M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASOPEN
```

```
BBBBBBBB      AAAAAA      SSSSSSSS    000000    PPPPPPPP    EEEEEEEEEE  NN        NN
BBBBBBBB      AAAAAA      SSSSSSSS    000000    PPPPPPPP    EEEEEEEEEE  NN        NN
BB      BB  AA      AA  SS          00      00  PP      PP  EE          NN        NN
BB      BB  AA      AA  SS          00      00  PP      PP  EE          NN        NN
BB      BB  AA      AA  SS          00      00  PP      PP  EE          NNNN      NN
BB      BB  AA      AA  SS          00      00  PP      PP  EE          NNNN      NN
BBBBBBBB    AA      AA    SSSSSS    00      00  PPPPPPPP    EEEEEEEE    NN  NN    NN
BBBBBBBB    AA      AA    SSSSSS    00      00  PPPPPPPP    EEEEEEEE    NN  NN    NN
BB      BB  AAAAAAAAAA          SS  00      00  PP          EE          NN      NNNN
BB      BB  AAAAAAAAAA          SS  00      00  PP          EE          NN      NNNN
BB      BB  AA      AA          SS  00      00  PP          EE          NN        NN      ....
BB      BB  AA      AA          SS  00      00  PP          EE          NN        NN      ....
BBBBBBBB    AA      AA  SSSSSSSS      000000    PP          EEEEEEEEEE  NN        NN      ....
BBBBBBBB    AA      AA  SSSSSSSS      000000    PP          EEEEEEEEEE  NN        NN      ....

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
0 MODULE BAS$OPEN (                                  ! OPEN a BASIC channel
0               IDENT = '1-113'         ! File: BASOPEN.B32 Edit: KC1113
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1983, 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-11 BASIC I/O Processing
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the BAS$OPEN routine, which opens a file
1 !       for a VAX-11 BASIC program.
1 !
1 ! ENVIRONMENT:  VAX-11 User Mode
1
1 ! AUTHOR: John Sauter, CREATION DATE: 30-NOV-78
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  JBS 30-NOV-78
1 ! 1-002 - Change REQUIRE file name from FOR... to OTS...  JBS 06-DEC-78
1 ! 1-003 - Change OPEN$K symbols to LUB$K.  JBS 08-DEC-78
1 ! 1-004 - REQUIRE BASOPN to get default record length.  JBS 12-DEC-78
1 ! 1-005 - Call BAS$CB_PUSH and POP instead of FOR$$.  JBS 29-DEC-78
1 ! 1-006 - Update this module to use the new OPEN calling sequence
1 !          and semantics decided on in the 12-FEB-1979 meeting.
1 !         JBS 14-FEB-1979
1 ! 1-007 - Set LUB$V_USER_RBUF if the record buffer is taken from the
1 !         MAP parameter instead of allocated from virtual storage.
1 !         JBS 16-FEB-1979
1 ! 1-008 - Clear the LOCATE bit in the RAB if the user provides his own
1 !         buffer, so the data will always be placed in it.  JBS 19-FEB-1979
```

```
1 ! 1-009 - Use I/O error codes from BASIOERR.REQ.  JBS 20-FEB-1979
1 ! 1-010 - If the device being opened is a terminal, set the right
1 !          margin based on the BLS field of the FAB.  JBS 22-FEB-1979
1 ! 1-011 - Support the ALLOW SCRATCH clause; it implies no sharing.
1 !         JBS 01-MAR-1979
1 ! 1-012 - Add ORGANIZATION INDEXED.  JBS 08-MAR-1979
1 ! 1-013 - Don't use summary XABs until I can learn how.  JBS 08-MAR-1979
1 ! 1-014 - Set up the key buffer and size fields of the RAB.  JBS 26-MAR-1979
1 ! 1-015 - If the user provides a MAP, null it.  JBS 04-APR-1979
1 ! 1-016 - Accept key data types, if the user provides them.
1 !          JBS 06-APR-1979
1 ! 1-017 - Set up LUB$B_ORGAN on a new file.  JBS 06-APR-1979
1 ! 1-018 - Verify keys properly for an existing indexed file.  JBS 06-APR-1979
1 ! 1-019 - If we allocate a prompt buffer, set its current length
1 !          to zero.  JBS 09-APR-1979
1 ! 1-020 - OPEN on channel 0 gives an error message, and on any other
1 !          open channel closes the channel first.  JBS 12-APR-1979
1 ! 1-021 - The default record format for VIRTUAL files is FIXED.
1 !          JBS 19-APR-1979
1 ! 1-022 - Implement STREAM, since it is in the compiler.  Note: the
1 !          rest of the RTL does not implement STREAM.  JBS 19-APR-1979
1 ! 1-023 - Add LUB$B_RAT, for the FSP$ function.  JBS 19-APR-1979
1 ! 1-024 - Add STATUS.  JBS 19-APR-1979
1 ! 1-025 - Set LUB$V_TERM_FOR if this is a terminal format file.
1 !          JBS 14-MAY-1979
1 ! 1-026 - If RECORDSIZE is specified on a terminal format file, set
1 !          the MARGIN and DEFAULT MARGIN to it.  JBS 18-MAY-1979
1 ! 1-027 - Don't allow a recordsize of 0 on an old file.
1 !          JBS 22-MAY-1979
1 ! 1-028 - Set MRS to the computed record size, in case it defaulted.
1 !          JBS 24-MAY-1979
1 ! 1-029 - Correct an obvious bug in STATUS.  It has not been tested.
1 !          JBS 24-MAY-1979
1 ! 1-030 - Set LUB$V_FORCIBLE if the device is a terminal.
1 !          JBS 24-MAY-1979
1 ! 1-031 - Set RAB$V_UIF for virtual files.  JBS 25-MAY-1979
1 ! 1-032 - Change from LUB$V_NO_KEYS to LUB$V_KEYED.  JBS 30-MAY-1979
1 ! 1-033 - Change margin to 16 bits.  JBS 30-MAY-1979
1 ! 1-034 - Add BAS$$STATU_INIT.  JBS 04-JUN-1979
1 ! 1-035 - Set the language byte in the LUB, so only BASIC I/O
1 !          statements can be used on files opened in BASIC.
1 !          This restriction may be relaxed in some future
1 !          release.  JBS 30-JUN-1979
1 ! 1-036 - If the device is a terminal, change to PRN format so it
1 !          is forcible.  JBS 10-JUL-1979
1 ! 1-037   Implement STREAM for real (see edit 022).  JBS 12-JUL-1979
1 ! [illegible]  PRN files must be VFC format.  JBS 17-JUL-1979
1 ! 1-0[illegible]  Add the unit number as the third argument to USEROPEN.
1 !          JBS 25-JUL-1979
1 ! 1-040 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
1 ! 1-041 - Make LUB$Q_BFA_QUEUE empty.  JBS 30-JUL-1979
1 ! 1-042 - Don't allow ACCESS READ to create a file. JBS 30-JUL-1979
1 ! 1-043 - Make sure the LUB/ISB/RAB gets deallocated if we must
1 !          "bail out".  JBS 31-JUL-1979
1 ! 1-044 - Don't fool with LUB$B_RSL and LUB$A_RSN until we are sure
1 !          the LUN is not open.  Any error messages should show the
1 !          previous file name.  JBS 08-AUG-1979
```

```
1 ! 1-045 - Set NOTSEQORG for all but terminal and sequential files,
1 !          so BAS$$IO_BEG can use it.  JBS 08-AUG-1979
1 ! 1-046 - If ACCESS READ, assume FOR INPUT.  JBS 08-AUG-1979
1 ! 1-047 - Use the BASIC-specific exit handler.  JBS 17-AUG-1979
1 ! 1-048 - Be sure to RMS close the file if an error is detected after
1 !          the RMS OPEN is successful.  JBS 23-AUG-1979
1 ! 1-049 - Rearrange MARGIN and RECORDSIZE defaults for compatability
1 !          with the PDP-11.  Now, terminal format files on disk have
1 !          margin of 72, whereas terminal format files on terminals
1 !          have infinite margin.  JBS 24-AUG-1979
1 ! 1-050 - Correct a typo in the computation of bucket size.  JBS 04-SEP-1979
1 ! 1-051 - Disable locate mode until we get a chance to debug it.
1 !          JBS 12-SEP-1979
1 ! 1-052 - Remove STREAM and add record attributes.  JBS 13-SEP-1979
1 ! 1-053 - Re-enable locate mode.  JBS 13-SEP-1979
1 ! 1-054 - Deafult record atrribute for an old virtual file may be NONE or CR.
1 !          JBS 15-SEP-1979
1 ! 1-055 - Give an error message for the CONNECT clause until it is implemented.
1 !          JBS 19-SEP-1979
1 ! 1-056 - Remove references to LUB$Q_BFA_QUEUE, no longer needed.
1 !          JBS 19-SEP-1979
1 ! 1-057 - Implement CONNECT for indexed files.  JBS 30-SEP-1979
1 ! 1-058 - Improve the error message for mismatch of record attributes.
1 !          JBS 03-OCT-1979
1 ! 1-059 - Don't demand any particular record format if the organization is
1 !          is UNDEFINED.  JBS 12-OCT-1979
1 ! 1-060 - Allow any record format, even UNDEFINED, if the organization is
1 !          undefined.  JBS 12-OCT-1979
1 ! 1-061 - If this is a VFC file, make sure the VFC field size is right.
1 !          JBS 15-OCT-1979
1 ! 1-062 - If the argument list says FOR OUTPUT, set the FAB$V_SUP bit, so
1 !          that an explicit version number in the file name will delete
1 !          and recreate an existing file.  QAR N11-02971  JBS 22-OCT-1979
1 ! 1-063 - Round the block size up to a multiple of 4 bytes.  JBS 25-OCT-1979
1 ! 1-064 - Fix an error message.  JBS 07-NOV-1979
1 ! 1-065 - Improve the interface to USEROPEN.  JBS 07-NOV-1979
1 ! 1-066 - Change virtual arrays to automatic record locking.  JBS 09-NOV-1979
1 ! 1-067 - Set up LIB$A_UBF.  JBS 13-NOV-1979
1 ! 1-068 - Make sure the record buffer is at least as large as the user's
1 !          declared "buffer size", or the default for the organization,
1 !          if none was declared.  JBS 27-NOV-1979
1 ! 1-069 - Don't ever turn on RMS Locate Mode.  DGP 29-Nov-79
1 ! 1-070 - Correct an error in edit 1-068.  JBS 05-DEC-1979
1 ! 1-071 - If the record size test fails on an existing file, give error
1 !         BAD RECORDSIZE VALUE ON OPEN rather than FILE ATTRIBUTES NOT
1 !         MATCHED.  This error was found by FEATS.  JBS 28-DEC-1979
1 ! 1-072 - Change "Run Time Syntax Error" to "Program Lost Sorry".  DGP 07-Jan-80
1 ! 1-073 - Change OPEN_HANDLER to clean up the I/O data base before signalling
1          in the event of a severe error.  DGP 08-Jan-80
1 ! 1-074 - Complete edit 1-073.  JBS 09-JAN-1980
1 ! 1-075 - Improve compatability with the PDP-11: round block size up to 20,
1 !          don't set FAB$V_RWC and don't check FAB$B_RFM if it is defaulted.
1 !          JBS 14-JAN-1980
1 ! 1-076 - Another compatability change: If the ALLOW clause is omitted, let
1 !          RMS default the FAB$B_SHR field.  JBS 15-JAN-1980
1 ! 1-077 - The bucket size field is a word, not a byte.  JBS 01-FEB-1980
1 ! 1-078 - Use 1 for default blocksize in Open.  DGP 12-Feb-1980
```

```
1 ! 1-079 - Always round recordsize for virtual files to next higher multiple
1 !         of 512.  Check file attributes against temp RSZ.  DGP 12-Feb-80
1 ! 1-080 - Complete edit 1-079.  JBS 13-FEB-1980
1 ! 1-081 - A Virtual file must be sequential and have fixed-length records.
1 !          The buffer size (record size) must be greater than or equal to
1 !          512 bytes.  If it is less, round it up unless the user supplied a
1 !          map, in which case we have an error.  Round the record size down
1 !          if necessary to make it even.  JBS 12-JUN-1980
1 ! 1-082 - Put FAB$L_ALQ in LUB$L_ALQ and initialize LUB$L_REC_MAX. FM 22-SEP-80
1 ! 1-083 - In order to be able to OPEN a spooled terminal just like any other
1 !         terminal, make the $OPEN macro a $CREATE macro after we see that
1 !         the file is a terminal and we close and reopen (reCREATE). FM 4-FEB-81
1 ! 1-084 - If organization is UNDEFINED, and recordsize is 0 then signal
1 !         BADRECVAL. FM 5-FEB-81
1 ! 1-085 - Set FAC to null before invoking macro $FAB_INIT.  If this is not done
1 !         the opener gets read access automatically.  PL 20-AUG-81
1 ! 1-086 - Set FAB$V_MSE unconditionally if organization is indexed, so a
1 !         subsequent CONNECT can work.  Also, if caller is trying to CONNECT
1 !         to a child, instead of a parent signal INVFILOPT.  FM 20-AUG-81
1 ! 1-087 - LIB$STOP should be declared EXTERNAL.  PLL 20-Nov-81
1 ! 1-088 - If the user specifies a recordsize less than the file record size,
1 !         do not give an error.  PLL 13-Jan-82
1 ! 1-089 - Check the recordsize of a variable length record file only if
1 !         MRS is set.  PLL 22-Feb-82
1 ! 1-090 - Check 089 should be made only if the file is opened for write access.
1 !         Rather than checking the recordsize of a variable length record file,
1 !         just let RMS catch the error.  PLL 9-Mar-1982
1 ! 1-091 - Yet another fix for recordsize.  RBUF can be larger than the recordsize
1 !         specified in the OPEN block.  PLL 3-May-1982
1 ! 1-092 - Add support for manual record locking.  If the user specifies "UNLOCK
1 !         EXPLICIT" in the OPEN statement, set the ULK bit in the RAB ROP.
1 !         PLL 2-Jun-1982
1 ! 1-093 - Add check for recordsize that exceeds the MAP buffer size.  PLL 2-Jun-1982
1 ! 1-094 - Add support for segmented keys.  PLL 3-Jun-1982
1 ! 1-095 - Fix bug in setting RAB ROP ULK bit.  PLL 8-Jun-1982
1 ! 1-096 - For V2, if the RECORDSIZE is 0, use the MAP size.  PLL 9-Jun-1982
1 ! 1-097 - For segmented keys, the lengths and the positions of all segments
1 !         specified by the user must be checked.  PLL 9-Jun-1982
1 ! 1-098 - For V2, if there is no MAP don't compare the MAP size to the
1 !         recordsize.  PLL 11-Jun-1982
1 ! 1-099 - Fix problem of two files being queued when output file is a spooled
1 !         terminal.  Instead of CLOSEing and re-CREATEing after initial CREATE
1 !         tells us the device is a spooled terminal, simply do a PARSE to see
1 !         if it is, and fix up the FAB and RAB accordingly prior to the CREATE.
1 !         MDL 16-Jun-1982
1 ! 1-100 - Add check for printer as output "file" on OPEN statement; set default
1 !         and right margins to printer width if it is.  MDL 17-Jun-1982
1 ! 1-101 - Move recordsize/map size check so that it includes new files also.
1 !         PLL 2-Jul-1982
1 ! 1-102 - when CONNECT is specified but indexed is not, allow check of
1 !         parent file attributes (to see if its indexed & therefore OK)
1 !         before giving an error.  MDL 19-Jul-1982
1 ! 1-103 - when UNLOCK EXPLICIT is specified for a non-512 byte fixed-length
1 !         sequential file, issue an error.  MDL 20-Jul-1982
1 ! 1-104 - only check for UNLOCK EXPLICIT on V2 programs.  MDL 21-Sep-1982
1 ! 1-105 - enabled commented-out code on summary XABs.  Added check of declared
1 !         key type vs. existing key type for indexed files.
```

```
1 !             MDL  28-Sep-1982
1 ! 1-106 - for V2 programs, if there is not a recordsize but there is a map
1 !             size, then the map size should be used to as the default right
1 !             margin rather than the BASIC default.  MDL 29-Sep-1982
1 ! 1-107 - for indexed files, only give an error if the number of keys the
1 !             user specifies is MORE than the number of keys on the file, rather
1 !             than looking for an exact match.  MDL 15-Oct-1982
1 ! 1-108 - update UNWIND_CCB when we have to drop R11 and pick it up again,
1 !             in the case of opening a channel that's already open somewhere
1 !             else (and therefore needs to be closed first).  MDL 5-Jan-1983
1 ! 1-109 - for V2 programs, signal BADRECVAL if the user attempts to open
1 !             a file with MRS with a MAP that is too small.  MDL 6-Jan-1983
1 ! 1-110 - don't be quite so picky about key data types matching existing
1 !             indexed files - Basic has no unsigned data type, but indexed files
1 !             can have unsigned keys.  MDL 29-Aug-1983
1 ! 1-111 - allow OPEN of SYS$INPUT with ORG=UNDEFINED.  also, look at FAB$W_BLS
1 !             to determine actual recordsize, along with MRS and LRL.  MDL 22-Feb-1984
1 ! 1-112 - take recordsize from the file if it isn't specified.  MDL 3-May-1984
1 ! 1-113 - Fix 1-112.  Use recordsize from the file only if the recordsize
1 !             was not specified and the maximum recordsize (FAB$W_MRS) is
1 !             not zero.  KC and MDL and AKS 07-Sep-1984.
1 !--
1
1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:OTSLNK';                              ! define linkages
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$OPEN : NOVALUE,                               ! Open a file
1     OPEN_HANDLER,                                     ! Condition handler for OPEN
1     BAS$STATUS,                                       ! Status of last file opened
1     BAS$$STATU_INIT : NOVALUE;                        ! Initialize status for RUN command
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLIN:RTLPSECT';                             ! Macros for defining psects
1
1 REQUIRE 'RTLML:OTSLUB';                               ! Logical Unit Block definitions
1
1 REQUIRE 'RTLML:OTSISB';                               ! ISB definitions
1
1 REQUIRE 'RTLIN:BASOPN';                               ! OPEN literals
1
1 REQUIRE 'RTLIN:BASIOERR';                             ! I/O error codes
1
1 LIBRARY 'RTLSTARLE';                                  ! system definitions
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !+
1 ! The following symbols refer to arguments of BAS$OPEN.
1 !-
1
1 LITERAL
1     OPN$K_ORG_TERMI = 0,                              ! no organization specified
1     OPN$K_ORG_VIRTU = 1,                              ! ORGANIZATION VIRTUAL specified
1     OPN$K_ORG_SEQUE = 2,                              ! ORGANIZATION SEQUENTIAL specified
1     OPN$K_ORG_RELAT = 3,                              ! ORGANIZATION RELATIVE specified
1     OPN$K_ORG_INDEX = 4,                              ! ORGANIZATION INDEXED specified
1     OPN$K_ORG_UNDEF = 5,                              ! ORGANIZATION UNDEFINED specified
1     OPN$K_ACC_DEFAU = 0,                              ! no access specified ( = MODIFY)
```

```
    OPN$K_ACC_MODIF = 1,                        ! ACCESS MODIFY specified
    OPN$K_ACC_WRITE = 2,                        ! ACCESS WRITE specified
    OPN$K_ACC_READ = 3,                         ! ACCESS READ specified
    OPN$K_ACC_SCRAT = 4,                        ! ACCESS SCRATCH specified
    OPN$K_ACC_APPEN = 5,                        ! ACCESS APPEND specified
    OPN$K_RFM_DEFAU = 0,                        ! record format not specified
    OPN$K_RFM_FIXED = 1,                        ! FIXED specified
    OPN$K_RFM_VARIA = 2,                        ! VARIABLE specified
    OPN$K_RFM_VFC = 3,                          ! VFC specified
    OPN$K_RFM_STREA = 4,                        ! STREAM specified
    OPN$K_ALL_DEFAU = 0,                        ! no ALLOW clause
    OPN$K_ALL_MODIF = 1,                        ! ALLOW MODIFY specified
    OPN$K_ALL_WRITE = 2,                        ! ALLOW WRITE specified
    OPN$K_ALL_READ = 3,                         ! ALLOW READ specified
    OPN$K_ALL_SCRAT = 4,                        ! ALLOW SCRATCH specified
    OPN$K_ALL_NONE = 6,                         ! ALLOW NONE specified
    OPN$K_RAT_DEFAU = 0,                        ! no RECORDATTR clause
    OPN$K_RAT_FORTR = 1,                        ! RECORDATTR FORTRAN specified
    OPN$K_RAT_CRLF = 2,                         ! RECORDATTR CRLF specified
    OPN$K_RAT_NONE = 3,                         ! RECORDATTR NONE specified
    OPN$K_RAT_PRINT = 4,                        ! RECORDATTR PRINT specified
    OPN$K_RAT_ANY = 5;                          ! RECORDATTR ANY specified

!+
! The following codes specify how OPEN_HANDLER is to act on an UNWIND.
!-

LITERAL
    UNWIND_MIN = 0,                             ! Minimum UNWIND code
    UNWIND_NOP = 0,                             ! Do nothing
    UNWIND_POP = 1,                             ! POP the CCB
    UNWIND_DEALLOC = 2,                         ! Deallocate (and POP) the CCB
    UNWIND_CLOSE = 3,                           ! RMS CLOSE the CCB, then deallocate and POP
    UNWIND_MAX = 3;                             ! Maximum UNWIND code

LITERAL
    K_V1_BLK_SIZE = 52;                         ! Size of V1 OPEN argument block

!
! PSECTS:
!
DECLARE_PSECTS (BAS);                           ! Declare psects for BAS$ facility
!
! OWN STORAGE:
!

OWN
    L_STATUS : INITIAL (0);                     ! Status of last file opened.

!
! EXTERNAL REFERENCES:
!

EXTERNAL
    BAS$$L_XIT_LOCK;                            ! Once-only flag for the exit handler.

EXTERNAL ROUTINE
```

```
1      LIB$STOP : NOVALUE,                        ! Signal fatal error
1      BAS$$STOP : NOVALUE,                       ! Signals fatal error
1      BAS$$STOP_IO : NOVALUE,                    ! Signals fatal I/O error
1      BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,        ! loads register CCB
1      BAS$$CB_POP : JSB_CB_POP NOVALUE,          ! completes LUN processing
1      LIB$GET_VM,                                ! get virtual storage
1      LIB$FREE_VM,                               ! free virtual storage
1      BAS$$DECL_EXITH : NOVALUE,                 ! Declare exit handler
1      OTS$$TAKE_LUN,                             ! Sign out a logical unit number
1      OTS$$CLOSE_FILE,                           ! RMS Close a file
1      LIB$MATCH_COND;                            ! Match a condition code
1
1 !+
1 ! The following are the error codes used in this module.
1 !-
1
1 EXTERNAL LITERAL
1      BAS$K_RECOVEMAP : UNSIGNED (8),            ! RECORDSIZE overflows MAP buffer
1      BAS$K_PROLOSSOR : UNSIGNED (8),            ! Program lost, sorry
1      BAS$K_ILLILLACC : UNSIGNED (8),            ! Illegal or illogical access
1      BAS$K_ILLIO_CHA : UNSIGNED (8),            ! Illegal I/O channel
1      BAS$K_IO_CHAALR : UNSIGNED (8),            ! I/O channel already open
1      BAS$K_FATSYSIO_ : UNSIGNED (8),            ! Fatal system I/O error
1      BAS$K_FILATTNOT : UNSIGNED (8),            ! File attributes not matched
1      BAS$K_RECATTNOT : UNSIGNED (8),            ! Record attributes not matched
1      BAS$K_MAXMEMEXC : UNSIGNED (8),            ! Maximum memory exceeded
1      BAS$K_BADRECVAL : UNSIGNED (8),            ! Bad recordsize value on OPEN
1      BAS$K_NOTIMP : UNSIGNED (8),               ! Not implemented
1      BAS$K_INVFILOPT : UNSIGNED (8),            ! Invalid file option
1      BAS$K_IO_CHANOT : UNSIGNED (8),            ! I/O Channel not open
1      BAS$K_REQRECSIZ : UNSIGNED (8);            ! unlock explicit requires recordsize 512
1
1 !<BLF/PAGE>
```

```
1 !+
1 ! The following FIELD describes the OPEN_ARG_BLK argument of
1 ! BAS$OPEN.
1 !-
1
1 FIELD
1     OPN$ARG_BLOCK =
1         SET
1         OPN$B_CNT = [0, 0, 8, 0],               ! number of remaining bytes
1         OPN$B_ORG = [1, 0, 8, 1],               ! ORGANIZATION clause
1         OPN$B_ACCESS = [2, 0, 8, 1],            ! ACCESS clause
1         OPN$B_RFM = [3, 0, 8, 1],               ! record format
1         OPN$B_ALLOW = [4, 0, 8, 1],             ! ALLOW clause
1         OPN$V_NOSPAN = [5, 0, 1, 0],            ! NO SPAN specified
1         OPN$V_CONTIGUOU = [5, 1, 1, 0],         ! CONTIGUOUS specified
1         OPN$V_NO_REWIND = [5, 2, 1, 0],         ! NO REWIND specified
1         OPN$V_TEMPORARY = [5, 3, 1, 0],         ! TEMPORARY specified
1         OPN$V_FOR_OUTPU = [5, 4, 1, 0],         ! FOR OUTPUT specified
1         OPN$V_FOR_INPUT = [5, 5, 1, 0],         ! FOR INPUT specified
1         OPN$V_CHAN_REF = [5, 6, 1, 0],          ! OPN$L_CHANNEL points to channel number
1         OPN$V_CHAN_WORD = [5, 7, 1, 0],         ! Channel is 16 bits (ref only)
1         OPN$B_RAT = [6, 0, 8, 1],               ! RECORDATTR clause
1         OPN$B_FSZ = [7, 0, 8, 0],               ! Size of VFC buffer
1         OPN$T_FILE_SPEC = [8, 0, %BPADDR, 0],   ! descriptor for file name string
1         OPN$W_RECORDSIZ = [12, 0, 16, 0],       ! size of record buffer in bytes
1         OPN$W_BLOCKSIZE = [14, 0, 16, 0],       ! size of magtape block in bytes
1         OPN$A_MAP = [16, 0, %BPADDR, 0],        ! address of user record buffer
1         OPN$L_FILESIZE = [20, 0, %BPVAL, 1],    ! size of file in blocks
1         OPN$W_EXTENDSIZ = [24, 0, 16, 1],       ! extend size
1         OPN$W_BUCKETSIZ = [26, 0, 16, 1],       ! bucket size in records
1         OPN$B_WINDOWSIZ = [28, 0, 8, 1],        ! number of retrieval pointers
1         OPN$W_CONNECT = [30, 0, 16, 1],         ! channel (LUN) to connect to
1         OPN$B_MULTIBUFF = [32, 0, 8, 1],        ! number of I/O buffers
1         OPN$T_DEFAULTNA = [36, 0, %BPADDR, 0],  ! default name string descriptor
1         OPN$A_USEROPEN = [40, 0, %BPADDR, 0],   ! address of user open procedure
1         OPN$L_CHANNEL = [44, 0, %BPVAL, 0],     ! Channel number, or its address
1         OPN$A_VFC = [48, 0, %BPADDR, 0],        ! Address of VFC buffer
1         OPN$W_MAP_SIZE = [52, 0, 16, 0],        ! Size of MAP buffer
1         OPN$V_UNLOCK = [54, 0, 1, 0]            ! set for manual record locking
1         TES;
1
1 !+
1 ! The following two fields describe the header on the KEY_INFO_BLK parameter
1 ! and each key, respectively.
1 !-
1
1 FIELD
1     OPN$KEYH_BLOCK =
1         SET
1         KEYH$B_KEYNUM = [0, 0, 8, 0],           ! Number of keys in this block
1         KEYH$B_LEN = [1, 0, 8, 0]               ! Length of each KEY field
1         TES;
1
1 LITERAL
1     KEYH$K_LENGTH = 4;                          ! Length of the key header
1
1 FIELD
```

```
1      OPN$KEY_BLOCK =
1          SET
1          KEY$B_LEN = [0, 0, 8, 0],                  ! Length of this key
1          KEY$B_FLAGS = [1, 0, 8, 0],                ! Flags, as follows:
1          KEY$V_DUP = [1, 0, 1, 0],                  ! If set, duplicates are allowed
1          KEY$V_CHG = [1, 1, 1, 0],                  ! If set, changes are allowed
1          KEY$W_OFFSET = [2, 0, 16, 0],              ! Offset of this key in the record
1          KEY$B_DTYPE = [4, 0, 8, 0],                ! Data type of key
1          KEY$B_NUM_SEG = [5, 0, 8, 0],              ! Additional # of segments
1          KEY$B_LEN1 = [6, 0, 8, 0],                 ! length segment 1
1          KEY$W_OFFSET1 = [7, 0, 16, 0],             ! position segment 1
1          KEY$B_LEN2 = [9, 0, 8, 0],                 ! length segment 2
1          KEY$W_OFFSET2 = [10, 0, 16, 0],            ! position segment 2
1          KEY$B_LEN3 = [12, 0, 8, 0],                ! length segment 3
1          KEY$W_OFFSET3 = [13, 0, 16, 0],            ! position segment 3
1          KEY$B_LEN4 = [15, 0, 8, 0],                ! length segment 4
1          KEY$W_OFFSET4 = [16, 0, 16, 0],            ! position segment 4
1          KEY$B_LEN5 = [18, 0, 8, 0],                ! length segment 5
1          KEY$W_OFFSET5 = [19, 0, 16, 0],            ! position segment 5
1          KEY$B_LEN6 = [21, 0, 8, 0],                ! length segment 6
1          KEY$W_OFFSET6 = [22, 0, 16, 0],            ! position segment 6
1          KEY$B_LEN7 = [24, 0, 8, 0],                ! length segment 7
1          KEY$W_OFFSET7 = [25, 0, 16, 0]             ! position segment 7
1          TES;
1
```

