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**RMSSTATS                              L  4







       RRRRRRRR    MM       MM    SSSSSSSS    SSSSSSSS  TTTTTTTTTT    AAAAAA    TTTTTTTTTT    SSSSSSSS
       RRRRRRRR    MM       MM    SSSSSSSS    SSSSSSSS  TTTTTTTTTT    AAAAAA    TTTTTTTTTT    SSSSSSSS
       RR      RR  MMMM   MMMM  SS          SS              TT      AA      AA      TT      SS
       RR      RR  MMMM   MMMM  SS          SS              TT      AA      AA      TT      SS
       RR      RR  MM  MM   MM  SS          SS              TT      AA      AA      TT      SS
       RR      RR  MM  MM   MM  SS          SS              TT      AA      AA      TT      SS
       RRRRRRRR    MM       MM    SSSSSS      SSSSSS        TT      AA      AA      TT        SSSSSS
       RRRRRRRR    MM       MM    SSSSSS      SSSSSS        TT      AA      AA      TT        SSSSSS
       RR  RR      MM       MM          SS          SS      TT      AAAAAAAAAA      TT              SS
       RR  RR      MM       MM          SS          SS      TT      AAAAAAAAAA      TT              SS
       RR    RR    MM       MM          SS          SS      TT      AA      AA      TT              SS
       RR    RR    MM       MM          SS          SS      TT      AA      AA      TT              SS    ....
       RR      RR  MM       MM  SSSSSSSS    SSSSSSSS        TT      AA      AA      TT      SSSSSSSS      ....
       RR      RR  MM       MM  SSSSSSSS    SSSSSSSS        TT      AA      AA      TT      SSSSSSSS      ....

       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 'RMSSTATS - Calculate and Report File Statistics'
0         module rmsstats (
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, Calculate and Report File Statistics
1 !
1 ! Abstract:     This module is responsible for accumulating, calculating, and
1 !               reporting statistics about RMS files.  This includes both the
1 !               analysis primaries for FDL files, and statistics reports.
1 !
1 !
1 ! Environment:
1 !
1 ! Author: Paul C. Anagnostopoulos, Creation Date: 11 August 1981
1 !
1 ! Modified By:
1 !
1 !       V03-005 PCA1012         Paul C. Anagnostopoulos  6-Apr-1983
1 !               Do not generate data record compression statistic for
1 !               secondary keys, because data record compression is
1 !               meaningless for them.
1 !               Duplicates per SIDR should count just duplicates, not
1 !               including the first pointer.
1 !               Add support for the index level 1 record count statistic.
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 PCA0064         Paul Anagnostopoulos    29-Mar-1982
```

```
1 !                     Use quadword arithmetic when calculating percentages
1 !                     so large files don't result in garbage numbers.
1 !
1 !             V03-002 PCA0063         Paul Anagnostopoulos    29-Mar-1982
1 !                     Change calculation of index record statistics so they
1 !                     are parallel to data record statistics.
1 !
1 !             V03-001 PCA0050         Paul Anagnostopoulos    26-Mar-1982
1 !                     Fix a bug in the way data record compression was
1 !                     calculated.
1 !--
```

```
;    70         0069   1 %sbttl 'Module Declarations'
;    71         0070   1 !
;    72         0071   1 ! Libraries and Requires:
;    73         0072   1 !
;    74         0073   1
;    75         0074   1 library 'lib';
;    76         0075   1 require 'rmsreq';
;    77         0584   1
;    78         0585   1 !
;    79         0586   1 ! Table of Contents:
;    80         0587   1 !
;    81         0588   1
;    82         0589   1 forward routine
;    83         0590   1         anl$fdl_analysis_of_area: novalue,
;    84         0591   1         anl$fdl_analysis_of_key: novalue,
;    85         0592   1         anl$area_statistics: novalue,
;    86         0593   1         anl$key_st:tistics: novalue,
;    87         0594   1         anl$percentage,
;    88         0595   1         anl$bucket_callback: novalue,
;    89         0596   1         anl$reclaimed_bucket_callback: novalue,
;    90         0597   1         anl$index_callback: novalue,
;    91         0598   1         anl$data_callback: novalue;
;    92         0599   1
;    93         0600   1 !
;    94         0601   1 ! External References:
;    95         0602   1 !
;    96         0603   1
;    97         0604   1 external routine
;    98         0605   1         anl$format_line,
;    99         0606   1         anl$format_skip;
;   100         0607   1
;   101         0608   1 external
;   102         0609   1         anl$gl_fat: ref block[,byte],
;   103         0610   1         anl$gw_prolog: word;
```

```
1 !
1 ! Own Variables:
1 !
1 ! The following variables are used to accumulate statistical data about
1 ! the file we are currently analyzing.  The following abbreviations are
1 ! used in the variable names:
1 !
1 !       blk     block
1 !       c       count
1 !       comp    compression
1 !       len     length
1 !       max     maximum
1 !       rec     record
1 !       t       total
1
1 own
1       statistics_accumulators: vector[20,long] initial(rep 20 of (0));
1
1 ! This variable accumulates data about a single area of an indexed file.
1
1 bind
1       reclaimed_blk_c                 = statistics_accumulators[ 0]: long;
1
1 ! These variables accumulate data about a single key of an indexed file.
1 ! There is a set of variables for the primary data level and a set for
1 ! the index levels.
1
1 bind
1       data_blk_c                      = statistics_accumulators[ 1]: long,
1       data_fill_t                     = statistics_accumulators[ 2]: long,
1       data_rec_c                      = statistics_accumulators[ 3]: long,
1       data_rec_len_t                  = statistics_accumulators[ 4]: long,
1       data_comp_key_len_t             = statistics_accumulators[ 5]: long,
1       data_comp_rec_len_t             = statistics_accumulators[ 6]: long,
1       data_sidr_ptr_c                 = statistics_accumulators[ 7]: long,
1
1       index_blk_c                     = statistics_accumulators[ 8]: long,
1       index_fill_t                    = statistics_accumulators[ 9]: long,
1       index_rec_c                     = statistics_accumulators[10]: long,
1       index_level1_rec_c              = statistics_accumulators[11]: long,
1       index_rec_len_t                 = statistics_accumulators[12]: long,
1       index_comp_rec_len_t            = statistics_accumulators[13]: long,
1       max_depth                       = statistics_accumulators[14]: long;
```

```
1 %sbttl 'ANL$FDL_ANALYSIS_OF_AREA - Generate ANALYSIS_OF_AREA Primary'
1 !++
1 ! Functional Description:
1 !       This routine is responsible for generating an analysis primary for
1 !       an indexed file area.  This routine is called after the appropriate
1 !       statistics have been accumulated via the statistics callback
1 !       mechanism described below.
1 !
1 ! Formal Parameters:
1 !       area_id         The ID of this area.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_analysis_of_area(area_id): novalue = begin
2
2
2 ! We begin with a heading which includes the area ID.
2
2 anl$format_skip(0);
2 anl$format_line(0,0,anlrms$_fdlanalarea,.area_id);
2
2 ! Now we put out the statistic that says how many blocks worth of
2 ! reclaimed space there is.
2
2 anl$format_line(0,1,anlrms$_fdlrecl,.reclaimed_blk_c);
2
2 ! Clear the statistics accumulators for the next area.
2
2 ch$fill(%x'00', %allocation(statistics_accumulators),statistics_accumulators);
2
2 return;
2
1 end;


                                                              .TITLE  RMSSTATS RMSSTATS - Calculate and Report File S
;                                                                              tatistics
                                                              .IDENT  \V04-000\

                                                              .PSECT  $OWN$,NOEXE,2

                                     00000000# 00000 STATISTICS_ACCUMULATORS:
                                                              .LONG   0[20]                                          ;

