     2$                               ;
    50        63              01                   07  EF 00081         EXTZV    #7, #1, (R3), R0                 ; 0929
                                                 6740  DD 00086         PUSHL    YES_NO[R0]                       ;
                                    00000000G      8F  DD 00089         PUSHL    #AN[RMS$_FDLDATARECCOMPB         ; 0928
                                                   01  DD 0008F         PUSHL    #1                               ;
                                                   7E  D4 00091         CLRL     -(SP)                            ;
                              66                   04  FB 00093         CALLS    #4, ANL$FORMAT_LINE              ;
                              7E           08      A2  9A 00096  2$:    MOVZBL   8(SP), -(SP)                     ; 0932
                                    00000000G      8F  DD 0009A         PUSHL    #ANLRMS$_FDLDATAAREA             ;
                                                   01  DD 000A0         PUSHL    #1                               ;
                                                   7E  D4 000A2         CLRL     -(SP)                            ;
                              66                   04  FB 000A4         CALLS    #4, ANL$FORMAT_LINE              ;
                              51           1A      A2  3C 000A7         MOVZWL   26(SP), R1                       ; 0933
                              51     00000064      8F  C4 000AB         MULL2    #100, R1                         ;
                              50           0B      A2  9A 000B2         MOVZBL   11(SP), R0                       ; 0934
              50              50                   09  78 000B6         ASHL     #9, R0, R0                       ;
              7E              51                   50  C7 000BA         DIVL3    R0, R1, -(SP)                    ;
                                    00000000G      8F  DD 000BE         PUSHL    #ANLRMS$_FDLDATAFILL             ; 0933
                                                   01  DD 000C4         PUSHL    #1                               ;
                                                   7E  D4 000C6         CLRL     -(SP)                            ;
                              66                   04  FB 000C8         CALLS    #4, ANL$FORMAT_LINE              ;
    50        63              01                   00  EF 000CB         EXTZV    #0, #1, (R3), R0                 ; 0935
                                                 6740  DD 000D0         PUSHL    YES_NO[R0]                       ;
                                    00000000G      8F  DD 000D3         PUSHL    #AN[RMS$_FDLDUPS                 ;
                                                   01  DD 000D9         PUSHL    #1                               ;
                                                   7E  D4 000DB         CLRL     -(SP)                            ;
                              66                   04  FB 000DD         CALLS    #4, ANL$FORMAT_LINE              ;
                              7E           06      A2  9A 000E0         MOVZBL   6(SP), -(SP)                     ; 0936
                                    00000000G      8F  DD 000E4         PUSHL    #ANLRMS$_FDLINDEXAREA            ;
```