```
1 GLOBAL ROUTINE BAS$OPEN (                                  ! Open a file
1           OPEN_ARG_BLK,                                     ! Argument list
1           KEY_INFO_BLK                                      ! Keys for ISAM
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Open a file for BASIC.  This will always be an RMS file.
1 !         Not all combinations of input options are valid.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         OPEN_ARG_BLK.mz.r         A long list of OPEN options.
1 !         KEY_INFO_BLK.rl.ra        The keys, for ISAM opens.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         A lot of fields in the LUB.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         Either opens a file, thus permitting use of the channel number
1 !         by BASIC I/O statements, or calls BAS$$STOP, thus not returning
1 !         to its caller.
1 !
1 !--
1
2     BEGIN
2
2     BUILTIN
2         ACTUALCOUNT;
2
2     MAP
2         OPEN_ARG_BLK : REF BLOCK [0, BYTE] FIELD (OPN$ARG_BLOCK),
2         KEY_INFO_BLK : REF BLOCK [0, BYTE] FIELD (OPN$KEYA_BLOCK);
2
2     LOCAL
2         RSZ,                                               ! Computed record size
2         BKS,                                               ! Computed bucket size
2         BLS,                                               ! Computed blocksize
2         NO_MAP_REC_SPECIFIED,                              ! Flag for W_RBUF_SIZE
2                                                            ! computation. Set in RSZ
2                                                            ! computation
2         OPEN_STATUS,                                       ! RMS status returned by $OPEN
2         CONNECT_STATUS,                                    ! RMS status returned by $CONNECT
2         CHANNEL,                                           ! The channel number
2         FAB_BLOCK : $FAB_DECL,                             !  ocal FAB
```

```
2           FAB : REF $FAB_DECL,                    ! pointer to FAB
2           NAM_BLOCK : $NAM_DECL,                  ! local NAM block
2           XABFHC : $XABFHC_DECL,                  ! local XABFHC block
2           XABSUM : $XABSUM_DECL,                  ! local XABSUM block
2           FILE_NAME_DESC : REF BLOCK [0, BYTE],   ! descriptor for file name
2           FILE_NAME : BLOCK [NAM$C_MAXRSS, BYTE], ! text for file name
2           UNWIND_ACTION : VOLATILE,               ! What to do on UNWIND
2           UNWIND_CCB : VOLATILE;                  ! What CCB to do it to
2
2       GLOBAL REGISTER
2           CCB = K_CCB_REG : REF BLOCK [0, BYTE];  ! points to LUB
2
2   !+
2   ! Maintenance note: When detecting an error, the best routine to call is
2   ! BAS$$STOP_IO, since it will print the channel number and file name.
2   ! However, it should not be called until after the file name and channel
2   ! number are stored in the LUB and FAB.  The best time to call BAS$$STOP_IO
2   ! is after the RMS $OPEN, since the file name will then be the resultant or
2   ! expanded name from RMS, which will have defaults merged and logical names
2   ! translated.  Thus, defering recognizing errors until after the RMS $OPEN
2   ! is worth while.  Of course, some errors cannot reasonably be deferred to
2   ! that point, such as a channel number being out of range, and for these
2   ! the BAS$$STOP routine is used to signal errors.
2   !-
2   !+
2   ! Set up condition handler for UNWINDs
2   !-
2
2       ENABLE
2           OPEN_HANDLER (UNWIND_ACTION, UNWIND_CCB);
2
2   !+
2   ! Initial action is NOP
2   !-
2       UNWIND_ACTION = UNWIND_NOP;
2   !+
2   ! Fetch the channel number from the user.  Later we may need the
2   ! address.
2   !-
2       CHANNEL = .OPEN_ARG_BLK [OPN$L_CHANNEL];
2
3       IF (.OPEN_ARG_BLK [OPN$V_CHAN_REF])
2       THEN
2
3           IF (.OPEN_ARG_BLK [OPN$V_CHAN_WORD])
2           THEN
2               CHANNEL = .BLOCK [.CHANNEL, 0, 0, 16, 1]
2           ELSE
2               CHANNEL = .BLOCK [.CHANNEL, 0, 0, %BPVAL, 1];
2
2   !+
2   ! If the channel number is negative or too large, we have a fatal error.
2   !-
2
2       IF ((.CHANNEL LSS LUB$K_LUN_MIN) OR (.CHANNEL GTR LUB$K_LUN_MAX)) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
2
2   !+
2 ! If the channel number is zero, the OPEN fails.
2 !-
2
2       IF (.CHANNEL EQL 0) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
2
2 !+
2 ! Compute some auxiliary variables which we will need later.
2 !-
2       NO_MAP_REC_SPECIFIED = 0;
2       RSZ = .OPEN_ARG_BLK [OPN$W_RECORDSIZ];
2
3       IF (.RSZ EQL 0)
2       THEN
3           BEGIN
3 !+
3 ! Provide a reasonable default value for the record size, when we
3 ! can determine one.  For V2 programs, the MAP size can be used if
3 ! no RECORDSIZE was specified.
3 !-
3           IF .OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE AND
3              .OPEN_ARG_BLK [OPN$W_MAP_SIZE] NEQ 0
3           THEN
3               RSZ = .OPEN_ARG_BLK [OPN$W_MAP_SIZE]
3           ELSE
4               BEGIN
4               NO_MAP_REC_SPECIFIED = 1;
5               RSZ = (CASE .OPEN_ARG_BLK [OPN$B_ORG] FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
5                   SET
5                   [OPN$K_ORG_TERMI, OPN$K_ORG_SEQUE] : BAS$K_DEF_RECLE;
5                   [OPN$K_ORG_VIRTU] : 512;
5                   [INRANGE, OUTRANGE] : 0;
4                   TES);
4               END
3           END
2       ELSE
2 !+
2 ! For virtual files, round the record size down if necessary to make
2 ! it even.
2 !-
2
3           IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU)
2           THEN
3               BEGIN
3               RSZ = (.RSZ AND -2);
3 !+
3 ! If there is no map, round the record size up to 512 if it is less.
3 ! We don't want to give an error message here because the file name is
3 ! not yet set up and the error message needs to include the file name,
3 ! so we don't check here for having a map less than
3 ! 512 bytes long; instead a check is made after the RMS $OPEN for the
3 ! record size still being less than 512 bytes.
3 !-
3
3               IF (.OPEN_ARG_BLK [OPN$A_MAP] EQLA 0) THEN RSZ = MAX (.RSZ, 512);
3
2               END;
2
```

```
2     BKS = .OPEN_ARG_BLK [OPN$W_BUCKETSIZ];
2
3     IF (.BKS NEQ 0)
2     THEN
2 !+
2 ! Compute the bucket size in RMS terms
2 !-
5         BKS = (511 + (.BKS*(.RSZ +
6         BEGIN
6
6         IF (.OPEN_ARG_BLK [OPN$B_RFM] EQL OPN$K_RFM_VARIA) THEN 2 ELSE 0
6
6         END
5         +
6         BEGIN
6
6         IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX) THEN 7 ELSE 1
6
6         END
3         )) +
4         BEGIN
4
4         IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX) THEN 15 ELSE 0
4
4         END
2         )/512;
2
2 !+
2 ! Compute the block size in RMS terms in case this is a new file on mag tape.
2 ! The size is rounded up to the nearest 4 bytes for compatability with the
2 ! PDP-11.
2 !-
5     BLS = MAX ((((MAX (1, .OPEN_ARG_BLK [OPN$W_BLOCKSIZE])*
6         BEGIN
6
6         IF (.OPEN_ARG_BLK [OPN$B_RFM] EQL OPN$K_RFM_VARIA) THEN .RSZ + 4 ELSE .RSZ
6
6         END
2         ) + 3) AND ( NOT 3)), 20);
2 !+
2 ! Allocate the LUB, ISB and RAB if they have not already been
2 ! allocated for this LUN.  Push down if there is already an I/O
2 ! statement in progress on another unit.  On return, CCB points
2 ! to the current control block.
2 !-
2     BAS$$CB_PUSH (.CHANNEL, LUB$K_LUN_MIN);
2 !+
2 ! Arrange to POP the CCB if we get an error.
2 !-
2     UNWIND_CCB = .CCB;
2     UNWIND_ACTION = UNWIND_POP;
2 !+
2 ! If the LUN is already open, close it.  The call to BAS$$STOP_IO here
2 ! will print the former file name (that is, the name of the file that
2 ! could not be closed).
2 !-
2
```

```
3       IF (.CCB [LUB$V_OPENED])
2       THEN
3           BEGIN
3
3           IF ( NOT OTS$$CLOSE_FI E ()) THEN BAS$$STOP_IO (BAS$K_IOERR_REC);
3
3   !+
3   ! We must drop R11 and pick it back up so that the CLOSE can complete.
3   ! Otherwise the CLOSE will 'hang fire' until the recursive I/O is
3   ! done.
3   !-
3           BAS$$CB_POP ();
3           BAS$$CB_PUSH (.CHANNEL, LUB$K_LUN_MIN);
3           !+
3           ! update which CCB to unwind in case of future errors
3           !-
3           UNWIND_CCB = .CCB;
2           END;
2
2   !+
2   ! If the channel is still open (possibly due to recursive I/O), fail.
2   ! Also fail if there is a FAB associated with the LUB, which means that
2   ! the FORTRAN compatability routines have been called to provide implicit
2   ! inputs to OPEN.  Again, the call to BAS$$STOP_IO will print the former
2   ! file name.
2   !-
2
3       IF (.CCB [LUB$V_OPENED] OR .CCB [LUB$V_DEALLOC] OR (.CCB [LUB$A_FAB] NEQA 0))
2       THEN
2           BAS$$STOP_IO (BAS$K_IO_CHAALR);
2
2   !+
2   ! Make sure the file name fields are set up, in case of an error.
2   ! From here to the RMS $OPEN, a call to BAS$$STOP_IO will print the name
2   ! of this file, as supplied in the call.
2   !-
2       FILE_NAME_DESC = .OPEN_ARG_BLK [OPN$T_FILE_SPEC];
2       CCB [LUB$A_RSN] = .FILE_NAME_DESC [DSC$A_POINTER];
2       CCB [LUB$B_RSL] = MIN (.FILE_NAME_DESC [DSC$W_LENGTH], 255);
2   !+
2   ! Sign out the logical unit number, so an AST won't try to open it.
2   !-
2
2       IF ( NOT OTS$$TAKE_LUN (CHANNEL)) THEN BAS$$STOP_IO (BAS$K_IO_CHAALR);
2
2   !+
2   ! Now that we are sure the CCB is ours, if we get an error, deallocate it.
2   !-
2       UNWIND_ACTION = UNWIND_DEALLOC;
2   !<BLF/PAGE>
```

```
2 !+
2 ! Set up the FAB.  The fields are set up in alphabetical order.
2 !-
2       FAB = FAB_BLOCK;
2       $FAB_INIT (FAB = .FAB, FAC = );
2       FAB [FAB$L_ALQ] = .OPEN_ARG_BLK [OPN$L_FILESIZE];    ! length of file in blocks
2
3       IF ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_RELAT) OR (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX))
2       THEN
2           FAB [FAB$B_BKS] = MIN (2>5, .BKS);       ! number of blocks in each bucket
2
5       IF ( NOT ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_RELAT)   !
3           OR (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX)))
2       THEN
2           FAB [FAB$W_BLS] = .BLS;                  ! mag tape block size
2
2       FAB [FAB$W_DEQ] = .OPEN_ARG_BLK [OPN$W_EXTENDSIZ];   ! blocks to add when extending file
2 !+
2 ! Store pointer to default file name, as specified by the user.
2 !-
2       FILE_NAME_DESC = .OPEN_ARG_BLK [OPN$T_DEFAULTNA];
2
3       IF (.FILE_NAME_DESC NEQ 0)
2       THEN
3           BEGIN
3           FAB [FAB$L_DNA] = .FILE_NAME_DESC [DSC$A_POINTER];
3           FAB [FAB$B_DNS] = MIN (255, .FILE_NAME_DESC [DSC$W_LENGTH]);
2           END;
2
2 !+
2 ! Set the FAC field.
2 !-
2
2       SELECT (.OPEN_ARG_BLK [OPN$B_ACCESS]) OF
2           SET
2
2           [OPN$K_ACC_DEFAU, OPN$K_ACC_MODIF] :
2               FAB [FAB$V_DEL] = 1;                 ! allow deletion of records
2
2           [OPN$K_ACC_DEFAU, OPN$K_ACC_READ, OPN$K_ACC_MODIF, OPN$K_ACC_SCRAT] :
2               FAB [FAB$V_GET] = 1;                 ! allow reading records
2
2           [OPN$K_ACC_DEFAU, OPN$K_ACC_WRITE, OPN$K_ACC_MODIF, OPN$K_ACC_SCRAT, OPN$K_ACC_APPEN] :
2               FAB [FAB$V_PUT] = 1;                 ! allow writing records
2
2           [OPN$K_ACC_SCRAT] :
2               FAB [FAB$V_TRN] = 1;                 ! allow truncate (scratch)
2
2           [OPN$K_ACC_DEFAU, OPN$K_ACC_MODIF, OPN$K_ACC_SCRAT] :
2               FAB [FAB$V_UPD] = 1;                 ! allow updating records
2
2           [OTHERWISE] :
2               BAS$$STOP_IO (BAS$K_PROLOSSOR);
2           TES;
2
2 !+
2 ! Store pointer to file name, as specified by the user.
```

```
  2 !-
  2     FILE_NAME_DESC = .OPEN_ARG_BLK [OPN$T_FILE_SPEC];
  2     FAB [FAB$L_FNA] = .FILE_NAME_DESC [DSC$A_POINTER];
  2     FAB [FAB$B_FNS] = MIN (255, .FILE_NAME_DESC [DSC$W_LENGTH]);
  2 !+
  2 ! Set up the FOP field.
  2 !-
  2 !+
  2 ! If the user specified a file size but did not say CONTIGUOUS, do a
  2 ! "best effort" attempt to get contiguous space.  This is compatable
  2 ! with FORTRAN.
  2 ! Note: this code has been disabled pending review.  Because a search
  2 ! is made for the requested space, this could cause a performance
  2 ! problem.
  2 !
C 2     %(
C 2     IF ((.OPEN_ARG_BLK [OPN$L_FILESIZE] NEQ 0) AND ( NOT .OPEN_ARG_BLK [OPN$V_CONTIGUOU]))
C 2     THEN
C 2         FAB [FAB$V_CBT] = 1;
  2     )%
  2 !+
  2 ! If the user specified neither FOR INPUT nor FOR OUTPUT, then set
  2 ! the CIF bit, so that the $CREATE system service will use an
  2 ! existing file if one is present.  (Otherwise it will create a new
  2 ! file, as usual.)
  2 !-
  2
  2     IF ( NOT (.OPEN_ARG_BLK [OPN$V_FOR_INPUT] OR .OPEN_ARG_BLK [OPN$V_FOR_OUTPU])) THEN FAB [FAB$V_CIF] = 1;
  2
  2 !+
  2 ! Set the 'contiguous best try' flag if the user said CONTIGUOUS.
  2 !       ( Perhaps we should set CTG? )
  2 !-
  2
  2     IF (.OPEN_ARG_BLK [OPN$V_CONTIGUOU]) THEN FAB [FAB$V_CBT] = 1;
  2
  2 !+
  2 ! Set "deferred write" unless the ALLOW option implies sharing which
  2 ! does not permit it.
  2 !-
  2
  4     IF ((.OPEN_ARG_BLK [OPN$B_ALLOW] NEQ OPN$K_ALL_WRITE) AND (.OPEN_ARG_BLK [OPN$B_ALLOW] NEQ
  3         OPN$K_ALL_MODIF))
  2     THEN
  2         FAB [FAB$V_DFW] = 1;
  2
  2 !+
  2 ! If the user did not say APPEND, set the NEF flag to prevent
  2 ! positioning a mag tape to end-of-file on open.
  2 !-
  2
  2     IF (.OPEN_ARG_BLK [OPN$B_ACCESS] NEQ OPN$K_ACC_APPEN) THEN FAB [FAB$V_NEF] = 1;
  2
  2 !+
  2 ! If NOREWIND is not specified, cause the tape to rewind on open.
  2 !-
  2
```

```
;  2       IF ( NOT .OPEN_ARG_ LK [OPN$V_NO_REWIND]) THEN FAB [FAB$V_RWO] = 1;
;  2
;  2   !+
;  2   ! In terminal format files, only sequential operations are allowed.
;  2   ! Tell RMS that we will only be doing sequential operations in case
;  2   ! it can gain some efficiency by knowing this.
;  2   !-
;  2
;  2       IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_TERMI) THEN FAB [FAB$V_SQO] = 1;
;  2
;  2   !+
;  2   ! If the user said "FOR OUTPUT", set the SUP bit so that the $CREATE will
;  2   ! supercede any existing file.  Note that this will happen only if a version
;  2   ! number is specified, since the default for the version number on $CREATE is
;  2   ! the next higher number.
;  2   !-
;  2
;  2       IF (.OPEN_ARG_BLK [OPN$V_FOR_OUTPU]) THEN FAB [FAB$V_SUP] = 1;
;  2
;  2   !+
;  2   ! If the user said TEMPORARY, mark the file as "temporary, delete when
;  2   ! closed".
;  2   !-
;  2
;  2       IF (.OPEN_ARG_BLK [OPN$V_TEMPORARY]) THEN FAB [FAB$V_TMD] = 1;
;  2
;  2       FAB [FAB$B_FSZ] = .OPEN_ARG_BLK [OPN$B_FSZ];
;  2       FAB [FAB$W_MRS] = .RSZ;
;  2
;  2   !+
;  2   ! Set up ORG field.
;  2   !-
;  2
;  2       CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
;  2           SET
;  2
;  2           [OPN$K_ORG_TERMI] :                         ! Terminal format file
;  2               FAB [FAB$B_ORG] = FAB$C_SEQ;
;  2
;  2           [OPN$K_ORG_VIRTU] :                         ! Virtual array file
;  2               FAB [FAB$B_ORG] = FAB$C_SEQ;
;  2
;  2           [OPN$K_ORG_SEQUE] :                         ! Sequential file
;  2               FAB [FAB$B_ORG] = FAB$C_SEQ;
;  2
;  2           [OPN$K_ORG_RELAT] :                         ! Relative file
;  2               FAB [FAB$B_ORG] = FAB$C_REL;
;  2
;  2           [OPN$K_ORG_INDEX] :                         ! Indexed file
;  2               FAB [FAB$B_ORG] = FAB$C_IDX;
;  2
;  2           [OPN$K_ORG_UNDEF] :                         ! Unspecified organization
;  2               FAB [FAB$B_ORG] = FAB$C_SEQ;
;  2
;  2           [OUTRANGE] :
;  2               BAS$$STOP_IO (BAS$K_PROLOSSOR);
;  2           TES;
;  935          1858  2
;  936          1859  2  !+
;  937          1860  2  ! Set up RAT field.  PRN will be set after OPEN if this is a terminal device and a
;  938          1861  2  ! terminal format file with default record attributes.
;  939          1862  2  !-
;  940          1863  2
;  941          1864  2      IF (.OPEN_ARG_BLK [OPN$V_NOSPAN]) THEN FAB [FAB$V_BLK] = 1;
;  942          1865  2
;  943          1866  2      CASE .OPEN_ARG_BLK [OPN$B_RAT] FROM OPN$K_RAT_DEFAU TO OPN$K_RAT_ANY OF
;  944          1867  2          SET
;  945          1868  2
;  946          1869  2          [OPN$K_RAT_DEFAU, OPN$K_RAT_ANY] :
;  947          1870  2
;  948          1871  2              IF (.OPEN_ARG_BLK [OPN$B_ORG] NEQ OPN$K_ORG_VIRTU) THEN FAB [FAB$V_CR] = 1;
;  949          1872  2
;  950          1873  2          [OPN$K_RAT_FORTR] :
;  951          1874  2              FAB [FAB$V_FTN] = 1;
;  952          1875  2
;  953          1876  2          [OPN$K_RAT_CRLF] :
;  954          1877  2              FAB [FAB$V_CR] = 1;
;  955          1878  2
;  956          1879  2          [OPN$K_RAT_NONE] :
;  957          1880  3              BEGIN
;  958          1881  3              0
;  959          1882  2              END;
;  960          1883  2
;  961          1884  2          [OPN$K_RAT_PRINT] :
;  962          1885  2              FAB [FAB$V_PRN] = 1;
;  963          1886  2
;  964          1887  2          [OUTRANGE] :
;  965          1888  2              BAS$$STOP_IO (BAS$K_PROLOSSOR);
;  966          1889  2          TES;
;  967          1890  2
;  968          1891  2  !+
;  969          1892  2  ! Set up the RFM field.
;  970          1893  2  !-
;  971          1894  2
;  972          1895  2      CASE (.OPEN_ARG_BLK [OPN$B_RFM]) FROM OPN$K_RFM_DEFAU TO OPN$K_RFM_STREA OF
;  973          1896  2          SET
;  974          1897  2
;  975          1898  2          [OPN$K_RFM_DEFAU] :                          ! Default to variable unless VIRTUAL
;  976          1899  2              FAB [FAB$B_RFM] =
;  977          1900  3              BEGIN
;  978          1901  3
;  979          1902  3              IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU) THEN FAB$C_FIX ELSE FAB$C_VAR
;  980          1903  3
;  981          1904  2              END;
;  982          1905  2
;  983          1906  2          [OPN$K_RFM_FIXED] :
;  984          1907  2              FAB [FAB$B_RFM] = FAB$C_FIX;         ! fixed-length record format
;  985          1908  2
;  986          1909  2          [OPN$K_RFM_VARIA] :
;  987          1910  2              FAB [FAB$B_RFM] = FAB$C_VAR;         ! variable-length record format
;  988          1911  2
;  989          1912  2          [OPN$K_RFM_VFC] :
;  990          1913  2              FAB [FAB$B_RFM] = FAB$C_VFC;         ! variable-length with fixed control record format
;  991          1914  2
```

```
2           [OPN$K_RFM_STREA, OUTRANGE] :
                BAS$$STOP_IO (BAS$K_PROLOSSOR);
            TES;

2       FAB [FAB$B_RTV] = .OPEN_ARG_BLK [OPN$B_WINDOWSIZ];
2 !+
2 ! Set the SHR field.  Note that no bits are set in the default case, which causes
2 ! RMS to default sharing based on how the file is to be used: read sharing if we
2 ! are only reading the file, no sharing otherwise.
2 !-
2
2       SELECT (.OPEN_ARG_BLK [OPN$B_ALLOW]) OF
2           SET
2
2           [OPN$K_ALL_MODIF] :
2               FAB [FAB$V_SHRDEL] = 1;             ! allow sharers to delete
2
2           [OPN$K_ALL_READ, OPN$K_ALL_MODIF] :
2               FAB [FAB$V_SHRGET] = 1;             ! allow sharers to read
2
2           [OPN$K_ALL_NONE, OPN$K_ALL_SCRAT] :
2               FAB [FAB$V_NIL] = 1;                ! forbid sharing
2
2           [OPN$K_ALL_MODIF, OPN$K_ALL_WRITE] :
2               FAB [FAB$V_SHRPUT] = 1;             ! allow sharers to write
2
2           [OPN$K_ALL_MODIF] :
2               FAB [FAB$V_SHRUPD] = 1;             ! allow sharers to update
2
2           [OPN$K_ALL_DEFAU] :                     ! use RMS defaults: don't set FAB$B_SHR field
3               BEGIN
3               0
2               END;
2
2           [OTHERWISE] :
2               BAS$$STOP_IO (BAS$K_PROLOSSOR);
2           TES;
2
2 !+
2 ! If organization is indexed then set FAB$V_MSE unconditionally.
2 !-
2       IF .OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX THEN FAB [FAB$V_MSE] = 1;
2
2 !<BLF/PAGE>
```

```
2  !+
2  ! Set up the LUB.
2  !-
2     CCB [LUB$A_FAB] = .FAB;                          ! store pointer to FAB in LUB
2  !+
2  ! Store pointer to user-supplied file name in LUB for error processing
2  !-
2     CCB [LUB$A_RSN] = .FAB [FAB$L_FNA];
2     CCB [LUB$B_RSL] = .FAB [FAB$B_FNS];
2  !+
2  ! Set the READ_ONLY, SCRATCH and APPEND bits in the LUB based on the
2  ! access.  If READ_ONLY is set, set OLD_FILE, since there is no point
2  ! in creating a file which can only be read.
2  !-
2
2     CASE (.OPEN_ARG_BLK [OPN$B_ACCESS]) FROM OPN$K_ACC_DEFAU TO OPN$K_ACC_APPEN OF
2         SET
2
2         [OPN$K_ACC_DEFAU, OPN$K_ACC_MODIF, OPN$K_ACC_WRITE] :
3             BEGIN                                    ! set none of the three bits
3             CCB [LUB$V_READ_ONLY] = 0;
3             CCB [LUB$V_SCRATCH] = 0;
3             CCB [LUB$V_APPEND] = 0;
2             END;
2
2         [OPN$K_ACC_READ] :
3             BEGIN                                    ! set READ_ONLY
3             CCB [LUB$V_OLD_FILE] = 1;                ! Do not create file
3             CCB [LUB$V_READ_ONLY] = 1;
3             CCB [LUB$V_SCRATCH] = 0;
3             CCB [LUB$V_APPEND] = 0;
2             END;
2
2         [OPN$K_ACC_SCRAT] :
3             BEGIN                                    ! set SCRATCH
3             CCB [LUB$V_READ_ONLY] = 0;
3             CCB [LUB$V_SCRATCH] = 1;
3             CCB [LUB$V_APPEND] = 0;
2             END;
2
2         [OPN$K_ACC_APPEN] :
3             BEGIN                                    ! set APPEND.  It will be cleared if the file is NEW
3             CCB [LUB$V_READ_ONLY] = 0;
3             CCB [LUB$V_SCRATCH] = 0;
3             CCB [LUB$V_APPEND] = 1;
2             END;
2         TES;
2
2  !+
2  ! Set the "not sequential" flag in the LUB based on the file organization.
2  !-
2
2     CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
2         SET
2
2         [OPN$K_ORG_TERMI, OPN$K_ORG_SEQUE] :
2             CCB [LUB$V_NOTSEQORG] = 0;
```

```
2
2          [OPN$K_ORG_RELAT, OPN$K_ORG_INDEX, OPN$K_ORG_VIRTU, OPN$K_ORG_UNDEF] :
2              CCB [LUB$V_NOTSEQORG] = 1;
2          TES;
2
2  !+
2  ! Set the 'formatted' and 'unformatted' bits in the LUB.
2  ! These bits are used only by FORTRAN I/O statements.
2  !-
2
2      CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
2          SET
2
2          [OPN$K_ORG_TERMI, OPN$K_ORG_SEQUE, OPN$K_ORG_RELAT, OPN$K_ORG_INDEX] :
2              CCB [LUB$V_FORMATTED] = 1;
2
2          [OPN$K_ORG_VIRTU, OPN$K_ORG_UNDEF] :
2              CCB [LUB$V_UNFORMAT] = 1;
2          TES;
2
2  !+
2  ! Set the 'fixed' flag in the LUB.
2  !-
2
2      IF (.FAB [FAB$B_RFM] EQL FAB$C_FIX) THEN CCB [LUB$V_FIXED] = 1;
2
2  !+
2  ! Set the 'old file' flag in the LUB.
2  ! It will also be set after the $OPEN if the file turns out to already
2  ! exist.
2  !-
2
2      IF (.OPEN_ARG_BLK [OPN$V_FOR_INPUT]) THEN CCB [LUB$V_OLD_FILE] = 1;
2
2  !+
2  ! Set the 'direct' flag in the LUB.  This is used only by FORTRAN I/O.
2  !-
2
2      CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
2          SET
2
2          [OPN$K_ORG_TERMI, OPN$K_ORG_SEQUE, OPN$K_ORG_RELAT, OPN$K_ORG_INDEX] :
2              CCB [LUB$V_DIRECT] = 0;               ! not direct access
2
2          [OPN$K_ORG_VIRTU, OPN$K_ORG_UNDEF] :
2              CCB [LUB$V_DIRECT] = 1;               ! direct access.  Assigning UNDEF here is arbitrary.
2          TES;
2
2  !+
2  ! Set up the right margin and default right margin.
2  !-
2
2      CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
2          SET
2
2          [OPN$K_ORG_TERMI] :
3              BEGIN
```