                                                    RECLAIMED_BLK_C=     STATISTICS_ACCUMULATORS
```

```
DATA_BLK_C=              STATISTICS_ACCUMULATORS+4
DATA_FILL_T=             STATISTICS_ACCUMULATORS+8
DATA_REC_C=              STATISTICS_ACCUMULATORS+12
DATA_REC_LEN_T=          STATISTICS_ACCUMULATORS+16
DATA_COMP_KEY_LEN_T=STATISTICS_ACCUMULATORS+20
DATA_COMP_REC_LEN_T=STATISTICS_ACCUMULATORS+24
DATA_SIDR_PTR_C=         STATISTICS_ACCUMULATORS+28
INDEX_BLK_C=             STATISTICS_ACCUMULATORS+32
INDEX_FILL_T=            STATISTICS_ACCUMULATORS+36
INDEX_REC_C=             STATISTICS_ACCUMULATORS+40
INDEX_LEVEL1_REC_C= STATISTICS_ACCUMULATORS+44
INDEX_REC_LEN_T=         STATISTICS_ACCUMULATORS+48
INDEX_COMP_REC_LEN_T=
                         STATISTICS_ACCUMULATORS+52
MAX_DEPTH=               STATISTICS_ACCUMULATORS+56
        .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$FORMAT_LINE
                                                                        .EXTRN  ANL$FORMAT_SKIP
                                                                        .EXTRN  ANL$GL_FAT, ANL$GW_PROLOG

                                                                        .PSECT  $CODE$,NOWRT,2

                                                  003C 00000            .ENTRY  ANL$FDL_ANALYSIS_OF_AREA, Save R2,R3,R4,R5   ; 0679
                                               7E   D4 00002            CLRL    -(SP)                                        ; 0684
                          0000G  CF            01   FB 00004            CALLS   #1, ANL$FORMAT_SKIP                          ;
                                         04    AC   DD 00009            PUSHL   AREA_ID                                      ; 0685
                                  00000000G    8F   DD 0000C            PUSHL   #ANLRMS$_FDLANALAREA                         ;
                                               7E   7C 00012            CLRQ    -(SP)                                        ;
                          0000G  CF            04   FB 00014            CALLS   #4, ANL$FORMAT_LINE                          ;
                                      0000'    CF   DD 00019            PUSHL   RECLAIMED_BLK_C                              ; 0690
                                  00000000G    8F   DD 0001D            PUSHL   #ANLRMS$_FDLRECL                             ;
                                               01   DD 00023            PUSHL   #1                                           ;
                                               7E   D4 00025            CLRL    -(SP)                                        ;
                          0000G  CF            04   FB 00027            CALLS   #4, ANL$FORMAT_LINE                          ;
     0050   8F            00            6E     00   2C 0002C            MOVC5   #0, (SP), #0, #80, STATISTICS_ACCUMULATORS   ; 0694
                                      0000'    CF      00033                                                                 ;
                                                    04 00036            RET                                                  ; 0698

; Routine Size:  55 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %sbttl 'ANL$FDL_ANALYSIS_OF_KEY - Generate FDL Analysis Primary'
1 !++
1 ! Functional Description:
1 !       This routine is responsible for generating an analysis primary for
1 !       an indexed file key.  This routine is called after the appropriate
1 !       statistics have been accumulated via the statistics callback
1 !       mechanism described below.
1 !
1 ! Formal Parameters:
1 !       key_bsd         Address of BSD describing the key.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$fdl_analysis_of_key(key_bsd): novalue = begin
2
2 bind
2       k = .key_bsd: bsd;
2
2 local
2       kp: ref block[,byte],
2       data_bytes: long;
2
2
2 ! We begin with a heading which includes the key ID.
2
2 kp = .k[bsd$l_bufptr] + .k[bsd$l_offset];
2 anl$format_skip(0);
2 anl$format_line(0,0,anlrms$_fdlanalkey,.kp[key$b_keyref]);
2
2 ! If there are no data records in this index because it is uninitialized,
2 ! then let's just put out a comment and quit.  Otherwise we'll have a
2 ! lot of trouble with dividing by zero.
2
3 if .data_rec_c eqlu 0 then (
3       anl$format_line(0,1,anlrms$_fdlnorecs);
3       return;
2 );
2
2 ! Now we put out the statistics for this index in alphabetical order.
2 ! Some items only apply to primary or secondary indices.  Others only
2 ! apply to prologue 3.
2
2 anl$format_line(0,1,anlrms$_fdldatafill,
2       anl$percentage(.data_fill_t,.data_blk_c*512));
3 if .anl$gw_prolog eqlu plg$c_ver_3 then (
;  252          0756  3          anl$format_line(0,1,anlrms$_fdldatakeycomp,
;  253          0757  3                  anl$percentage(.data_rec_c*.kp[key$b_keysz] - .data_comp_key_len_t,.data_rec_c*.kp[key$b_key
;  254          0758  4          if .kp[key$b_keyref] eqlu 0 then (
;  255          0759  4                  data_bytes = .data_rec_len_t - .data_rec_c*.kp[key$b_keysz];
;  256          0760  4                  anl$format_line(0,1,anl$rms$_fdldatareccomp,
;  257          0761  4                          anl$percentage(.data_bytes - .data_comp_rec_len_t,.data_bytes));
;  258          0762  3          );
;  259          0763  2 );
;  260          0764  2 anl$format_line(0,1,anlrms$_fdldatarecs,
;  261          0765  2          .data_rec_c);
;  262          0766  2 anl$format_line(0,1,anlrms$_fdldataspace,
;  263          0767  2          .data_blk_c);
;  264          0768  2 anl$format_line(0,1,anlrms$_fdldepth,
;  265          0769  2          .max_depth);
;  266          0770  2 if .kp[key$b_keyref] nequ 0 then
;  267          0771  2          anl$format_line(0,1,anlrms$_fdldupsper,
;  268          0772  2                  (.data_sidr_ptr_c + .data_rec_c/2) /
;  269          0773  2                          .data_rec_c                     - 1);
;  270          0774  2 if .anl$gw_prolog eqlu plg$c_ver_3 then
;  271          0775  2          anl$format_line(0,1,anlrms$_fdlidxcomp,
;  272          0776  2                  anl$percentage(.index_rec_len_t - .index_comp_rec_len_t,.index_rec_len_t));
;  273          0777  2 anl$format_line(0,1,anlrms$_fdlidxfill,
;  274          0778  2          anl$percentage(.index_fill_t,.index_blk_c*512));
;  275          0779  2 anl$format_line(0,1,anlrms$_fdlidxspace,
;  276          0780  2          .index_blk_c);
;  277          0781  2 anl$format_line(0,1,anlrms$_fdlidxl1recs,
;  278          0782  2          .index_level1_rec_c);
;  279          0783  2 anl$format_line(0,1,anlrms$_fdldatalenmean,
;  280          0784  2          (.data_rec_len_t + .data_rec_c/2) /
;  281          0785  2                  .data_rec_c);
;  282          0786  2 anl$format_line(0,1,anlrms$_fdlidxlenmean,
;  283          0787  2          (.index_rec_len_t + .index_rec_c/2) /
;  284          0788  2                  .index_rec_c);
;  285          0789  2
;  286          0790  2 ! Clear the statistics accumulators for the next key.
;  287          0791  2
;  288          0792  2 ch$fill(%x'00', %allocation(statistics_accumulators),statistics_accumulators);
;  289          0793  2
;  290          0794  2 return;
;  291          0795  2
;  292          0796  1 end;



                                           01FC 00000         .ENTRY  ANL$FDL_ANALYSIS_OF_KEY, Save R2,R3,R4,R5,- ; 0724
                                                                      R6,R7,R8                                    ;
                            58      0000V  CF  9E 00002       MOVAB   ANL$PERCENTAGE, R8                          ;
                            57      0000G  CF  9E 00007       MOVAB   ANL$FORMAT_LINE, R7                         ;
                            56      0000'  CF  9E 0000C       MOVAB   DATA_REC_C, R6                              ;
                            50         04  AC  D0 00011       MOVL    KEY_BSD, R0                                 ; 0727
                52      0C  A0         08  A0  C1 00015       ADDL3   8(R0), 12(R0), KP                           ; 0736
                                           7E  D4 0001B       CLRL    -(SP)                                       ; 0737
                     0000G  CF             01  FB 0001D       CALLS   #1, ANL$FORMAT_SKIP                         ;
                            7E         15  A2  9A 00022       MOVZBL  21(KP), -(SP)                               ; 0738
                               00000000G   8F  DD 00026       PUSHL   #ANLRMS$_FDLANALKEY                         ;
```