```
                                                      01 DD 000EA       PUSHL   #1                                      ;
                                                      7E D4 000EC       CLRL    -(SP)                                   ;
                                        66            04 FB 000EE       CALLS   #4, ANL$FORMAT_LINE                     ;
                                        03            68 B1 000F1       CMPW    ANL$GW_PROLOG, #3                       ; 0940
                                                      15 12 000F4       BNEQ    3$                                      ;
            50            63            01            03 EF 000F6       EXTZV   #3, #1, (R3), R0                        ; 0941
                                                    6740 DD 000FB       PUSHL   YES_NO[R0]                              ;
                                          00000000G   8F DD 000FE       PUSHL   #ANLRMS$_FDLINDEXCOMPB                  ;
                                                      01 DD 00104       PUSHL   #1                                      ;
                                                      7E D4 00106       CLRL    -(SP)                                   ;
                                        66            04 FB 00108       CALLS   #4, ANL$FORMAT_LINE                     ;
                                        51       18   A2 3C 0010B 3$:   MOVZWL  24(SP), R1                              ; 0943
                                        51 00000064   8F C4 0010F       MULL2   #100, R1                                ;
                                        50       0A   A2 9A 00116       MOVZBL  10(SP), R0                              ; 0944
                          50            50            09 78 0011A       ASHL    #9, R0, R0                              ;
                          7E            51            50 C7 0011E       DIVL3   R0, R1, -(SP)                           ;
                                          00000000G   8F DD 00122       PUSHL   #ANLRMS$_FDLINDEXFILL                   ; 0943
                                                      01 DD 00128       PUSHL   #1                                      ;
                                                      7E D4 0012A       CLRL    -(SP)                                   ;
                                        66            04 FB 0012C       CALLS   #4, ANL$FORMAT_LINE                     ;
                                        7E       07   A2 9A 0012F       MOVZBL  7(SP), -(SP)                            ; 0945
                                          00000000G   8F DD 00133       PUSHL   #ANLRMS$_FDLL1INDEXAREA                 ;
                                                      01 DD 00139       PUSHL   #1                                      ;
                                                      7E D4 0013B       CLRL    -(SP)                                   ;
                                        66            04 FB 0013D       CALLS   #4, ANL$FORMAT_LINE                     ;
                                        6E       42   8F 9A 00140       MOVZBL  #66, STRING_BUF                         ; 0953
                                    04  AE       08   AE 9E 00144       MOVAB   STRING_BUF+8, STRING_BUF+4              ;
                                    4C  AE       20   D0 00149          MOVL    #32, NAME_DSC                           ; 0955
                                    50  AE       34   A2 9E 0014D       MOVAB   52(R2), NAME_DSC+4                      ;
                                                      5E DD 00152       PUSHL   SP                                      ; 0956
                                                 50   AE 9F 00154       PUSHAB  NAME_DSC                                ;
                                 0000G  CF            02 FB 00157       CALLS   #2, ANL$PREPARE_QUOTED_STRING           ;
                                                      5E DD 0015C       PUSHL   SP                                      ; 0957
                                          00000000G   8F DD 0015E       PUSHL   #ANLRMS$_FDLKEYNAME                     ;
                                                      01 DD 00164       PUSHL   #1                                      ;
                                                      7E D4 00166       CLRL    -(SP)                                   ;
                                        66            04 FB 00168       CALLS   #4, ANL$FORMAT_LINE                     ;
            50            63            01            02 EF 0016B       EXTZV   #2, #1, (R3), R0                        ; 0960
                                                    6740 DD 00170       PUSHL   YES_NO[R0]                              ;
                                          00000000G   8F DD 00173       PUSHL   #ANLRMS$_FDLNULLKEY                     ;
                                                      01 DD 00179       PUSHL   #1                                      ;
                                                      7E D4 0017B       CLRL    -(SP)                                   ;
                                        66            04 FB 0017D       CALLS   #4, ANL$FORMAT_LINE                     ;
                          11            63            02 E1 00180       BBC     #2, (R3), 4$                            ; 0961
                                        7E       13   A2 9A 00184       MOVZBL  19(SP), -(SP)                           ; 0962
                                          00000000G   8F DD 00188       PUSHL   #ANLRMS$_FDLNULLVALUE                   ;
                                                      01 DD 0018E       PUSHL   #1                                      ;
                                                      7E D4 00190       CLRL    -(SP)                                   ;
                                        66            04 FB 00192       CALLS   #4, ANL$FORMAT_LINE                     ;
                                                      55 D5 00195 4$:   TSTL    ID                                      ; 0966
                                                      10 12 00197       BNEQ    5$                                      ;
                                        7E            68 3C 00199       MOVZWL  ANL$GW_PROLOG, -(SP)                    ; 0967
                                          00000000G   8F DD 0019C       PUSHL   #ANLRMS$_FDLPROLOG                      ;
                                                      01 DD 001A2       PUSHL   #1                                      ;
                                                      7E D4 001A4       CLRL    -(SP)                                   ;
                                        66            04 FB 001A6       CALLS   #4, ANL$FORMAT_LINE                     ;
                                        54       12   A2 9A 001A9 5$:   MOVZBL  18(SP), R4                              ; 0977
```