```
3 !+
3 ! Terminal format file.  We assume that this will be a disk file.  If it turns out
3 ! to be a terminal, the margin will be reset to "infinite" after the OPEN.
3 !-
3                   CCB [LUB$W_D_MARGIN] =
4                   BEGIN
4
5                   IF (.OPEN_ARG_BLK [OPN$W_RECORDSIZ] NEQ 0)
4                   THEN
4                       .OPEN_ARG_BLK [OPN$W_RECORDSIZ]
4                   ELSE
4                       !+
4                       ! for V2 programs, use the MAP size if it was specified.
4                       ! (and there was no recordsize)
4                       !-
5                       IF ( (.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE) AND
5                            (.OPEN_ARG_BLK [OPN$W_MAP_SIZE] NEQ 0) )
4                       THEN .OPEN_ARG_BLK [OPN$W_MAP_SIZE]
4                       ELSE LUB$K_D_MARGIN
4
3                   END;
3                   CCB [LUB$W_R_MARGIN] = .CCB [LUB$W_D_MARGIN];
3                   CCB [LUB$V_NOMARGIN] = 0;
2                   END;
2
2               [INRANGE] :
3                   BEGIN
3 !+
3 ! This is not a terminal format file.  The margin is only important for sequential files,
3 ! so we will set up the margin for sequential files.
3 !-
3                   CCB [LUB$W_D_MARGIN] =
4                   BEGIN
4
5                   IF (.OPEN_ARG_BLK [OPN$W_RECORDSIZ] NEQ 0)
4                   THEN
4                       .OPEN_ARG_BLK [OPN$W_RECORDSIZ]
4                   ELSE
4                       !+
4                       ! for V2 programs, use the MAP size if it was specified.
4                       ! (and there was no recordsize)
4                       !-
5                       IF ( (.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE) AND
5                            (.OPEN_ARG_BLK [OPN$W_MAP_SIZE] NEQ 0) )
4                       THEN .OPEN_ARG_BLK [OPN$W_MAP_SIZE]
4                       ELSE LUB$K_D_MARGIN
4
3                   END;
3                   CCB [LUB$W_R_MARGIN] = 0;
3                   CCB [LUB$V_NOMARGIN] = 1;
2                   END;
2
2               [OUTRANGE] :
2                   BAS$$STOP_IO (BAS$K_PROLOSSOR);
2               TES;
2
2 !+
2 ! Set a flag in the LUB if this file was opened with  keys.  This
2 ! is important for processing indexed files.
2 !-
2     CCB [LUB$V_KEYED] = (ACTUALCOUNT () GEQ 2);
2 !+
2 ! Set a flag in the LUB if this is a multi-stream connect.
2 !-
2
3     IF (.OPEN_ARG_BLK [OPN$W_CONNECT] NEQ 0)
2     THEN
2         CCB [LUB$V_M_STREAM] = 1
2     ELSE
2         CCB [LUB$V_M_STREAM] = 0;
2
2 !+
2 ! Set the terminal format file bit in the LUB if this is a terminal
2 ! format file.  This is used to bias RECOUNT.
2 !-
2
2     IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_TERMI) THEN CCB [LUB$V_TERM_FOR] = 1;
2
2 !<BLF/PAGE>
```

```
; 1231          2152  2 !+
; 1232          2153  2 ! Set up the RAB.
; 1233          2154  2 !-
; 1234          2155  2     CCB [RAB$B_BID] = RAB$C_BID;                  ! this is a RAB
; 1235          2156  2     CCB [RAB$B_BLN] = RAB$C_BLN;                  ! length of a RAB
; 1236          2157  2     CCB [RAB$L_FAB] = .FAB;                       ! pointer to File Access Block
; 1237          2158  2 !+
; 1238          2159  2 ! Set up the key buffer and size fields.  If this is a terminal device
; 1239          2160  2 ! these will be overwritten by the prompt buffer and size fields,
; 1240          2161  2 ! but this is OK, since you cannot do keyed access to a terminal device.
; 1241          2162  2 !-
; 1242          2163  2     CCB [RAB$L_KBF] = CCB [LUB$L_LOG_RECNO];
; 1243          2164  2     CCB [RAB$B_KSZ] = 4;
; 1244          2165  2     CCB [RAB$B_MBF] = .OPEN_ARG_BLK [OPN$B_MULTIBUFF];  ! number of buffers
; 1245          2166  2
; 1246          2167  2     IF (.OPEN_ARG_BLK [OPN$B_FSZ] NEQ 0) THEN CCB [RAB$L_RHB] = .OPEN_ARG_BLK [OPN$A_VFC];
; 1247          2168  2
; 1248          2169  2 !+
; 1249          2170  2 ! Set up the ROP field.
; 1250          2171  2 !-
; 1251          2172  2
; 1252          2173  2 !+
; 1253          2174  2 ! for V2 programs, enable manual record locking if specified.
; 1254          2175  2 !-
; 1255          2176  3     IF (.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE)
; 1256          2177  2     THEN
; 1257          2178  3     IF (.OPEN_ARG_BLK [OPN$V_UNLOCK] NEQ 0)
; 1258          2179  2     THEN
; 1259          2180  3         BEGIN
; 1260          2181  3
; 1261          2182  3         !+
; 1262          2183  3         ! if this is a sequential file, and the recordsize is not 512,
; 1263          2184  3         ! manual record locking is not allowed; otherwise, it is.
; 1264          2185  3         !-
; 1265          2186  4         IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_SEQUE  AND
; 1266          2187  4             .OPEN_ARG_BLK [OPN$W_RECORDSIZ] NEQ 512)
; 1267          2188  3         THEN    BAS$$STOP_IO (BAS$K_REQRECSIZ)
; 1268          2189  3
; 1269          2190  3         ELSE    CCB [RAB$V_ULK] = 1;
; 1270          2191  3                                         ! enable manual record locking
; 1271          2192  2         END;
; 1272          2193  2
; 1273          2194  2 !+
; 1274          2195  2 ! If we are to append to the file, position to EOF when opening it.
; 1275          2196  2 !-
; 1276          2197  2
; 1277          2198  2     IF (.OPEN_ARG_BLK [OPN$B_ACCESS] EQL OPN$K_ACC_APPEN) THEN CCB [RAB$V_EOF] = 1;
; 1278          2199  2
; 1279          2200  2 !+
; 1280          2201  2 ! Always turn off Locate mode.  This is done because GETs of variable length
; 1281          2202  2 ! records are to null fill the remainder of the buffer.  This cannot be done
; 1282          2203  2 ! if the data is in an RMS buffer in system space.
; 1283          2204  2 !-
; 1284          2205  2     CCB [RAB$V_LOC] = 0;
; 1285          2206  2 !+
; 1286          2207  2 ! Don't truncate the file on a PUT to its middle; give an error message
; 1287          2208  2 ! instead.
```

```
2 !-
2       CCB [RAB$V_TPT] = 0;
2 !+
2 ! Don't convert PUT to UPDATE except on VIRTUAL files.  If the user wants
2 ! to do an UPDATE to any other kind of file, he must issue an explicit
2 ! UPDATE statement.
2 !-
2       CCB [RAB$V_UIF] = (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU);
2 !+
2 ! Initialize the local NAM block and point the FAB to it so that
2 ! RMS will return the full name of the file being accessed.  This
2 ! will improve error messages.
2 !-
2       $NAM_INIT (NAM = NAM_BLOCK);
2       NAM_BLOCK [NAM$L_RSA] = NAM_BLOCK [NAM$L_ESA] = FILE_NAME;
2       NAM_BLOCK [NAM$B_RSS] = NAM_BLOCK [NAM$B_ESS] = NAM$C_MAXRSS;
2       FAB [FAB$L_NAM] = NAM_BLOCK;
2 !+
2 ! Initialize the local FHC XAB and point the FAB to it.  This is
2 ! used to get the size of the longest record in an old file, so
2 ! that the buffer size can be defaulted properly.
2 ! Also initialize the SUMMARY XAB and add it into the XAB chain.
2 ! This will be used to set total number of keys for indexed files.
2 !-
2       $XABFHC_INIT (XAB = XABFHC, NXT = XABSUM);
2       $XABSUM_INIT (XAB = XABSUM);
2       XABSUM [XAB$L_NXT] = 0;                         ! make sure SUM is end of chain
2       FAB [FAB$L_XAB] = XABFHC;                       ! link FAB to XAB
2 !+
2 ! Initialize the KEY XABs if any keys are provided in this call to OPEN.
2 !-
2
3       IF (ACTUALCOUNT () GEQ 2)
2       THEN
3           BEGIN
3
3           LOCAL
3               KEY_PTR : REF BLOCK [, BYTE] FIELD (OPN$KEY_BLOCK),
3               XABKEY : REF $XABKEY_DECL;
3
3 !+
3 ! The keys are arranged in order by key number.
3 ! They are allocated in reverse order because RMS-32 requires them to
3 ! be chained in forward order.
3 !-
3
3           DECR KEYNO FROM .KEY_INFO_BLK [KEYH$B_KEYNUM] - 1 TO 0 DO
4               BEGIN
4               KEY_PTR = (.KEYNO*.KEY_INFO_BLK [KEYH$B_LEN]) + KEYH$K_LENGTH + .KEY_INFO_BLK;
5               BEGIN
5
5               LOCAL
5                   GET_VM_STATUS;
5
5               GET_VM_STATUS = LIB$GET_VM (%REF (XAB$C_KEYLEN), XABKEY);
5
5               IF ( NOT .GET_VM_STATUS) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
```

```
5
4               END;
4               $XABKEY_INIT (XAB = .XABKEY);
4       !+
4       ! Fill in the key XAB from the parameter.
4       !-
4               XABKEY [XAB$B_REF] = .KEYNO;            ! Key of reference
4               XABKEY [XAB$W_POS0] = .KEY_PTR [KEY$W_OFFSET];      ! Position in record
4               XABKEY [XAB$B_SIZ0] = .KEY_PTR [KEY$B_LEN]; ! Total key length
4                                                           ! if NOT segmented;
4                                                           ! otherwise length of
4                                                           ! first segment.
4               XABKEY [XAB$V_CHG] = .KEY_PTR [KEY$V_CHG];  ! Key can be changed
4               XABKEY [XAB$V_DUP] = .KEY_PTR [KEY$V_DUP];  ! Key can be duplicated
4
4       !+
4       ! also fill in the summary XAB.
4       !-
4               XABSUM [XAB$B_NOK] = .KEY_INFO_BLK [KEYH$B_KEYNUM];
4
4       !+
4       ! The V2 compiler supports segmented keys, the V1 compiler does not.  If this
4       ! is a V2 program, check for multiple key segments.
4       !-
5               IF (.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE)
4               THEN
5                   BEGIN
5                   IF .KEY_PTR [KEY$B_NUM_SEG] GTR 0
5                   THEN
6                       BEGIN
6                       INCR KEY_NUM FROM 1 TO .KEY_PTR [KEY$B_NUM_SEG] DO
7                           BEGIN
7                           CASE .KEY_NUM FROM 1 TO 7 OF
7                           SET
7                               [1]:
8                               BEGIN
8                               XABKEY [XAB$B_SIZ1] = .KEY_PTR [KEY$B_LEN1];
8                               XABKEY [XAB$W_POS1] = .KEY_PTR [KEY$W_OFFSET1];
7                               END;
7
7                               [2]:
8                               BEGIN
8                               XABKEY [XAB$B_SIZ2] = .KEY_PTR [KEY$B_LEN2];
8                               XABKEY [XAB$W_POS2] = .KEY_PTR [KEY$W_OFFSET2];
7                               END;
7
7                               [3]:
8                               BEGIN
8                               XABKEY [XAB$B_SIZ3] = .KEY_PTR [KEY$B_LEN3];
8                               XABKEY [XAB$W_POS3] = .KEY_PTR [KEY$W_OFFSET3];
7                               END;
7
7                               [4]:
8                               BEGIN
8                               XABKEY [XAB$B_SIZ4] = .KEY_PTR [KEY$B_LEN4];
8                               XABKEY [XAB$W_POS4] = .KEY_PTR [KEY$W_OFFSET4];
7                               END;
```

```
7
7                                   [5]:
8                                   BEGIN
8                                   XABKEY [XAB$B_SIZ5] = .KEY_PTR [KEY$B_LEN5];
8                                   XABKEY [XAB$W_POS5] = .KEY_PTR [KEY$W_OFFSET5];
7                                   END;
7
7                                   [6]:
8                                   BEGIN
8                                   XABKEY [XAB$B_SIZ6] = .KEY_PTR [KEY$B_LEN6];
8                                   XABKEY [XAB$W_POS6] = .KEY_PTR [KEY$W_OFFSET6];
7                                   END;
7
7                                   [7]:
8                                   BEGIN
8                                   XABKEY [XAB$B_SIZ7] = .KEY_PTR [KEY$B_LEN7];
8                                   XABKEY [XAB$W_POS7] = .KEY_PTR [KEY$W_OFFSET7];
7                                   END;
7
7                                   [OUTRANGE]:
7                                   BAS$$STOP_IO (BAS$K_NOTIMP);
7
7                               TES;
6                               END;
5                           END;
4                       END;
4
4  !+
4  ! Fill in the data type field.  The compiler deals in VAX standard
4  ! codes for the data types, so we must translate them to the RMS   .
4  ! codes.  VAX codes which do not translate are treated as strings.
4  !-
4                   XABKEY [XAB$B_DTP] =
5                   BEGIN
5
5                   CASE .KEY_PTR [KEY$B_DTYPE] FROM DSC$K_DTYPE_WU TO DSC$K_DTYPE_P OF
5                       SET
5
5                       [DSC$K_DTYPE_WU] :
5                           XAB$C_BN2;
5
5                       [DSC$K_DTYPE_LU] :
5                           XAB$C_BN4;
5
5                       [DSC$K_DTYPE_W] :
5                           XAB$C_IN2;
5
5                       [DSC$K_DTYPE_L] :
5                           XAB$C_IN4;
5
5                       [DSC$K_DTYPE_T] :
5                           XAB$C_STG;
5
5                       [DSC$K_DTYPE_P] :
5                           XAB$C_PAC;
5
5                       [INRANGE, OUTRANGE] :
```

```
; 1459          2380  5                          XAB$C_STG;
; 1460          2381  5                      TES
; 1461          2382  5
; 1462          2383  4                  END;
; 1463          2384  4  !+
; 1464          2385  4  ! define the total key size of this key XAB, since RMS
; 1465          2386  4  ! doesn't set it for us in the case of a $CREATE.
; 1466          2387  4  !-
; 1467          2388  5                  XABKEY [XAB$B_TKS] = (IF .KEY_PTR [KEY$B_NUM_SEG] EQL 0
; 1468          2389  5                                        THEN
; 1469          2390  5                                            .XABKEY [XAB$B_SIZ0]
; 1470          2391  5                                        ELSE
; 1471          2392  5                                            .XABKEY [XAB$B_SIZ0] + .XABKEY [XAB$B_SIZ1] +
; 1472          2393  5                                            .XABKEY [XAB$B_SIZ2] + .XABKEY [XAB$B_SIZ3] +
; 1473          2394  5                                            .XABKEY [XAB$B_SIZ4] + .XABKEY [XAB$B_SIZ5] +
; 1474          2395  4                                            .XABKEY [XAB$B_SIZ6] + .XABKEY [XAB$B_SIZ7]);
; 1475          2396  4
; 1476          2397  4  !+
; 1477          2398  4  ! Link this key XAB to the FAB.
; 1478          2399  4  ! The XAB chain is set up so that key XABs are placed, in order,
; 1479          2400  4  ! in front of any other existing XABs on the file.
; 1480          2401  4  !-
; 1481          2402  4                  XABKEY [XAB$L_NXT] = .FAB [FAB$L_XAB];
; 1482          2403  4                  FAB [FAB$L_XAB] = .XABKEY;
; 1483          2404  3                  END;
; 1484          2405  3
; 1485          2406  2              END;
; 1486          2407  2
; 1487          2408  2  !<BLF/PAGE>
```

```
2 !+
2 ! Set up the FAB from the parent file if this is a multi-stream connect.
2 ! This is needed because we will not be doing an OPEN.
2 !-
2
3           IF (.CCB [LUB$V_M_STREAM])
2           THEN
3               BEGIN
3
3               LOCAL
3                   PARENT_IFI,
3                   PARENT_ORG,
3                   PARENT_MRS,
3                   PARENT_RAT,
3                   PARENT_RFM,
3                   PARENT_BKS,
3                   PARENT_BLS,
3                   CONNECTED,
3                   OUR_CCB : REF BLOCK [, BYTE];
3
3               OUR_CCB = .CCB;
3               BAS$$CB_PUSH (.OPEN_ARG_BLK [OPN$W_CONNECT], LUB$K_LUN_MIN);
3               PARENT_IFI = (IF (.CCB [LUB$V_OPENED]) THEN .CCB [LUB$W_IFI] ELSE 0);
3               PARENT_ORG = .CCB [LUB$B_ORGAN];
3               PARENT_MRS = .CCB [LUB$W_RBUF_SIZE];
3               PARENT_RAT = .CCB [LUB$B_RAT];
3               PARENT_RFM = .CCB [LUB$B_RFM];
3               PARENT_BKS = .CCB [LUB$B_BKS];
3               PARENT_BLS = .CCB [LUB$W_BLS];
3 !+
3 ! Mark that there may be a connect to this file, for CLOSE.
3 ! If this is already connected; a child instead of a parent; then remember to
3 ! complain later.
3 !-
3               IF .CCB [LUB$V_M_STREAM] EQL 1
3               THEN
3                   CONNECTED = 1
3               ELSE
4                   BEGIN
4                   CONNECTED = 0;
4                   CCB [LUB$V_M_STR_C] = 1;
3                   END;
3
3               BAS$$CB_POP ();
3               CCB = .OUR_CCB;
3
3               IF (.CONNECTED) THEN BAS$$STOP_IO (BAS$K_INVFILOPT);
3
3               IF (.PARENT_IFI EQL 0) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
3
3               IF (.PARENT_ORG NEQ LUB$K_ORG_INDEX) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
3
3               FAB [FAB$W_IFI] = .PARENT_IFI;
3               FAB [FAB$W_MRS] = .PARENT_MRS;
3               FAB [FAB$B_RAT] = .PARENT_RAT;
3               FAB [FAB$B_RFM] = .PARENT_RFM;
3               FAB [FAB$B_BKS] = .PARENT_BKS;
```

```
; 1546          2466  3         FAB [FAB$W_BLS] = .PARENT_BLS;
; 1547          2467  2         END;
; 1548          2468  2
; 1549          2469  2 !<BLF/PAGE>
```

```
2 !+
2 ! If the USEROPEN value is non-zero, call the user-supplied procedure
2 ! to do the $OPEN and $CONNECT.  It will return an RMS status code
2 ! as its value.  Otherwise we do the $OPEN and $CONNECT ourselves.
2 ! If we call USEROPEN, set a flag in the LUB to help software support
2 ! if they get an SPR.
2 !-
2
3     IF (.OPEN_ARG_BLK [OPN$A_USEROPEN] NEQA 0)
2     THEN
3         BEGIN
3         CCB [LUB$V_USEROPEN] = 1;
3         OPEN_STATUS = (.OPEN_ARG_BLK [OPN$A_USEROPEN]) (.FAB, .CCB, %REF (.CCB [LUB$W_LUN]));
3         CONNECT_STATUS = SS$_NORMAL;
3
4         IF ( NOT .OPEN_STATUS)
3         THEN
4             BEGIN
4 !+
4 ! Try to construct the correct values for OPEN_STATUS and CONNECT_STATUS.
4 !-
4
5             IF (.FAB [FAB$L_STS])
4             THEN
5                 BEGIN
5 !+
5 ! The $OPEN was OK, how about the $CONNECT.
5 !-
5
6                 IF ( NOT .CCB [RAB$L_STS])
5                 THEN
6                     BEGIN
6 !+
6 ! The $OPEN succeeded but the $CONNECT failed.
6 !-
6                     OPEN_STATUS = .FAB [FAB$L_STS];
6                     CONNECT_STATUS = .CCB [RAB$L_STS];
6                     END
5                 ELSE
5 !+
5 ! Both the RMS values look ok, just signal the error.
5 !-
5                     LIB$STOP (.OPEN_STATUS);
5
4                 END;
4
3             END;
3
3         END
2     ELSE
3         BEGIN
3 !+
3 ! Not USEROPEN.  If an old file is explicitly wanted
3 ! (user said FOR INPUT) do a $OPEN.  Otherwise do a $CREATE.
3 ! However, if this is just a CONNECT, do a $DISPLAY to set up the XABs.
3 !-
3
```

```
4          IF (.CCB [LUB$V_M_STREAM])
3          THEN
4              OPEN_STATUS = $DISPLAY (FAB = .FAB)
4
3          ELSE
4              BEGIN
4
4 !+
4 ! check for a terminal format file on a terminal device,
4 ! and change to PRN format if so.
4 ! This is so that the terminal is forcible.
4 !-
4              OPEN_STATUS = $PARSE (FAB = .FAB);
4
5              IF (.OPEN_STATUS)
4              THEN
5                  BEGIN
5
7                  IF (((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)     !
7                      AND (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_TERMI) !
7                      AND (.OPEN_ARG_BLK [OPN$B_RFM] EQL OPN$K_RFM_DEFAU) !
6                      AND (.OPEN_ARG_BLK [OPN$B_RAT] EQL OPN$K_RAT_DEFAU))
5                  THEN
6                      BEGIN
6 !+
6 ! Turn off CR and turn on PRN.
6 !-
6                      FAB [FAB$V_CR] = 0;
6                      FAB [FAB$V_PRN] = 1;
6 !+
6 ! Change from VAR to VFC, so we can support PRN.
6 !-
6                      FAB [FAB$B_RFM] = FAB$C_VFC;
6                      FAB [FAB$B_FSZ] = 2;
6 !+
6 ! Any VFC field provided by the user is ignored.
6 !-
6                      CCB [RAB$L_RHB] = CCB [LUB$W_BAS_VFC];
6 !+
6 ! make the terminal forcible.
6 !-
6                      CCB [LUB$V_FORCIBLE] = 1;
5                      END;
5
4                  END;
4
5              IF (.CCB [LUB$V_OLD_FILE])
4              THEN
5                  OPEN_STATUS = $OPEN (FAB = .FAB)
4              ELSE
4                  OPEN_STATUS = $CREATE (FAB = .FAB);
3              END;
3
3          IF (.OPEN_STATUS) THEN CONNECT_STATUS = $CONNECT (RAB = .CCB);
3
2          END;
2
```

```
2 !+
2 ! If FAB$V_CIF is set, set the 'old file' bit in the LUB based on
2 ! whether or not an existing file was found.  If this is a CONNECT, always
2 ! set 'old file', for error checking.
2 !-
2
3     IF (.CCB [LUB$V_M_STREAM] OR (.FAB [FAB$V_CIF] AND (.FAB [FAB$L_STS] NEQU RMS$_CREATED)))
2     THEN
2         CCB [LUB$V_OLD_FILE] = 1;
2
2 !+
2 ! Store away the Directory ID in case CLOSE needs to delete the file.
2 ! Also save the IFI.
2 !-
2     CH$MOVE (NAM$S_DID, NAM_BLOCK [NAM$W_DID], CCB [LUB$W_DID]);
2     CCB [LUB$W_IFI] = .FAB [FAB$W_IFI];
2 !+
2 ! If we have an expanded name string or a resultant name string, point
2 ! the LUB to it instead of the user-supplied name, to improve error
2 ! messages.
2 !-
2
3     IF (.NAM_BLOCK [NAM$B_RSL] NEQA 0)
2     THEN
3         BEGIN
3         CCB [LUB$A_RSN] = .NAM_BLOCK [NAM$L_RSA];
3         CCB [LUB$B_RSL] = .NAM_BLOCK [NAM$B_RSL];
3         END
2     ELSE
2
3         IF (.NAM_BLOCK [NAM$B_ESL] NEQA 0)
2         THEN
3             BEGIN
3             CCB [LUB$A_RSN] = .NAM_BLOCK [NAM$L_ESA];
3             CCB [LUB$B_RSL] = .NAM_BLOCK [NAM$B_ESL];
2             END;
2
2 !+
2 ! From here to the end of this routine, a call to BAS$$STOP_IO prints the
2 ! expanded or resultant name from RMS, hence any error detection should be
2 ! done after this point, if this is reasonable.
2 !-
2 !+
2 ! If OPEN or CREATE got an error, give an appropriate error message.
2 !-
2
2     IF ( NOT .OPEN_STATUS) THEN BAS$$STOP_IO (BAS$K_IOERR_OPE);
2
2 !+
2 ! Since the CCB has been RMS OPENed, CLOSE it if we detect an error hereafter.
2 ! Note that this will actually do a DISCONNECT if LUB$V_M_STREAM is set.
2 !-
2     UNWIND_ACTION = UNWIND_CLOSE;
2
2     IF ( NOT .CONNECT_STATUS) THEN BAS$$STOP_IO (BAS$K_IOERR_CON);
2
2 !+
2 ! If the device opened is a terminal, set the TERM_DEV bit in the LUB
2 ! and allocate a prompt buffer.  Also, set the margin to infinite.
2 ! Note that CONNECT is not permitted on terminals (because indexed only).
2 !-
2
3     IF ( NOT .CCB [LUB$V_M_STREAM])
2     THEN
3         BEGIN
3
4         IF ((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)
3         THEN
4             BEGIN
4
4             LOCAL
4                 GET_VM_RESULT;
4
4             CCB [LUB$V_TERM_DEV] = 1;
4
5             IF ( NOT (GET_VM_RESULT = LIB$GET_VM (%REF (LUB$K_PBUF_SIZ), CCB [RAB$L_PBF])))
4             THEN
4                 BAS$$STOP_IO (BAS$K_MAXMEMEXC);
4
4             CCB [RAB$B_PSZ] = 0;
4             CCB [RAB$V_PMT] = 1;
4
5             IF (.OPEN_ARG_BLK [OPN$W_RECORDSIZ] EQL 0)
4             THEN CCB [LUB$W_D_MARGIN] = .FAB [FAB$W_BLS];
4
4             CCB [LUB$W_R_MARGIN] = 0;
4             CCB [LUB$V_NOMARGIN] = 1;
3             END;
3
3 !+
3 ! if the device opened is a line printer, set the margin to the printer's
3 ! width (if the user didn't specify RECORDSIZE).
3 !
3 ! the way we check to see if its a line printer is to see if it IS a
3 ! carriage-control device, IS an output device, IS record-oriented,
3 ! IS NOT an input device, IS NOT a mailbox, and IS NOT a terminal.
3 !-
4         IF (    ( (.FAB [FAB$L_DEV] AND DEV$M_CCL) NEQ 0 ) AND
4                 ( (.FAB [FAB$L_DEV] AND DEV$M_ODV) NEQ 0 ) AND
4                 ( (.FAB [FAB$L_DEV] AND DEV$M_REC) NEQ 0 ) AND
4                 ( (.FAB [FAB$L_DEV] AND DEV$M_IDV) EQL 0 ) AND
4                 ( (.FAB [FAB$L_DEV] AND DEV$M_MBX) EQL 0 ) AND
4                 ( (.FAB [FAB$L_DEV] AND DEV$M_TRM) EQL 0 )      )
3         THEN
4             BEGIN
4
5             IF (.OPEN_ARG_BLK [OPN$W_RECORDSIZ] EQL 0)
4             THEN CCB [LUB$W_D_MARGIN] = .FAB [FAB$W_BLS];
4
4             CCB [LUB$W_R_MARGIN] = 0;
4             CCB [LUB$V_NOMARGIN] = 1;
3             END;
3
2         END;
```

```
; 1779          2698  2
; 1780          2699  2 !<BLF/PAGE>
```

```
2  !+
2  ! Version 2 stores the MAP buffer size and the recordsize in the OPEN block.
2  ! If this is a version 2 program, make the check here for a recordsize longer
2  ! than the user's buffer.
2  !-
2
2          IF (.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE) AND
3             (.OPEN_ARG_BLK [OPN$A_MAP] NEQA 0)
2          THEN
3              IF (.OPEN_ARG_BLK [OPN$W_MAP_SIZE] LSSU .OPEN_ARG_BLK [OPN$W_RECORDSIZ])
2              THEN
2                  BAS$$STOP_IO (BAS$K_RECOVEMAP);
2
2  !+
2  ! If the file just opened was already in existence, perform
2  ! consistency checks between the file's attributes and the
2  ! open parameters.
2  !-
2
3      IF (.CCB [LUB$V_OLD_FILE])
2      THEN
3          BEGIN
3  !+
3  ! Organization check: If the user did not specify an organization
3  ! with this OPEN, use the attributes from the file.  Otherwise,
3  ! check that the user's specification agrees with the file.
3  ! Store the organization in the LUB if the user did not specify one.
3  !-
3
3          CASE (.OPEN_ARG_BLK [OPN$B_ORG]) FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
3              SET
3
3              [OPN$K_ORG_TERMI] :
4                  BEGIN
4                  CCB [LUB$B_ORGAN] = LUB$K_ORG_TERMI;
4
4                  IF (.FAB [FAB$B_ORG] NEQ FAB$C_SEQ) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
3                  END;
3
3              [OPN$K_ORG_VIRTU] :
4                  BEGIN
4                  CCB [LUB$B_ORGAN] = LUB$K_ORG_VIRTU;
4
4                  IF (.FAB [FAB$B_ORG] NEQ FAB$C_SEQ) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
3                  END;
3
3              [OPN$K_ORG_SEQUE] :
4                  BEGIN
4                  CCB [LUB$B_ORGAN] = LUB$K_ORG_SEQUE;
4
4                  IF (.FAB [FAB$B_ORG] NEQ FAB$C_SEQ) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
3                  END;
3
3              [OPN$K_ORG_RELAT] :
```

```
4                    BEGIN
4                    CCB [LUB$B_ORGAN] = LUB$K_ORG_RELAT;
4
4                    IF (.FAB [FAB$B_ORG] NEQ FAB$C_REL) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
3                    END;
3
3               [OPN$K_ORG_INDEX] :
4                    BEGIN
4                    CCB [LUB$B_ORGAN] = LUB$K_ORG_INDEX;
4
4                    IF (.FAB [FAB$B_ORG] NEQ FAB$C_IDX) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
3                    END;
3
3               [OPN$K_ORG_UNDEF] :
3 !+
3 ! If the user does not specify the organization, accept whatever
3 ! is in the file.
3 !-
4                    BEGIN
4
4                    SELECTONE (.FAB [FAB$B_ORG]) OF
4                        SET
4
4                        [FAB$C_SEQ] :
4                            CCB [LUB$B_ORGAN] = LUB$K_ORG_SEQUE;
4
4                        [FAB$C_REL] :
4                            CCB [LUB$B_ORGAN] = LUB$K_ORG_RELAT;
4
4                        [FAB$C_IDX] :
4                            CCB [LUB$B_ORGAN] = LUB$K_ORG_INDEX;
4
4                        [OTHERWISE] :
4                            BAS$$STOP_IO (BAS$K_FILATTNOT);
4                        TES;
4
3                    END;
3               TES;
3
3 !+
3 ! Verify that the user-declared bucket size agrees with the file.
3 ! If the user specified zero, we accept the file attribute.
3 !-
3
5           IF ((.OPEN_ARG_BLK [OPN$W_BUCKETSIZ] NEQ 0)       !
6               AND ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_RELAT)         !
4               OR (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX)))
3           THEN
3
3               IF (.BKS NEQ .FAB [FAB$B_BKS]) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
3
3 !+
3 ! Verify that the user-declared block size agrees with the file.
3 ! If the user specified zero, we accept the file attribute.
3 !-
```