```
                                 7E  7C 0002C          CLRQ    -(SP)                                          ;
                 67              04  FB 0002E          CALLS   #4, ANL$FORMAT_LINE                            ;
                                 66  D5 00031          TSTL    DATA_REC_C                                     ; 0744
                                 0E  12 00033          BNEQ    1$                                             ;
                      00000000G  8F  DD 00035          PUSHL   #ANLRMS$_FDLNORECS                             ; 0745
                                 01  DD 0003B          PUSHL   #1                                             ;
                                 7E  D4 0003D          CLRL    -(SP)                                          ;
                 67              03  FB 0003F          CALLS   #3, ANL$FORMAT_LINE                            ;
                                     04 00042          RET                                                    ; 0744
  7E       F8    A6              09  78 00043 1$:      ASHL    #9, DATA_BLK_C, -(SP)                          ; 0754
                            FC   A6  DD 00048          PUSHL   DATA_FILL_T                                    ;
                 68              02  FB 0004B          CALLS   #2, ANL$PERCENTAGE                             ;
                                 50  DD 0004E          PUSHL   R0                                             ;
                      00000000G  8F  DD 00050          PUSHL   #ANLRMS$_FDLDATAFILL                           ; 0753
                                 01  DD 00056          PUSHL   #1                                             ;
                                 7E  D4 00058          CLRL    -(SP)                                          ;
                 67              04  FB 0005A          CALLS   #4, ANL$FORMAT_LINE                            ;
                 03         0000G  CF  B1 0005D        CMPW    ANL$GW_PROLOG, #3                              ; 0755
                                 4A  12 00062          BNEQ    2$                                             ;
                 50         14   A2  9A 00064          MOVZBL  20(KP), R0                                     ; 0757
                 50              66  C4 00068          MULL2   DATA_REC_C, R0                                 ;
                                 50  DD 0006B          PUSHL   R0                                             ;
  7E             50         08   A6  C3 0006D          SUBL3   DATA_COMP_KEY_LEN_T, R0, -(SP)                 ;
                 68              02  FB 00072          CALLS   #2, ANL$PERCENTAGE                             ;
                                 50  DD 00075          PUSHL   R0                                             ;
                      00000000G  8F  DD 00077          PUSHL   #ANLRMS$_FDLDATAKEYCOMP                        ; 0756
                                 01  DD 0007D          PUSHL   #1                                             ;
                                 7E  D4 0007F          CLRL    -(SP)                                          ;
                 67              04  FB 00081          CALLS   #4, ANL$FORMAT_LINE                            ;
                            15   A2  95 00084          TSTB    21(KP)                                         ; 0758
                                 25  12 00087          BNEQ    2$                                             ;
                 50         14   A2  9A 00089          MOVZBL  20(KP), R0                                     ; 0759
                 50              66  C4 0008D          MULL2   DATA_REC_C, R0                                 ;
  50       04    A6              50  C3 00090          SUBL3   R0, DATA_REC_LEN_T, DATA_BYTES                 ;
                                 50  DD 00095          PUSHL   DATA_BYTES                                     ; 0761
  7E             50         0C   A6  C3 00097          SUBL3   DATA_COMP_REC_LEN_T, DATA_BYTES, -(SP)         ;
                 68              02  FB 0009C          CALLS   #2, ANL$PERCENTAGE                             ;
                                 50  DD 0009F          PUSHL   R0                                             ;
                      00000000G  8F  DD 000A1          PUSHL   #ANLRMS$_FDLDATARECCOMP                        ; 0760
                                 01  DD 000A7          PUSHL   #1                                             ;
                                 7E  D4 000A9          CLRL    -(SP)                                          ;
                 67              04  FB 000AB          CALLS   #4, ANL$FORMAT_LINE                            ;
                                 66  DD 000AE 2$:      PUSHL   DATA_REC_C                                     ; 0765
                      00000000G  8F  DD 000B0          PUSHL   #ANLRMS$_FDLDATARECS                           ; 0764
                                 01  DD 000B6          PUSHL   #1                                             ;
                                 7E  D4 000B8          CLRL    -(SP)                                          ;
                 67              04  FB 000BA          CALLS   #4, ANL$FORMAT_LINE                            ;
                            F8   A6  DD 000BD          PUSHL   DATA_BLK_C                                     ; 0767
                      00000000G  8F  DD 000C0          PUSHL   #ANLRMS$_FDLDATASPACE                          ; 0766
                                 01  DD 000C6          PUSHL   #1                                             ;
                                 7E  D4 000C8          CLRL    -(SP)                                          ;
                 67              04  FB 000CA          CALLS   #4, ANL$FORMAT_LINE                            ;
                            2C   A6  DD 000CD          PUSHL   MAX_DEPTH                                      ; 0769
                      00000000G  8F  DD 000D0          PUSHL   #ANLRMS$_FDLDEPTH                              ; 0768
                                 01  DD 000D6          PUSHL   #1                                             ;
                                 7E  D4 000D8          CLRL    -(SP)                                          ;
                 67              04  FB 000DA          CALLS   #4, ANL$FORMAT_LINE                            ;
```

```
                                  15  A2  95 000DD          TSTB    21(KP)                                         ; 0770
                                      1B  13 000E0          BEQL    3$                                             ;
                 50         66        02  C7 000E2          DIVL3   #2, DATA_REC_C, R0                             ; 0772
                            50    10  A6  C0 000E6          ADDL2   DATA_SIDR_PTR_C, R0                            ;
                            50        66  C6 000EA          DIVL2   DATA_REC_C, R0                                 ; 0773
                                  FF  A0  9F 000ED          PUSHAB  -1(R0)                                         ;
                           00000000G  8F  DD 000F0          PUSHL   #ANLRMS$_FDLDUPSPER                            ; 0771
                                      01  DD 000F6          PUSHL   #1                                             ;
                                      7E  D4 000F8          CLRL    -(SP)                                          ;
                            67        04  FB 000FA          CALLS   #4, ANL$FORMAT_LINE                            ;
                            03     0000G  CF  B1 000FD 3$:  CMPW    ANL$GW_PROLOG, #3                              ; 0774
                                      1B  12 00102          BNEQ    4$                                             ;
                                  24  A6  DD 00104          PUSHL   INDEX_REC_LEN_T                                ; 0776
                 7E      24  A6   28  A6  C3 00107          SUBL3   INDEX_COMP_REC_LEN_T, INDEX_REC_LEN_T, -       ;
                                                                    -(SP)                                          ;
                            68        02  FB 0010D          CALLS   #2, ANL$PERCENTAGE                             ;
                                      50  DD 00110          PUSHL   R0                                             ;
                           00000000G  8F  DD 00112          PUSHL   #ANLRMS$_FDLIDXCOMP                            ; 0775
                                      01  DD 00118          PUSHL   #1                                             ;
                                      7E  D4 0011A          CLRL    -(SP)                                          ;
                            67        04  FB 0011C          CALLS   #4, ANL$FORMAT_LINE                            ;
                 7E      14  A6       09  78 0011F 4$:      ASHL    #9, INDEX_BLK_C, -(SP)                         ; 0778
                                  18  A6  DD 00124          PUSHL   INDEX_FILL_T                                   ;
                            68        02  FB 00127          CALLS   #2, ANL$PERCENTAGE                             ;
                                      50  DD 0012A          PUSHL   R0                                             ;
                           00000000G  8F  DD 0012C          PUSHL   #ANLRMS$_FDLIDXFILL                            ; 0777
                                      01  DD 00132          PUSHL   #1                                             ;
                                      7E  D4 00134          CLRL    -(SP)                                          ;
                            67        04  FB 00136          CALLS   #4, ANL$FORMAT_LINE                            ;
                                  14  A6  DD 00139          PUSHL   INDEX_BLK_C                                    ; 0780
                           00000000G  8F  DD 0013C          PUSHL   #ANLRMS$_FDLIDXSPACE                           ; 0779
                                      01  DD 00142          PUSHL   #1                                             ;
                                      7E  D4 00144          CLRL    -(SP)                                          ;
                            67        04  FB 00146          CALLS   #4, ANL$FORMAT_LINE                            ;
                                  20  A6  DD 00149          PUSHL   INDEX_LEVEL1_REC_C                             ; 0782
                           00000000G  8F  DD 0014C          PUSHL   #ANLRMS$_FDLIDXLTRECS                          ; 0781
                                      01  DD 00152          PUSHL   #1                                             ;
                                      7E  D4 00154          CLRL    -(SP)                                          ;
                            67        04  FB 00156          CALLS   #4, ANL$FORMAT_LINE                            ;
                 50         66        02  C7 00159          DIVL3   #2, DATA_REC_C, R0                             ; 0784
                            50    04  A6  C0 0015D          ADDL2   DATA_REC_LEN_T, R0                             ;
                 7E         50        66  C7 00161          DIVL3   DATA_REC_C, R0, -(SP)                          ; 0785
                           00000000G  8F  DD 00165          PUSHL   #ANLRMS$_FDLDATALENMEAN                        ; 0783
                                      01  DD 0016B          PUSHL   #1                                             ;
                                      7E  D4 0016D          CLRL    -(SP)                                          ;
                            67        04  FB 0016F          CALLS   #4, ANL$FORMAT_LINE                            ;
                 50      1C  A6       02  C7 00172          DIVL3   #2, INDEX_REC_C, R0                            ; 0787
                            50    24  A6  C0 00177          ADDL2   INDEX_REC_LEN_T, R0                            ;
                 7E         50    1C  A6  C7 0017B          DIVL3   INDEX_REC_C, R0, -(SP)                         ; 0788
                           00000000G  8F  DD 00180          PUSHL   #ANLRMS$_FDLIDXLENMEAN                         ; 0786
                                      01  DD 00186          PUSHL   #1                                             ;
                                      7E  D4 00188          CLRL    -(SP)                                          ;
                            67        04  FB 0018A          CALLS   #4, ANL$FORMAT_LINE                            ;
   0050   8F     00         6E        00  2C 0018D          MOVC5   #0, (SP), #0, #80, STATISTICS_ACCUMULATORS     ; 0792
                                  F4  A6     00194                                                                 ;
                                          04 00196          RET                                                    ; 0796
```