```
                              54  D7 001AD          DECL    R4                                        :
                              53  D4 001AF          CLRL    I                                         :
                              2A  11 001B1          BRB     7$                                        :
             7E         2C A243  9A 001B3 6$:       MOVZBL  44(SP)[I], -(SP)                          : 0978
                              53  DD 001B8          PUSHL   I                                         :
                    00000000G 8F  DD 001BA          PUSHL   #ANLRMS$_FDLSEGLENGTH                     :
                              01  DD 001C0          PUSHL   #1                                        :
                              7E  D4 001C2          CLRL    -(SP)                                     :
             66               05  FB 001C4          CALLS   #5, ANL$FORMAT_LINE                       :
             7E         1C A243  3C 001C7          MOVZWL  28(SP)[I], -(SP)                          : 0979
                              53  DD 001CC          PUSHL   I                                         :
                    00000000G 8F  DD 001CE          PUSHL   #ANLRMS$_FDLSEGPOS                        :
                              01  DD 001D4          PUSHL   #1                                        :
                              7E  D4 001D6          CLRL    -(SP)                                     :
             66               05  FB 001D8          CALLS   #5, ANL$FORMAT_LINE                       :
                              53  D6 001DB          INCL    I                                         : 0977
             54               53  D1 001DD 7$:      CMPL    I, R4                                     :
                              D1  1B 001E0          BLEQU   6$                                        :
             50           11  A2  9A 001E2          MOVZBL  17(SP), R0                                : 0985
                         08 A740  DD 001E6          PUSHL   TYPES[R0]                                 :
                    00000000G 8F  DD 001EA          PUSHL   #ANLRMS$_FDLSEGTYPE                       :
                              01  DD 001F0          PUSHL   #1                                        :
                              7E  D4 001F2          CLRL    -(SP)                                     :
             66               04  FB 001F4          CALLS   #4, ANL$FORMAT_LINE                       :
                              7E  7C 001F7          CLRQ    -(SP)                                     : 0990
                              55  DD 001F9          PUSHL   ID                                        :
                          60  AE  9F 001FB          PUSHAB  P                                         :
        0000G  CF             04  FB 001FE          CALLS   #4, ANL$KEY_DESCRIPTOR                    :
               05             50  E9 00203          BLBC    R0, 8$                                    :
                              55  D6 00206          INCL    ID                                        : 0911
                            FE25  31 00208          BRW     1$                                        :
             7E               01  CE 0020B 8$:      MNEGL   #1, -(SP)                                 : 0993
                          58  AE  9F 0020E          PUSHAB  P                                         :
        0000G  CF             02  FB 00211          CALLS   #2, ANL$BUCKET                            :
                                  04 00216          RET                                               : 0996