```
3
4        IF ((.OPEN_ARG_BLK [OPN$W_BLOCKSIZE] NEQ 0) AND (.FAB [FAB$B_ORG] EQL FAB$C_SEQ))
3        THEN
3
3            IF (.BLS NEQ .FAB [FAB$W_BLS]) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
3
3  !+
3  ! Verify that the user-declared record size agrees with the file.
3  ! If the user specified zero, we accept the file attribute.
3  !-
3        CCB [LUB$W_RBUF_SIZE] = MAXU (  .FAB [FAB$W_MRS],
3                                        .XABFHC [XAB$W_LRL]     );
3
3  !+
3  ! If the file's purported record size is zero, use a reasonable size.
3  !-
3
3        IF (.CCB [LUB$W_RBUF_SIZE] EQL 0) THEN CCB [LUB$W_RBUF_SIZE] = .RSZ;
3
3  !+
3  ! if the record buffer is still zero-length at this point, try looking
3  ! at FAB$W_BLS.
3  !-
3        IF (.CCB [LUB$W_RBUF_SIZE] EQL 0) THEN CCB [LUB$W_RBUF_SIZE] = .FAB [FAB$W_BLS];
3
3  !+
3  ! If the user specifies a record size, make sure that the file has that record size
3  ! (for files with fixed-length records).  For files with variable-length
3  ! records,  the check is only necessary if the MRS is set and the file is opened
3  ! for write access.  No checking is done here for variable length record files
3  ! since RMS will catch the error at write time.
3  !-
4        IF (.OPEN_ARG_BLK [OPN$W_RECORDSIZ] NEQ 0)
3        THEN
4            IF (.FAB [FAB$B_RFM] EQL FAB$C_FIX)
3            THEN
3                IF (.RSZ NEQ .CCB [LUB$W_RBUF_SIZE]) THEN BAS$$STOP_IO (BAS$K_BADRECVAL);
3
3  !+
3  ! for V2 programs, check the MAP size too, since if no RECORDSIZE is specified
3  ! but a MAP size is, OPEN_ARG_BLK[OPN$W_RECORDSIZ] will still be zero.  This
3  ! is different from V1, where the compiler would give us the MAP value in the
3  ! RECORDSIZE field.
3  !-
4        IF ((.OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE) AND
4                (.OPEN_ARG_BLK [OPN$A_MAP] NEQA 0))
3        THEN
3            IF (.RSZ LSS .CCB [LUB$W_RBUF_SIZE]) THEN BAS$$STOP_IO (BAS$K_BADRECVAL);
3
3  !+
3  ! The buffer size we actually use must be at least as large as the user
3  ! declared.
3  !
3  ! If the user did not declare a buffer size, then we want to
3  ! use the maximum record size from the file.  If no MRS is set for the file and
3  ! the user did not declare a buffer size,  then we will use the max of the
3  ! default recordsize and the longest record length.  This guards against
```

```
3 ! opening a file with only short records in it (and MRS=0) and being unable
3 ! to write a large record.  1-113
3 !-
4       IF ((.NO_MAP_REC_SPECIFIED)  AND (.FAB [FAB$W_MRS] NEQ 0))
3       THEN
3           0
3       ELSE
3           CCB [LUB$W_RBUF_SIZE] = MAXU (.CCB [LUB$W_RBUF_SIZE], .RSZ);
3 !+
3 ! If the user is using a MAP, the record size must not be longer than the
3 ! space in the map.
3 !-
3
4       IF ((.OPEN_ARG_BLK [OPN$A_MAP] NEQA 0) AND          !
4           (.CCB [LUB$W_RBUF_SIZE] LSSU .OPEN_ARG_BLK [OPN$W_RECORDSIZ]))
3       THEN
3           BAS$$STOP_IO (BAS$K_BADRECVAL);
3
3 !+
3 ! If the organization is virtual, the buffer size must be at least 512 bytes.
3 !-
3
4       IF ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU) AND          !
4           (.CCB [LUB$W_RBUF_SIZE] LSSU 512))
3       THEN
3           BAS$$STOP_IO (BAS$K_BADRECVAL);
3
3 !+
3 ! Verify that the user-declared record format agrees with the file.
3 ! If the user did not specify a record format, we accept the file
3 ! attribute.
3 !-
3
3       CASE (.OPEN_ARG_BLK [OPN$B_RFM]) FROM OPN$K_RFM_DEFAU TO OPN$K_RFM_STREA OF
3           SET
3
3           [OPN$K_RFM_DEFAU] :
3 !+
3 ! Don't check for virtual and undefined.
3 !-
3
5               IF ((.OPEN_ARG_BLK [OPN$B_ORG] NEQ OPN$K_ORG_VIRTU) AND (.OPEN_ARG_BLK [OPN$B_ORG] NEQ
4                   OPN$K_ORG_UNDEF))
3               THEN
3 !+
3 ! If the device is a terminal, it is opened in PRN format.
3 !-
3
5                   IF (((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)          !
5                       AND (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_TERMI)      !
5                       AND (.OPEN_ARG_BLK [OPN$B_RFM] EQL OPN$K_RFM_DEFAU)      !
4                       AND (.OPEN_ARG_BLK [OPN$B_RAT] EQL OPN$K_RAT_DEFAU))
3                   THEN
4                       BEGIN
4
4                       IF (.FAB [FAB$B_RFM] NEQ FAB$C_VFC) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
4
```

```
                        IF (.FAB [FAB$B_FSZ] NEQ 2) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);

                        END;

            [OPN$K_RFM_FIXED] :

                IF (.FAB [FAB$B_RFM] NEQ FAB$C_FIX) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);

            [OPN$K_RFM_VARIA] :

                IF ((.FAB [FAB$B_RFM] NEQ FAB$C_VAR) AND (.FAB [FAB$B_RFM] NEQ FAB$C_VFC))
                THEN
                    BAS$$STOP_IO (BAS$K_FILATTNOT);

            [OPN$K_RFM_VFC] :
                BEGIN

                IF (.FAB [FAB$B_RFM] NEQ FAB$C_VFC) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);

                IF (.FAB [FAB$B_FSZ] NEQ .OPEN_ARG_BLK [OPN$B_FSZ]) THEN BAS$$STOP_IO (BAS$K_FILATTNOT);

                END;

            [OPN$K_RFM_STREA, OUTRANGE] :
                BAS$$STOP_IO (BAS$K_FILATTNOT);
            TES;

!+
! Verify that the user-declared record attributes agree with the file.
!-

        CASE (.OPEN_ARG_BLK [OPN$B_RAT]) FROM OPN$K_RAT_DEFAU TO OPN$K_RAT_ANY OF
            SET

            [OPN$K_RAT_DEFAU] :
!+
! If the device is a terminal, it is opened in PRN format.
!-

                IF (((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)        !
                    AND (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_TERMI) !
                    AND (.OPEN_ARG_BLK [OPN$B_RFM] EQL OPN$K_RFM_DEFAU) !
                    AND (.OPEN_ARG_BLK [OPN$B_RAT] EQL OPN$K_RAT_DEFAU))
                THEN
                    BEGIN

                    IF ( NOT .FAB [FAB$V_PRN]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT)

                    END
                ELSE
!+
! If the organization is virtual, it should have RAT NONE or CR.
!-

                    IF (.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU)
                    THEN
                        BEGIN
```

```
; 2067          2985  4
; 2068          2986  4                             IF (.FAB [FAB$V_FTN] OR .FAB [FAB$V_PRN]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT)
; 2069          2987  4
; 2070          2988  4                             END
; 2071          2989  3                         ELSE
; 2072          2990  3
; 2073          2991  3                             IF ( NOT .FAB [FAB$V_CR]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2074          2992  3
; 2075          2993  3                 [OPN$K_RAT_FORTR] :
; 2076          2994  3
; 2077          2995  3                     IF ( NOT .FAB [FAB$V_FTN]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2078          2996  3
; 2079          2997  3                 [OPN$K_RAT_CRLF] :
; 2080          2998  3
; 2081          2999  3                     IF ( NOT .FAB [FAB$V_CR]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2082          3000  3
; 2083          3001  3                 [OPN$K_RAT_NONE] :
; 2084          3002  3
; 2085          3003  4                     IF (.FAB [FAB$V_CR] OR .FAB [FAB$V_FTN] OR .FAB [FAB$V_PRN])
; 2086          3004  3                     THEN
; 2087          3005  3                         BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2088          3006  3
; 2089          3007  3                 [OPN$K_RAT_PRINT] :
; 2090          3008  3
; 2091          3009  3                     IF ( NOT .FAB [FAB$V_PRN]) THEN BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2092          3010  3
; 2093          3011  3                 [OPN$K_RAT_ANY] :
; 2094          3012  4                     BEGIN
; 2095          3013  4                     0
; 2096          3014  3                     END;
; 2097          3015  3
; 2098          3016  3                 [OUTRANGE] :
; 2099          3017  3                     BAS$$STOP_IO (BAS$K_RECATTNOT);
; 2100          3018  3                 TES;
; 2101          3019  3
; 2102          3020  3 !+
; 2103          3021  3 ! Mark the file as being in PRN format, if it is.
; 2104          3022  3 !-
; 2105          3023  3
; 2106          3024  3             IF (.FAB [FAB$V_PRN]) THEN CCB [LUB$V_PRN] = 1;
; 2107          3025  3
; 2108          3026  3 !+
; 2109          3027  3 ! Check keys for indexed file organization.
; 2110          3028  3 !-
; 2111          3029  3
; 2112          3030  4             IF ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_INDEX) AND (ACTUALCOUNT () GEQ 2))
; 2113          3031  3             THEN
; 2114          3032  4                 BEGIN
; 2115          3033  4
; 2116          3034  4                 LOCAL
; 2117          3035  4                     XABKEY : REF $XABKEY_DECL,
; 2118          3036  4                     KEY_PTR : REF BLOCK [0, BYTE] FIELD (OPN$KEY_BLOCK),
; 2119          3037  4                     KEYNO;
; 2120          3038  4
; 2121          3039  4 !+
; 2122          3040  4 ! The number of keys specified in the argument list must be less than or
; 2123          3041  4 ! equal to the number of keys in the file, as recorded in the summary XAB.
; 2124     3042  4 ! If the number of keys in the argument list is more than the number of
; 2125     3043  4 ! keys in the file, then give the user an error.
; 2126     3044  4 !-
; 2127     3045  5             IF (.KEY_INFO_BLK [KEYH$B_KEYNUM] GTRU .XABSUM [XAB$B_NOK])
; 2128     3046  4             THEN
; 2129     3047  4                 BAS$$STOP_IO (BAS$K_FILATTNOT);
; 2130     3048  4
; 2131     3049  4 !+
; 2132     3050  4 ! Each key must match the argument list.  The keys are stored in the XABKEY
; 2133     3051  4 ! blocks that were set up before the OPEN.  Search through the XABKEY blocks
; 2134     3052  4 ! matching each with the argument list.
; 2135     3053  4 !-
; 2136     3054  4             XABKEY = .FAB [FAB$L_XAB];
; 2137     3055  4
; 2138     3056  4             WHILE (.XABKEY NEQA 0) DO
; 2139     3057  5                 BEGIN
; 2140     3058  5
; 2141     3059  6                 IF (.XABKEY [XAB$B_COD] EQL XAB$C_KEY)
; 2142     3060  5                 THEN
; 2143     3061  6                     BEGIN
; 2144     3062  6                     KEYNO = .XABKEY [XAB$B_REF];
; 2145     3063  6                     KEY_PTR = (.KEY_INFO_BLK [KEYH$B_LEN]*.KEYNO) + KEYH$K_LENGTH + .KEY_INFO_BLK;
; 2146     3064  6
; 2147     3065  6                     !+
; 2148     3066  6                     ! Check the size, position, CHANGES and DUPLICATES of
; 2149     3067  6                     ! each key.
; 2150     3068  6                     !-
; 2151     3069  8                     IF ((.XABKEY [XAB$W_POS0] NEQ .KEY_PTR [KEY$W_OFFSET])        !
; 2152     3070  8                         OR (.XABKEY [XAB$B_SIZ0] NEQ .KEY_PTR [KEY$B_LEN])        !
; 2153     3071  8                         OR (.XABKEY [XAB$V_CHG] NEQ .KEY_PTR [KEY$V_CHG])         !
; 2154     3072  7                         OR (.XABKEY [XAB$V_DUP] NEQ .KEY_PTR [KEY$V_DUP]))
; 2155     3073  6                     THEN
; 2156     3074  6                         BAS$$STOP_IO (BAS$K_FILATTNOT);
; 2157     3075  6
; 2158     3076  6                     !+
; 2159     3077  6                     ! Check the data type of each key.
; 2160     3078  6                     ! Basic has no unsigned data type, so we must allow signed
; 2161     3079  6                     ! word & longword keys to open a file with unsigned word and
; 2162     3080  6                     ! longword keys.
; 2163     3081  6                     !-
; 2164     3082  6                     CASE .KEY_PTR [KEY$B_DTYPE] FROM DSC$K_DTYPE_W TO DSC$K_DTYPE_P OF
; 2165     3083  6                         SET
; 2166     3084  6
; 2167     3085  6                         !+
; 2168     3086  6                         ! user program has signed word key
; 2169     3087  6                         !-
; 2170     3088  6                         [DSC$K_DTYPE_W] :
; 2171     3089  6                             IF .XABKEY [XAB$B_DTP] NEQ XAB$C_IN2 ! signed word
; 2172     3090  6                            AND .XABKEY [XAB$B_DTP] NEQ XAB$C_BN2 ! unsigned word
; 2173     3091  6                             THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
; 2174     3092  6
; 2175     3093  6                         !+
; 2176     3094  6                         ! user program has signed longword key
; 2177     3095  6                         !-
; 2178     3096  6                         [DSC$K_DTYPE_L] :
; 2179     3097  6                             IF .XABKEY [XAB$B_DTP] NEQ XAB$C_IN4 ! signed longword
; 2180     3098  6                            AND .XABKEY [XAB$B_DTP] NEQ XAB$C_BN4 ! unsigned longword
```

```
 6                  THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
 6
 6              !+
 6              ! user program has text key
 6              !-
 6              [DSC$K_DTYPE_T] :
 6                  IF .XABKEY [XAB$B_DTP] NEQ XAB$C_STG ! string
 6                  THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
 6
 6              !+
 6              ! user program has packed decimal key
 6              !-
 6              [DSC$K_DTYPE_P] :
 6                  IF .XABKEY [XAB$B_DTP] NEQ XAB$C_PAC ! packed decimal
 6                  THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
 6
 6              !+
 6              ! user program has some other data type key
 6              !-
 6              [INRANGE, OUTRANGE] :
 6                  IF .XABKEY [XAB$B_DTP] NEQ XAB$C_STG ! string
 6                  THEN BAS$$STOP_IO (BAS$K_FILATTNOT);
 6
 6              TES;
 6
 6          !+
 6          ! For V2 programs, check all the key segments if there are
 6          ! multiples.
 6          !-
 6          IF .OPEN_ARG_BLK [OPN$B_CNT] GTR K_V1_BLK_SIZE
 6          THEN
 7              BEGIN
 7              IF .KEY_PTR [KEY$B_NUM_SEG] GTR 0
 7              THEN
 8                  BEGIN
 8                  INCR KEY_NUM FROM 1 TO .KEY_PTR [KEY$B_NUM_SEG] DO
 9                      BEGIN
 9                      CASE .KEY_NUM FROM 1 TO 7 OF
 9                          SET
 9                          [1]:
 9                          IF (.XABKEY [XAB$W_POS1] NEQ .KEY_PTR [KEY$W_OFFSET1]) OR
10                             (.XABKEY [XAB$B_SIZ1] NEQ .KEY_PTR [KEY$B_LEN1])
 9                          THEN
 9                              BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                          [2]:
 9                          IF (.XABKEY [XAB$W_POS2] NEQ .KEY_PTR [KEY$W_OFFSET2]) OR
10                             (.XABKEY [XAB$B_SIZ2] NEQ .KEY_PTR [KEY$B_LEN2])
 9                          THEN
 9                              BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                          [3]:
 9                          IF (.XABKEY [XAB$W_POS3] NEQ .KEY_PTR [KEY$W_OFFSET3]) OR
10                             (.XABKEY [XAB$B_SIZ3] NEQ .KEY_PTR [KEY$B_LEN3])
 9                          THEN
 9                              BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                              [4]:
 9                              IF (.XABKEY [XAB$W_POS4] NEQ .KEY_PTR [KEY$W_OFFSET4]) OR
10                                 (.XABKEY [XAB$B_SIZ4] NEQ .KEY_PTR [KEY$B_LEN4])
 9                              THEN
 9                                  BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                              [5]:
 9                              IF (.XABKEY [XAB$W_POS5] NEQ .KEY_PTR [KEY$W_OFFSET5]) OR
10                                 (.XABKEY [XAB$B_SIZ5] NEQ .KEY_PTR [KEY$B_LEN5])
 9                              THEN
 9                                  BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                              [6]:
 9                              IF (.XABKEY [XAB$W_POS6] NEQ .KEY_PTR [KEY$W_OFFSET6]) OR
10                                 (.XABKEY [XAB$B_SIZ6] NEQ .KEY_PTR [KEY$B_LEN6])
 9                              THEN
 9                                  BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                              [7]:
 9                              IF (.XABKEY [XAB$W_POS7] NEQ .KEY_PTR [KEY$W_OFFSET7]) OR
10                                 (.XABKEY [XAB$B_SIZ7] NEQ .KEY_PTR [KEY$B_LEN7])
 9                              THEN
 9                                  BAS$$STOP_IO (BAS$K_FILATTNOT);
 9
 9                              TES;
 8                          END;                    ! end of processing each segment
 7                      END;                        ! end of if segmented key
 6                  END;                            ! end of if V2 program
 6
 5              END;                                ! end of if key XAB
 5
 5          XABKEY = .XABKEY [XAB$L_NXT];
 4          END;                                    ! end of processing this XAB
 4
 3      END;                                        ! end of old indexed file processing
 3
 3    END                                           ! end of old file processing
 2  ELSE
 3    BEGIN
 3 !+
 3 ! This is a new file.  If it is organized relative or index, the user
 3 ! must have specified a record size.
 3 !-
 3
 3    IF (.RSZ EQL 0) THEN BAS$$STOP_IO (BAS$K_BADRECVAL) ELSE CCB [LUB$W_RBUF_SIZE] = .RSZ;
 3
 3 !+
 3 ! Make sure the LUB 'append' flag is off so FORTRAN BACKSPACE will work.
 3 !-
 3    CCB [LUB$V_APPEND] = 0;
 3 !+
 3 ! Set LUB$B_ORGAN based on the OPEN argument.  This is legitimate since
 3 ! we just created the file, so it must agree with the OPEN argument.
 3 !-
 3
 3    CASE .OPEN_ARG_BLK [OPN$B_ORG] FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
 3        SET
```

```
3
3             [OPN$K_ORG_TERMI] :
3                 CCB [LUB$B_ORGAN] = LUB$K_ORG_TERMI;
3
3             [OPN$K_ORG_VIRTU] :
3                 CCB [LUB$B_ORGAN] = LUB$K_ORG_VIRTU;
3
3             [OPN$K_ORG_SEQUE] :
3                 CCB [LUB$B_ORGAN] = LUB$K_ORG_SEQUE;
3
3             [OPN$K_ORG_RELAT] :
3                 CCB [LUB$B_ORGAN] = LUB$K_ORG_RELAT;
3
3             [OPN$K_ORG_INDEX] :
3                 CCB [LUB$B_ORGAN] = LUB$K_ORG_INDEX;
3
3             [OPN$K_ORG_UNDEF] :
3                 BAS$$STOP_IO (BAS$K_FILATTNOT);
3             TES;
3
3 !+
3 ! Don't allow an open with ACCESS READ to create a file.
3 !-
3
3         IF (.OPEN_ARG_BLK [OPN$B_ACCESS] EQL OPN$K_ACC_READ) THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
3
3 !+
3 ! A virtual file's record size must not be less than 512 bytes.
3 !-
3
4         IF ((.OPEN_ARG_BLK [OPN$B_ORG] EQL OPN$K_ORG_VIRTU) AND          !
4             (.RSZ [SS 512))
3         THEN
3             BAS$$STOP_IO (BAS$K_BADRECVAL);
3
2         END;                                        ! End of new file processing
2
2 !+
2 ! Validate the record format.  It must be one of those the run-time
2 ! library can process.  In particular, we don't permit UNDEFINED unless
2 ! the organization is UNDEFINED.
2 !-
2
3     IF (.OPEN_ARG_BLK [OPN$B_ORG] NEQ OPN$K_ORG_UNDEF)
2     THEN
2
2         SELECTONE (.FAB [FAB$B_RFM]) OF
2             SET
2
2             [FAB$C_FIX, FAB$C_VAR, FAB$C_VFC] :          ! This is ok.
2             ;
2
2             [OTHERWISE] :
2                 BAS$$STOP_IO (BAS$K_FILATTNOT);
2             TES;
2
2 !+
2 ! Record the record attribute, record format block size and bucket
2 ! size in the LUB, for the FSP$ function and for connect.
2 !-
2     CCB [LUB$B_RAT] = .FAB [FAB$B_RAT];
2     CCB [LUB$B_RFM] = .FAB [FAB$B_RFM];
2     CCB [LUB$B_BKS] = .FAB [FAB$B_BKS];
2     CCB [LUB$W_BLS] = .FAB [FAB$W_BLS];
2     CCB [LUB$L_ALQ] = .FAB [FAB$L_ALQ];
2     CCB [LUB$L_REC_MAX] =
3     BEGIN
3
3     CASE .OPEN_ARG_BLK [OPN$B_ORG] FROM OPN$K_ORG_TERMI TO OPN$K_ORG_UNDEF OF
3         SET
3
3         [OPN$K_ORG_VIRTU] :
3             .FAB [FAB$L_ALQ];
3
3         [OPN$K_ORG_UNDEF] :
3             .FAB [FAB$L_MRN];
3
3         [INRANGE] :
3             0;
3         TES
3
2     END;
2 !+
2 ! Remember the device characteristics, in case the user calls
2 ! STATUS.
2 !-
2     L_STATUS = .FAB [FAB$L_DEV];
2 !+
2 ! Free the key XABs, since they were allocated from virtual storage.
2 !-
3     BEGIN
3
3     LOCAL
3         XABKEY : REF $XABKEY_DECL,
3         FREE_VM_STATUS,
3         XAB_PTR;
3
3     XAB_PTR = FAB [FAB$L_XAB];
3
3     WHILE (..XAB_PTR NEQA 0) DO
4         BEGIN
4         XABKEY = ..XAB_PTR;
4
5         IF (.XABKEY [XAB$B_COD] EQL XAB$C_KEY)
4         THEN
5             BEGIN
5 !+
5 ! We have found a key XAB.  Unlink it from the XAB chain and free it.
5 ! We have remembered XAB_PTR as its chain location.
5 !-
5             .XAB_PTR = .XABKEY [XAB$L_NXT];
5             FREE_VM_STATUS = LIB$FREE_VM (%REF (XAB$C_KEYLEN), XABKEY);
5
5             IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP_IO (BAS$K_PROLOSSOR);
```

```
5
5              END
4          ELSE
4              XAB_PTR = XABKEY [XAB$L_NXT];
4
3          END;
3
2      END;
2  !+
2  ! Allocate and clear a record buffer unless the user provided one.
2  !-
2
3      IF (.OPEN_ARG_BLK [OPN$A_MAP] EQLA 0)
2      THEN
3          BEGIN
3
3          LOCAL
3              GET_VM_RESULT;
3
3  !+
3  ! If recordsize is still 0 then signal an error.
3  !-
3
3          IF (.CCB [LUB$W_RBUF_SIZE] EQL 0) THEN BAS$$STOP_IO (BAS$K_BADRECVAL);
3
3          GET_VM_RESULT = LIB$GET_VM (%REF (.CCB [LUB$W_RBUF_SIZE]), CCB [LUB$A_RBUF_ADR]);
3
3          IF ( NOT .GET_VM_RESULT) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
3
3  !+
3  ! Make sure the buffer is null, in case the user fetches from it before
3  ! the first GET.
3  !-
3          CH$FILL (0, .CCB [LUB$W_RBUF_SIZE], .CCB [LUB$A_RBUF_ADR]);
3          END
2      ELSE
3          BEGIN
3          CCB [LUB$A_RBUF_ADR] = .OPEN_ARG_BLK [OPN$A_MAP];
3          CCB [LUB$V_USER_RBUF] = 1;
2          END;
2
2  !+
2  ! Allocate dynamic storage for the file name so that the name can be
2  ! used later for error diagnostics.  Point the LUB to the new location.
2  ! Indicate that the space pointed to must be deallocated when the file
2  ! is closed.
2  !-
3      BEGIN
3
3      LOCAL
3          GET_VM_RESULT,
3          OLD_ADDRESS;
3
3      OLD_ADDRESS = .CCB [LUB$A_RSN];
3      GET_VM_RESULT = LIB$GET_VM (%REF (.CCB [LUB$B_RSL]), CCB [LUB$A_RSN]);
3
3      IF ( NOT .GET_VM_RESULT) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
```

```
; 2466          3384  3
; 2467          3385  3       CH$MOVE (.CCB [LUB$B_RSL], .OLD_ADDRESS, .CCB [LUB$A_RSN]);
; 2468          3386  3       CCB [LUB$V_VIRT_RSN] = 1;
; 2469          3387  2       END;
; 2470          3388  2   !+
; 2471          3389  2   ! Set those RAB fields that seldom change.
; 2472          3390  2   !-
; 2473          3391  2       CCB [RAB$L_UBF] = .CCB [LUB$A_RBUF_ADR];
; 2474          3392  2       CCB [RAB$W_USZ] = .CCB [LUB$W_RBUF_SIZE];
; 2475          3393  2       CCB [LUB$A_UBF] = .CCB [LUB$A_RBUF_ADR];
; 2476          3394  2   !+
; 2477          3395  2   ! Clear LUB$A_FAB to indicate that the FAB is no longer present.
; 2478          3396  2   !-
; 2479          3397  2       CCB [LUB$A_FAB] = 0;
; 2480          3398  2       CCB [RAB$L_FAB] = 0;
; 2481          3399  2   !+
; 2482          3400  2   ! Indicate that the file is now open for BASIC.
; 2483          3401  2   !-
; 2484          3402  2       CCB [LUB$B_LANGUAGE] = LUB$K_LANG_BAS;
; 2485          3403  2       CCB [LUB$V_OPENED] = 1;
; 2486          3404  2   !+
; 2487          3405  2   ! Make sure that the BASIC exit handler will be called when the image
; 2488          3406  2   ! exits to purge the file's I/O buffers and close it, if necessary.
; 2489          3407  2   ! This can happen if the user's last PRINT statement ends with a
; 2490          3408  2   ! comma or a semicolon.
; 2491          3409  2   !-
; 2492          3410  2
; 2493          3411  2       IF ( NOT .BAS$$L_XIT_LOCK) THEN BAS$$DECL_EXITH ();
; 2494          3412  2
; 2495          3413  2   !+
; 2496          3414  2   ! Pop back previous LUB or indicate that no I/O statement
; 2497          3415  2   ! is currently active.
; 2498          3416  2   !-
; 2499          3417  2       BAS$$CB_POP ();
; 2500          3418  2       RETURN;
; 2501          3419  1       END;                                        ! end of BAS$OPEN


                                                                            .TITLE  BAS$OPEN
                                                                            .IDENT  \1-113\

                                                                            .PSECT  _BAS$DATA,NOEXE,  PIC,2

                                               00000000  00000 L_STATUS:
                                                                            .LONG   0                                        ;

                                                                            .EXTRN  BAS$$L_XIT_LOCK
                                                                            .EXTRN  LIB$STOP, BAS$$STOP
                                                                            .EXTRN  BAS$$STOP_IO, BAS$$CB_PUSH
                                                                            .EXTRN  BAS$$CB_POP, LIB$GET_VM
                                                                            .EXTRN  LIB$FREE_VM, BAS$$DECL_EXITH
                                                                            .EXTRN  OTS$$TAKE_LUN, OTS$$CLOSE_FILE
                                                                            .EXTRN  LIB$MATCH_COND, BAS$K_RECOVEMAP
                                                                            .EXTRN  BAS$K_PROCOSSOR
                                                                            .EXTRN  BAS$K_ILLILLACC
                                                                            .EXTRN  BAS$K_ILLIO_CHA
                                                                            .EXTRN  BAS$K_IO_CHAALR
```

```
                                                                  .EXTRN  BAS$K_FATSYSIO
                                                                  .EXTRN  BAS$K_FILATTNOT
                                                                  .EXTRN  BAS$K_RECATTNOT
                                                                  .EXTRN  BAS$K_MAXMEMEXC
                                                                  .EXTRN  BAS$K_BADRECVAL
                                                                  .EXTRN  BAS$K_NOTIMP, BAS$K_INVFILOPT
                                                                  .EXTRN  BAS$K_IO_CHANOT
                                                                  .EXTRN  BAS$K_REQRECSIZ
                                                                  .EXTRN  SYS$DISPLAY, SYS$PARSE
                                                                  .EXTRN  SYS$OPEN, SYS$CREATE
                                                                  .EXTRN  SYS$CONNECT