```
; Routine Size:  407 bytes,    Routine Base:  $CODE$ + 0037
;  294          0797  1 %sbttl 'ANL$AREA_STATISTICS - Print Statistics for Area'
;  295          0798  1 !++
;  296          0799  1 ! Functional Description:
;  297          0800  1 !       This routine is called during /STATISTICS mode to print the
;  298          0801  1 !       statistics for an indexed file area.  We are called after the
;  299          0802  1 !       appropriate statistics have been accumulated via the statistics
;  300          0803  1 !       callback mechanism described below.
;  301          0804  1 !
;  302          0805  1 ! Formal Parameters:
;  303          0806  1 !       area_id         The ID of this area.
;  304          0807  1 !
;  305          0808  1 ! Implicit Inputs:
;  306          0809  1 !       global data
;  307          0810  1 !
;  308          0811  1 ! Implicit Outputs:
;  309          0812  1 !       global data
;  310          0813  1 !
;  311          0814  1 ! Returned Value:
;  312          0815  1 !       none
;  313          0816  1 !
;  314          0817  1 ! Side Effects:
;  315          0818  1 !
;  316          0819  1 !--
;  317          0820  1
;  318          0821  1
;  319          0822  2 global routine anl$area_statistics(area_id): novalue = begin
;  320          0823  2
;  321          0824  2
;  322          0825  2 ! We begin with a heading which includes the area ID.
;  323          0826  2
;  324          0827  2 anl$format_skip(0);
;  325          0828  2 anl$format_line(0,0,anlrms$_statarea,.area_id);
;  326          0829  2 anl$format_skip(0);
;  327          0830  2
;  328          0831  2 ! Now we put out the statistic that says how many blocks worth of
;  329          0832  2 ! reclaimed space there is.
;  330          0833  2
;  331          0834  2 anl$format_line(0,1,anlrms$_statrecl,.reclaimed_blk_c);
;  332          0835  2
;  333          0836  2 ! Clear the statistics accumulators for the next area.
;  334          0837  2
;  335          0838  2 ch$fill(%x'00', %allocation(statistics_accumulators),statistics_accumulators);
;  336          0839  2
;  337          0840  2 return;
;  338          0841  2
;  339          0842  1 end;



                                      003C 00000          .ENTRY  ANL$AREA_STATISTICS, Save R2,R3,R4,R5        ; 0822
                                   7E  D4 00002          CLRL    -(SP)                                        ; 0827
                  0000G  CF        01  FB 00004          CALLS   #1, ANL$FORMAT_SKIP                          ;
                               04  AC  DD 00009          PUSHL   AREA_ID                                      ; 0828
                        00000000G  8F  DD 0000C          PUSHL   #ANLRMS$_STATAREA                            ;
                                   7E  7C 00012          CLRQ    -(SP)                                        ;
                        0000G  CF           04  FB 00014        CALLS   #4, ANL$FORMAT_LINE                        ;
                                            7E  D4 00019        CLRL    -(SP)                                      ; 0829
                        0000G  CF           01  FB 0001B        CALLS   #1, ANL$FORMAT_SKIP                        ;
                                  0000'  CF     DD 00020        PUSHL   RECLAIMED_BLK_C                            ; 0834
                              00000000G  8F     DD 00024        PUSHL   #ANLRMS$_STATRECL                          ;
                                            01  DD 0002A        PUSHL   #1                                         ;
                                            7E  D4 0002C        CLRL    -(SP)                                      ;
                        0000G  CF           04  FB 0002E        CALLS   #4, ANL$FORMAT_LINE                        ;
  0050   8F           00             6E     00  2C 00033        MOVC5   #0, (SP), #0, #80, STATISTICS_ACCUMULATORS ; 0838
                                  0000'  CF        0003A                                                           ;
                                            04     0003D        RET                                                ; 0842