; Routine Size:  535 bytes,    Routine Base:  $CODE$ + 0260
```

```
1 %sbttl 'ANL$ANALYZE_AREAS - Generate Analysis Primaries for Areas'
1 !++
1 ! Functional Description:
1 !       This routine is responsible for generating the analysis_of_area
1 !       primaries, one for each area.  This primary contains useful
1 !       statistics about an area.
1 !
1 ! Formal Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$analyze_areas: novalue = begin
2
2 local
2       p: bsd,
2       sp: ref block[,byte],
2       area_vbn: long,
2       id: long,
2       r: bsd;
2
2
2 ! We begin by setting up a BSD for the prolog and reading it in.
2
2 init_bsd(p);
2 p[bsd$w_size] = 1;
2 p[bsd$l_vbn] = 1;
2 anl$bucket(p,0);
2
2 ! Save the VBN of the first area descriptor for later use.
2
2 sp = .p[bsd$l_bufptr];
2 area_vbn = .sp[plg$b_avbn];
2
2 ! Now we will loop through the area descriptors and generate an
2 ! analysis of them.  We move from one to the next manually, rather
2 ! than by calling anl$area_descriptor, because we don't want to
2 ! check them again.
2
2 init_bsd(r);
2
3 incru id from 0 to .sp[plg$b_amax]-1 do (
3
3       ! Compute the VBN and offset of this area descriptor.  Get the
3       ! descriptor and set up a pointer SP to it.
;  552      1054  3
;  553      1055  3               p[bsd$l_vbn] = .area_vbn + .id / (512/area$c_bln);
;  554      1056  3               p[bsd$l_offset] = .id mod (512/area$c_bln) * area$c_bln;
;  555      1057  3               anl$bucket(p,0);
;  556      1058  3               sp = .p[bsd$l_bufptr] + .p[bsd$l_offset];
;  557      1059  3
;  558      1060  3               ! If the area contains any reclaimed buckets, we want to count
;  559      1061  3               ! them.  Only prolog 3 files have such buckets.
;  560      1062  3
;  561      1063  4               if .sp[area$l_avail] nequ 0 then (
;  562      1064  4
;  563      1065  4                       ! Get the first reclaimed bucket, using BSD R.
;  564      1066  4
;  565      1067  4                       r[bsd$w_size] = .sp[area$b_arbktsz];
;  566      1068  4                       r[bsd$l_vbn] = .sp[area$l_avail];
;  567      1069  4                       anl$bucket(r,0);
;  568      1070  4
;  569      1071  4                       ! To accumulate the statistics for this area, we will check
;  570      1072  4                       ! the validity of the reclaimed bucket chain, as if we were
;  571      1073  4                       ! in /CHECK mode.  This causes statistics to be accumulated
;  572      1074  4                       ! via the statistics callback mechanism (see module RMSSTATS).
;  573      1075  4
;  574      1076  4                       while anl$3reclaimed_bucket_header(r,false) do;
;  575      1077  3               );
;  576      1078  3
;  577      1079  3               ! Now we can generate the analysis primary.
;  578      1080  3
;  579      1081  3               anl$fdl_analysis_of_area(.id);
;  580      1082  2       );
;  581      1083  2
;  582      1084  2       anl$bucket(p,-1);
;  583      1085  2       anl$bucket(r,-1);
;  584      1086  2       return;
;  585      1087  2
;  586      1088  1       end;



                                                 01FC 00000            .ENTRY   ANL$ANALYZE_AREAS, Save R2,R3,R4,R5,R6,R7,-  ; 1021
                                                                                R8                                             ;
                                58     0000G  CF   9E 00002            MOVAB    ANL$BUCKET, R8                                 ;
                                5E            30   C2 00007            SUBL2    #48, SP                                        ;
          18           00       6E            00   2C 0000A            MOVC5    #0, (SP), #0, #24, P                           ; 1033
                                        18    AE      0000F
                           1A   AE            01   B0 00011            MOVW     #1, P+2                                        ; 1034
                           1C   AE            01   D0 00015            MOVL     #1, P+4                                        ; 1035
                                              7E   D4 00019            CLRL     -(SP)                                          ; 1036
                                        1C    AE   9F 0001B            PUSHAB   P                                              ;
                                68            02   FB 0001E            CALLS    #2, ANL$BUCKET                                 ;
                                56      24    AE   D0 00021            MOVL     P+12, SP                                       ; 1040
                                57      66    A6   9A 00025            MOVZBL   102(SP), AREA_VBN                              ; 1041
          18           00       6E            00   2C 00029            MOVC5    #0, (SP), #0, #24, R                           ; 1048
                                              6E      0002E
                                53      67    A6   9A 0002F            MOVZBL   103(SP), R3                                    ; 1050
                                              53   D7 00033            DECL     R3                                             ;
```

```
                                   52 D4 00035         CLRL    ID                                          ; 1055
                                   53 11 00037         BRB     4$                                          ;
              50          52       08 C7 00039  1$:    DIVL3   #8, ID, R0                                  ;
          1C  AE          50       57 C1 0003D         ADDL3   AREA_VBN, R0, P+4                           ;
     7E       00          52       01 7A 00042         EMUL    #1, ID, #0, -(SP)                           ; 1056
     50       50          8E       08 7B 00047         EDIV    #8, (SP)+, R0, R0                           ;
          20  AE          50       06 78 0004C         ASHL    #6, R0, P+8                                 ;
                                   7E D4 00051         CLRL    -(SP)                                       ; 1057
                              1C   AE 9F 00053         PUSHAB  P                                           ;
                          68       02 FB 00056         CALLS   #2, ANL$BUCKET                              ;
              56    24    AE  20   AE C1 00059         ADDL3   P+8, P+12, SP                               ; 1058
                              08   A6 D5 0005F         TSTL    8(SP)                                       ; 1063
                                   1F 13 00062         BEQL    3$                                          ;
                    02    AE  03   A6 9B 00064         MOVZBW  3(SP), R+2                                  ; 1067
                    04    AE  08   A6 D0 00069         MOVL    8(SP), R+4                                  ; 1068
                                   7E D4 0006E         CLRL    -(SP)                                       ; 1069
                              04   AE 9F 00070         PUSHAB  R                                           ;
                          68       02 FB 00073         CALLS   #2, ANL$BUCKET                              ;
                                   7E D4 00076  2$:    CLRL    -(SP)                                       ; 1076
                              04   AE 9F 00078         PUSHAB  R                                           ;
                   0000G  CF       02 FB 0007B         CALLS   #2, ANL$3RECLAIMED_BUCKET_HEADER            ;
                          F3       50 E8 00080         BLBS    R0, 2$                                      ;
                                   52 DD 00083  3$:    PUSHL   ID                                          ; 1081
                   0000G  CF       01 FB 00085         CALLS   #1, ANL$FDL_ANALYSIS_OF_AREA                ;
                                   52 D6 0008A         INCL    ID                                          ; 1050
                          53       52 D1 0008C  4$:    CMPL    ID, R3                                      ;
                                   A8 1B 0008F         BLEQU   1$                                          ;
                          7E       01 CE 00091         MNEGL   #1, -(SP)                                   ; 1084
                              1C   AE 9F 00094         PUSHAB  P                                           ;
                          68       02 FB 00097         CALLS   #2, ANL$BUCKET                              ;
                          7E       01 CE 0009A         MNEGL   #1, -(SP)                                   ; 1085
                              04   AE 9F 0009D         PUSHAB  R                                           ;
                          68       02 FB 000A0         CALLS   #2, ANL$BUCKET                              ;
                                      04 000A3         RET                                                 ; 1088

; Routine Size:  164 bytes,    Routine Base:  $CODE$ + 0477
```

```
%sbttl 'ANL$ANALYZE_KEYS - Generate Analysis Primaries for Keys'
!++
! Functional Description:
!       This routine is responsible for generating the analysis_of_key
!       primaries, one for each key.  This primary contains useful
!       statistics about a key.
!
! Formal Parameters:
!       none
!
! Implicit Inputs:
!       global data
!
! Implicit Outputs:
!       global data
!
! Returned Value:
!       none
!
! Side Effects:
!
!--


global routine anl$analyze_keys: novalue = begin

local
        p: bsd,
        id: long,
        sp: ref block[,byte],
        i: long;


! We will be looking at all of the key descriptors.  Set up a BSD for the
! first one.

init_bsd(p);
p[bsd$w_size] = 1;
p[bsd$l_vbn] = 1;
p[bsd$l_offset] = 0;

! Now we can loop through the key descriptors.  We move from one to the
! next manually, rather than by calling anl$key_descriptor, because we
! don't want to check them again.

incru id from 0 do (

        ! Get the key descriptor and set up SP to point at it.

        anl$bucket(p,0);
        sp = .p[bsd$l_bufptr] + .p[bsd$l_offset];

        ! Now we want to calculate the statistics for this index.  We do
        ! this by "pretending" to check the index structure.
        ! It can't be done if the index is uninitialized.

        if not .sp[key$v_initidx] then
;  645          1146  3                        anl$idx_check_key_stuff(.sp[key$l_rootvbn],p,.sp[key$b_rootlev]);
;  646          1147  3
;  647          1148  3             ! Now we can generate the analysis primary.
;  648          1149  3
;  649          1150  3             anl$fdl_analysis_of_key(p);
;  650          1151  3
;  651          1152  3             ! Now we can go on to the next descriptor, if there is one.
;  652          1153  3
;  653          1154  3 exitif (.sp[key$l_idxfl] eqlu 0);
;  654          1155  3             p[bsd$l_vbn] = .sp[key$l_idxfl];
;  655          1156  3             p[bsd$l_offset] = .sp[key$w_noff];
;  656          1157  2 );
;  657          1158  2
;  658          1159  2 anl$bucket(p,-1);
;  659          1160  2 return;
;  660          1161  2
;  661          1162  1 end;



                                                  003C 00000          .ENTRY  ANL$ANALYZE_KEYS, Save R2,R3,R4,R5                ; 1113
                                   5E          18  C2 00002          SUBL2   #24, SP
            18         00          6E          00  2C 00005          MOVC5   #0, (SP), #0, #24, P                               ; 1125
                                               6E     0000A                                                                     ;
                              02   AE          01  B0 0000B          MOVW    #1, P+2                                            ; 1126
                              04   AE          01  7D 0000F          MOVQ    #1, P+4                                            ; 1127
                                               53  D4 00013          CLRL    ID                                                 ; 1134
                                               7E  D4 00015 1$:      CLRL    -(SP)                                              ; 1138
                                         04    AE  9F 00017          PUSHAB  P                                                  ;
                           0000G   CF          02  FB 0001A          CALLS   #2, ANL$BUCKET
                       52     0C   AE    08    AE  C1 0001F          ADDL3   P+8, P+12, SP                                      ; 1139
                       0F     10   A2          04  E0 00025          BBS     #4, 16(SP), 2$                                     ; 1145
                                   7E    09    A2  9A 0002A          MOVZBL  9(SP), -(SP)                                       ; 1146
                                         04    AE  9F 0002E          PUSHAB  P                                                  ;
                                         0C    A2  DD 00031          PUSHL   12(SP)                                             ;
                           0000G   CF          03  FB 00034          CALLS   #3, ANL$IDX_CHECK_KEY_STUFF
                                               5E  DD 00039 2$:      PUSHL   SP                                                 ; 1150
                           0000G   CF          01  FB 0003B          CALLS   #1, ANL$FDL_ANALYSIS_OF_KEY
                                               62  D5 00040          TSTL    (SP)                                               ; 1154
                                               0D  13 00042          BEQL    3$                                                 ;
                              04   AE          62  D0 00044          MOVL    (SP), P+4                                          ; 1155
                              08   AE    04    A2  3C 00048          MOVZWL  4(SP), P+8                                         ; 1156
                                               53  D6 0004D          INCL    ID                                                 ; 1134
                                               C4  11 0004F          BRB     1$                                                 ;
                                   7E          01  CE 00051 3$:      MNEGL   #1, -(SP)                                          ; 1159
                                         04    AE  9F 00054          PUSHAB  P                                                  ;
                           0000G   CF          02  FB 00057          CALLS   #2, ANL$BUCKET
                                                   04 0005C          RET                                                        ; 1162

; Routine Size:  93 bytes,    Routine Base:  $CODE$ + 051B


;  662          1163  1
;  663          1164  0 end eludom
;                                   PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;    $PLIT$                        143  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $OWN$                          40  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $CODE$                       1400  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                             Library Statistics
;
;                                              -------- Symbols --------       Pages       Processing
;       File                                   Total    Loaded   Percent       Mapped      Time
;
;    _$255$DUA28:[SYSLIB]LIB.L32;1             18619       61          0       1000          00:01.8




;                                     COMMAND QUALIFIERS

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

; Size:           1400 code + 183 data bytes
; Run Time:          00:25.4
; Elapsed Time:      01:29.2
; Lines/CPU Min:      2750
; Lexemes/CPU-Min: 15984
; Memory Used:  248 pages
; Compilation Complete
```

0008 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RMSINTER
LIS

RMSCHECKA
LIS

RMSFDL
LIS

RMSCHECKB
LIS

RMSINPUT
LIS

RMSMSG
LIS