                                                                  .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                                  0FFC 00000      .ENTRY  BAS$OPEN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 1409
                                                                          R11                                         ;
                              5E      FDC0   CE   9E 00002        MOVAB   -576(SP), SP                                ;
                                        50   AE   7C 00007        CLRQ    UNWIND_CCB                                  ; 1446
                              6D      10E4   CF   DE 0000A        MOVAL   298$, (FP)                                  ;
                                        54   AE   D4 0000F        CLRL    UNWIND_ACTION                               ; 1500
                 50       04  AC             2C   C1 00012        ADDL3   #44, OPEN_ARG_BLK, R0                       ; 1505
                          44  AE             60   D0 00017        MOVL    (R0), CHANNEL                               ;
                 50       04  AC             05   C1 0001B        ADDL3   #5, OPEN_ARG_BLK, R0                        ; 1507
                          2C  AE             60   9E 00020        MOVAB   (R0), 44(SP)                                ;
                 11       2C  BE             06   E1 00024        BBC     #6, @44(SP), 2$                             ;
                                        2C   BE   95 00029        TSTB    @44(SP)                                     ; 1510
                                             07   18 0002C        BGEQ    1$                                          ;
                          44  AE        44   BE   32 0002E        CVTWL   @CHANNEL, CHANNEL                           ; 1512
                                             05   11 00033        BRB     2$                                          ;
                          44  AE        44   BE   D0 00035 1$:    MOVL    @CHANNEL, CHANNEL                           ; 1514
                              52        44   AE   D0 0003A 2$:    MOVL    CHANNEL, R2                                 ; 1520
                                             09   19 0003E        BLSS    3$                                          ;
               00000077       8F             52   D1 00040        CMPL    R2, #119                                    ;
                                             0B   15 00047        BLEQ    4$                                          ;
                              7E       00G   8F   9A 00049 3$:    MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                     ;
               00000000G      00             01   FB 0004D        CALLS   #1, BAS$$STOP                               ;
                                             52   D5 00054 4$:    TSTL    R2                                          ; 1526
                                             0B   12 00056        BNEQ    5$                                          ;
                              7E       00G   8F   9A 00058        MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                     ;
               00000000G      00             01   FB 0005C        CALLS   #1, BAS$$STOP                               ;
                                        3C   AE   D4 00063 5$:    CLRL    NO_MAP_REC_SPECIFIED                        ; 1531
                 50       04  AC             0C   C1 00066        ADDL3   #12, OPEN_ARG_BLK, R0                       ; 1532
                          14  AE             60   3C 0006B        MOVZWL  (R0), 20(SP)                                ;
                              58        14   AE   D0 0006F        MOVL    20(SP), RSZ                                 ;
                                             43   12 00073        BNEQ    11$                                         ; 1534
                              34        04   BC   91 00075        CMPB    @OPEN_ARG_BLK, #52                          ; 1542
                                             13   1B 00079        BLEQU   6$                                          ;
                 50       04  AC             34   C1 0007B        ADDL3   #52, OPEN_ARG_BLK, R0                       ; 1543
                                             60   B5 00080        TSTW    (R0)                                        ;
                                             0A   13 00082        BEQL    6$                                          ;
                 50       04  AC             34   C1 00084        ADDL3   #52, OPEN_ARG_BLK, R0                       ; 1545
                              58             60   3C 00089        MOVZWL  (R0), RSZ                                   ;
                                             54   11 0008C        BRB     13$                                         ;
                          3C  AE             01   D0 0008E 6$:    MOVL    #1, NO_MAP_REC_SPECIFIED                    ; 1548
                 50       04  AC             01   C1 00092        ADDL3   #1, OPEN_ARG_BLK, R0                        ; 1549
                 05       00                 60   8F 00097        CASEB   (R0), #0, #5                                ;
      000C     0010     0016            0010         0009B 7$:    .WORD   9$-7$,-                                     ;
```

```
            000C           000C     000A3                  10$-7$,-                              ;
                                                           9$-7$,-                               ;
                                                           8$-7$,-                               ;
                                                           8$-7$,-                               ;
                                                           8$-7$                                 ;
                             58  D4 000A7 8$:      CLRL    RSZ                                   ;
                             37  11 000A9          BRB     13$                                   ;
               58       84   8F  9A 000AB 9$:      MOVZBL  #132, RSZ                             ;
                             31  11 000AF          BRB     13$                                   ;
               58     0200   8F  3C 000B1 10$:     MOVZWL  #512, RSZ                             ;
                             2A  11 000B6          BRB     13$                                   ; 1536
50       04    AC            01  C1 000B8 11$:     ADDL3   #1, OPEN_ARG_BLK, R0                  ; 1563
               01            60  91 000BD          CMPB    (R0), #1                              ;
                             20  12 000C0          BNEQ    13$                                   ;
               58            01  8A 000C2          BICB2   #1, RSZ                               ; 1566
50       04    AC            10  C1 000C5          ADDL3   #16, OPEN_ARG_BLK, R0                 ; 1576
                             60  D5 000CA          TSTL    (R0)                                  ;
                             14  12 000CC          BNEQ    13$                                   ;
               50            58  D0 000CE          MOVL    RSZ, R0                               ;
   00000200    8F            50  D1 000D1          CMPL    R0, #512                              ;
                             05  18 000D8          BGEQ    12$                                   ;
               50     0200   8F  3C 000DA          MOVZWL  #512, R0                              ;
               58            50  D0 000DF 12$:     MOVL    R0, RSZ                               ;
50       04    AC            1A  C1 000E2 13$:     ADDL3   #26, OPEN_ARG_BLK, R0                 ; 1580
               5A            60  32 000E7          CVTWL   (R0), BKS                             ;
                             48  13 000EA          BEQL    20$                                   ; 1582
50       04    AC            03  C1 000EC          ADDL3   #3, OPEN_ARG_BLK, R0                  ; 1590
               02            60  91 000F1          CMPB    (R0), #2                              ;
                             05  12 000F4          BNEQ    14$                                   ;
               50            02  D0 000F6          MOVL    #2, R0                                ;
                             02  11 000F9          BRB     15$                                   ;
                             50  D4 000FB 14$:     CLRL    R0                                    ;
               50            58  C0 000FD 15$:     ADDL2   RSZ, R0                               ; 1587
                             53  D4 00100          CLRL    R3                                    ; 1596
51       04    AC            01  C1 00102          ADDL3   #1, OPEN_ARG_BLK, R1                  ;
               04            61  91 00107          CMPB    (R1), #4                              ;
                             07  12 0010A          BNEQ    16$                                   ;
                             53  D6 0010C          INCL    R3                                    ;
               51            07  D0 0010E          MOVL    #7, R1                                ;
                             03  11 00111          BRB     17$                                   ;
               51            01  D0 00113 16$:     MOVL    #1, R1                                ;
               50            51  C0 00116 17$:     ADDL2   R1, R0                                ; 1594
               50            5A  C4 00119          MULL2   BKS, R0                               ; 1587
               05            53  E9 0011C          BLBC    R3, 18$                               ; 1602
               51            0F  D0 0011F          MOVL    #15, R1                               ;
                             02  11 00122          BRB     19$                                   ;
                             51  D4 00124 18$:     CLRL    R1                                    ;
               50     01FF C140  9E 00126 19$:     MOVAB   511(R1)[R0], R0                       ; 1587
5A             50 00000200   8F  C7 0012C          DIVL3   #512, R0, BKS                         ; 1605
50       04    AC            0E  C1 00134 20$:     ADDL3   #14, OPEN_ARG_BLK, R0                 ; 1612
               51            60  3C 00139          MOVZWL  (R0), R1                              ;
                             03  14 0013C          BGTR    21$                                   ;
               51            01  D0 0013E          MOVL    #1, R1                                ;
50       04    AC            03  C1 00141 21$:     ADDL3   #3, OPEN_ARG_BLK, R0                  ; 1615
         24    AE            60  98 00146          CVTBL   (R0), 36(SP)                          ;
               02       24   AE  91 0014A          CMPB    36(SP), #2                            ;
                             06  12 0014E          BNEQ    22$                                   ;
```

```
                              50        04   A8  9E 00150        MOVAB   4(R8), R0                               ;
                                             03  11 00154        BRB     23$                                     ;
                              50             58  D0 00156 22$:   MOVL    RSZ, R0                                 ;
                              50             51  C4 00159 23$:   MULL2   R1, R0                                  ; 1613
                              50             03  C0 0015C        ADDL2   #3, R0                                  ; 1612
                              50             03  8A 0015F        BICB2   #3, R0                                  ; 1618
                              14             50  D1 00162        CMPL    R0, #20                                 ; 1612
                                             03  18 00165        BGEQ    24$                                     ;
                              50             14  D0 00167        MOVL    #20, R0                                 ;
                         40   AE             50  D0 0016A 24$:   MOVL    R0, BLS                                 ;
                                             50  D4 0016E        CLRL    R0                                      ; 1625
                                00000000G    00  16 00170        JSB     BAS$$CB_PUSH                            ;
                         50   AE             5B  D0 00176        MOVL    CCB, UNWIND_CCB                         ; 1629
                         54   AE             01  D0 0017A        MOVL    #1, UNWIND_ACTION                       ; 1630
                              26        FC   AB  E9 0017E        BLBC    -4(CCB), 26$                            ; 1637
                  00000000G   00             00  FB 00182        CALLS   #0, OTS$$CLOSE_FILE                     ; 1641
                              0A             50  E8 00189        BLBS    R0, 25$                                 ;
                              7E             01  CE 0018C        MNEGL   #1, -(SP)                               ;
                  00000000G   00             01  FB 0018F        CALLS   #1, BAS$$STOP_IO                        ;
                                00000000G    00  16 00196 25$:   JSB     BAS$$CB_POP                             ; 1648
                                             50  D4 0019C        CLRL    R0                                      ; 1649
                                00000000G    00  16 0019E        JSB     BAS$$CB_PUSH                            ;
                         50   AE             5B  D0 001A4        MOVL    CCB, UNWIND_CCB                         ; 1653
                              59        FC   AB  9E 001A8 26$:   MOVAB   -4(CCB), R9                             ; 1664
                              0A             69  E8 001AC        BLBS    (R9), 27$                               ;
              05        FF    AB             04  E0 001AF        BBS     #4, -1(CCB), 27$                        ;
                                        E8   AB  D5 001B4        TSTL    -24(CCB)                                ;
                                             0B  13 001B7        BEQL    28$                                     ;
                              7E       00G   8F  9A 001B9 27$:   MOVZBL  #BAS$K_IO_CHAALR, -(SP)                 ; 1666
                  00000000G   00             01  FB 001BD        CALLS   #1, BAS$$STOP_IO                        ;
              50         04   AC             08  C1 001C4 28$:   ADDL3   #8, OPEN_ARG_BLK, R0                    ; 1673
                              57             60  D0 001C9        MOVL    (R0), FILE_NAME_DESC                    ;
                         F8   AB        04   A7  D0 001CC        MOVL    4(FILE_NAME_DESC), -8(CCB)              ; 1674
                              50             67  3C 001D1        MOVZWL  (FILE_NAME_DESC), R0                    ; 1675
                       00FF   8F             50  B1 001D4        CMPW    R0, #255                                ;
                                             04  1B 001D9        BLEQU   29$                                     ;
                              50        FF   8F  9A 001DB        MOVZBL  #255, R0                                ;
                         F7   AB             50  90 001DF 29$:   MOVB    R0, -9(CCB)                             ;
                                        44   AE  9F 001E3        PUSHAB  CHANNEL                                 ; 1680
                  00000000G   00             01  FB 001E6        CALLS   #1, OTS$$TAKE_LUN                       ;
                              0B             50  E8 001ED        BLBS    R0, 30$                                 ;
                              7E       00G   8F  9A 001F0        MOVZBL  #BAS$K_IO_CHAALR, -(SP)                 ;
                  00000000G   00             01  FB 001F4        CALLS   #1, BAS$$STOP_IO                        ;
                         54   AE             02  D0 001FB 30$:   MOVL    #2, UNWIND_ACTION                       ; 1685
                              56        B0   AD  9E 001FF        MOVAB   FAB_BLOCK, FAB                          ; 1690
   0050    8F         00      6E             00  2C 00203        MOVC5   #0, (SP), #0, #80, (FAB)                ; 1691
                                             66     0020A                                                        ;
                              66      5003   8F  B0 0020B        MOVW    #20483, (FAB)                           ;
                         1F   A6             02  90 00210        MOVB    #2, 31(FAB)                             ;
              50         04   AC             14  C1 00214        ADDL3   #20, OPEN_ARG_BLK, R0                   ; 1692
                         10   A6             60  D0 00219        MOVL    (R0), 16(FAB)                           ;
              50         04   AC             01  C1 0021D        ADDL3   #1, OPEN_ARG_BLK, R0                    ; 1694
                         08   AE             60  98 00222        CVTBL   (R0), 8(SP)                             ;
                                        38   AE  D4 00226        CLRL    56(SP)                                  ;
                              03        08   AE  91 00229        CMPB    8(SP), #3                               ;
                                             05  12 0022D        BNEQ    31$                                     ;
                                        38   AE  D6 0022F        INCL    56(SP)                                  ;
```

```
                                  06  11 00232       BRB     32$                          :
                 04         08    AE  91 00234 31$:  CMPB    8(SP), #4                    :
                                  14  12 00238       BNEQ    34$                          :
                 50               5A  D0 0023A 32$:  MOVL    BKS, R0                      : 1696
         000000FF 8F              50  D1 0023D       CMPL    R0, #255                     :
                                  04  15 00244       BLEQ    33$                          :
                 50         FF    8F  9A 00246       MOVZBL  #255, R0                     :
            3E   A6               50  90 0024A 33$:  MOVB    R0, 62(FAB)                  :
                 0B         38    AE  E8 0024E 34$:  BLBS    56(SP), 35$                  : 1698
                 04         08    AE  91 00252       CMPB    8(SP), #4                    : 1699
                                  05  13 00256       BEQL    35$                          :
            3C   A6         40    AE  B0 00258       MOVW    BLS, 60(FAB)                 : 1701
      50    04   AC               18  C1 0025D 35$:  ADDL3   #24, OPEN_ARG_BLK, R0        : 1703
            14   A6               60  B0 00262       MOVW    (R0), 20(FAB)                :
      50    04   AC               24  C1 00266       ADDL3   #36, OPEN_ARG_BLK, R0        : 1707
                 57               60  D0 0026B       MOVL    (R0), FILE_NAME_DESC         :
                                  17  13 0026E       BEQL    37$                          : 1709
            30   A6         04    A7  D0 00270       MOVL    4(FILE_NAME_DESC), 48(FAB)   : 1712
                 50               67  3C 00275       MOVZWL  (FILE_NAME_DESC), R0         : 1713
          00FF   8F               50  B1 00278       CMPW    R0, #255                     :
                                  04  1B 0027D       BLEQU   36$                          :
                 50         FF    8F  9A 0027F       MOVZBL  #255, R0                     :
            35   A6               50  90 00283 36$.  MOVB    R0, 53(FAB)                  :
      51    04   AC               02  C1 00287 37$:  ADDL3   #2, OPEN_ARG_BLK, R1         : 1720
                 6E               61  98 0028C       CVTBL   (R1), (SP)                   :
                 50               01  D0 0028F       MOVL    #1, R0                       :
                                  6E  D5 00292       TSTL    (SP)                         : 1723
                                  0B  19 00294       BLSS    38$                          :
                 01               6E  91 00296       CMPB    (SP), #1                     :
                                  06  14 00299       BGTR    38$                          :
                                  50  D4 0029B       CLRL    R0                           :
            16   A6               04  88 0029D       BISB2   #4, 22(FAB)                  : 1724
                                  6E  D5 002A1 38$:  TSTL    (SP)                         : 1726
                                  05  19 002A3       BLSS    39$                          :
                 01               6E  91 002A5       CMPB    (SP), #1                     :
                                  0A  15 002A8       BLEQ    40$                          :
                 03               6E  91 002AA 39$:  CMPB    (SP), #3                     :
                                  0B  19 002AD       BLSS    41$                          :
                 04               6E  91 002AF       CMPB    (SP), #4                     :
                                  06  14 002B2       BGTR    41$                          :
                                  50  D4 002B4 40$:  CLRL    R0                           :
            16   A6               02  88 002B6       BISB2   #2, 22(FAB)                  : 1727
                                  6E  D5 002BA 41$:  TSTL    (SP)                         : 1729
                                  05  19 002BC       BLSS    42$                          :
                 02               6E  91 002BE       CMPB    (SP), #2                     :
                                  0A  15 002C1       BLEQ    43$                          :
                 04               6E  91 002C3 42$:  CMPB    (SP), #4                     :
                                  0B  19 002C6       BLSS    44$                          :
                 05               6E  91 002C8       CMPB    (SP), #5                     :
                                  06  14 002CB       BGTR    44$                          :
                                  50  D4 002CD 43$:  CLRL    R0                           :
            16   A6               01  88 002CF       BISB2   #1, 22(FAB)                  : 1730
                 04               6E  91 002D3 44$:  CMPB    (SP), #4                     : 1732
                                  06  12 002D6       BNEQ    45$                          :
                                  50  D4 002D8       CLRL    R0                           :
            16   A6               10  88 002DA       BISB2   #16, 22(FAB)                 : 1733
                                  6E  D5 002DE 45$:  TSTL    (SP)                         : 1735
```

```
                                   05  19 002E0         BLSS    46$                                  ;
                       01          6E  91 002E2         CMPB    (SP), #1                             ;
                                   05  15 002E5         BLEQ    47$                                  ;
                       04          6E  91 002E7 46$:    CMPB    (SP), #4                             ;
                                   06  12 002EA         BNEQ    48$                                  ;
                                   50  D4 002EC 47$:    CLRL    R0                                   ;
                  16   A6          08  88 002EE         BISB2   #8, 22(FAB)                          ; 1736
                       0B          50  E9 002F2 48$:    BLBC    R0, 49$                              ; 1738
                       7E     00G  8F  9A 002F5         MOVZBL  #BAS$K_PROLOSSOR, -(SP)              ; 1739
           00000000G   00          01  FB 002F9         CALLS   #1, BAS$$STOP_IO                     ;
        50        04   AC          08  C1 00300 49$:    ADDL3   #8, OPEN_ARG_BLK, R0                 ; 1745
                       57          60  D0 00305         MOVL    (R0), FILE_NAME_DESC                 ;
                  2C   A6     04   A7  D0 00308         MOVL    4(FILE_NAME_DESC), 44(FAB)           ; 1746
                       57          67  3C 0030D         MOVZWL  (FILE_NAME_DESC), R7                 ; 1747
                0OFF   8F          57  B1 00310         CMPW    R7, #255                             ;
                                   04  1B 00315         BLEQU   50$                                  ;
                       57     FF   8F  9A 00317         MOVZBL  #255, R7                             ;
                  34   A6          57  90 0031B 50$:    MOVB    R7, 52(FAB)                          ;
        09        2C   BE          05  E0 0031F         BBS     #5, @44(SP), 51$                     ; 1771
        04        2C   BE          04  E0 00324         BBS     #4, @44(SP), 51$                     ;
                  07   A6          02  88 00329         BISB2   #2, 7(FAB)                           ;
        04        2C   BE          01  E1 0032D 51$:    BBC     #1, @44(SP), 52$                     ; 1778
                  06   A6          20  88 00332         BISB2   #32, 6(FAB)                          ;
        50        04   AC          04  C1 00336 52$:    ADDL3   #4, OPEN_ARG_BLK, R0                 ; 1785
                       52          60  98 0033B         CVTBL   (R0), R2                             ;
                       02          52  91 0033E         CMPB    R2, #2                               ;
                                   09  13 00341         BEQL    53$                                  ;
                       01          52  91 00343         CMPB    R2, #1                               ;
                                   04  13 00346         BEQL    53$                                  ;
                  04   A6          20  88 00348         BISB2   #32, 4(FAB)                          ; 1788
                       05          6E  91 0034C 53$:    CMPB    (SP), #5                             ; 1795
                                   04  13 0034F         BEQL    54$                                  ;
                  05   A6          04  88 00351         BISB2   #4, 5(FAB)                           ;
        05        2C   BE          02  E0 00355 54$:    BBS     #2, @44(SP), 55$                     ; 1801
        04        A6          80   8F  88 0035A         BISB2   #128, 4(FAB)                         ;
                              34   AE  D4 0035F 55$:    CLRL    52(SP)                               ; 1809
                              08   AE  D5 00362         TSTL    8(SP)                                ;
                                   08  12 00365         BNEQ    56$                                  ;
                              34   AE  D6 00367         INCL    52(SP)                               ;
                  04   A6     40   8F  88 0036A         BISB2   #64, 4(FAB)                          ;
        04        2C   BE          04  E1 0036F 56$:    BBC     #4, @44(SP), 57$                     ; 1818
                  04   A6          04  88 00374         BISB2   #4, 4(FAB)                           ;
        04        2C   BE          03  E1 00378 57$:    BBC     #3, @44(SP), 58$                     ; 1825
                  04   A6          10  88 0037D         BISB2   #16, 4(FAB)                          ;
        50        04   AC          07  C1 00381 58$:    ADDL3   #7, OPEN_ARG_BLK, R0                 ; 1827
                  3F   A6          60  90 00386         MOVB    (R0), 63(FAB)                        ;
                  36   A6          58  B0 0038A         MOVW    RSZ, 54(FAB)                         ; 1828
        05             00     08   AE  8F 0038E         CASEB   8(SP), #0, #5                        ; 1834
0019    0025         0025        0025     00393 59$:    .WORD   62$-59$,-                            ;
                     0025        001F     0039B                 62$-59$,-                            ;
                                                                62$-59$,-                            ;
                                                                60$-59$,-                            ;
                                                                61$-59$,-                            ;
                                                                62$-59$                              ;
                       7E     00G  8F  9A 0039F         MOVZBL  #BAS$K_PROLOSSOR, -(SP)              ; 1856
           00000000G   00          01  FB 003A3         CALLS   #1, BAS$$STOP_IO                     ;
                                   0F  11 003AA         BRB     63$                                  ;
```

```
                        1D   A6        10  90 003AC 60$:    MOVB    #16, 29(FAB)                        ; 1847
                                       09  11 003B0         BRB     63$                                 ;
                        1D   A6        20  90 003B2 61$:    MOVB    #32, 29(FAB)                        ; 1850
                                       03  11 003B6         BRB     63$                                 ;
                                  1D   A6  94 003B8 62$:    CLRB    29(FAB)                             ; 1853
                             04   2C   BE  E9 003BB 63$:    BLBC    @44(SP), 64$                        ; 1864
                        1E   A6        08  88 003BF         BISB2   #8, 30(FAB)                         ;
               50       04   AC        06  C1 003C3 64$:    ADDL3   #6, OPEN_ARG_BLK, R0                ; 1866
                        10   AE        60  98 003C8         CVTBL   (R0), 16(SP)                        ;
               05            00   10   AE  8F 003CC         CASEB   16(SP), #0, #5                      ;
       0031    0027        0021      0019     003D1 65$:    .WORD   66$-65$,-                           ;
                           0019      002D     003D9                 67$-65$,-                           ;
                                                                    68$-65$,-                           ;
                                                                    70$-65$,-                           ;
                                                                    69$-65$,-                           ;
                                                                    66$-65$                             ;
                             7E   00G  8F  9A 003DD         MOVZBL  #BAS$K_PROLOSSOR, -(SP)             ; 1888
                  00000000G  00        01  FB 003E1         CALLS   #1, BAS$$STOP_IO                    ;
                                       18  11 003E8         BRB     70$                                 ;
                             01   08   AE  91 003EA 66$:    CMPB    8(SP), #1                           ; 1871
                                       12  13 003EE         BEQL    70$                                 ;
                                       06  11 003F0         BRB     68$                                 ;
                        1E   A6        01  88 003F2 67$:    BISB2   #1, 30(FAB)                         ; 1874
                                       0A  11 003F6         BRB     70$                                 ;
                        1E   A6        02  88 003F8 68$:    BISB2   #2, 30(FAB)                         ; 1877
                                       04  11 003FC         BRB     70$                                 ;
                        1E   A6        04  88 003FE 69$:    BISB2   #4, 30(FAB)                         ; 1885
               04            00   24   AE  8F 00402 70$:    CASEB   36(SP), #0, #4                      ; 1895
       0037    0031        002B      0017     00407 71$:    .WORD   73$-71$,-                           ;
                                     000A     0040F                 76$-71$,-                           ;
                                                                    77$-71$,-                           ;
                                                                    78$-71$,-                           ;
                                                                    72$-71$                             ;
                             7E   00G  8F  9A 00411 72$:    MOVZBL  #BAS$K_PROLOSSOR, -(SP)             ; 1916
                  00000000G  00        01  FB 00415         CALLS   #1, BAS$$STOP_IO                    ;
                                       24  11 0041C         BRB     79$                                 ;
                             01   08   AE  91 0041E 73$:    CMPB    8(SP), #1                           ; 1902
                                       05  12 00422         BNEQ    74$                                 ;
                             50        01  D0 00424         MOVL    #1, R0                              ;
                                       03  11 00427         BRB     75$                                 ;
                             50        02  D0 00429 74$:    MOVL    #2, R0                              ;
                        1F   A6        50  90 0042C 75$:    MOVB    R0, 31(FAB)                         ; 1900
                                       10  11 00430         BRB     79$                                 ; 1899
                        1F   A6        01  90 00432 76$:    MOVB    #1, 31(FAB)                         ; 1907
                                       0A  11 00436         BRB     79$                                 ;
                        1F   A6        02  90 00438 77$:    MOVB    #2, 31(FAB)                         ; 1910
                                       04  11 0043C         BRB     79$                                 ;
                        1F   A6        03  90 0043E 78$:    MOVB    #3, 31(FAB)                         ; 1913
               50       04   AC        1C  C1 00442 79$:    ADDL3   #28, OPEN_ARG_BLK, R0               ; 1919
                        1C   A6        60  90 00447         MOVB    (R0), 28(FAB)                       ;
                             50        01  D0 0044B         MOVL    #1, R0                              ; 1926
                             01        52  91 0044E         CMPB    R2, #1                              ; 1929
                                       06  12 00451         BNEQ    80$                                 ;
                                       50  D4 00453         CLRL    R0                                  ;
                        17   A6        04  88 00455         BISB2   #4, 23(FAB)                         ; 1930
                             01        52  91 00459 80$:    CMPB    R2, #1                              ; 1932
                                       05  13 0045C         BEQL    81$                                 ;
                        03         52  91 0045E          CMPB    R2, #3                                   ;
                                   06  12 00461          BNEQ    82$                                      ;
                                   50  D4 00463 81$:     CLRL    R0                                       ;
                    17  A6         02  88 00465          BISB2   #2, 23(FAB)                              ; 1933
                        04         52  91 00469 82$:     CMPB    R2, #4                                   ; 1935
                                   05  13 0046C          BEQL    83$                                      ;
                        06         52  91 0046E          CMPB    R2, #6                                   ;
                                   06  12 00471          BNEQ    84$                                      ;
                                   50  D4 00473 83$:     CLRL    R0                                       ;
                    17  A6         20  88 00475          BISB2   #32, 23(FAB)                             ; 1936
                                   52  D5 00479 84$:     TSTL    R2                                       ; 1938
                                   0B  15 0047B          BLEQ    85$                                      ;
                        02         52  91 0047D          CMPB    R2, #2                                   ;
                                   06  14 00480          BGTR    85$                                      ;
                                   50  D4 00482          CLRL    R0                                       ;
                    17  A6         01  88 00484          BISB2   #1, 23(FAB)                              ; 1939
                        01         52  91 00488 85$:     CMPB    R2, #1                                   ; 1941
                                   06  12 0048B          BNEQ    86$                                      ;
                                   50  D4 0048D          CLRL    R0                                       ;
                    17  A6         08  88 0048F          BISB2   #8, 23(FAB)                              ; 1942
                                   52  D5 00493 86$:     TSTL    R2                                       ; 1944
                                   02  12 00495          BNEQ    87$                                      ;
                                   50  D4 00497          CLRL    R0                                       ;
                        0B         50  E9 00499 87$:     BLBC    R0, 88$                                  ; 1949
                        7E    00G  8F  9A 0049C          MOVZBL  #BAS$K_PROLOSSOR, -(SP)                  ; 1950
             00000000G  00         01  FB 004A0          CALLS   #1, BAS$$STOP_IO                         ;
                              30   AE  D4 004A7 88$:     CLRL    48(SP)                                   ; 1956
                        04    08   AE  91 004AA          CMPB    8(SP), #4                                ;
                                   07  12 004AE          BNEQ    89$                                      ;
                              30   AE  D6 004B0          INCL    48(SP)                                   ;
                    17  A6         10  88 004B3          BISB2   #16, 23(FAB)                             ;
                    E8  AB         56  D0 004B7 89$:     MOVL    FAB, -24(CCB)                            ; 1962
                    F8  AB    2C   A6  D0 004BB          MOVL    44(FAB), -8(CCB)                         ; 1966
                    F7  AB    34   A6  90 004C0          MOVB    52(FAB), -9(CCB)                         ; 1967
               05       00         6E  8F 004C5          CASEB   (SP), #0, #5                             ; 1974
   0011        000C   000C      000C      004C9 90$:     .WORD   91$-90$,-                                ;
                      0025      0019      004D1                  91$-90$,-                                ;
                                                                 91$-90$,-                                ;
                                                                 92$-90$,-                                ;
                                                                 94$-90$,-                                ;
                                                                 96$-90$                                  ;
                        69         04  8A 004D5 91$:     BICB2   #4, (R9)                                 ; 1979
                                   03  11 004D8          BRB     93$                                      ; 1980
                        69         0C  88 004DA 92$:     BISB2   #12, (R9)                                ; 1987
                        69         20  8A 004DD 93$:     BICB2   #32, (R9)                                ; 1988
                                   06  11 004E0          BRB     95$                                      ; 1989
                        69         04  8A 004E2 94$:     BICB2   #4, (R9)                                 ; 1994
                        69         20  88 004E5          BISB2   #32, (R9)                                ; 1995
                    01  A9         20  8A 004E8 95$:     BICB2   #32, 1(R9)                               ; 1996
                                   07  11 004EC          BRB     97$                                      ; 1974
                        69         24  8A 004EE 96$:     BICB2   #36, (R9)                                ; 2002
                    01  A9         20  88 004F1          BISB2   #32, 1(R9)                               ; 2003
               05       00    08   AE  8F 004F5 97$:     CASEB   8(SP), #0, #5                            ; 2011
   0012        000C   0012      000C      004FA 98$:     .WORD   99$-98$,-                                ;
                      0012      0012      00502                  100$-98$,-                               ;
                                                                 99$-98$,-                                ;
                                                                 100$-98$,-                               ;
                                                                 100$-98$,-                        ;
                                                                 100$-98$                          ;
                         A1   AB        08  8A 00506  99$:  BICB2   #8, -95(CCB)                   ; 2015
                                        04  11 0050A        BRB     101$                           ;
                         A1   AB        08  88 0050C 100$:  BISB2   #8, -95(CCB)                   ; 2018
                    05        00   08   AE  8F 00510 101$:  CASEB   8(SP), #0, #5                  ; 2026
      000C        000C      0012      000C     00515 102$:  .WORD   103$-102$,-                    ;
                            0012      000C     0051D                104$-102$,-                    ;
                                                                    103$-102$,-                    ;
                                                                    103$-102$,-                    ;
                                                                    103$-102$,-                    ;
                                                                    104$-102$                      ;
                         01   A9        01  88 00521 103$:  BISB2   #1, 1(R9)                      ; 2030
                                        04  11 00525        BRB     105$                           ;
                         01   A9        02  88 00527 104$:  BISB2   #2, 1(R9)                      ; 2033
                         04   AE   1F   A6  9E 0052B 105$:  MOVAB   31(FAB), 4(SP)                 ; 2040
                              01   04   BE  91 00530        CMPB    @4(SP), #1                     ;
                                        04  12 00534        BNEQ    106$                           ;
                         01   A9        04  88 00536        BISB2   #4, 1(R9)                      ;
                    03   2C   BE        05  E1 0053A 106$:  BBC     #5, @44(SP), 107$              ; 2048
                              69        08  88 0053F        BISB2   #8, (R9)                       ;
                    05        00   08   AE  8F 00542 107$:  CASEB   8(SP), #0, #5                  ; 2054
      000C        000C      0011      000C     00547 108$:  .WORD   109$-108$,-                    ;
                            0011      000C     0054F                110$-108$,-                    ;
                                                                    109$-108$,-                    ;
                                                                    109$-108$,-                    ;
                                                                    109$-108$,-                    ;
                                                                    110$-108$                      ;
                              69        10  8A 00553 109$:  BICB2   #16, (R9)                      ; 2058
                                        03  11 00556        BRB     111$                           ;
                              69        10  88 00558 110$:  BISB2   #16, (R9)                      ; 2061
                    05        00   08   AE  8F 0055B 111$:  CASEB   8(SP), #0, #5                  ; 2068
      0050        0050      0050      0019     00560 112$:  .WORD   113$-112$,-                    ;
                            0050      0050     00568                117$-112$,-                    ;
                                                                    117$-112$,-                    ;
                                                                    117$-112$,-                    ;
                                                                    117$-112$,-                    ;
                                                                    117$-112$                      ;
                              7E  00G   8F  9A 0056C        MOVZBL  #BAS$K_PROLOSSOR, -(SP)        ; 2126
                     00000000G  00      01  FB 00570        CALLS   #1, BAS$$STOP_IO               ;
                                        6A  11 00577        BRB     121$                           ;
                                   14   AE  D5 00579 113$:  TSTL    20(SP)                         ; 2080
                                        06  13 0057C        BEQL    114$                           ;
                              50   14   AE  D0 0057E        MOVL    20(SP), R0                     ; 2082
                                        1D  11 00582        BRB     116$                           ;
                              34   04   BC  91 00584 114$:  CMPB    @OPEN_ARG_BLK, #52             ; 2088
                                        13  1B 00588        BLEQU   115$                           ;
                    50   04   AC        34  C1 0058A        ADDL3   #52, OPEN_ARG_BLK, R0          ; 2089
                                        60  B5 0058F        TSTW    (R0)                           ;
                                        0A  13 00591        BEQL    115$                           ;
                    51   04   AC        34  C1 00593        ADDL3   #52, OPEN_ARG_BLK, R1          ; 2090
                              50        61  3C 00598        MOVZWL  (R1), R0                       ;
                                        04  11 0059B        BRB     116$                           ;
                              50   48   8F  9A 0059D 115$:  MOVZBL  #72, R0                        ; 2088
                         D6   AB        50  B0 005A1 116$:  MOVW    R0, -42(CCB)                   ; 2078
                         D4   AB   D6   AB  B0 005A5        MOVW    -42(CCB), -44(CCB)             ; 2094
                         A1   AB        02  8A 005AA        BICB2   #2, -95(CCB)                   ; 2095
```

```
                                                  33  11 005AE          BRB     121$                                  ; 2068
                                        14        AE  D5 005B0 117$:    TSTL    20(SP)                                ; 2107
                                                  06  13 005B3          BEQL    118$                                  ;
                                 50     14        AE  D0 005B5          MOVL    20(SP), R0                            ; 2109
                                                  1D  11 005B9          BRB     120$                                  ;
                                 34     04        BC  91 005BB 118$:    CMPB    @OPEN_ARG_BLK, #52                    ; 2115
                                                  13  1B 005BF          BLEQU   119$                                  ;
                   50     04     AC               34  C1 005C1          ADDL3   #52, OPEN_ARG_BLK, R0                 ; 2116
                                                  60  B5 005C6          TSTW    (R0)                                  ;
                                                  0A  13 005C8          BEQL    119$                                  ;
                   51     04     AC               34  C1 005CA          ADDL3   #52, OPEN_ARG_BLK, R1                 ; 2117
                                 50               61  3C 005CF          MOVZWL  (R1), R0                              ;
                                                  04  11 005D2          BRB     120$                                  ;
                                 50     48        8F  9A 005D4 119$:    MOVZBL  #72, R0                               ; 2115
                          D6     AB               50  B0 005D8 120$:    MOVW    R0, -42(CCB)                          ; 2105
                                        D4        AB  B4 005DC          CLRW    -44(CCB)                              ; 2121
                          A1     AB               02  88 005DF          BISB2   #2, -95(CCB)                          ; 2122
                                                  50  D4 005E3 121$:    CLRL    R0                                    ; 2133
                                 02               6C  91 005E5          CMPB    (AP), #2                              ;
                                                  02  1F 005E8          BLSSU   122$                                  ;
                                                  50  D6 005EA          INCL    R0                                    ;
          69       01            0F               50  F0 005EC 122$:    INSV    R0, #15, #1, (R9)                     ;
                   50     04     AC               1E  C1 005F1          ADDL3   #30, OPEN_ARG_BLK, R0                 ; 2138
                                                  60  B5 005F6          TSTW    (R0)                                  ;
                                                  06  13 005F8          BEQL    123$                                  ;
                          FF     AB               04  88 005FA          BISB2   #4, -1(CCB)                           ; 2140
                                                  04  11 005FE          BRB     124$                                  ;
                          FF     AB               04  8A 00600 123$:    BICB2   #4, -1(CCB)                           ; 2142
                                 04     34        AE  E9 00604 124$:    BLBC    52(SP), 125$                          ; 2149
                          FE     AB               10  88 00608          BISB2   #16, -2(CCB)                          ;
                                 6B   4401        8F  B0 0060C 125$:    MOVW    #17409, (CCB)                         ; 2155
                          3C     AB               56  D0 00611          MOVL    FAB, 60(CCB)                          ; 2157
                          30     AB     E0        AB  9E 00615          MOVAB   -32(R11), 48(CCB)                     ; 2163
                          34     AB               04  90 0061A          MOVB    #4, 52(CCB)                           ; 2164
                   50     04     AC               20  C1 0061E          ADDL3   #32, OPEN_ARG_BLK, R0                 ; 2165
                          36     AB               60  90 00623          MOVB    (R0), 54(CCB)                         ;
                   50     04     AC               07  C1 00627          ADDL3   #7, OPEN_ARG_BLK, R0                  ; 2167
                                                  60  95 0062C          TSTB    (R0)                                  ;
                                                  09  13 0062E          BEQL    126$                                  ;
                   50     04     AC               30  C1 00630          ADDL3   #48, OPEN_ARG_BLK, R0                 ;
                          2C     AB               60  D0 00635          MOVL    (R0), 44(CCB)                         ;
                                        2C        AE  D4 00639 126$:    CLRL    44(SP)                                ; 2176
                                 34     04        BC  91 0063C          CMPB    @OPEN_ARG_BLK, #52                    ;
                                                  2A  1B 00640          BLEQU   128$                                  ;
                                        2C        AE  D6 00642          INCL    44(SP)                                ;
                   50     04     AC               36  C1 00645          ADDL3   #54, OPEN_ARG_BLK, R0                 ; 2178
                                 1F               60  E9 0064A          BLBC    (R0), 128$                            ;
                                 02     08        AE  91 0064D          CMPB    8(SP), #2                             ; 2186
                                                  15  12 00651          BNEQ    127$                                  ;
                        0200     8F     14        AE  B1 00653          CMPW    20(SP), #512                          ; 2187
                                                  0D  13 00659          BEQL    127$                                  ;
                                 7E    00G        8F  9A 0065B          MOVZBL  #BAS$K_REQRECSIZ, -(SP)               ; 2188
                   00000000G     00               01  FB 0065F          CALLS   #1, BAS$$STOP_IO                      ;
                                                  04  11 00666          BRB     128$                                  ;
                          06     AB               04  88 00668 127$:    BISB2   #4, 6(CCB)                            ; 2190
                                 05               6E  91 0066C 128$:    CMPB    (SP), #5                              ; 2198
                                                  04  12 0066F          BNEQ    129$                                  ;
                   05   AB           01  88 00671        BISB2   #1, 5(CCB)                          :
                   04   AB 00010002  8F  CA 00675 129$: BICL2   #65538, 4(CCB)                      : 2205
                              28     AE  D4 0067D        CLRL    40(SP)                              : 2216
                   01         08     AE  91 00680        CMPB    8(SP), #1                           :
                                     03  12 00684        BNEQ    130$                                :
                              28     AE  D6 00686        INCL    40(SP)                              :
    04   AB    01         04  28     AE  F0 00689 130$: INSV    40(SP), #4, #1, 4(CCB)              :
  0060   8F    00         6E         00  2C 00690        MOVC5   #0, (SP), #0, #96, $RMS_PTR         : 2222
                            FF50     CD     00697                                                    :
                 FF50   CD  6002     8F  B0 0069A        MOVW    #24578, $RMS_PTR                    :
                        50    58     AE  9E 006A1        MOVAB   FILE_NAME, R0                       : 2223
                 FF5C   CD           50  D0 006A5        MOVL    R0, NAM_BLOCK+12                    :
                 FF54   CD           50  D0 006AA        MOVL    R0, NAM_BLOCK+4                     :
                 FF5A   CD           01  8E 006AF        MNEGB   #1, NAM_BLOCK+10                    : 2224
                 FF52   CD           01  8E 006B4        MNEGB   #1, NAM_BLOCK+2                     :
                   28   A6  FF50     CD  9E 006B9        MOVAB   NAM_BLOCK, 40(FAB)                  : 2225
         2C    00         6E         00  2C 006BF        MOVC5   #0, (SP), #0, #44, $RMS_PTR         : 2233
                            FF24     CD     006C4                                                    :
                 FF24   CD  2C1D     8F  B0 006C7        MOVW    #11293, $RMS_PTR                    :
                 FF28   CD  FF18     CD  9E 006CE        MOVAB   XABSUM, $RMS_PTR+4                  :
         0C    00         6E         00  2C 006D5        MOVC5   #0, (SP), #0, #12, $RMS_PTR         : 2234
                            FF18     CD     006DA                                                    :
                 FF18   CD  0C16     8F  B0 006DD        MOVW    #3094, $RMS_PTR                     :
                            FF1C     CD  D4 006E4        CLRL    XABSUM+4                            : 2235
                   0C   AE    24     A6  9E 006E8        MOVAB   36(FAB), 12(SP)                     : 2236
                   0C   BE  FF24     CD  9E 006ED        MOVAB   XABFHC, @12(SP)                     :
                        02           6C  91 006F3        CMPB    (AP), #2                            : 2241
                                     03  1E 006F6        BGEQU   131$                                :
                                   0228  31 006F8        BRW     161$                                :
                   20   AE    08     AC  D0 006FB 131$: MOVL    KEY_INFO_BLK, 32(SP)                : 2257
                        59    08     BC  9A 00700        MOVZBL  @KEY_INFO_BLK, KEYNO                :
                                   0214  31 00704        BRW     159$                                :
               51      20     AE     01  C1 00707 132$: ADDL3   #1, 32(SP), R1                      :
                        50           61  9A 0070C        MOVZBL  (R1), R0                            :
                        50           59  C4 0070F        MULL2   KEYNO, R0                           :
                        50    08     AC  C0 00712        ADDL2   KEY_INFO_BLK, R0                    :
                        57    04     A0  9E 00716        MOVAB   4(R0), KEY_PTR                      :
                        48           AE  9F 0071A        PUSHAB  XABKEY                              : 2263
                   20   AE    4C     8F  9A 0071D        MOVZBL  #76, 32(SP)                         :
                              20     AE  9F 00722        PUSHAB  32(SP)                              :
            00000000G   00           02  FB 00725        CALLS   #2, LIB$GET_VM                      :
                        0B           50  E8 0072C        BLBS    GET_VM_STATUS, 133$                 : 2265
                        7E    00G    8F  9A 0072F        MOVZBL  #BAS$K_MAXMEMEXC, -(SP)             :
            00000000G   00           01  FB 00733        CALLS   #1, BAS$$STOP_IO                    :
  004C   8F    00         6E         00  2C 0073A 133$: MOVC5   #0, (SP), #0, #76, @XABKEY          : 2268
                              48     BE     00741                                                    :
                   48   BE           15  90 00743        MOVB    #21, @XABKEY                        :
               50      48     AE     01  C1 00747        ADDL3   #1, XABKEY, R0                      :
                        60    4C     8F  90 0074C        MOVB    #76, (R0)                           :
               50      48     AE     13  C1 00750        ADDL3   #19, XABKEY, R0                     :
                                     60  94 00755        CLRB    (R0)                                :
               50      48     AE     17  C1 00757        ADDL3   #23, XABKEY, R0                     : 2272
                        60           59  90 0075C        MOVB    KEYNO, (R0)                         :
               50      48     AE     1E  C1 0075F        ADDL3   #30, XABKEY, R0                     : 2273
                        60    02     A7  B0 00764        MOVW    2(KEY_PTR), (R0)                    :
               50      48     AE     2E  C1 00768        ADDL3   #46, XABKEY, R0                     : 2274
                        60           67  90 0076D        MOVB    (KEY_PTR), (R0)                     :
```

```
     50       01   A7           01         01  EF 00770          EXTZV     #1, #1, 1(KEY_PTR), R0                    ; 2278
                   51      48   AE         12  C1 00776          ADDL3     #18, XABKEY, R1                           ;
     61            01           01         50  F0 0077B          INSV      R0, #1, #1, (R1)                          ;
                   50      48   AE         12  C1 00780          ADDL3     #18, XABKEY, R0                           ; 2279
     60            01           00     01  A7  F0 00785          INSV      1(KEY_PTR), #0, #1, (R0)                  ;
                         FF21   CD     08  BC  90 0078B          MOVB      @KEY_INFO_BLK, XABSUM+9                   ; 2284
                                03     2C  AE  E8 00791          BLBS      44(SP), 135$                              ; 2290
                                        00B9   31 00795 134$:    BRW       148$                                      ;
                                53     05  A7  9A 00798 135$:    MOVZBL    5(KEY_PTR), R3                            ; 2293
                                           F7  15 0079C          BLEQ      134$                                      ;
                                           52  D4 0079E          CLRL      KEY_NUM                                   ; 2296
                                           6D  11 007A0          BRB       142$                                      ;
                   06           01         52  CF 007A2 136$:    CASEL     KEY_NUM, #1, #6                           ; 2298
   0057          0043         002F      001B      007A6 137$:    .WORD     138$-137$,-                               ;
                 0093         007F      006B      007AE                    139$-137$,-                               ;
                                                                           140$-137$,-                               ;
                                                                           141$-137$,-                               ;
                                                                           143$-137$,-                               ;
                                                                           144$-137$,-                               ;
                                                                           146$-137$                                 ;
                                7E    00G  8F  9A 007B4          MOVZBL    #BAS$K_NOTIMP, -(SP)                      ; 2343
                   000000000G   00         01  FB 007B8          CALLS     #1, BAS$$STOP_IO                          ;
                                           76  11 007BF          BRB       145$                                      ;
                   50      48   AE         2F  C1 007C1 138$:    ADDL3     #47, XABKEY, R0                           ; 2302
                                60     06  A7  90 007C6          MOVB      6(KEY_PTR), (R0)                          ;
                   50      48   AE         20  C1 007CA          ADDL3     #32, XABKEY, R0                           ; 2303
                                60     07  A7  B0 007CF          MOVW      7(KEY_PTR), (R0)                          ;
                                           76  11 007D3          BRB       147$                                      ; 2298
                   50      48   AE         30  C1 007D5 139$:    ADDL3     #48, XABKEY, R0                           ; 2308
                                60     09  A7  90 007DA          MOVB      9(KEY_PTR), (R0)                          ;
                   50      48   AE         22  C1 007DE          ADDL3     #34, XABKEY, R0                           ; 2309
                                60     0A  A7  B0 007E3          MOVW      10(KEY_PTR), (R0)                         ;
                                           62  11 007E7          BRB       147$                                      ; 2298
                   50      48   AE         31  C1 007E9 140$:    ADDL3     #49, XABKEY, R0                           ; 2314
                                60     0C  A7  90 007EE          MOVB      12(KEY_PTR), (R0)                         ;
                   50      48   AE         24  C1 007F2          ADDL3     #36, XABKEY, R0                           ; 2315
                                60     0D  A7  B0 007F7          MOVW      13(KEY_PTR), (R0)                         ;
                                           4E  11 007FB          BRB       147$                                      ; 2298
                   50      48   AE         32  C1 007FD 141$:    ADDL3     #50, XABKEY, R0                           ; 2320
                                60     0F  A7  90 00802          MOVB      15(KEY_PTR), (R0)                         ;
                   50      48   AE         26  C1 00806          ADDL3     #38, XABKEY, R0                           ; 2321
                                60     10  A7  B0 0080B          MOVW      16(KEY_PTR), (R0)                         ;
                                           3A  11 0080F 142$:    BRB       147$                                      ; 2298
                   50      48   AE         33  C1 00811 143$:    ADDL3     #51, XABKEY, R0                           ; 2326
                                60     12  A7  90 00816          MOVB      18(KEY_PTR), (R0)                         ;
                   50      48   AE         28  C1 0081A          ADDL3     #40, XABKEY, R0                           ; 2327
                                60     13  A7  B0 0081F          MOVW      19(KEY_PTR), (R0)                         ;
                                           26  11 00823          BRB       147$                                      ; 2298
                   50      48   AE         34  C1 00825 144$:    ADDL3     #52, XABKEY, R0                           ; 2332
                                60     15  A7  90 0082A          MOVB      21(KEY_PTR), (R0)                         ;
                   50      48   AE         2A  C1 0082E          ADDL3     #42, XABKEY, R0                           ; 2333
                                60     16  A7  B0 00833          MOVW      22(KEY_PTR), (R0)                         ;
                                           12  11 00837 145$:    BRB       147$                                      ; 2298
                   50      48   AE         35  C1 00839 146$:    ADDL3     #53, XABKEY, R0                           ; 2338
                                60     18  A7  90 0083E          MOVB      24(KEY_PTR), (R0)                         ;
                   50      48   AE         2C  C1 00842          ADDL3     #44, XABKEY, R0                           ; 2339
                                60     19  A7  B0 00847          MOVW      25(KEY_PTR), (R0)                         ;
```

```
    FF51          52          01          53  F1 0084B 147$:    ACBL    R3, #1, KEY_NUM, 136$                          ; 2296
                  12          03      04  A7  8F 00851 148$:    CASEB   4(KEY_PTR), #3, #18                            ; 2358
    003C        003C        002D        0028     00856 149$:    .WORD   150$-149$,-                                    ;
    003C        003C        0037        0032     0085E                  151$-149$,-                                    ;
    003C        003C        003C        003C     00866                  154$-149$,-                                    ;
    003C        003C        003C        003C     0086E                  154$-149$,-                                    ;
                0040        003C        003C     00876                  152$-149$,-                                    ;
                                                                        153$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        154$-149$,-                                    ;
                                                                        155$-149$                                      ;
                                          14  11 0087C          BRB     154$                                           ;
                              50          02  D0 0087E 150$:    MOVL    #2, R0                                         ;
                                          16  11 00881          BRB     156$                                           ;
                              50          04  D0 00883 151$:    MOVL    #4, R0                                         ;
                                          11  11 00886          BRB     156$                                           ;
                              50          01  D0 00888 152$:    MOVL    #1, R0                                         ;
                                          0C  11 0088B          BRB     156$                                           ;
                              50          03  D0 0088D 153$:    MOVL    #3, R0                                         ;
                                          07  11 00890          BRB     156$                                           ;
                                          50  D4 00892 154$:    CLRL    R0                                             ;
                                          03  11 00894          BRB     156$                                           ;
                              50          05  D0 00896 155$:    MOVL    #5, R0                                         ;
                  51          48  AE      13  C1 00899 156$:    ADDL3   #19, XABKEY, R1                                ; 2356
                                  61      50  90 0089E          MOVB    R0, (R1)                                       ;
                                      05  A7  95 008A1          TSTB    5(KEY_PTR)                                     ; 2388
                                          0A  12 008A4          BNEQ    157$                                           ;
                  52          48  AE      2E  C1 008A6          ADDL3   #46, XABKEY, R2                                ; 2390
                                  50      62  9A 008AB          MOVZBL  (R2), R0                                       ;
                                          55  11 008AE          BRB     158$                                           ;
                  53          48  AE      2E  C1 008B0 157$:    ADDL3   #46, XABKEY, R3                                ; 2392
                                  51      63  9A 008B5          MOVZBL  (R3), R1                                       ;
                  54          48  AE      2F  C1 008B8          ADDL3   #47, XABKEY, R4                                ;
                                  50      64  9A 008BD          MOVZBL  (R4), R0                                       ;
                                  50      51  C0 008C0          ADDL2   R1, R0                                         ;
                  53          48  AE      30  C1 008C3          ADDL3   #48, XABKEY, R3                                ; 2393
                                  51      63  9A 008C8          MOVZBL  (R3), R1                                       ;
                                  50      51  C0 008CB          ADDL2   R1, R0                                         ; 2392
                  53          48  AE      31  C1 008CE          ADDL3   #49, XABKEY, R3                                ; 2393
                                  51      63  9A 008D3          MOVZBL  (R3), R1                                       ;
                                  50      51  C0 008D6          ADDL2   R1, R0                                         ;
                  53          48  AE      32  C1 008D9          ADDL3   #50, XABKEY, R3                                ; 2394
                                  51      63  9A 008DE          MOVZBL  (R3), R1                                       ;
                                  50      51  C0 008E1          ADDL2   R1, R0                                         ; 2393
                  53          48  AE      33  C1 008E4          ADDL3   #51, XABKEY, R3                                ; 2394
                                  51      63  9A 008E9          MOVZBL  (R3), R1                                       ;
                                  50      51  C0 008EC          ADDL2   R1, R0                                         ;
```

```
             53        48   AE               34  C1 008EF          ADDL3   #52, XABKEY, R3                         ; 2395
                            51               63  9A 008F4          MOVZBL  (R3), R1                                ;
                            50               51  C0 008F7          ADDL2   R1, R0                                  ; 2394
             53        48   AE               35  C1 008FA          ADDL3   #53, XABKEY, R3                         ; 2395
                            51               63  9A 008FF          MOVZBL  (R3), R1                                ;
                            50               51  C0 00902          ADDL2   R1, R0                                  ;
             55        48   AE               16  C1 00905  158$:   ADDL3   #22, XABKEY, R5                         ; 2388
                            65               50  90 0090A          MOVB    R0, (R5)                                ;
             50        48   AE               04  C1 0090D          ADDL3   #4, XABKEY, R0                          ; 2402
                            60         0C    BE  D0 00912          MOVL    @12(SP), (R0)                           ;
                       0C   BE         48    AE  D0 00916          MOVL    XABKEY, @12(SP)                         ; 2403
                            02               59  F4 0091B  159$:   SOBGEQ  KEYNO, 160$                             ; 2255
                                             03  11 0091E          BRB     161$                                    ;
                                           FDE4  31 00920  160$:   BRW     132$                                    ;
             03        FF   AB               02  E0 00923  161$:   BBS     #2, -1(CCB), 162$                       ; 2414
                                           009B  31 00928          BRW     170$                                    ;
                            54               5B  D0 0092B  162$:   MOVL    CCB, OUR_CCB                            ; 2429
                                             50  D4 0092E          CLRL    R0                                      ; 2430
             53        04   AC               1E  C1 00930          ADDL3   #30, OPEN_ARG_BLK, R3                   ;
                            52               63  32 00935          CVTWL   (R3), R2                                ;
                                  00000000G  00  16 00938          JSB     BAS$$CB_PUSH                            ;
                            06         FC    AB  E9 0093E          BLBC    -4(CCB), 163$                           ; 2431
                            53         D0    AB  3C 00942          MOVZWL  -48(CCB), PARENT_IFI                    ;
                                             02  11 00946          BRB     164$                                    ;
                                             53  D4 00948  163$:   CLRL    PARENT_IFI                              ;
                       20   AE         C4    AB  9A 0094A  164$:   MOVZBL  -60(CCB), PARENT_ORG                    ; 2432
                       1C   AE         D2    AB  3C 0094F          MOVZWL  -46(CCB), PARENT_MRS                    ; 2433
                       18   AE         F6    AB  9A 00954          MOVZBL  -10(CCB), PARENT_RAT                    ; 2434
                            59         D9    AB  9A 00959          MOVZBL  -39(CCB), PARENT_RFM                    ; 2435
                            57         C5    AB  9A 0095D          MOVZBL  -59(CCB), PARENT_BKS                    ; 2436
                            55         A2    AB  3C 00961          MOVZWL  -94(CCB), PARENT_BLS                    ; 2437
             05        FF   AB               02  E1 00965          BBC     #2, -1(CCB), 165$                       ; 2443
                            52               01  D0 0096A          MOVL    #1, CONNECTED                           ; 2445
                                             06  11 0096D          BRB     166$                                    ;
                                             52  D4 0096F  165$:   CLRL    CONNECTED                               ; 2448
                       FF   AB               08  88 00971          BISB2   #8, -1(CCB)                             ; 2449
                                  00000000G  00  16 00975  166$:   JSB     BAS$$CB_POP                             ; 2452
                            5B               54  D0 0097B          MOVL    OUR_CCB, CCB                            ; 2453
                            0B               52  E9 0097E          BLBC    CONNECTED, 167$                         ; 2455
                            7E         00G   8F  9A 00981          MOVZBL  #BAS$K_INVFILOPT, -(SP)                 ;
                 00000000G  00               01  FB 00985          CALLS   #1, BAS$$STOP_IO                        ;
                                             53  D5 0098C  167$:   TSTL    PARENT_IFI                              ; 2457
                                             0B  12 0098E          BNEQ    168$                                    ;
                            7E         00G   8F  9A 00990          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                 ;
                 00000000G  00               01  FB 00994          CALLS   #1, BAS$$STOP_IO                        ;
                            03         20    AE  D1 0099B  168$:   CMPL    PARENT_ORG, #3                          ; 2459
                                             0B  13 0099F          BEQL    169$                                    ;
                            7E         00G   8F  9A 009A1          MOVZBL  #BAS$K_FILATTNOT, -(SP)                 ;
                 00000000G  00               01  FB 009A5          CALLS   #1, BAS$$STOP_IO                        ;
                        02  A6               53  B0 009AC  169$:   MOVW    PARENT_IFI, 2(FAB)                      ; 2461
                        36  A6         1C    AE  B0 009B0          MOVW    PARENT_MRS, 54(FAB)                     ; 2462
                        1E  A6         18    AE  90 009B5          MOVB    PARENT_RAT, 30(FAB)                     ; 2463
                        04  BE               59  90 009BA          MOVB    PARENT_RFM, @4(SP)                      ; 2464
                        3E  A6               57  90 009BE          MOVB    PARENT_BKS, 62(FAB)                     ; 2465
                        3C  A6               55  B0 009C2          MOVW    PARENT_BLS, 60(FAB)                     ; 2466
             50        04   AC               28  C1 009C6  170$:   ADDL3   #40, OPEN_ARG_BLK, R0                   ; 2478
                                             60  D5 009CB          TSTL    (R0)                                    ;
```

```
                                       3E  13 009CD         BEQL    173$                                      ;
                A1    AB               04  88 009CF         BISB2   #4, -95(CCB)                              ; 2481
                20    AE         C6    AB  32 009D3         CVTWL   -58(CCB), 32(SP)                          ; 2482
                                 20    AE  9F 009D8         PUSHAB  32(SP)                                    ;
                               0840    8F  BB 009DB         PUSHR   #^M<R6,R11>                               ;
      52        04    AC               28  C1 009DF         ADDL3   #40, OPEN_ARG_BLK, R2                     ;
                      92               03  FB 009E4         CALLS   #3, @(R2)                                 ;
                      59               50  D0 009E7         MOVL    R0, OPEN_STATUS                           ;
                      57               01  D0 009EA         MOVL    #1, CONNECT_STATUS                        ; 2483
                      1B               59  E8 009ED         BLBS    OPEN_STATUS, 172$                         ; 2485
                      17         08    A6  E9 009F0         BLBC    8(FAB), 172$                              ; 2492
                      0A         08    AB  E8 009F4         BLBS    8(CCB), 171$                              ; 2499
                      59         08    A6  D0 009F8         MOVL    8(FAB), OPEN_STATUS                       ; 2505
                      57         08    AB  D0 009FC         MOVL    8(CCB), CONNECT_STATUS                    ; 2506
                                       09  11 00A00         BRB     172$                                      ; 2499
                                       59  DD 00A02 171$:   PUSHL   OPEN_STATUS                               ; 2512
          00000000G   00               01  FB 00A04         CALLS   #1, LIB$STOP                              ;
                                       77  11 00A09 172$:   BRB     178$                                      ; 2492
      0B        FF    AB               02  E1 00A0D 173$:   BBC     #2, -1(CCB), 174$                         ; 2527
                                       56  DD 00A12         PUSHL   FAB                                       ; 2529
          00000000G   00               01  FB 00A14         CALLS   #1, SYS$DISPLAY                           ;
                                       55  11 00A1B         BRB     177$                                      ;
                                       56  DD 00A1D 174$:   PUSHL   FAB                                       ; 2539
          00000000G   00               01  FB 00A1F         CALLS   #1, SYS$PARSE                             ;
                      59               50  D0 00A26         MOVL    R0, OPEN_STATUS                           ;
                      2D               59  E9 00A29         BLBC    OPEN_STATUS, 175$                         ; 2541
      28        40    A6               02  E1 00A2C         BBC     #2, 64(FAB), 175$                         ; 2545
                      24         34    AE  E9 00A31         BLBC    52(SP), 175$                              ; 2546
                                 24    AE  D5 00A35         TSTL    36(SP)                                    ; 2547
                                       1F  12 00A38         BNEQ    175$                                      ;
                                 10    AE  D5 00A3A         TSTL    16(SP)                                    ; 2548
                                       1A  12 00A3D         BNEQ    175$                                      ;
                1E    A6               02  8A 00A3F         BICB2   #2, 30(FAB)                               ; 2554
                1E    A6               04  88 00A43         BISB2   #4, 30(FAB)                               ; 2555
                04    BE               03  90 00A47         MOVB    #3, @4(SP)                                ; 2559
                3F    A6               02  90 00A4B         MOVB    #2, 63(FAB)                               ; 2560
                2C    AB         DA    AB  9E 00A4F         MOVAB   -38(R11), 44(CCB)                         ; 2564
                FE    AB         40    8F  88 00A54         BISB2   #64, -2(CCB)                              ; 2568
      0B        FC    AB               03  E1 00A59 175$:   BBC     #3, -4(CCB), 176$                         ; 2573
                                       56  DD 00A5E         PUSHL   FAB                                       ; 2575
          00000000G   00               01  FB 00A60         CALLS   #1, SYS$OPEN                              ;
                                       09  11 00A67         BRB     177$                                      ;
                                       56  DD 00A69 176$:   PUSHL   FAB                                       ; 2577
          00000000G   00               01  FB 00A6B         CALLS   #1, SYS$CREATE                            ;
                      59               50  D0 00A72 177$:   MOVL    R0, OPEN_STATUS                           ;
                      0C               59  E9 00A75         BLBC    OPEN_STATUS, 178$                         ; 2580
                                       5B  DD 00A78         PUSHL   CCB                                       ;
          00000000G   00               01  FB 00A7A         CALLS   #1, SYS$CONNECT                           ;
                      57               50  D0 00A81         MOVL    R0, CONNECT_STATUS                        ;
                20    AE         FE    AB  9E 00A84 178$:   MOVAB   -2(CCB), 32(SP)                           ; 2590
      0F        20    BE               0A  E0 00A89         BBS     #10, @32(SP), 179$                        ;
      0E        07    A6               01  E1 00A8E         BBC     #1, 7(FAB), 180$                          ;
          00010619    8F         08    A6  D1 00A93         CMPL    8(FAB), #67097                            ;
                                       04  13 00A9B         BEQL    180$                                      ;
                FC    AB               08  88 00A9D 179$:   BISB2   #8, -4(CCB)                               ; 2592
 F0   AB      FF7A    CD               06  28 00AA1 180$:   MOVC3   #6, NAM_BLOCK+42, -16(CCB)                ; 2598
                D0    AB         02    A6  B0 00AA8         MOVW    2(FAB), -48(CCB)                          ; 2599
```

```
                    50      FF53   CD  9A 00AAD         MOVZBL  NAM_BLOCK+3, R0                         ; 2606
                                   08  13 00AB2         BEQL    181$                                    ;
              F8    AB      FF54   CD  D0 00AB4         MOVL    NAM_BLOCK+4, -8(CCB)                    ; 2609
                                   0D  11 00ABA         BRB     182$                                    ; 2610
                    50      FF5B   CD  9A 00ABC  181$:  MOVZBL  NAM_BLOCK+11, R0                        ; 2614
                                   0A  13 00AC1         BEQL    183$                                    ;
              F8    AB      FF5C   CD  D0 00AC3         MOVL    NAM_BLOCK+12, -8(CCB)                   ; 2617
              F7    AB             50  90 00AC9  182$:  MOVB    R0, -9(CCB)                             ; 2618
                    0A             59  E8 00ACD  183$:  BLBS    OPEN_STATUS, 184$                       ; 2630
                    7E             02  CE 00AD0         MNEGL   #2, -(SP)                               ;
       00000000G    00             01  FB 00AD3         CALLS   #1, BAS$$STOP_IO                        ;
              54    AE             03  D0 00ADA  184$:  MOVL    #3, UNWIND_ACTION                       ; 2636
                    0A             57  E8 00ADE         BLBS    CONNECT_STATUS, 185$                    ; 2638
                    7E             03  CE 00AE1         MNEGL   #3, -(SP)                               ;
       00000000G    00             01  FB 00AE4         CALLS   #1, BAS$$STOP_IO                        ;
  6D          20    BE             0A  E0 00AEB  185$:  BBS     #10, @32(SP), 190$                      ; 2646
                    52      40     A6  D0 00AF0         MOVL    64(FAB), R2                             ; 2650
  3D                52             02  E1 00AF4         BBC     #2, R2, 188$                            ;
              20    BE             20  88 00AF8         BISB2   #32, @32(SP)                            ; 2657
                            30     AB  9F 00AFC         PUSHAB  48(CCB)                                 ; 2659
              20    AE      50     8F  9A 00AFF         MOVZBL  #80, 32(SP)                             ;
                            20     AE  9F 00B04         PUSHAB  32(SP)                                  ;
       00000000G    00             02  FB 00B07         CALLS   #2, LIB$GET_VM
                    0B             50  E8 00B0E         BLBS    GET_VM_RESULT, 186$                     ;
                    7E      00G    8F  9A 00B11         MOVZBL  #BAS$K_MAXMEMEXC, -(SP)                 ; 2661
       00000000G    00             01  FB 00B15         CALLS   #1, BAS$$STOP_IO                        ;
                            34     AB  94 00B1C  186$:  CLRB    52(CCB)                                 ; 2663
              07    AB      40     8F  88 00B1F         BISB2   #64, 7(CCB)                             ; 2664
                            14     AE  D5 00B24         TSTL    20(SP)                                  ; 2666
                                   05  12 00B27         BNEQ    187$                                    ;
              D6    AB      3C     A6  B0 00B29         MOVW    60(FAB), -42(CCB)                       ; 2667
                            D4     AB  B4 00B2E  187$:  CLRW    -44(CCB)                                ; 2669
              A1    AB             02  88 00B31         BISB2   #2, -95(CCB)                            ; 2670
  24                52             01  E1 00B35  188$:  BBC     #1, R2, 190$                            ; 2681
  20                52             1B  E1 00B39         BBC     #27, R2, 190$                           ; 2682
                    1D             52  E9 00B3D         BLBC    R2, 190$                                ; 2683
  19                52             1A  E0 00B40         BBS     #26, R2, 190$                           ; 2684
  15                52             14  E0 00B44         BBS     #20, R2, 190$                           ; 2685
  11                52             02  E0 00B48         BBS     #2, R2, 190$                            ; 2686
                            14     AE  D5 00B4C         TSTL    20(SP)                                  ; 2690
                                   05  12 00B4F         BNEQ    189$                                    ;
              D6    AB      3C     A6  B0 00B51         MOVW    60(FAB), -42(CCB)                       ; 2691
                            D4     AB  B4 00B56  189$:  CLRW    -44(CCB)                                ; 2693
              A1    AB             02  88 00B59         BISB2   #2, -95(CCB)                            ; 2694
                    1F      2C     AE  E9 00B5D  190$:  BLBC    44(SP), 191$                            ; 2706
  50          04    AC             10  C1 00B61         ADDL3   #16, OPEN_ARG_BLK, R0                   ; 2707
                                   60  D5 00B66         TSTL    (R0)                                    ;
                                   16  13 00B68         BEQL    191$                                    ;
  50          04    AC             34  C1 00B6A         ADDL3   #52, OPEN_ARG_BLK, R0                   ; 2709
              14    AE             60  B1 00B6F         CMPW    (R0), 20(SP)                            ;
                                   0B  1E 00B73         BGEQU   191$                                    ;
                    7E      00G    8F  9A 00B75         MOVZBL  #BAS$K_RECOVEMAP, -(SP)                 ; 2711
       00000000G    00             01  FB 00B79         CALLS   #1, BAS$$STOP_IO                        ;
  03          FC    AB             03  E0 00B80  191$:  BBS     #3, -4(CCB), 192$                       ; 2719
                                 0395  31 00B85         BRW     269$                                    ;
                    50      1D     A6  9E 00B88  192$:  MOVAB   29(FAB), R0                             ; 2736
  05                00      08     AE  8F 00B8C         CASEB   8(SP), #0, #5                           ; 2729
```

```
   0021        0018        0012        000C     00B91 193$:    .WORD     194$-193$,-                          ;
                           003B        002D     00B99                    195$-193$,-                          ;
                                                                         196$-193$,-                          ;
                                                                         198$-193$,-                          ;
                                                                         199$-193$,-                          ;
                                                                         201$-193$                            ;
                       C4    AB          04  90 00B9D 194$:    MOVB      #4, -60(CCB)                         ; 2734
                                         0A  11 00BA1          BRB       197$                                 ; 2736
                       C4    AB          05  90 00BA3 195$:    MOVB      #5, -60(CCB)                         ; 2742
                                         04  11 00BA7          BRB       197$                                 ; 2744
                       C4    AB          01  90 00BA9 196$:    MOVB      #1, -60(CCB)                         ; 2750
                             52          60  9A 00BAD 197$:    MOVZBL    (R0), R2                             ; 2752
                                         16  11 00BB0          BRB       200$                                 ;
                       C4    AB          02  90 00BB2 198$:    MOVB      #2, -60(CCB)                         ; 2758
                             52          60  9A 00BB6          MOVZBL    (R0), R2                             ; 2760
                             10          52  91 00BB9          CMPB      R2, #16                              ;
                                         0A  11 00BBC          BRB       200$                                 ;
                       C4    AB          03  90 00BBE 199$:    MOVB      #3, -60(CCB)                         ; 2766
                             52          60  9A 00BC2          MOVZBL    (R0), R2                             ; 2768
                             20          52  91 00BC5          CMPB      R2, #32                              ;
                                         2E  13 00BC8 200$:    BEQL      205$                                 ;
                                         21  11 00BCA          BRB       204$                                 ;
                             52          60  9A 00BCC 201$:    MOVZBL    (R0), R2                             ; 2779
                                         06  12 00BCF          BNEQ      202$                                 ; 2782
                       C4    AB          01  90 00BD1          MOVB      #1, -60( .B)                         ; 2783
                                         21  11 00BD5          BRB       205$                                 ;
                             10          52  91 00BD7 202$:    CMPB      R2, #16                              ; 2785
                                         06  12 00BDA          BNEQ      203$                                 ;
                       C4    AB          02  90 00BDC          MOVB      #2, -60(CCB)                         ; 2786
                                         16  11 00BE0          BRB       205$                                 ;
                             20          52  91 00BE2 203$:    CMPB      R2, #32                              ; 2788
                                         06  12 00BE5          BNEQ      204$                                 ;
                       C4    AB          03  90 00BE7          MOVB      #3, -60(CCB)                         ; 2789
                                         0B  11 00BEB          BRB       205$                                 ;
                             7E    00G   8F  9A 00BED 204$:    MOVZBL    #BAS$K_FILATTNOT, -(SP)              ; 2792
                  00000000G  00          01  FB 00BF1          CALLS     #1, BAS$$STOP_IO                     ;
            50           04  AC          1A  C1 00BF8 205$:    ADDL3     #26, OPEN_ARG_BLK, R0                ; 2803
                                         60  B5 00BFD          TSTW      (R0)                                 ;
                                         1B  13 00BFF          BEQL      207$                                 ;
                             04    38    AE  E8 00C01          BLBS      56(SP), 206$                         ; 2804
                             13    30    AE  E9 00C05          BLBC      48(SP), 207$                         ; 2805
   5A          3E    A6      08          00  ED 00C09 206$:    CMPZV     #0, #8, 62(FAB), BKS                 ; 2808
                                         0B  13 00C0F          BEQL      207$                                 ;
                             7E    00G   8F  9A 00C11          MOVZBL    #BAS$K_FILATTNOT, -(SP)              ;
                  00000000G  00          01  FB 00C15          CALLS     #1, BAS$$STOP_IO                     ;
            50           04  AC          0E  C1 00C1C 207$:    ADDL3     #14, OPEN_ARG_BLK, R0                ; 2815
                                         60  B5 00C21          TSTW      (R0)                                 ;
                                         18  13 00C23          BEQL      208$                                 ;
                                         52  D5 00C25          TSTL      R2                                   ;
                                         14  12 00C27          BNEQ      208$                                 ;
40    AE       3C    A6      10          00  ED 00C29          CMPZV     #0, #16, 60(FAB), BLS                ; 2818
                                         0B  13 00C30          BEQL      208$                                 ;
                             7E    00G   8F  9A 00C32          MOVZBL    #BAS$K_FILATTNOT, -(SP)              ;
                  00000000G  00          01  FB 00C36          CALLS     #1, BAS$$STOP_IO                     ;
                             52    D2    AB  9E 00C3D 208$:    MOVAB     -46(CCB), R2                         ; 2824
                             50    36    A6  3C 00C41          MOVZWL    54(FAB), R0                          ; 2825
                             50  FF2E    CD  B1 00C45          CMPW      XABFHC+10, R0                        ;
```

```
                                   05  1B 00C4A          BLEQU   209$                                   ;
                      50     FF2E  CD  3C 00C4C          MOVZWL  XABFHC+10, R0                          ;
                      62           50  B0 00C51 209$:    MOVW    R0, (R2)                               ; 2824
                                   09  12 00C54          BNEQ    210$                                   ; 2831
                      62           58  B0 00C56          MOVW    RSZ, (R2)                              ;
                                   04  12 00C59          BNEQ    210$                                   ; 2837
                      62       3C  A6  B0 00C5B          MOVW    60(FAB), (R2)                          ;
                               14  AE  D5 00C5F 210$:    TSTL    20(SP)                                 ; 2846
                                   18  13 00C62          BEQL    211$                                   ;
                      01       04  BE  91 00C64          CMPB    @4(SP), #1                             ; 2848
                      12           12  12 00C68          BNEQ    211$                                   ;
   58         62      10           00  ED 00C6A          CMPZV   #0, #16, (R2), RSZ                     ; 2850
                                   0B  13 00C6F          BEQL    211$                                   ;
                      7E      00G  8F  9A 00C71          MOVZBL  #BAS$K_BADRECVAL, -(SP)                ;
              00000000G  00        01  FB 00C75          CALLS   #1, BAS$$STOP_IO                       ;
                      1B       2C  AE  E9 00C7C 211$:    BLBC    44(SP), 212$                           ; 2858
              50      04  AC       10  C1 00C80          ADDL3   #16, OPEN_ARG_BLK, R0                  ; 2859
                                   60  D5 00C85          TSTL    (R0)                                   ;
                                   12  13 00C87          BEQL    212$                                   ;
   58         62      10           00  ED 00C89          CMPZV   #0, #16, (R2), RSZ                     ; 2861
                                   0B  15 00C8E          BLEQ    212$                                   ;
                      7E      00G  8F  9A 00C90          MOVZBL  #BAS$K_BADRECVAL, -(SP)                ;
              00000000G  00        01  FB 00C94          CALLS   #1, BAS$$STOP_IO                       ;
                      05       3C  AE  E9 00C9B 212$:    BLBC    NO_MAP_REC_SPECIFIED, 213$             ; 2874
                               36  A6  B5 00C9F          TSTW    54(FAB)                                ;
                                   0E  12 00CA2          BNEQ    215$                                   ;
                      50           62  3C 00CA4 213$:    MOVZWL  (R2), R0                               ; 2878
                      58           50  D1 00CA7          CMPL    R0, RSZ                                ;
                                   03  1E 00CAA          BGEQU   214$                                   ;
                      50           58  D0 00CAC          MOVL    RSZ, R0                                ;
                      62           50  B0 00CAF 214$:    MOVW    R0, (R2)                               ;
              50      04  AC       10  C1 00CB2 215$:    ADDL3   #16, OPEN_ARG_BLK, R0                  ; 2884
                                   60  D5 00CB7          TSTL    (R0)                                   ;
                                   11  13 00CB9          BEQL    216$                                   ;
                      14  AE       62  B1 00CBB          CMPW    (R2), 20(SP)                           ; 2885
                                   0B  1E 00CBF          BGEQU   216$                                   ;
                      7E      00G  8F  9A 00CC1          MOVZBL  #BAS$K_BADRECVAL, -(SP)                ; 2887
              00000000G  00        01  FB 00CC5          CALLS   #1, BAS$$STOP_IO                       ;
                      12       28  AE  E9 00CCC 216$:    BLBC    40(SP), 217$                           ; 2893
                   0200  8F        62  B1 00CD0          CMPW    (R2), #512                             ; 2894
                                   0B  1E 00CD5          BGEQU   217$                                   ;
                      7E      00G  8F  9A 00CD7          MOVZBL  #BAS$K_BADRECVAL, -(SP)                ; 2896
              00000000G  00        01  FB 00CDB          CALLS   #1, BAS$$STOP_IO                       ;
                  04  00       24  AE  8F 00CE2 217$:    CASEB   36(SP), #0, #4                         ; 2904
 0054           0048      0042     000C     00CE7 218$:  .WORD   219$-218$,-                            ;
                                   0070     00CEF                221$-218$,-                            ;
                                                                 222$-218$,-                            ;
                                                                 223$-218$,-                            ;
                                                                 226$-218$                              ;
                                   64  11 00CF1          BRB     226$                                   ; 2952
                      01       08  AE  91 00CF3 219$:    CMPB    8(SP), #1                              ; 2912
                                   69  13 00CF7          BEQL    227$                                   ;
                      05       08  AE  91 00CF9          CMPB    8(SP), #5                              ;
                                   63  13 00CFD          BEQL    227$                                   ;
              5E      40  A6       02  E1 00CFF          BBC     #2, 64(FAB), 227$                      ; 2919
                      5A       34  AE  E9 00D04          BLBC    52(SP), 227$                           ; 2920
                               24  AE  D5 00D08          TSTL    36(SP)                                 ; 2921
```

```
                                             55  12 000DB         BNEQ    227$                                  ;
                                    10       AE  D5 000DD         TSTL    16(SP)                                ; 2922
                                             50  12 00D10         BNEQ    227$                                  ;
                               03   04       BE  91 00D12         CMPB    @4(SP), #3                            ; 2926
                                             0B  13 00D16         BEQL    220$                                  ;
                               7E   00G      8F  9A 00D18         MOVZBL  #BAS$K_FILATTNOT, -(SP)               ;
                      00000000G    00        01  FB 00D1C         CALLS   #1, BAS$$STOP_IO                      ;
                               02   3F       A6  91 00D23 220$:   CMPB    63(FAB), #2                           ; 2928
                                             2C  11 00D27         BRB     225$                                  ;
                               01   04       BE  91 00D29 221$:   CMPB    @4(SP), #1                            ; 2934
                                             26  11 00D2D         BRB     225$                                  ;
                               02   04       BE  91 00D2F 222$:   CMPB    @4(SP), #2                            ; 2938
                                             2D  13 00D33         BEQL    227$                                  ;
                               03   04       BE  91 00D35         CMPB    @4(SP), #3                            ;
                                             1A  11 00D39         BRB     225$                                  ;
                               03   04       BE  91 00D3B 223$:   CMPB    @4(SP), #3                            ; 2945
                                             0B  13 00D3F         BEQL    224$                                  ;
                               7E   00G      8F  9A 00D41         MOVZBL  #BAS$K_FILATTNOT, -(SP)               ;
                      00000000G    00        01  FB 00D45         CALLS   #1, BAS$$STOP_IO                      ;
                  50        04     AC        07  C1 00D4C 224$:   ADDL3   #7, OPEN_ARG_BLK, R0                  ; 2947
                                   60   3F   A6  91 00D51         CMPB    63(FAB), (R0)                         ;
                                             0B  13 00D55 225$:   BEQL    227$                                  ;
                               7E   00G      8F  9A 00D57 226$:   MOVZBL  #BAS$K_FILATTNOT, -(SP)               ;
                      00000000G    00        01  FB 00D5B         CALLS   #1, BAS$$STOP_IO                      ;
                  05               00   10   AE  8F 00D62 227$:   CASEB   16(SP), #0, #5                        ; 2959
      003D        0030           0037      000E     00D67 228$:   .WORD   229$-228$,-                           ;
                                 0054      0044     00D6F                 233$-228$,-                           ;
                                                                          232$-228$,-                           ;
                                                                          234$-228$,-                           ;
                                                                          235$-228$,-                           ;
                                                                          237$-228$                             ;
                                             3B  11 00D73         BRB     236$                                  ; 3017
                  0E        40     A6        02  E1 00D75 229$:   BBC     #2, 64(FAB), 230$                     ; 2967
                                   0A   34   AE  E9 00D7A         BLBC    52(SP), 230$                          ; 2968
                                        24   AE  D5 00D7E         TSTL    36(SP)                                ; 2969
                                             05  12 00D81         BNEQ    230$                                  ;
                                        10   AE  D5 00D83         TSTL    16(SP)                                ; 2970
                                             23  13 00D86         BEQL    235$                                  ;
                                   0B   28   AE  E9 00D88 230$:   BLBC    40(SP), 232$                          ; 2982
                                   20   1E   A6  E8 00D8C 231$:   BLBS    30(FAB), 236$                         ; 2986
                  2F        1E     A6        02  E1 00D90         BBC     #2, 30(FAB), 239$                     ;
                                             19  11 00D95         BRB     236$                                  ;
                  1F        1E     A6        01  E0 00D97 232$:   BBS     #1, 30(FAB), 237$                     ; 2991
                                             12  11 00D9C         BRB     236$                                  ;
                                   19   1E   A6  E8 00D9E 233$:   BLBS    30(FAB), 237$                         ; 2995
                                             0C  11 00DA2         BRB     236$                                  ;
                  07        1E     A6        01  E0 00DA4 234$:   BBS     #1, 30(FAB), 236$                     ; 3003
                                             E1  11 00DA9         BRB     231$                                  ;
                  10        1E     A6        02  E0 00DAB 235$:   BBS     #2, 30(FAB), 238$                     ; 3009
                                   7E   00G  8F  9A 00DB0 236$:   MOVZBL  #BAS$K_RECATTNOT, -(SP)               ;
                      00000000G    00        01  FB 00DB4         CALLS   #1, BAS$$STOP_IO                      ;
                  04        1E     A6        02  E1 00DBB 237$:   BBC     #2, 30(FAB), 239$                     ; 3024
                            A1     AB        01  88 00DC0 238$:   BISB2   #1, -95(CCB)                          ;
                                   03   30   AE  E8 00DC4 239$:   BLBS    48(SP), 241$                          ; 3030
                                           01CD  31 00DC8 240$:   BRW     281$                                  ;
                                   02        6C  91 00DCB 241$:   CMPB    (AP), #2                              ;
                                             F8  1F 00DCE         BLSSU   240$                                  ;
                                   54      08   AC   D0 000D0          MOVL     KEY_INFO_BLK, R4                       ; 3045
                         FF21      CD           64   91 000D4          CMPB     (R4), XABSUM+9                         ;
                                                0B   1B 000D9          BLEQU    242$                                   ;
                                   7E      00G  8F   9A 000DB          MOVZBL   #BAS$K_FILATTNOT, -(SP)                ; 3047
                    00000000G      00           01   FB 000DF          CALLS    #1, BAS$$STOP_IO                       ;
                                   52      0C   BE   D0 000E6 242$:    MOVL     @12(SP), XABKEY                        ; 3054
                                                DC   13 000EA 243$:    BEQL     240$                                   ; 3056
                                   15           62   91 000EC          CMPB     (XABKEY), #21                          ; 3059
                                                03   13 000EF          BEQL     244$                                   ;
                                              0122   31 000F1          BRW      268$                                   ;
                                   57      17   A2   9A 000F4 244$:    MOVZBL   23(XABKEY), KEYNO                      ; 3062
                                   50      01   A4   9A 000F8          MOVZBL   1(R4), R0                              ; 3063
                                   50           57   C4 000FC          MULL2    KEYNO, R0                              ;
                                   53      04 A440   9E 000FF          MOVAB    4(R4)[R0], KEY_PTR                     ;
                           02      A3      1E   A2   B1 00E04          CMPW     30(XABKEY), 2(KEY_PTR)                 ; 3069
                                                19   12 00E09          BNEQ     245$                                   ;
                                   63      2E   A2   91 00E0B          CMPB     46(XABKEY), (KEY_PTR)                  ; 3070
                                                13   12 00E0F          BNEQ     245$                                   ;
                   50      01      A3      12   A2   8D 00E11          XORB3    18(XABKEY), 1(KEY_PTR), R0             ; 3071
                   09              50           01   E0 00E17          BBS      #1, R0, 245$                           ;
                   50      01      A3      12   A2   8D 00E1B          XORB3    18(XABKEY), 1(KEY_PTR), R0             ; 3072
                                   0B           50   E9 00E21          BLBC     R0, 246$                               ;
                                   7E      00G  8F   9A 00E24 245$:    MOVZBL   #BAS$K_FILATTNOT, -(SP)                ; 3074
                    00000000G      00           01   FB 00E28          CALLS    #1, BAS$$STOP_IO                       ;
                                   50      13   A2   9E 00E2F 246$:    MOVAB    19(XABKEY), R0                         ; 3119
                   0E              07      04   A3   8F 00E33          CASEB    4(KEY_PTR), #7, #14                    ; 3082
    0034           0034            002A         0020    00E38 247$:    .WORD    248$-247$,-                            ;
    0034           0034            0034         0034    00E40                   249$-247$,-                            ;
    0034           0034            0034         0034    00E48                   250$-247$,-                            ;
                   0038            0034         0034    00E50                   250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                250$-247$,-                            ;
                                                                                251$-247$                              ;
                                                14   11 00E56          BRB      250$                                   ; 3119
                                   01           60   91 00E58 248$:    CMPB     (R0), #1                               ; 3089
                                                23   13 00E5B          BEQL     253$                                   ;
                                   02           60   91 00E5D          CMPB     (R0), #2                               ; 3090
                                                11   11 00E60          BRB      252$                                   ;
                                   03           60   91 00E62 249$:    CMPB     (R0), #3                               ; 3097
                                                19   13 00E65          BEQL     253$                                   ;
                                   04           60   91 00E67          CMPB     (R0), #4                               ; 3098
                                                07   11 00E6A          BRB      252$                                   ;
                                                60   95 00E6C 250$:    TSTB     (R0)                                   ; 3105
                                                03   11 00E6E          BRB      252$                                   ;
                                   05           60   91 00E70 251$:    CMPB     (R0), #5                               ; 3112
                                                0B   13 00E73 252$:    BEQL     253$                                   ;
                                   7E      00G  8F   9A 00E75          MOVZBL   #BAS$K_FILATTNOT, -(SP)                ; 3113
                    00000000G      00           01   FB 00E79          CALLS    #1, BAS$$STOP_IO                       ;
                                   03      2C   AE   E8 00E80 253$:    BLBS     44(SP), 255$                           ; 3128
```