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 01CE
```

```
1 %sbttl 'ANL$KEY_STATISTICS - Print Statistics for a Key'
1 !++
1 ! Functional Description:
1 !       This routine is called during /STATISTICS mode to print the
1 !       statistics for an indexed file key.  We are called after the
1 !       appropriate statistics have been accumulated via the statistics
1 !       callback mechanism described below.
1 !
1 ! Formal Parameters:
1 !       key_bsd         Address of BSD describing the key.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine anl$key_statistics(key_bsd): novalue = begin
2
2 bind
2       k = .key_bsd: bsd;
2
2 local
2       kp: ref block[,byte],
2       data_bytes: long;
2
2
2 ! We begin with a heading which includes the key ID.
2
2 kp = .k[bsd$l_bufptr] + .k[bsd$l_offset];
2 anl$format_skip(0);
2 anl$format_line(0,0,anlrms$_statkey,.kp[key$b_keyref]);
2 anl$format_skip(0);
2
2 ! If there are no data records in this index because it is uninitialized,
2 ! then let's just put out a comment and quit.  Otherwise we'll have a
2 ! lot of trouble with dividing by zero.
2
3 if .data_rec_c eqlu 0 then (
3       anl$format_line(0,1,anlrms$_fdlnorecs);
3       return;
2 );
2
2 ! Now we put out the statistics for the index levels in a logical order.
2 ! Some values only make sense for prologue 3.
2
2 anl$format_line(0,1,anlrms$_statdepth,
2       .max_depth);
2 anl$format_line(0,1,anlrms$_statidxl1recs,
;  398          0900  2             .index_level1_rec_c);
;  399          0901  2     anl$format_line(0,1,anlrms$_statidxlenmean,
;  400          0902  2             (.index_rec_len_t + .index_rec_c/2) /
;  401          0903  2                     .index_rec_c);
;  402          0904  2     anl$format_line(0,1,anlrms$_statidxspace,
;  403          0905  2             .index_blk_c);
;  404          0906  2     anl$format_line(0,1,anlrms$_statidxfill,
;  405          0907  2             anl$percentage(.index_fill_t,.index_blk_c*512));
;  406          0908  2     if .anl$gw_prolog eqlu plg$c_ver_3 then
;  407          0909  2             anl$format_line(0,1,anlrms$_statidxcomp,
;  408          0910  2                     anl$percentage(.index_rec_len_t - .index_comp_rec_len_t,.index_rec_len_t));
;  409          0911  2
;  410          0912  2     ! Now we put out the statistics for the data records in a logical order.
;  411          0913  2     ! Some statistics only make sense for primary or secondary indices.
;  412          0914  2     ! Others make sense only for prologue 3.
;  413          0915  2
;  414          0916  2     anl$format_skip(0);
;  415          0917  2     anl$format_line(0,1,anlrms$_statdatarecs,
;  416          0918  2             .data_rec_c);
;  417          0919  2     if .kp[key$b_keyref] nequ 0 then
;  418          0920  2             anl$format_line(0,1,anlrms$_statdupsper,
;  419          0921  2                     .data_sidr_ptr_c /
;  420          0922  2                       .data_rec_c       - 1);
;  421          0923  2     anl$format_line(0,1,anlrms$_statdatalenmean,
;  422          0924  2             (.data_rec_len_t + .data_rec_c/2) /
;  423          0925  2                     .data_rec_c);
;  424          0926  2     anl$format_line(0,1,anlrms$_statdataspace,
;  425          0927  2             .data_blk_c);
;  426          0928  2     anl$format_line(0,1,anlrms$_statdatafill,
;  427          0929  2             anl$percentage(.data_fill_t,.data_blk_c*512));
;  428          0930  3     if .anl$gw_prolog eqlu plg$c_ver_3 then (
;  429          0931  3             anl$format_line(0,1,anlrms$_statdatakeycomp,
;  430          0932  3                     anl$percentage(.data_rec_c*.kp[key$b_keysz] - .data_comp_key_len_t,.data_rec_c*.kp[key$b_key
;  431          0933  4             if .kp[key$b_keyref] eqlu 0 then (
;  432          0934  4                     data_bytes = .data_rec_len_t - .data_rec_c*.kp[key$b_keysz];
;  433          0935  4                     anl$format_line(0,1,anlrms$_statdatareccomp,
;  434          0936  4                             anl$percentage(.data_bytes - .data_comp_rec_len_t,.data_bytes));
;  435          0937  3             );
;  436          0938  2     );
;  437          0939  2
;  438          0940  2     ! If this is the primary key, then print an indication of the overall
;  439          0941  2     ! space efficiency.
;  440          0942  2
;  441          0943  2     anl$format_skip(0);
;  442          0944  2     if .kp[key$b_keyref] eqlu 0 then
;  443          0945  2             anl$format_line(0,1,anlrms$_statefficiency,
;  444          0946  2                     anl$percentage(.data_rec_len_t,.anl$gl_fat[fat$l_hiblk]*512));
;  445          0947  2
;  446          0948  2     ! Clear the statistics accumulators for the next key.
;  447          0949  2
;  448          0950  2     ch$fill(%x'00', %allocation(statistics_accumulators),statistics_accumulators);
;  449          0951  2
;  450          0952  2     return;
;  451          0953  2
;  452          0954  1     end;
```

```
                                                     F  6
RMSSTATS        RMSSTATS - Calculate and Report File Statistics 16-Sep-1984 00:12:49    VAX-11 Bliss-32 V4.0-742          Page  20
V04-000         ANL$KEY_STATISTICS - Print Statistics for a Key 14-Sep-1984 11:53:02    [ANALYZ.SRC]RMSSTATS.B32;1              (7)



                                             03FC 00000          .ENTRY  ANL$KEY_STATISTICS, Save R2,R3,R4,R5,R6,R7,-; 0868
                                                                         R8,R9                                      ;
                          59       0000G  CF   9E 00002          MOVAB   ANL$FORMAT_SKIP, R9                        ;
                          58       0000V  CF   9E 00007          MOVAB   ANL$PERCENTAGE, R8                         ;
                          57       0000G  CF   9E 0000C          MOVAB   ANL$FORMAT_LINE, R7                        ;
                          56       0000'  CF   9E 00011          MOVAB   DATA_REC_C, R6                             ;
                          50          04  AC   D0 00016          MOVL    KEY_BSD, R0                                ; 0871
       52         0C      A0          08  A0   C1 0001A          ADDL3   8(R0), 12(R0), KP                          ; 0880
                                          7E   D4 00020          CLRL    -(SP)                                      ; 0881
                          69              01   FB 00022          CALLS   #1, ANL$FORMAT_SKIP                        ;
                          7E          15  A2   9A 00025          MOVZBL  21(KP), -(SP)                              ; 0882
                             00000000G    8F   DD 00029          PUSHL   #ANLRMS$_STATKEY                           ;
                                          7E   7C 0002F          CLRQ    -(SP)                                      ;
                          67              04   FB 00031          CALLS   #4, ANL$FORMAT_LINE                        ;
                                          7E   D4 00034          CLRL    -(SP)                                      ; 0883
                          69              01   FB 00036          CALLS   #1, ANL$FORMAT_SKIP                        ;
                                          66   D5 00039          TSTL    DATA_REC_C                                 ; 0889
                                          0E   12 0003B          BNEQ    1$                                         ;
                             00000000G    8F   DD 0003D          PUSHL   #ANLRMS$_FDLNORECS                         ; 0890
                                          01   DD 00043          PUSHL   #1                                         ;
                                          7E   D4 00045          CLRL    -(SP)                                      ;
                          67              03   FB 00047          CALLS   #3, ANL$FORMAT_LINE                        ;
                                               04 0004A          RET                                                ; 0889
                                      2C  A6   DD 0004B 1$:      PUSHL   MAX_DEPTH                                  ; 0898
                             00000000G    8F   DD 0004E          PUSHL   #ANLRMS$_STATDEPTH                         ; 0897
                                          01   DD 00054          PUSHL   #1                                         ;
                                          7E   D4 00056          CLRL    -(SP)                                      ;
                          67              04   FB 00058          CALLS   #4, ANL$FORMAT_LINE                        ;
                                      20  A6   DD 0005B          PUSHL   INDEX_LEVEL1_REC_C                         ; 0900
                             00000000G    8F   DD 0005E          PUSHL   #ANLRMS$_STATIDXL1RECS                     ; 0899
                                          01   DD 00064          PUSHL   #1                                         ;
                                          7E   D4 00066          CLRL    -(SP)                                      ;
                          67              04   FB 00068          CALLS   #4, ANL$FORMAT_LINE                        ;
       50         1C      A6              02   C7 0006B          DIVL3   #2, INDEX_REC_C, R0                        ; 0902
                          50          24  A6   C0 00070          ADDL2   INDEX_REC_LEN_T, R0                        ;
       7E                 50          1C  A6   C7 00074          DIVL3   INDEX_REC_C, R0, -(SP)                     ; 0903
                             00000000G    8F   DD 00079          PUSHL   #ANLRMS$_STATIDXLENMEAN                    ; 0901
                                          01   DD 0007F          PUSHL   #1                                         ;
                                          7E   D4 00081          CLRL    -(SP)                                      ;
                          67              04   FB 00083          CALLS   #4, ANL$FORMAT_LINE                        ;
                                      14  A6   DD 00086          PUSHL   INDEX_BLK_C                                ; 0905
                             00000000G    8F   DD 00089          PUSHL   #ANLRMS$_STATIDXSPACE                      ; 0904
                                          01   DD 0008F          PUSHL   #1                                         ;
                                          7E   D4 00091          CLRL    -(SP)                                      ;
                          67              04   FB 00093          CALLS   #4, ANL$FORMAT_LINE                        ;
       7E         14      A6              09   78 00096          ASHL    #9, INDEX_BLK_C, -(SP)                     ; 0907
                                      18  A6   DD 0009B          PUSHL   INDEX_FILL_T                               ;
                          68              02   FB 0009E          CALLS   #2, ANL$PERCENTAGE                         ;
                                          50   DD 000A1          PUSHL   R0                                         ;
                             00000000G    8F   DD 000A3          PUSHL   #ANLRMS$_STATIDXFILL                       ; 0906
                                          01   DD 000A9          PUSHL   #1                                         ;
                                          7E   D4 000AB          CLRL    -(SP)                                      ;
                          67              04   FB 000AD          CALLS   #4, ANL$FORMAT_LINE                        ;
                          03       0000G  CF   B1 000B0          CMPW    ANL$GW_PROLOG, #3                          ; 0908
```