```
                                    008F  31 00E84 254$:     BRW     268$                                        ;
                  58          05    A3  9A 00E87 255$:     MOVZBL  5(KEY_PTR), R8                            ; 3131
                                    F7  15 00E8B           BLEQ    254$                                        ;
                                    55  D4 00E8D           CLRL    KEY_NUM                                     ; 3134
                                    7F  11 00E8F           BRB     267$                                        ;
           06             01        55  CF 00E91 256$:     CASEL   KEY_NUM, #1, #6                             ; 3136
 0038      002A           001C     000E    00E95 257$:     .WORD   258$-257$,-                                 ;
           0062           0054     0046    00E9D                   259$-257$,-                                 ;
                                                                   260$-257$,-                                 ;
                                                                   261$-257$,-                                 ;
                                                                   262$-257$,-                                 ;
                                                                   263$-257$,-                                 ;
                                                                   264$-257$                                   ;
                  07     A3   20    A2  B1 00EA3 258$:     CMPW    32(XABKEY), 7(KEY_PTR)                      ; 3139
                                    5B  12 00EA8           BNEQ    266$                                        ;
                  06     A3   2F    A2  91 00EAA           CMPB    47(XABKEY), 6(KEY_PTR)                      ; 3140
                                    52  11 00EAF           BRB     265$                                        ;
                  0A     A3   22    A2  B1 00EB1 259$:     CMPW    34(XABKEY), 10(KEY_PTR)                     ; 3145
                                    4D  12 00EB6           BNEQ    266$                                        ;
                  09     A3   30    A2  91 00EB8           CMPB    48(XABKEY), 9(KEY_PTR)                      ; 3146
                                    44  11 00EBD           BRB     265$                                        ;
                  0D     A3   24    A2  B1 00EBF 260$:     CMPW    36(XABKEY), 13(KEY_PTR)                     ; 3151
                                    3F  12 00EC4           BNEQ    266$                                        ;
                  0C     A3   31    A2  91 00EC6           CMPB    49(XABKEY), 12(KEY_PTR)                     ; 3152
                                    36  11 00ECB           BRB     265$                                        ;
                  10     A3   26    A2  B1 00ECD 261$:     CMPW    38(XABKEY), 16(KEY_PTR)                     ; 3157
                                    31  12 00ED2           BNEQ    266$                                        ;
                  0F     A3   32    A2  91 00ED4           CMPB    50(XABKEY), 15(KEY_PTR)                     ; 3158
                                    28  11 00ED9           BRB     265$                                        ;
                  13     A3   28    A2  B1 00EDB 262$:     CMPW    40(XABKEY), 19(KEY_PTR)                     ; 3163
                                    23  12 00EE0           BNEQ    266$                                        ;
                  12     A3   33    A2  91 00EE2           CMPB    51(XABKEY), 18(KEY_PTR)                     ; 3164
                                    1A  11 00EE7           BRB     265$                                        ;
                  16     A3   2A    A2  B1 00EE9 263$:     CMPW    42(XABKEY), 22(KEY_PTR)                     ; 3169
                                    15  12 00EEE           BNEQ    266$                                        ;
                  15     A3   34    A2  91 00EF0           CMPB    52(XABKEY), 21(KEY_PTR)                     ; 3170
                                    0C  11 00EF5           BRB     265$                                        ;
                  19     A3   2C    A2  B1 00EF7 264$:     CMPW    44(XABKEY), 25(KEY_PTR)                     ; 3175
                                    07  12 00EFC           BNEQ    266$                                        ;
                  18     A3   35    A2  91 00EFE           CMPB    53(XABKEY), 24(KEY_PTR)                     ; 3176
                                    0B  13 00F03 265$:     BEQL    267$                                        ;
                         7E   00G   8F  9A 00F05 266$:     MOVZBL  #BAS$K_FILATTNOT, -(SP)                     ; 3178
              00000000G  00         01  FB 00F09           CALLS   #1, BAS$$STOP_IO                            ;
 FF7B          55        01         58  F1 00F10 267$:     ACBL    R8, #1, KEY_NUM, 256$                       ; 3134
                         52   04    A2  D0 00F16 268$:     MOVL    4(XABKEY), XABKEY                           ; 3187
                                  FECD  31 00F1A           BRW     243$                                        ; 3056
                                    58  D5 00F1D 269$:     TSTL    RSZ                                         ; 3200
                                    0D  12 00F1F           BNEQ    270$                                        ;
                         7E   00G   8F  9A 00F21           MOVZBL  #BAS$K_BADRECVAL, -(SP)                     ;
              00000000G  00         01  FB 00F25           CALLS   #1, BAS$$STOP_IO                            ;
                                    04  11 00F2C           BRB     271$                                        ;
                  D2     AB         58  B0 00F2E 270$:     MOVW    RSZ, -46(CCB)                               ;
                  FD     AB         20  8A 00F32 271$:     BICB2   #32, -3(CCB)                                ; 3205
           05            00   08    AE  8F 00F36           CASEB   8(SP), #0, #5                               ; 3211
 001E      0018         0012       000C    00F3B 272$:     .WORD   273$-272$,-                                 ;
                        002A       0024    00F43                   274$-272$,-                                 ;
                                                                   275$-272$,-                                 ;
```

```
                                                                      276$-272$,-                         :
                                                                      277$-272$,-                         :
                                                                      278$-272$                           :
                       C4    AB         04  90 00F47 273$:   MOVB     #4, -60(CCB)                        : 3215
                                        23  11 00F4B         BRB      279$                                :
                       C4    AB         05  90 00F4D 274$:   MOVB     #5, -60(CCB)                        : 3218
                                        1D  11 00F51         BRB      279$                                :
                       C4    AB         01  90 00F53 275$:   MOVB     #1, -60(CCB)                        : 3221
                                        17  11 00F57         BRB      279$                                :
                       C4    AB         02  90 00F59 276$:   MOVB     #2, -60(CCB)                        : 3224
                                        11  11 00F5D         BRB      279$                                :
                       C4    AB         03  90 00F5F 277$:   MOVB     #3, -60(CCB)                        : 3227
                                        0B  11 00F63         BRB      279$                                :
                             7E    00G  8F  9A 00F65 278$:   MOVZBL   #BA_$K_FILATTNOT, -(SP)             : 3230
                  00000000G  00         01  FB 00F69         CALLS    #1, BAS$$STOP_IO                    :
                             03         6E  91 00F70 279$:   CMPB     (SP), #3                            : 3237
                                        0B  12 00F73         BNEQ     280$                                :
                             7E    00G  8F  9A 00F75         MOVZBL   #BAS$K_ILLILLACC, -(SP)             :
                  00000000G  00         01  FB 00F79         CALLS    #1, BAS$$STOP_IO                    :
                             14    28   AE  E9 00F80 280$:   BLBC     40(SP), 281$                        : 3243
                  00000200   8F         58  D1 00F84         CMPL     RSZ, #512                           : 3244
                                        0B  18 00F8B         BGEQ     281$                                :
                             7E    00G  8F  9A 00F8D         MOVZBL   #BAS$K_BADRECVAL, -(SP)             : 3246
                  00000000G  00         01  FB 00F91         CALLS    #1, BAS$$STOP_IO                    :
                             05    08   AE  91 00F98 281$:   CMPB     8(SP), #5                           : 3256
                                        16  13 00F9C         BEQL     283$                                :
                                   04   BE  95 00F9E         TSTB     @4(SP)                              : 3262
                                        06  13 00FA1         BEQL     282$                                :
                             03    04   BE  91 00FA3         CMPB     @4(SP), #3                          :
                                        0B  1B 00FA7         BLEQU    283$                                :
                             7E    00G  8F  9A 00FA9 282$:   MOVZBL   #BAS$K_FILATTNOT, -(SP)             : 3266
                  00000000G  00         01  FB 00FAD         CALLS    #1, BAS$$STOP_IO                    :
                         F6  AB    1E   A6  90 00FB4 283$:   MOVB     30(FAB), -10(CCB)                   : 3273
                         D9  AB    04   BE  90 00FB9         MOVB     @4(SP), -39(CCB)                    : 3274
                         C5  AB    3E   A6  90 00FBE         MOVB     62(FAB), -59(CCB)                   : 3275
                         A2  AB    3C   A6  B0 00FC3         MOVW     60(FAB), -94(CCB)                   : 3276
                         DC  AB    10   A6  D0 00FC8         MOVL     16(FAB), -36(CCB)                   : 3277
                 05          00    08   AE  8F 00FCD         CASEB    8(SP), #0, #5                       : 3281
   0018          0018      000C       0018     00FD2 284$:   .WORD    287$-284$,-                         :
                           0012       0018     00FDA                  285$-284$,-                         :
                                                                      287$-284$,-                         :
                                                                      287$-284$,-                         :
                                                                      287$-284$,-                         :
                                                                      286$-284$                           :
                             50    10   A6  D0 00FDE 285$:   MOVL     16(FAB), R0                         : 3285
                                        08  11 00FE2         BRB      288$                                :
                             50    38   A6  D0 00FE4 286$:   MOVL     56(FAB), R0                         : 3288
                                        02  11 00FE8         BRB      288$                                :
                                        50  D4 00FEA 287$:   CLRL     R0                                  : 3281
                         E4  AB         50  D0 00FEC 288$:   MOVL     R0, -28(CCB)                        : 3279
                  00000000'  EF    40   A6  D0 00FF0         MOVL     64(FAB), L_STATUS                   : 3299
                             52    0C   AE  D0 00FF8         MOVL     12(SP), XAB_PTR                     : 3310
                                        62  D5 00FFC 289$:   TSTL     (XAB_PTR)                           : 3312
                                        3D  13 00FFE         BEQL     291$                                :
                         4C  AE         62  D0 01000         MOVL     (XAB_PTR), XABKEY                   : 3314
                             50    4C   AE  D0 01004         MOVL     XABKEY, R0                          : 3323
                             15    4C   BE  91 01008         CMPB     @XABKEY, #21                        : 3316
```

```
                                                     29 12 0100C          BNEQ    290$                                ;
                                       62         04 A0 D0 0100E          MOVL    4(R0), (XAB_PTR)                    ; 3323
                                                  4C AE 9F 01012          PUSHAB  XABKEY                              ; 3324
                                   44  AE         4C 8F 9A 01015          MOVZBL  #76, 68(SP)                         ;
                                                  44 AE 9F 0101A          PUSHAB  68(SP)                              ;
                            00000000G  00            02 FB 0101D          CALLS   #2, LIB$FREE_VM                     ;
                                       53            50 D0 01024          MOVL    R0, FREE_VM_STATUS                  ;
                                       D2            53 E8 01027          BLBS    FREE_VM_STATUS, 289$                ; 3326
                                       7E        00G 8F 9A 0102A          MOVZBL  #BAS$K_PROLOSSOR, -(SP)             ;
                            00000000G  00            01 FB 0102E          CALLS   #1, BAS$$STOP_IO                    ;
                                                     C5 11 01035          BRB     289$                                ; 3316
                                       52         04 A0 9E 01037 290$:    MOVAB   4(R0), XAB_PTR                      ; 3330
                                                     BF 11 0103B          BRB     289$                                ; 3312
                                       57         EC AB 9E 0103D 291$:    MOVAB   -20(CCB), R7                        ; 3352
                       50          04  AC            10 C1 01041          ADDL3   #16, OPEN_ARG_BLK, R0               ; 3339
                                                     60 D5 01046          TSTL    (R0)                                ;
                                                     39 12 01048          BNEQ    294$                                ;
                                                  D2 AB B5 0104A          TSTW    -46(CCB)                            ; 3350
                                                     0B 12 0104D          BNEQ    292$                                ;
                                       7E        00G 8F 9A 0104F          MOVZBL  #BAS$K_BADRECVAL, -(SP)             ;
                            00000000G  00            01 FB 01053          CALLS   #1, BAS$$STOP_IO                    ;
                                                     57 DD 0105A 292$:    PUSHL   R7                                  ; 3352
                                   44  AE         D2 AB 3C 0105C          MOVZWL  -46(CCB), 68(SP)                    ;
                                                  44 AE 9F 01061          PUSHAB  68(SP)                              ;
                            00000000G  00            02 FB 01064          CALLS   #2, LIB$GET_VM                      ;
                                       0B            50 E8 0106B          BLBS    GET_VM_RESULT, 293$                 ; 3354
                                       7E        00G 8F 9A 0106E          MOVZBL  #BAS$K_MAXMEMEXC, -(SP)             ;
                            00000000G  00            01 FB 01072          CALLS   #1, BAS$$STOP_IO                    ;
       D2    AB           00           6E            00 2C 01079 293$:    MOVC5   #0, (SP), #0, -46(CCB), @0(R7)      ; 3360
                                                  00 B7    0107F
                                                     0E 11 01081          BRB     295$                                ; 3339
                       50          04  AC            10 C1 01083 294$:    ADDL3   #16, OPEN_ARG_BLK, R0               ; 3364
                                       67            60 D0 01088          MOVL    (R0), (R7)                          ;
                                   20  BE       8000 8F A8 0108B          BISW2   #32768, @32(SP)                     ; 3365
                                       52         F8 AB D0 01091 295$:    MOVL    -8(CCB), OLD_ADDRESS                ; 3380
                                                  F8 AB 9F 01095          PUSHAB  -8(CCB)                             ; 3381
                                   44  AE         F7 AB 9A 01098          MOVZBL  -9(CCB), 68(SP)                     ;
                                                  44 AE 9F 0109D          PUSHAB  68(SP)                              ;
                            00000000G  00            02 FB 010A0          CALLS   #2, LIB$GET_VM                      ;
                                       0B            50 E8 010A7          BLBS    GET_VM_RESULT, 296$                 ; 3383
                                       7E        00G 8F 9A 010AA          MOVZBL  #BAS$K_MAXMEMEXC, -(SP)             ;
                            00000000G  00            01 FB 010AE          CALLS   #1, BAS$$STOP_IO                    ;
                                       50         F7 AB 9A 010B5 296$:    MOVZBL  -9(CCB), R0                         ; 3385
                  F8      BB           62            50 28 010B9          MOVC3   R0, (OLD_ADDRESS), @-8(CCB)         ;
                                   20  BE            01 88 010BE          BISB2   #1, @32(SP)                         ; 3386
                                   24  AB            67 D0 010C2          MOVL    (R7), 36(CCB)                       ; 3391
                                   20  AB         D2 AB B0 010C6          MOVW    -46(CCB), 32(CCB)                   ; 3392
                                   9C  AB            67 D0 010CB          MOVL    (R7), -100(CCB)                     ; 3393
                                                  E8 AB D4 010CF          CLRL    -24(CCB)                            ; 3397
                                                  3C AB D4 010D2          CLRL    60(CCB)                             ; 3398
                                   D8  AB            01 90 010D5          MOVB    #1, -40(CCB)                        ; 3402
                                   FC  AB            01 88 010D9          BISB2   #1, -4(CCB)                         ; 3403
                                       07  00000000G 00 E8 010DD          BLBS    BAS$$L_XIT_LOCK, 297$               ; 3411
                            00000000G  00            00 FB 010E4          CALLS   #0, BAS$$DECL_EXITH                 ;
                                           00000000G 00 16 010EB 297$:    JSB     BAS$$CB_POP                         ; 3417
                                                        04 010F1          RET                                         ; 3419
                                                      0000 010F2 298$:    .WORD   Save nothing                        ; 1446
```

```
                                    50        08   AC  D0 010F4         MOVL     8(AP), R0                                   ;
                                    50        04   A0  D0 010F8         MOVL     4(R0), R0                                   ;
                                            FE10   C0  9F 010FC         PUSHAB   UNWIND_CCB                                  ;
                                            FE14   C0  9F 01100         PUSHAB   UNWIND_ACTION                               ;
                                                   02  DD 01104         PUSHL    #2                                          ;
                                                   5E  DD 01106         PUSHL    SP                                          ;
                                    7E        04   AC  7D 01108         MOVQ     4(AP), -(SP)                                ;
                            0000V   CF             03  FB 0110C         CALLS    #3, OPEN_HANDLER                            ;
                                                       04 01111         RET                                                  ;

; Routine Size:  4370 bytes,    Routine Base:  _BAS$CODE + 0000


; 2502          3420  1
```

```
1 ROUTINE OPEN_HANDLER (                                 ! Handle an UNWIND from OPEN
1         SIG,                                           ! Signal vector
1         MECH,                                          ! Mechanism vector
1         ENBL                                           ! Enable vector
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       If we are unwinding, do the indicated OPEN cleanup, either nothing, POP
1 !       the CCB, mark the CCB for deallocation and POP it, or RMS CLOSE the CCB
1 !       and POP it.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIG.rl.a        A counted vector of parameters to LIB$SIGNAL/STOP
1 !       MECH.rl.a       A counted vector of info from CHF
1 !       ENBL.ra.a       A counted vector of ENABLE argument addresses.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       Always SS$_RESIGNAL, which is ignored when unwinding.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May RMS CLOSE or DISCONNECT the file, and may deallocate the CCB.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SIG : REF VECTOR,
2         MECH : REF VECTOR,
2         ENBL : REF VECTOR;
2
2     LOCAL
2         MY_UNWIND_ACT : VOLATILE,
2         MY_UNWIND_CCB : VOLATILE;
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2 !+
2 ! Define names for the two items in the ENABLE vector.
2 !-
2
2     BIND
2         UNWIND_ACTION = .ENBL [1],
2               UNWIND_CCB = .ENBL [2];
2
2       !+
2       ! We are our own handler, in case the CLOSE or DISCONNECT fails.
2       !-
2
2           ENABLE
2               OPEN_HANDLER (MY_UNWIND_ACT, MY_UNWIND_CCB);
2
2       !+
2       ! Don't do anything yet.
2       !-
2           MY_UNWIND_ACT = UNWIND_NOP;
2       !+
2       ! Just resignal if this is an UNWIND or if the error is not severe.  Otherwise
2       ! we clean up the I/O data base prior to signalling the SEVERE error.
2       !-
2
3           IF (LIB$MATCH_COND (SIG [1], %REF (SS$_UNWIND)) OR (.BLOCK [SIG [1], STS$V_SEVERITY] NEQ STS$K_SEVERE))
2           THEN
2               RETURN (SS$_RESIGNAL);
2
2       !+
2       ! Depending on the action selected, do things.
2       !-
2           CCB = .UNWIND_CCB;
2
2           CASE .UNWIND_ACTION FROM UNWIND_MIN TO UNWIND_MAX OF
2               SET
2
2               [UNWIND_NOP] :                          ! Do nothing.
3                   BEGIN
3                   0
2                   END;
2
2               [UNWIND_POP] :                          ! POP the specified CCB
2                   BAS$$CB_POP ();
2
2               [UNWIND_DEALLOC] :                      ! Mark the specified CCB for deallocation, then POP it
3                   BEGIN
3                   CCB [LUB$V_DEALLOC] = 1;
3                   BAS$$CB_POP ();
2                   END;
2       !+
2       ! RMS CLOSE or DISCONNECT the specified CCB (which marks it for deallocation), then POP it
2       ! (which will usually deallocate it).
2       !-
2
2               [UNWIND_CLOSE] :
3                   BEGIN
3       !+
3       ! If the CLOSE fails, deallocate and POP.
3       !-
3                   MY_UNWIND_CCB = .CCB;
3                   MY_UNWIND_ACT = UNWIND_DEALLOC;
3
3                   IF ( NOT OTS$$CLOSE_FILE ()) THEN BAS$$STOP_IO (BAS$K_IOERR_REC);
```

```
; 2618          3535  3
; 2619          3536  3            BAS$$CB_POP ();
; 2620          3537  2            END;
; 2621          3538  2         TES;
; 2622          3539  2
; 2623          3540  2      RETURN (SS$_RESIGNAL);
; 2624          3541  1      END;                                             ! end of OPEN_HANDLER



                                                  0804 00000 OPEN_HANDLER:
                                                                  .WORD   Save R2,R11                                       ; 3421
                              52          0C   AC  D0 00002       MOVL    ENBL, R2                                          ; 3477
                                               7E  7C 00006       CLRQ    MY_UNWIND_CCB                                     ; 3478
                              6D        0065   CF  DE 00008       MOVAL   6$, (FP)
                                          04   AE  D4 0000D       CLRL    MY_UNWIND_ACT                                     ; 3490
                              7E        0920   8F  3C 00010       MOVZWL  #2336, -(SP)                                      ; 3496
                                               5E  DD 00015       PUSHL   SP                                                ;
                              5B          04   AC  D0 00017       MOVL    SIG, R11                                          ;
                                          04   AB  9F 0001B       PUSHAB  4(R11)                                            ;
                   00000000G  00               02  FB 0001E       CALLS   #2, LIB$MATCH_COND                                ;
                              43               50  E8 00025       BLBS    R0, 5$                                            ;
      04        04   AB       03               00  ED 00028       CMPZV   #0, #3, 4(R11), #4                                ;
                              3B               12  12 0002E       BNEQ    5$                                                ;
                              5B          08   B2  D0 00030       MOVL    @8(R2), CCB                                       ; 3503
                       03     00          04   B2  CF 00034       CASEL   @4(R2), #0, #3                                    ; 3505
    0010             000A   002C            0032      00039 1$:   .WORD   5$-1$,-                                           ;
                                                                          4$-1$,-                                           ;
                                                                          2$-1$,-                                           ;
                                                                          3$-1$                                             ;
                                               28  11 00041       BRB     5$                                                ; 3509
                      FF      AB               10  88 00043 2$:   BISB2   #16, -1(CCB)                                      ; 3518
                                               1C  11 00047       BRB     4$                                                ; 3519
                      04      AE               5B  D0 00049 3$:   MOVL    CCB, MY_UNWIND_CCB                                ; 3531
                      08      AE               02  D0 0004D       MOVL    #2, MY_UNWIND_ACT                                 ; 3532
                   00000000G  00               00  FB 00051       CALLS   #0, OTS$$CLOSE_FILE                               ; 3534
                              0A               50  E8 00058       BLBS    R0, 4$                                            ;
                              7E               01  CE 0005B       MNEGL   #1, -(SP)                                         ;
                   00000000G  00               01  FB 0005E       CALLS   #1, BAS$$STOP_IO                                  ;
                                   00000000G   00  16 00065 4$:   JSB     BAS$$CB_POP                                       ; 3536
                              50        0918   8F  3C 0006B 5$:   MOVZWL  #2328, R0                                         ; 3540
                                                   04 00070       RET                                                       ; 3541
                                                 0000 00071 6$:   .WORD   Save nothing                                      ; 3478
                              50          08   AC  D0 00073       MOVL    8(AP), R0                                         ;
                              50          04   A0  D0 00077       MOVL    4(R0), R0                                         ;
                                          F8   A0  9F 0007B       PUSHAB  MY_UNWIND_CCB                                     ;
                                          FC   A0  9F 0007E       PUSHAB  MY_UNWIND_ACT                                     ;
                                               02  DD 00081       PUSHL   #2                                                ;
                                               5E  DD 00083       PUSHL   SP                                                ;
                              7E          04   AC  7D 00085       MOVQ    4(AP), -(SP)                                      ;
                      FF72    CF               03  FB 00089       CALLS   #3, OPEN_HANDLER                                  ;
                                                   04 0008E       RET                                                       ;

; Routine Size:  143 bytes,    Routine Base:  _BAS$CODE + 1112
; 2625          3542  1
```

```
1 GLOBAL ROUTINE BAS$STATUS                         ! Status of last file opened
1     =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Get the status of the last file opened.  The necessary bits
1 !       were saved by OPEN in L_STATUS.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       L_STATUS        A copy of FAB$L_DEV from the last OPEN, or 0
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Bits 0 through 4 reflect device characteristics, see below.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2 !+
2 ! The following field describes the status bits returned.
2 !-
2
2     FIELD
2         STATUS_BITS =
2             SET
2             STATUS_REC = [0, 0, 1, 0],            ! Record-oriented device
2             STATUS_CCL = [0, 1, 1, 0],            ! Carriage control device
2             STATUS_TRM = [0, 2, 1, 0],            ! Device is a terminal
2             STATUS_DIR = [0, 3, 1, 0],            ! Directory device (disk)
2             STATUS_SDI = [0, 4, 1, 0],            ! Single-directory device
2             STATUS_SQD = [0, 5, 1, 0]             ! Sequential, block-oriented device (magtape)
2             TES;
2
2     LOCAL
2         STATUS : BLOCK [2, BYTE] FIELD (STATUS_BITS);
2
2 !+
2 ! Clear all of the bits in STATUS, then set the appropriate ones.
2 !-
2     STATUS = 0;
2
2     IF ((.L_STATUS AND DEV$M_REC) NEQ 0) THEN STATUS [STATUS_REC] = 1;
```

```
; 2684          3600  2
; 2685          3601  2      IF ((.L_STATUS AND DEV$M_CCL) NEQ 0) THEN STATUS [STATUS_CCL] = 1;
; 2686          3602  2
; 2687          3603  2      IF ((.L_STATUS AND DEV$M_TRM) NEQ 0) THEN STATUS [STATUS_TRM] = 1;
; 2688          3604  2
; 2689          3605  2      IF ((.L_STATUS AND DEV$M_DIR) NEQ 0) THEN STATUS [STATUS_DIR] = 1;
; 2690          3606  2
; 2691          3607  2      IF ((.L_STATUS AND DEV$M_SDI) NEQ 0) THEN STATUS [STATUS_SDI] = 1;
; 2692          3608  2
; 2693          3609  2      IF ((.L_STATUS AND DEV$M_SQD) NEQ 0) THEN STATUS [STATUS_SQD] = 1;
; 2694          3610  2
; 2695          3611  2 !+
; 2696          3612  2 ! Return the bits as our value.
; 2697          3613  2 !-
; 2698          3614  2      RETURN (.STATUS);
; 2699          3615  1      END;                                           ! of routine BAS$STATUS


                                    0000 00000          .ENTRY  BAS$STATUS, Save nothing                    ; 3543
                                  51  B4 00002          CLRW    STATUS                                      ; 3597
                 50 00000000'     EF  D0 00004          MOVL    L_STATUS, R0                                ; 3599
                 03               50  E9 0000B          BLBC    R0, 1$                                      ;
                 51               01  88 0000E          BISB2   #1, STATUS                                  ;
          03     50               01  E1 00011 1$:      BBC     #1, R0, 2$                                  ; 3601
                 51               02  88 00015          BISB2   #2, STATUS                                  ;
          03     50               02  E1 00018 2$:      BBC     #2, R0, 3$                                  ; 3603
                 51               04  88 0001C          BISB2   #4, STATUS                                  ;
          03     50               03  E1 0001F 3$:      BBC     #3, R0, 4$                                  ; 3605
                 51               08  88 00023          BISB2   #8, STATUS                                  ;
          03     50               04  E1 00026 4$:      BBC     #4, R0, 5$                                  ; 3607
                 51               10  88 0002A          BISB2   #16, STATUS                                 ;
          03     50               05  E1 0002D 5$:      BBC     #5, R0, 6$                                  ; 3609
                 51               20  88 00031          BISB2   #32, STATUS                                 ;
                 50               51  3C 00034 6$:      MOVZWL  STATUS, R0                                  ; 3614
                                      04 00037          RET                                                 ; 3615

; Routine Size:  56 bytes,    Routine Base:  _BAS$CODE + 11A1


; 2700          3616  1
; 2702           3617  1 GLOBAL ROUTINE BAS$$STATU_INIT : NOVALUE =       ! Initialize status
; 2703           3618  1
; 2704           3619  1 !++
; 2705           3620  1 ! FUNCTIONAL DESCRIPTION:
; 2706           3621  1 !
; 2707           3622  1 !       Initialize the STATUS variable.  This is needed by the RUN command in case this
; 2708           3623  1 !       is not the first RUN command in this image.
; 2709           3624  1 !
; 2710           3625  1 ! FORMAL PARAMETERS:
; 2711           3626  1 !
; 2712           3627  1 !       NONE
; 2713           3628  1 !
; 2714           3629  1 ! IMPLICIT INPUTS:
; 2715           3630  1 !
; 2716           3631  1 !       NONE
; 2717           3632  1 !
; 2718           3633  1 ! IMPLICIT OUTPUTS:
; 2719           3634  1 !
; 2720           3635  1 !       L_STATUS, always zet to zero.
; 2721           3636  1 !
; 2722           3637  1 ! ROUTINE VALUE:
; 2723           3638  1 ! COMPLETION CODES:
; 2724           3639  1 !
; 2725           3640  1 !       NONE
; 2726           3641  1 !
; 2727           3642  1 ! SIDE EFFECTS:
; 2728           3643  1 !
; 2729           3644  1 !       NONE
; 2730           3645  1 !
; 2731           3646  1 !--
; 2732           3647  1
; 2733           3648  2     BEGIN
; 2734           3649  2     L_STATUS = 0;
; 2735           3650  1     END;                                      ! of routine BAS$$STATU_INIT



                                        0000 00000        .ENTRY  BAS$$STATU_INIT, Save nothing           ; 3617
                          00000000'  EF  D4 00002         CLRL    L_STATUS                                ; 3649
                                          04 00008        RET                                             ; 3650

; Routine Size:  9 bytes,    Routine Base:  _BAS$CODE + 11D9


; 2736           3651  1
; 2737           3652  1 END                                          ! end of mcdule BAS$OPEN
; 2738           3653  1
; 2739           3654  0 ELUDOM



;                                        PSECT SUMMARY
```

```
;
;       Name                           Bytes                         Attributes
;
;  _BAS$DATA                               4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _BAS$CODE                            4578  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)




;                                 Library Statistics
;
;                                                     -------- Symbols --------      Pages       Processing
;       File                                          Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1                   9776       188         1        581         00:01.2




;                                            COMMAND QUALIFIERS

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

; Size:         4578 code + 4 data bytes
; Run Time:        01:55.7
; Elapsed Time:    04:15.6
; Lines/CPU Min:    1894
; Lexemes/CPU-Min: 18938
; Memory Used:  1272 pages
; Compilation Complete
```

0029 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASOPEN
LIS

BASPOWJJ
LIS

BASPOS
LIS

BASPOWDJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWHH
LIS

BASPOWRJ
LIS

BASPOWII
LIS

BASPURIOB
LIS

BASPOWDD
LIS

BASOPENZE
LIS

BASPOWDR
LIS

BASPOWGJ
LIS

BASPOWRD
LIS

BASPOWHJ
LIS

BASPOWRR
LIS