```
                                         1B  12 000B5        BNEQ    2$                                                  ;
                               24        A6  DD 000B7        PUSHL   INDEX_REC_LEN_T                                     ; 0910
       7E         24     A6    28        A6  C3 000BA        SUBL3   INDEX_COMP_REC_LEN_T, INDEX_REC_LEN_T, -            ;
                                                                     -(SP)                                               ;
                         68              02  FB 000C0        CALLS   #2, ANL$PERCENTAGE                                  ;
                                         50  DD 000C3        PUSHL   R0                                                  ;
                              00000000G  8F  DD 000C5        PUSHL   #ANLRMS$_STATIDXCOMP                                ; 0909
                                         01  DD 000CB        PUSHL   #1                                                  ;
                                         7E  D4 000CD        CLRL    -(SP)                                               ;
                         67              04  FB 000CF        CALLS   #4, ANL$FORMAT_LINE                                 ;
                                         7E  D4 000D2 2$:    CLRL    -(SP)                                               ; 0916
                         69              01  FB 000D4        CALLS   #1, ANL$FORMAT_SKIP                                 ;
                                         66  DD 000D7        PUSHL   DATA_REC_C                                          ; 0918
                              00000000G  8F  DD 000D9        PUSHL   #ANLRMS$_STATDATARECS                               ; 0917
                                         01  DD 000DF        PUSHL   #1                                                  ;
                                         7E  D4 000E1        CLRL    -(SP)                                               ;
                         67              04  FB 000E3        CALLS   #4, ANL$FORMAT_LINE                                 ;
                               15        A2  95 000E6        TSTB    21(KP)                                              ; 0919
                                         15  13 000E9        BEQL    3$                                                  ;
       50         10     A6              66  C7 000EB        DIVL3   DATA_REC_C, DATA_SIDR_PTR_C, R0                     ; 0922
                               FF        A0  9F 000F0        PUSHAB  -1(R0)                                              ;
                              00000000G  8F  DD 000F3        PUSHL   #ANLRMS$_STATDUPSPER                                ; 0920
                                         01  DD 000F9        PUSHL   #1                                                  ;
                                         7E  D4 000FB        CLRL    -(SP)                                               ;
                         67              04  FB 000FD        CALLS   #4, ANL$FORMAT_LINE                                 ;
       50                66              02  C7 00100 3$:    DIVL3   #2, DATA_REC_C, R0                                  ; 0924
                         50    04        A6  C0 00104        ADDL2   DATA_REC_LEN_T, R0                                  ;
       7E                50              66  C7 00108        DIVL3   DATA_REC_C, R0, -(SP)                               ; 0925
                              00000000G  8F  DD 0010C        PUSHL   #ANLRMS$_STATDATALENMEAN                            ; 0923
                                         01  DD 00112        PUSHL   #1                                                  ;
                                         7E  D4 00114        CLRL    -(SP)                                               ;
                         67              04  FB 00116        CALLS   #4, ANL$FORMAT_LINE                                 ;
                               F8        A6  DD 00119        PUSHL   DATA_BLK_C                                          ; 0927
                              00000000G  8F  DD 0011C        PUSHL   #ANLRMS$_STATDATASPACE                              ; 0926
                                         01  DD 00122        PUSHL   #1                                                  ;
                                         7E  D4 00124        CLRL    -(SP)                                               ;
                         67              04  FB 00126        CALLS   #4, ANL$FORMAT_LINE                                 ;
       7E         F8     A6              09  78 00129        ASHL    #9, DATA_BLK_C, -(SP)                               ; 0929
                               FC        A6  DD 0012E        PUSHL   DATA_FILL_T                                         ;
                         68              02  FB 00131        CALLS   #2, ANL$PERCENTAGE                                  ;
                                         50  DD 00134        PUSHL   R0                                                  ;
                              00000000G  8F  DD 00136        PUSHL   #ANLRMS$_STATDATAFILL                               ; 0928
                                         01  DD 0013C        PUSHL   #1                                                  ;
                                         7E  D4 0013E        CLRL    -(SP)                                               ;
                         67              04  FB 00140        CALLS   #4, ANL$FORMAT_LINE                                 ;
                         03       0000G  CF  B1 00143        CMPW    ANL$GW_PROLOG, #3                                   ; 0930
                                         4A  12 00148        BNEQ    4$                                                  ;
                         50    14        A2  9A 0014A        MOVZBL  20(KP), R0                                          ; 0932
                         50              66  C4 0014E        MULL2   DATA_REC_C, R0                                      ;
                                         50  DD 00151        PUSHL   R0                                                  ;
       7E                50    08        A6  C3 00153        SUBL3   DATA_COMP_KEY_LEN_T, R0, -(SP)                      ;
                         68              02  FB 00158        CALLS   #2, ANL$PERCENTAGE                                  ;
                                         50  DD 0015B        PUSHL   R0                                                  ;
                              00000000G  8F  DD 0015D        PUSHL   #ANLRMS$_STATDATAKEYCOMP                            ; 0931
                                         01  DD 00163        PUSHL   #1                                                  ;
                                         7E  D4 00165        CLRL    -(SP)                                               ;
                         67              04  FB 00167        CALLS   #4, ANL$FORMAT_LINE                                 ;
```

```
                                     15  A2  95 0016A        TSTB    21(KP)                                       ; 0933
                                         25  12 0016D        BNEQ    4$                                           ;
                         50          14  A2  9A 0016F        MOVZBL  20(KP), R0                                   ; 0934
                         50              66  C4 00173        MULL2   DATA_REC_C, R0                               ;
           50      04    A6              50  C3 00176        SUBL3   R0, DATA_REC_LEN_T, DATA_BYTES               ;
                                         50  DD 0017B        PUSHL   DATA_BYTES                                   ; 0936
           7E            50          0C  A6  C3 0017D        SUBL3   DATA_COMP_REC_LEN_T, DATA_BYTES, -(SP)       ;
                         68              02  FB 00182        CALLS   #2, ANL$PERCENTAGE                           ;
                                         50  DD 00185        PUSHL   R0                                           ;
                              00000000G  8F  DD 00187        PUSHL   #ANLRMS$_STATDATARECCOMP                     ; 0935
                                         01  DD 0018D        PUSHL   #1                                           ;
                                         7E  D4 0018F        CLRL    -(SP)                                        ;
                         67              04  FB 00191        CALLS   #4, ANL$FORMAT_LINE                          ;
                                         7E  D4 00194 4$:    CLRL    -(SP)                                        ; 0943
                         69              01  FB 00196        CALLS   #1, ANL$FORMAT_SKIP                          ;
                                     15  A2  95 00199        TSTB    21(KP)                                       ; 0944
                                         1F  12 0019C        BNEQ    5$                                           ;
                         50       0000G  CF  D0 0019E        MOVL    ANL$GL_FAT, R0                               ; 0946
           7E      04    A0              09  78 001A3        ASHL    #9, 4(R0), -(SP)                             ;
                                     04  A6  DD 001A8        PUSHL   DATA_REC_LEN_T                               ;
                         68              02  FB 001AB        CALLS   #2, ANL$PERCENTAGE                           ;
                                         50  DD 001AE        PUSHL   R0                                           ;
                              00000000G  8F  DD 001B0        PUSHL   #ANLRMS$_STATEFFICIENCY                      ; 0945
                                         01  DD 001B6        PUSHL   #1                                           ;
                                         7E  D4 001B8        CLRL    -(SP)                                        ;
                         67              04  FB 001BA        CALLS   #4, ANL$FORMAT_LINE                          ;
  0050   8F        00    6E              00  2C 001BD 5$:    MOVC5   #0, (SP), #0, #80, STATISTICS_ACCUMULATORS   ; 0950
                                     F4  A6     001C4                                                         ;
                                             04 001C6        RET                                              ; 0954

; Routine Size:  455 bytes,    Routine Base:  $CODE$ + 020C
;  454          0955   1 %sbttl 'ANL$PERCENTAGE - Calculate a Percentage'
;  455          0956   1 !++
;  456          0957   1 ! Functional Description:
;  457          0958   1 !       This routine is called to calculate a percentage.
;  458          0959   1 !
;  459          0960   1 ! Formal Parameters:
;  460          0961   1 !       dividend        The dividend in the percentage.
;  461          0962   1 !       divisor         The divisor in the percentage.
;  462          0963   1 !
;  463          0964   1 ! Implicit Inputs:
;  464          0965   1 !       global data
;  465          0966   1 !
;  466          0967   1 ! Implicit Outputs:
;  467          0968   1 !       global data
;  468          0969   1 !
;  469          0970   1 ! Returned Value:
;  470          0971   1 !       The percentage.
;  471          0972   1 !
;  472          0973   1 ! Side Effects:
;  473          0974   1 !
;  474          0975   1 !--
;  475          0976   1
;  476          0977   1
;  477          0978   2 global routine anl$percentage(dividend,divisor) = begin
;  478          0979   2
;  479          0980   2 local
;  480          0981   2         quadword: block[8,byte],
;  481          0982   2         quotient: signed long, remainder: signed long;
;  482          0983   2
;  483          0984   2 builtin
;  484          0985   2         emul, ediv;
;  485          0986   2
;  486          0987   2
;  487          0988   2 ! If the divisor is zero, then just return zero.
;  488          0989   2
;  489          0990   2 if .divisor eqlu 0 then
;  490          0991   2         return 0;
;  491          0992   2
;  492          0993   2 ! Calculate the percentage using quadword arithmetic.
;  493          0994   2
;  494          0995   2 emul(dividend,%ref(100),%ref(0),quadword);
;  495          0996   2 ediv(divisor,quadword,quotient,remainder);
;  496          0997   2
;  497          0998   2 return .quotient;
;  498          0999   2
;  499          1000   1 end;


                                                 0000 00000        .ENTRY  ANL$PERCENTAGE, Save nothing              ; 0978
                                   5E          08   C2 00002        SUBL2   #8, SP                                    ;
                                          08   AC   D5 00005        TSTL    DIVISOR                                   ; 0990
                                               14   13 00008        BEQL    1$                                        ;
         6E          00 00000064   8F     04   AC   7A 0000A        EMUL    DIVIDEND, #100, #0, QUADWORD              ; 0995
         50          51            6E     08   AC   7B 00014        EDIV    DIVISOR, QUADWORD, QUOTIENT, REMAINDER    ; 0996
```

```
                                  50             51  D0 0001A          MOVL    QUOTIENT, R0                              ; 0998
                                                     04 0001D          RET                                               ;
                                                 50  D4 0001E 1$:      CLRL    R0                                        ; 1000
                                                     04 00020          RET                                               ;

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 03D3
```

```
1 !                             STATISTICS CALLBACK ROUTINES
1 !
1 ! The method by which we collect the statistics for an RMS file is known
1 ! as statistics callback routines.  These are routines in this module
1 ! which are "called back" by the various low-level structure analysis
1 ! routines.  The analysis routines pass enough information so that the
1 ! callback routines can accumulate raw statistical information.  After the
1 ! analysis is finished, one of the statistics formatting routines
1 ! (ANL$STATISTICS_FDL or ANL$STATISTICS_MODE) can calculate the final
1 ! statistics and format them.
1 !
1 ! When an analysis routine wants to call back a statistics routine, it
1 ! uses the STATISTICS_CALLBACK macro.  This macro takes an arbitrary
1 ! number of statements as its argument, and causes them to be executed
1 ! only if the current mode does statistics (/FDL or /STATISTICS).
```

```
1016  1 %sbttl 'ANL$BUCKET_CALLBACK - Statistics Callback for Buckets'
1017  1 !++
1018  1 ! Functional Description:
1019  1 !       This is the statistics callback routine for the analysis of a bucket.
1020  1 !       We keep track of information about the buckets.
1021  1 !
1022  1 ! Formal Parameters:
1023  1 !       level           The level of this bucket.
1024  1 !       blocks          Number of blocks in the bucket.
1025  1 !       fill            Fill count for this bucket.
1026  1 !
1027  1 ! Implicit Inputs:
1028  1 !       global data
1029  1 !
1030  1 ! Implicit Outputs:
1031  1 !       global data
1032  1 !
1033  1 ! Returned Value:
1034  1 !       none
1035  1 !
1036  1 ! Side Effects:
1037  1 !
1038  1 !--
1039  1
1040  1
1041  2 global routine anl$bucket_callback(level,blocks,fill): novalue = begin
1042  2
1043  2
1044  2 ! We need to keep track of the space occupied and the data fill count.
1045  2 ! It's done separately for data and index.
1046  2
1047  3 if .level eqlu 0 then (
1048  3       data_blk_c = .data_blk_c + .blocks;
1049  3       data_fill_t = .data_fill_t + .fill;
1050  3 ) else (
1051  3       index_blk_c = .index_blk_c + .blocks;
1052  3       index_fill_t = .index_fill_t + .fill;
1053  2 );
1054  2
1055  2 return;
1056  2
1057  1 end;



                                0000 00000        .ENTRY  ANL$BUCKET_CALLBACK, Save nothing          ; 1041
                      04    AC   D5 00002         TSTL    LEVEL                                      ; 1047
                            0D   12 00005         BNEQ    1$                                         ;
0000'   CF            08    AC   C0 00007         ADDL2   BLOCKS, DATA_BLK_C                         ; 1048
0000'   CF            0C    AC   C0 0000D         ADDL2   FILL, DATA_FILL_T                          ; 1049
                                 04 00013         RET                                                ; 1047
0000'   CF            08    AC   C0 00014 1$:     ADDL2   BLOCKS, INDEX_BLK_C                        ; 1051
0000'   CF            0C    AC   C0 0001A         ADDL2   FILL, INDEX_FILL_T                         ; 1052
                                 04 00020         RET                                                ; 1057
```

```
; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 03F4
;   560         1058   1 %sbttl 'ANL$RECLAIMED_BUCKET_CALLBACK - Statistics Callback for Reclaimed'
;   561         1059   1 !++
;   562         1060   1 ! Functional Description:
;   563         1061   1 !       This is the statistics callback routine for reclaimed buckets
;   564         1062   1 !       hanging off an area descriptor.
;   565         1063   1 !
;   566         1064   1 ! Formal Parameters:
;   567         1065   1 !       blocks          Number of blocks in the bucket.
;   568         1066   1 !
;   569         1067   1 ! Implicit Inputs:
;   570         1068   1 !       global data
;   571         1069   1 !
;   572         1070   1 ! Implicit Outputs:
;   573         1071   1 !       global data
;   574         1072   1 !
;   575         1073   1 ! Returned Value:
;   576         1074   1 !       none
;   577         1075   1 !
;   578         1076   1 ! Side Effects:
;   579         1077   1 !
;   580         1078   1 !--
;   581         1079   1
;   582         1080   1
;   583         1081   2 global routine anl$reclaimed_bucket_callback(blocks): novalue = begin
;   584         1082   2
;   585         1083   2
;   586         1084   2 ! All we need to do is keep track of the number of reclaimed blocks.
;   587         1085   2
;   588         1086   2 reclaimed_blk_c = .reclaimed_blk_c + .blocks;
;   589         1087   2
;   590         1088   2 return;
;   591         1089   2
;   592         1090   1 end;


                                        0000 00000          .ENTRY  ANL$RECLAIMED_BUCKET_CALLBACK, Save nothing ; 1081
                        0000'  CF      04   AC   C0 00002          ADDL2   BLOCKS, RECLAIMED_BLK_C                     ; 1086
                                          04 00008          RET                                                 ; 1090

; Routine Size:  9 bytes,    Routine Base:  $CODE$ + 0415
;   594          1091   1 %sbttl 'ANL$INDEX_CALLBACK - Statistics Callback for Index Record'
;   595          1092   1 !++
;   596          1093   1 ! Functional Description:
;   597          1094   1 !       This is the statistics callback routine for an index record.  We
;   598          1095   1 !       accumulate various information about the index.
;   599          1096   1 !
;   600          1097   1 ! Formal Parameters:
;   601          1098   1 !       level           Level of the index bucket.
;   602          1099   1 !       record_len      Nominal length of the entire index record.
;   603          1100   1 !       compressed_len  Compressed length of the entire index record.
;   604          1101   1 !
;   605          1102   1 ! Implicit Inputs:
;   606          1103   1 !       global data
;   607          1104   1 !
;   608          1105   1 ! Implicit Outputs:
;   609          1106   1 !       global data
;   610          1107   1 !
;   611          1108   1 ! Returned Value:
;   612          1109   1 !
;   613          1110   1 !
;   614          1111   1 ! Side Effects:
;   615          1112   1 !
;   616          1113   1 !--
;   617          1114   1
;   618          1115   1
;   619          1116   2 global routine anl$index_callback(level,record_len,compressed_len): novalue = begin
;   620          1117   2
;   621          1118   2
;   622          1119   2 ! Count this index record.  Keep a separate count of level 1 records.
;   623          1120   2
;   624          1121   2 increment (index_rec_c);
;   625          1122   2 if .level eqlu 1 then
;   626          1123   2       increment (index_level1_rec_c);
;   627          1124   2
;   628          1125   2 ! We have to keep track of the total nominal record length, and the total
;   629          1126   2 ! compressed record length.
;   630          1127   2
;   631          1128   2 index_rec_len_t = .index_rec_len_t + .record_len;
;   632          1129   2 index_comp_rec_len_t = .index_comp_rec_len_t + .compressed_len;
;   633          1130   2
;   634          1131   2 ! Now we must keep track of the maximum index depth.
;   635          1132   2
;   636          1133   2 max_depth = maxu(.max_depth,.level);
;   637          1134   2
;   638          1135   2 return;
;   639          1136   2
;   640          1137   1 end;



                                        0004 00000          .ENTRY  ANL$INDEX_CALLBACK, Save R2                      ; 1116
                      52     0000'  CF   9E 00002          MOVAB   MAX_DEPTH, R2                                     ;
                               F0   A2   D6 00007          INCL    INDEX_REC_C                                       ; 1121
                      01       04   AC   D1 0000A          CMPL    LEVEL, #1                                         ; 1122
                                    03   12 0000E          BNEQ    1$                                                ;
                                    F4     A2  D6 00010          INCL    INDEX_LEVEL1_REC_C                            ; 1123
                      F8    A2      08     AC  C0 00013 1$:      ADDL2   RECORD_LEN, INDEX_REC_LEN_T                   ; 1128
                      FC    A2      0C     AC  C0 00018          ADDL2   COMPRESSED_LEN, INDEX_COMP_REC_LEN_T          ; 1129
                            50             62  D0 0001D          MOVL    MAX_DEPTH, R0                                 ; 1133
                      04    AC             50  D1 00020          CMPL    R0, LEVEL                                     ;
                                           04  1E 00024          BGEQU   2$                                            ;
                            50      04     AC  D0 00026          MOVL    LEVEL, R0                                     ;
                            62             50  D0 0002A 2$:      MOVL    R0, MAX_DEPTH                                 ;
                                               04 0002D          RET                                                   ; 1137

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 041E
;  642          1138  1 %sbttl 'ANL$DATA_CALLBACK - Statistics Callback for Data Records'
;  643          1139  1 !++
;  644          1140  1 ! Functional Description:
;  645          1141  1 !       This is the statistics callback routine for data records.  We
;  646          1142  1 !       accumulate various information about the records.
;  647          1143  1 !
;  648          1144  1 ! Formal Parameters:
;  649          1145  1 !       record_len              Nominal length of this data record.
;  650          1146  1 !       compressed_key_len      Compressed length of the key.
;  651          1147  1 !       compressed_data_len     Compressed length of the data bytes.
;  652          1148  1 !       sidr_pointers           Count of pointers in this SIDR.
;  653          1149  1 !
;  654          1150  1 ! Implicit Inputs:
;  655          1151  1 !       global data
;  656          1152  1 !
;  657          1153  1 ! Implicit Outputs:
;  658          1154  1 !       global data
;  659          1155  1 !
;  660          1156  1 ! Returned Value:
;  661          1157  1 !       none
;  662          1158  1 !
;  663          1159  1 ! Side Effects:
;  664          1160  1 !
;  665          1161  1 !--
;  666          1162  1
;  667          1163  1
;  668          1164  2 global routine anl$data_callback(record_len,compressed_key_len,compressed_data_len,sidr_pointers): novalue =
;  669          1165  2
;  670          1166  2
;  671          1167  2 ! Count this data record.
;  672          1168  2
;  673          1169  2 increment (data_rec_c);
;  674          1170  2
;  675          1171  2 ! Keep track of the total nominal record length, total compressed key length,
;  676          1172  2 ! and total compressed data length.
;  677          1173  2
;  678          1174  2 data_rec_len_t = .data_rec_len_t + .record_len;
;  679          1175  2 data_comp_key_len_t = .data_comp_key_len_t + .compressed_key_len;
;  680          1176  2 data_comp_rec_len_t = .data_comp_rec_len_t + .compressed_data_len;
;  681          1177  2
;  682          1178  2 ! For secondary data, keep track of the total number of SIDR pointers.
;  683          1179  2
;  684          1180  2 data_sidr_ptr_c = .data_sidr_ptr_c + .sidr_pointers;
;  685          1181  2
;  686          1182  2 return;
;  687          1183  2
;  688          1184  1 end;



                                  0004 00000          .ENTRY  ANL$DATA_CALLBACK, Save R2                       ; 1164
              52      0000'  CF   9E 00002          MOVAB   DATA_REC_C, R2                                   ;
                             62   D6 00007          INCL    DATA_REC_C                                       ; 1169
        04    A2        04   AC   C0 00009          ADDL2   RECORD_LEN, DATA_REC_LEN_T                       ; 1174
        08    A2        08   AC   C0 0000E          ADDL2   COMPRESSED_KEY_LEN, DATA_COMP_KEY_LEN_T          ; 1175
```

```
                              0C   A2        0C   AC   C0 00013         AD'     COMPRESSED_DATA_LEN, DATA_COMP_REC_LEN_T     ; 1176
                              10   A2        10   AC   C0 00018         AC      SIDR_POINTERS, DATA_SIDR_PTR_C               ; 1180
                                                       04 0001D         RE1                                                  ; 1184

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 044C


;  689          1185  1
;  690          1186  0 end eludom



;                                  PSECT SUMMARY
;
;      Name                     Bytes                       Attributes
;
;  $OWN$                           80  NOVEC,  WRT,  RD ,NOEXE,NOSHR   LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1130  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       14          0      1000          00:01.8




;                                 COMMAND QUALIFIERS

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

; Size:          1130 code + 80 data bytes
; Run Time:         00:22.1
; Elapsed Time:     01:04.5
; Lines/CPU Min:      3222
; Lexemes/CPU-Min: 11831
; Memory Used:  222 pages
; Compilation Complete
```

0009 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY