F 7

```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC   KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC   KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC   KKK         KKK  UUU         UUU  PPPPPPPPPPPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC              KKK      KKK     UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC              KKK      KKK     UUU         UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC              KKK      KKK     UUU         UUU  PPP         PPP
BBBBBBBBBBBB     AAA         AAA  CCC              KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC              KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC              KKKKKKKKK        UUU         UUU  PPPPPPPPPPPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC              KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC              KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC              KKK      KKK     UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP
BBB         BBB  AAA         AAA  CCC              KKK         KKK  UUU         UUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC   KKK         KKK  UUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC   KKK         KKK  UUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC   KKK         KKK  UUUUUUUUUUUUUUU  PPP
```

```
**FILE**ID**FASTSCAN
```

```
FFFFFFFFFF    AAAAAA      SSSSSSSS  TTTTTTTTTT    SSSSSSSS    CCCCCCCC    AAAAAA    NN      NN
FFFFFFFFFF    AAAAAA      SSSSSSSS  TTTTTTTTTT    SSSSSSSS    CCCCCCCC    AAAAAA    NN      NN
FF          AA      AA  SS              TT      SS          CC          AA      AA  NN      NN
FF          AA      AA  SS              TT      SS          CC          AA      AA  NN      NN
FF          AA      AA  SS              TT      SS          CC          AA      AA  NNNN    NN
FF          AA      AA  SS              TT      SS          CC          AA      AA  NNNN    NN
FFFFFFFF    AA      AA    SSSSSS        TT        SSSSSS    CC          AA      AA  NN  NN  NN
FFFFFFFF    AA      AA    SSSSSS        TT        SSSSSS    CC          AA      AA  NN  NN  NN
FF          AAAAAAAAAA          SS      TT              SS  CC          AAAAAAAAAA  NN    NNNN
FF          AAAAAAAAAA          SS      TT              SS  CC          AAAAAAAAAA  NN    NNNN
FF          AA      AA          SS      TT              SS  CC          AA      AA  NN      NN      ....
FF          AA      AA          SS      TT              SS  CC          AA      AA  NN      NN      ....
FF          AA      AA  SSSSSSSS        TT      SSSSSSSS      CCCCCCCC  AA      AA  NN      NN      ....
FF          AA      AA  SSSSSSSS        TT      SSSSSSSS      CCCCCCCC  AA      AA  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 MODULE FASTSCAN (%TITLE 'Fast file scan'
0                  IDENT = 'V04-000'
0                  ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:
1 !       Backup/Restore
1 !
1 ! ABSTRACT:
1 !       This module contains the fast file scan routines.
1 !
1 ! ENVIRONMENT:
1 !       VAX/VMS user mode.
1 !--
1 !
1 ! AUTHOR: M. Jack, CREATION DATE: 19-Nov-1980
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-015 LMP0272         L. Mark Pilant,         6-Jul-1984  8:41
1 !               Modify BACKUP to always use a full length FIB.
1 !
1 !       V03-014 LY0499          Larry Yetto             25-JUN-1984 12:36
1 !               Modify the concealed root directory code to properly deal
1 !               with [000000]
1 !
1 !       V03-013 ACG0381         Andrew C. Goldstein,    9-Dec-1983  19:43
1 !               Fix byte count of initial MFD file name string
1 !               (broken by ACG0366)
1 !
1 !       V03-012 ACG0366         Andrew C. Goldstein,    11-Oct-1983  18:55
```

```
1 !                Fix rooted directory handling to track $PARSE changes
1 !
1 !        V03-011 ACG0325          Andrew C. Goldstein,    4-Apr-1983  15:51
1 !                Fix header area length validation
1 !
1 !        V03-010 MLJ0105          Martin L. Jack, 14-Feb-1983  17:07
1 !                Correct error in execute-only "no such file" path.
1 !
1 !        V03-009 ACG0313          Andrew C. Goldstein,    12-Feb-1983  16:12
1 !                Add routine subtitles
1 !
1 !        V03-008 MLJ0104          Martin L. Jack, 24-Jan-1983  18:51
1 !                Correct access control bits used for index file under
1 !                /IGNORE=INTERLOCK.  Allow access to execute-only directories
1 !                via ACP lookup.  Change bootblock handling to allow copying RSX
1 !                system disks.  Tighten V03-006 to save scanned but non-selected
1 !                directories only if the selection file specification is "*.*;*"
1 !                and the directory terminator is "*".
1 !
1 !        V03-007 MLJ0101          Martin L. Jack, 13-Nov-1982  19:46
1 !                Correct ODS-1 latest-version selection.
1 !
1 !        V03-006 MLJ48949         Martin L. Jack, 6-Sep-1982  17:17
1 !                Save scanned but non-selected directories only if the selection
1 !                file specification is "*.*;*".  This saves space in cases where
1 !                an incremental restore is meaningless anyway.
1 !
1 !        V03-005 MLJ0096          Martin L. Jack, 24-Aug-1982  15:50
1 !                Correct /IMAGE file number conflict bug.
1 !
1 !        V03-004 MLJ48500         Martin L. Jack, 18-Aug-1982  10:23
1 !                Correct bad test that caused scanned but non-selected
1 !                directories to be copied in a disk to disk operation.  Also
1 !                add a test to avoid extra accesses to directories.
1 !
1 !        V03-003 MLJ0087          Martin L. Jack, 08-Apr-1982  16:45
1 !                Do not truncate the index file for ODS-1 version 2 because it
1 !                cannot be re-extended to multi-header dynamically.  Fix setting
1 !                of DIR_STATUS.
1 !
1 !        V03-002 MLJ0083          Martin L. Jack, 22-Mar-1982  13:25
1 !                Inhibit saving of scanned but not selected directories under
1 !                /INTERCHANGE, to save space on distribution media.
1 !
1 !        V03-001 MLJ0082          Martin L. Jack, 16-Mar-1982  18:42
1 !                Initialize DIR_STATUS before index file scan to avoid incorrect
1 !                results from SELECT_INPUT_FILE.
1 !
1 !        V02-011 MLJ0078          Martin L. Jack, 10-Feb-1982  15:30
1 !                Correct error in exclusion of aliased files introduced by
1 !                V02-007.
1 !
1 !        V02-010 MLJ0077          Martin L. Jack, 8-Feb-1982  15:11
1 !                Implement negative version numbers.
1 !
1 !        V02-009 MLJ0075          Martin L. Jack, 28-Jan-1982  20:16
1 !                Implement DIR_VERLIM and VERLIMIT attributes to support version
```

```
1 !                    limit handling.  Add FIB$V_NORECORD to file accesses to support
1 !                    file expiration handling.
1 !
1 !          V02-008 MLJ0063          Martin L. Jack, 22-Dec-1981  4:20
1 !                  Support rooted directories.
1 !
1 !          V02-007 MLJ0062          Martin L. Jack, 4-Dec-1981  14:33
1 !                  Implement /INCREMENTAL.
1 !
1 !          V02-006 MLJ0054          Martin L. Jack, 15-Oct-1981  19:26
1 !                  Support segmented directory records.  Combine routines
1 !                  ODS1_VOLUME_ATTRIBUTES and ODS2_VOLUME_ATTRIBUTES.  Remove
1 !                  COM_IMP_NOBACK, as this must be computed on every file.
1 !                  Implement /VOLUME.  Implement /IGNORE=INTERLOCK.  Integrate
1 !                  GET_VM and FREE_VM jacket routines.
1 !
1 !          V02-005 MLJ0037          Martin L. Jack, 29-Aug-1981  16:04
1 !                  Correct file selection with /IMAGE.
1 !
1 !          V02-004 MLJ0036          Martin L. Jack, 28-Aug-1981  17:55
1 !                  Reimplement file scan.
1 !
1 !          V02-003 MLJ0025          Martin L. Jack, 8-May-1981  11:25
1 !                  Implement latest-version selection.  Make /RECORD restartable.
1 !
1 !          V02-002 MLJ0018          Martin L. Jack, 7-Apr-1981  21:08
1 !                  Restore expanded string length in input NAM block.
1 !
1 !          V02-001 MLJ0010          Martin L. Jack, 25-Mar-1981  16:33
1 !                  Reorganize global storage.  Reorganize so that index file
1 !                  processing occurs on one volume at a time to minimize global
1 !                  storage requirement.  Incorporate "slow" file scan into fast
1 !                  file scan so that BACKUP, not RMS, maintains directory context.
1 !                  Add attributes check on directories so that spurious
1 !                  "illegal format" errors are not produced.
1 !
1 !**
```

```
1 REQUIRE 'SRC$:COMMON';
1 LIBRARY 'SYS$LIBRARY:LIB';
1
1
1 FORWARD ROUTINE
1         FAST_FILE_SCAN: NOVALUE,          ! Scan with index file
1         SLOW_FILE_SCAN: NOVALUE,          ! Scan without index file
1         READ_HOMEBLOCK: NOVALUE,          ! Read and check home block
1         VERIFY_HEADER,                    ! Check a file header
1         PROCESS_FILE:   NOVALUE,          ! Process one file
1         DIR_SCAN:       NOVALUE,          ! Driver for directory scan
1         INIT_DIR_SCAN:  NOVALUE,          ! Initialize directory scan
1         RESET_DIR_SPEC: NOVALUE,          ! Reset selection filespec
1         FIND_NEXT,                        ! Scan for next matching file
1         FREE_DIR_DATA:  NOVALUE;          ! Free directory scan context
1
1
1 EXTERNAL ROUTINE
1         LIB$EXTRACT_CONCEALED: ADDRESSING_MODE(GENERAL),
1                                           ! Parse concealed device/root directory
1         CHECKSUM,                         ! Compute file header checksum
1         CHECKSUM2,                        ! Compute home block checksum
1         GEN_FID_RECORD: NOVALUE,          ! Write FID record
1         INIT_NAMEBLOCK: NOVALUE,          ! Initialize extended name block fields
1         VOLUME_ATTRIBUTES:                ! Write volume attribute record
1                         NOVALUE,
1         SAVE_ONE_FILE,                    ! Routine to copy one file
1         FILE_ERROR:     NOVALUE,          ! Signal file-related error
1         INIT_ATTR:      NOVALUE,          ! Initialize attributes area
1         LEFT_ONE,                         ! Find leftmost one bit in a word
1         MAKE_STRING,                      ! Convert ODS-1 filename to ASCII
1         INIT_SEL_INFO:  NOVALUE,          ! Initialize selection information
1         MATCH_DIRECTORY,                  ! Match directory specification
1         MATCH_FILENAME,                   ! Match file name, type, and version
1         SELECT_INPUT_FILE,                ! Select input file based on qualifiers
1         TERMINATE_SCAN,                   ! Test termination of directory scan
1         FREE_VM:        NOVALUE,          ! Free virtual memory
1         GET_VM,                           ! Get virtual memory
1         GET_ZERO_VM,                      ! Get and zero virtual memory
1         LIB$SIGNAL:     ADDRESSING_MODE(GENERAL);
1                                           ! Signal a condition
1
1
1 EXTERNAL LITERAL
1         BACKUP$_MAXVOLS,
1         BACKUP$_PROCINDEX,
1         BACKUP$_OPENDIR,
1         BACKUP$_OPENIN,
1         BACKUP$_READDIR,
1         BACKUP$_BADDIR,
1         BACKUP$_NOSUCHRVN,
1         LIB$_INVFILSPE;
1
1
1 G$DEFINE();                       ! Define global common area
1
1
```

```
1 LITERAL
1              MAX_VOLUMES=     255,                ! Largest volume set
1              DIR_BUF_COUNT=   16,                 ! Size of directory buffer
1              INDEX_BUF_COUNT=64;                  ! Size of index file buffer
1
1
1 BIND
1              MFD=             UPLIT BYTE (%ASCIC '000000');
1
1
1 BUILTIN
1              TESTBITCC,
1              TESTBITSC,
1              CALLG,
1              ROT;
```

```
1 %SBTTL 'FAST_FILE_SCAN - fast file scan main routine'
1 GLOBAL ROUTINE FAST_FILE_SCAN: NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is the driver for the fast file scan.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       INPUT_FAB       - Pointer to current input FAB.
1 !       INPUT_NAM       - Pointer to current input NAM block.
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       RVN,                                    ! Relative volume number
2       FIB:            BBLOCK[FIB$C_LENGTH],   ! FIB
2       FIB_DESC:       VECTOR[2],              ! Descriptor for FIB
2       STATUS,                                 ! Status variable
2       IOSB:           VECTOR[4,WORD];         ! I/O status block
2
2
2 ! Initialize FIB descriptor.
2 !
2 FIB_DESC[0] = FIB$C_LENGTH;
2 FIB_DESC[1] = FIB;
2
2
2 ! Allocate the general buffer.
2 !
2 COM_FLAGS[COM_DSBL_CHKPT] = TRUE;
2 FAST_BUFFER_SIZE = 512 * INDEX_BUF_COUNT;
2 FAST_BUFFER = GET_VM(512 * INDEX_BUF_COUNT);
2
2
2 ! Loop over all volumes in the volume set.
2 !
2 RVN = 1;
2 DO
3     BEGIN
3     LOCAL
```

```
  3            ATR_DESC:        BBLOCK[12],       ! Attribute list
  3            STATBLK:         BBLOCK[8],        ! Statistics block
  3            ACCTL,                             ! FIB$M_WRITE or 0
  3            EOF,                               ! Index file EOF position
  3            CLUSTER,                           ! Cluster factor
  3            BITMAP_OFFSET,                     ! Index file bitmap VBN offset
  3            VBN;                               ! Current index file VBN
  3
  3
  3        ! Access the index file on RVN 1.
  3        !
  3        CH$FILL (0, FIB$C_LENGTH, FIB);
  3        ACCTL = FIB$M_WRITE OR FIB$M_NORECORD;
  3        FIB[FIB$L_ACCTL] = .ACCTL;
  3        FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
  3        FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
  3        FIB[FIB$W_FID_RVN] = .RVN;
  3        ATR_DESC[0,0,16,0] = 8;
  3        ATR_DESC[2,0,16,0] = ATR$C_STATBLK;
  3        ATR_DESC[4,0,32,0] = STATBLK;
  3        ATR_DESC[8,0,32,0] = 0;
P 3        STATUS = $QIOW(
P 3            FUNC=IO$_ACCESS OR IO$M_ACCESS,
P 3            CHAN=.INPUT_CHAN,
P 3            IOSB=IOSB,
P 3            P1=FIB_DESC,
  3            P5=ATR_DESC);
  3        IF .STATUS THEN STATUS = .IOSB[0];
  3        IF .STATUS EQL SS$_WRITLCK
  3        THEN
  4            BEGIN
  4            CH$FILL (0, FIB$C_LENGTH, FIB);
  4            ACCTL = FIB$M_NORECORD;
  4            FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
  4            FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
  4            FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
  4            FIB[FIB$W_FID_RVN] = .RVN;
P 4            STATUS = $QIOW(
P 4                FUNC=IO$_ACCESS OR IO$M_ACCESS,
P 4                CHAN=.INPUT_CHAN,
P 4                IOSB=IOSB,
P 4                P1=FIB_DESC,
  4                P5=ATR_DESC);
  4            IF .STATUS THEN STATUS = .IOSB[0];
  3            END;
  3        IF NOT .STATUS
  3        THEN
  3            SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
  3
  3
  3        ! Read the home block.  If RVN 1, establish the size of the volume set
  3        ! and the structure level.
  3        !
  3        READ_HOMEBLOCK(.RVN, .FAST_BUFFER);
  3
  3
  3        ! Allocate the per-volume data.  Each variable between FAST_VOL_BEG and
```

```
! FAST_VOL_END is a pointer to a vector that varies with the number of
! volumes in the volume set.
!
IF .RVN EQL 1
THEN
    BEGIN
    INCRA A FROM FAST_VOL_BEG TO FAST_VOL_END-%UPVAL BY %UPVAL DO
        BEGIN
        .A = GET_ZERO_VM(.COM_I_SETCOUNT*%UPVAL);
        END;
    END;


! Initialize information from the home block.
!
IF .FAST_STRUCLEV EQL 2
THEN
    BEGIN
    CLUSTER = .FAST_BUFFER[HM2$W_CLUSTER];
    FAST_IMAP_SIZE[.RVN-1] = .FAST_BUFFER[HM2$W_IBMAPSIZE];
    BITMAP_OFFSET = .CLUSTER*4 + 1;
    FAST_HDR_OFFSET[.RVN-1] = .CLUSTER*4 + .FAST_BUFFER[HM2$W_IBMAPSIZE];
    END
ELSE
    BEGIN
    CLUSTER = 1;
    FAST_IMAP_SIZE[.RVN-1] = .FAST_BUFFER[HM1$W_IBMAPSIZE];
    BITMAP_OFFSET = 3;
    FAST_HDR_OFFSET[.RVN-1] = 2 + .FAST_BUFFER[HM1$W_IBMAPSIZE];
    END;


! Allocate memory for and read in index file bitmap.
!
FAST_IMAP[.RVN-1] = GET_VM(.FAST_IMAP_SIZE[.RVN-1] * 512);
STATUS = $QIOW(
    FUNC=IO$_READVBLK,
    CHAN=.INPUT_CHAN,
    IOSB=IOSB,
    P1=.FAST_IMAP[.RVN-1],
    P2=.FAST_IMAP_SIZE[.RVN-1] * 512,
    P3=.BITMAP_OFFSET);
IF .STATUS THEN STATUS = .IOSB[0];
IF NOT .STATUS
THEN
    SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);


! Read the index file header and get the EOF stored therein.
!
STATUS = $QIOW(
    FUNC=IO$_READVBLK,
    CHAN=.INPUT_CHAN,
    IOSB=IOSB,
    P1=.FAST_BUFFER + 512,
    P2=512,
    P3=.FAST_HDR_OFFSET[.RVN-1] + 1);
```

```
  3      IF .STATUS THEN STATUS = .IOSB[0];
  3      IF NOT .STATUS
  3      THEN
  3          SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
  3
  3
  3      EOF = 0;
  3      IF .FAST_STRUCLEV EQL 2
  3          THEN EOF = ROT(.BBLOCK[BBLOCK[.FAST_BUFFER + 512, FH2$W_RECATTR], FAT$L_EFBLK], 16) - 1
  3          ELSE IF .FAST_BUFFER[HM2$B_STRUCVER] EQL 2
  3              THEN EOF = ROT(.STATBLK[SBK$L_FILESIZE], 16);
  3
  3
  3      ! Now scan the volume's index file bitmap backwards, looking for the
  3      ! highest set bit.  The maximum of the EOF mark and the highest set
  3      ! bit is taken to be the true index file EOF.
  3      !
  3      DECR J FROM .FAST_IMAP_SIZE[.RVN-1] * 128 - 1 TO 0 DO
  4          BEGIN
  4          IF .VECTOR[.FAST_IMAP[.RVN-1], .J] NEQ 0
  4          THEN
  5              BEGIN
  5              EOF = MAXU(
  5                      .J*32 +
  5                      LEFT_ONE(.VECTOR[.FAST_IMAP[.RVN-1], .J]) +
  5                      .FAST_HDR_OFFSET[.RVN-1],
  5                  .EOF);
  5              EXITLOOP;
  4              END;
  3          END;
  3
  3
  3      IF .QUAL[QUAL_IMAG]
  3      THEN
  4          BEGIN
  4
  4          ! Read the boot block.
  4          !
P 4          STATUS = $QIOW(
P 4              FUNC=IO$_READVBLK,
P 4              CHAN=.INPUT_CHAN,
P 4              IOSB=IOSB,
P 4              P1=.FAST_BUFFER + 512,
P 4              P2=512,
  4              P3=1);
  4          IF .STATUS THEN STATUS = .IOSB[0];
  4          IF NOT .STATUS
  4          THEN
  4              SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
  4
  4
  4          ! If the volume is bootable, save the LBN from the boot block.
  4          !
  4          FAST_BOOT_LBN[.RVN-1] = -1;
  4          IF CH$RCHAR(.FAST_BUFFER + 512 + 5) EQL 0
  4          THEN
  4              FAST_BOOT_LBN[.RVN-1] = ROT(.VECTOR[.FAST_BUFFER + 512, 1], 16);
```

```
     3              END;
     3
     3
     3          ! Deaccess the index file.
     3          !
P    3          $QIOW(
P    3              FUNC=IO$_DEACCESS,
     3              CHAN=.INPUT_CHAN);
     3
     3
     3          IF .QUAL[QUAL_IMAG]
     3          THEN
     4              BEGIN
     4
     4              ! Access the bitmap file.
     4              !
     4              CH$FILL (0, FIB$C_LENGTH, FIB);
     4              FIB[FIB$L_ACCTL] = .ACCTL;
     4              FIB[FIB$W_FID_NUM] = FID$C_BITMAP;
     4              FIB[FIB$W_FID_SEQ] = FID$C_BITMAP;
     4              FIB[FIB$W_FID_RVN] = .RVN;
P    4              STATUS = $QIOW(
P    4                  FUNC=IO$_ACCESS OR IO$M_ACCESS,
P    4                  CHAN=.INPUT_CHAN,
P    4                  IOSB=IOSB,
     4                  P1=FIB_DESC);
     4              IF .STATUS THEN STATUS = .IOSB[0];
     4              IF NOT .STATUS
     4              THEN
     4                  SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
     4
     4
     4              ! Read the storage control block.
     4              !
P    4              STATUS = $QIOW(
P    4                  FUNC=IO$_READVBLK,
P    4                  CHAN=.INPUT_CHAN,
P    4                  IOSB=IOSB,
P    4                  P1=.FAST_BUFFER + 1024,
P    4                  P2=512,
     4                  P3=1);
     4              IF .STATUS THEN STATUS = .IOSB[0];
     4              IF NOT .STATUS
     4              THEN
     4                  SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
     4
     4
     4              ! Deaccess the bitmap file.
     4              !
P    4              $QIOW(
P    4                  FUNC=IO$_DEACCESS,
     4                  CHAN=.INPUT_CHAN);
     4
     4
     4              ! Generate the volume attribute record.
     4              !
     4              IF NOT .QUAL[QUAL_VOLU] OR .QUAL[QUAL_VOLU_VALUE] EQL .RVN
```

```
    4           THEN
    4               VOLUME_ATTRIBUTES(
    4                   .FAST_BUFFER,               ! home block
    4                   .FAST_BUFFER+512,           ! boot block
    4                   .FAST_BUFFER+1024,          ! storage control block
    4                   .EOF = .FAST_HDR_OFFSET[.RVN-1]);
    3           END;
    3
    3
    3       ! Access the index file.
    3       !
    3       CH$FILL (0, FIB$C_LENGTH, FIB);
    3       FIB[FIB$L_ACCTL] = .ACCTL;
    3       FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
    3       FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
    3       FIB[FIB$W_FID_RVN] = .RVN;
P   3       STATUS = $QIOW(
P   3           FUNC=IO$_ACCESS OR IO$M_ACCESS,
P   3           CHAN=.INPUT_CHAN,
P   3           IOSB=IOSB,
    3           P1=FIB_DESC);
    3       IF .STATUS THEN STATUS = .IOSB[0];
    3       IF NOT .STATUS
    3       THEN
    3           SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
    3
    3
    3       ! Loop for all headers in the index file.  Read multiple blocks into a
    3       ! buffer and process them one at a time.
    3       !
    3       DIR_STATUS = 0;                         ! Used by SELECT_INPUT_FILE
    3       VBN = .FAST_HDR_OFFSET[.RVN-1] + 1;
    3       UNTIL .VBN GTRU .EOF DO
    4           BEGIN
    4           LOCAL
    4               HEADER:     REF BBLOCK,         ! Pointer to header
    4               READ_COUNT;                     ! Blocks to read on current iteration
    4
    4
    4           ! Establish the count of blocks to read and execute the read.
    4           !
    4           READ_COUNT = MINU(INDEX_BUF_COUNT, .EOF + 1 - .VBN);
P   4           STATUS = $QIOW(
P   4               FUNC=IO$_READVBLK,
P   4               CHAN=.INPUT_CHAN,
P   4               IOSB=IOSB,
P   4               P1=.FAST_BUFFER,
P   4               P2=512 * .READ_COUNT,
    4               P3=.VBN);
    4           IF .STATUS THEN STATUS = .IOSB[0];
    4
    4
    4           ! If an error occurred, read each block separately, reporting any
    4           ! errors.
    4           !
    4           IF NOT .STATUS
    4           THEN
```

```
          BEGIN
          INCR XVBN FROM 0 TO .READ_COUNT-1 DO
              BEGIN
              LOCAL
                  HEADER:     REF BBLOCK;              ! Pointer to header


              HEADER = .FAST_BUFFER + .XVBN * 512;
              STATUS = $QIOW(
                  FUNC=IO$_READVBLK,
                  CHAN=.INPUT_CHAN,
                  IOSB=IOSB,
                  P1=.HEADER,
                  P2=512,
                  P3=.VBN + .XVBN);
              IF .STATUS THEN STATUS = .IOSB[0];
              IF NOT .STATUS
              THEN
                  CH$FILL(0, 512, .HEADER);
              END;
          END;


      ! For each header, verify that it is a valid file header.
      ! If it is, process it.
      !
      HEADER = .FAST_BUFFER;
      INCR XVBN FROM 0 TO .READ_COUNT-1 DO
          BEGIN
          LOCAL
              MAP_AREA:       REF BBLOCK,              ! Pointer to map area
              FILE_NUMBER,                             ! Current file number
              FILE_ID:        BBLOCK[FID$C_LENGTH];    ! Current file ID


          ! Get a clean file ID.
          !
          FILE_NUMBER = .VBN + .XVBN - .FAST_HDR_OFFSET[.RVN-1];
          FILE_ID[FID$W_NUM] = .FILE_NUMBER<0,16>;
          FILE_ID[FID$B_NMX] = .FILE_NUMBER<16,8>;
          IF .FAST_STRUCLEV EQL 2
              THEN FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_FID_SEQ]
              ELSE FILE_ID[FID$W_SEQ] = .HEADER[FH1$W_FID_SEQ];
          FILE_ID[FID$B_RVN] = .RVN;


          ! Generate the FID record for an image mode save.
          !
          IF
              .QUAL[QUAL_IMAG] AND
              .XVBN EQL 0 AND
              (NOT .QUAL[QUAL_VOLU] OR .QUAL[QUAL_VOLU_VALUE] EQL .RVN)
          THEN
              GEN_FID_RECORD(.HEADER, .READ_COUNT, .FILE_NUMBER, .RVN);


          ! Validate the header.  In ODS-2, a valid header is to be taken as
```

```
5        ! valid even if the index file bitmap shows it as available.  In
5        ! ODS-1, a header corresponding to a clear index file bitmap bit
5        ! is not to be examined.
5        !
5        STATUS = VERIFY_HEADER(.HEADER, FILE_ID);
5        IF
5            .FAST_STRUCLEV EQL 1 AND
5            NOT .BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1]
5        THEN
5            STATUS = FALSE;
5
5
5        ! Clear the index file bitmap bit until we determine that the
5        ! header is not an extension header.
5        !
5        BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
5
5
5        IF .STATUS
5        THEN
6            BEGIN
6
6            ! Header is valid.
6            ! Other processing executed only if not an extension header.
6            !
6            MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
6            IF
7                BEGIN
7                IF .FAST_STRUCLEV EQL 2
7                    THEN .HEADER[FH2$W_SEG_NUM] EQL 0
7                    ELSE .MAP_AREA[FM1$B_EX_SEGNUM] EQL 0
7                END
6            THEN
7                BEGIN
7                LOCAL
7                    FCH:    REF BBLOCK;     ! Pointer to characteristics
7
7
7                ! Make sure the index file bitmap indicates that the header
7                ! is valid.
7                !
7                BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1] = TRUE;
7
7
7                ! Get file characteristics pointer.
7                !
7                IF .FAST_STRUCLEV EQL 2
7                    THEN FCH = HEADER[FH2$L_FILECHAR]
7                    ELSE FCH = HEADER[FH1$W_FILECHAR];
7
7
7                ! Evaluate file selection criteria for header.  However,
7                ! always reject files marked for delete.  If this is /IMAGE
7                ! mode, select all other valid files.
7                !
7                IF
8                    BEGIN
```

```
  8                         IF .FCH[FCH$V_MARKDEL]
  8                         THEN
  8                             TRUE
  8                         ELSE
  9                             BEGIN
  9                             INIT_ATTR(.HEADER);
  9                             IF .QUAL[QUAL_IMAG]
  9                                 THEN FALSE
  9                                 ELSE NOT SELECT_INPUT_FILE(%B'001')
  9                             END
  8                         END
  7                     THEN
  7                         BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
  6                     END;
  5                 END;
  5
  5
  5             HEADER = .HEADER + 512;
  4             END;
  4
  4
  4         VBN = .VBN + INDEX_BUF_COUNT;
  3         END;
  3
  3
  3     ! Deaccess the index file.
  3     !
P 3     $QIOW(
P 3         FUNC=IO$_DEACCESS,
  3         CHAN=.INPUT_CHAN);
  3
  3
  3     RVN = .RVN + 1;
  3     END
  2 WHILE .RVN LEQU .COM_I_SETCOUNT;
  2
  2
  2 ! Free the general buffer.
  2 !
  2 FREE_VM(.FAST_BUFFER_SIZE, .FAST_BUFFER);
  2 FAST_BUFFER = FAST_BUFFER_SIZE = 0;
  2 COM_FLAGS[COM_DSBL_CHKPT] = FALSE;
  2
  2
  2 ! Scan all directories on all volumes.
  2 !
  2 INCR RVN FROM 1 TO .COM_I_SETCOUNT DO DIR_SCAN(.RVN);
  2
  2
  2 ! Allocate a buffer to read file headers.
  2 !
  2 FAST_BUFFER_SIZE = 512;
  2 FAST_BUFFER = GET_VM(512);
  2
  2
  2 ! If this is /IMAGE mode, scan for and process lost files.
  2 !
```

```
     2 IF .QUAL[QUAL_IMAG]
     2 THEN
     3     BEGIN
     3     INCR RVN FROM 1 TO .COM_I_SETCOUNT DO
     4         BEGIN
     4
     4         ! Scan bitmap for files not yet processed and process these.
     4         !
     4         INCR FILE_NUMBER FROM 1 TO .FAST_IMAP_SIZE[.RVN-1]*4096 DO
     5             BEGIN
     5             IF .BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1]
     5             THEN
     6                 BEGIN
     6                 LOCAL
     6                     NAME_LENGTH,                             ! Length of filename
     6                     NAME_ADDRESS,                            ! Address of filename
     6                     RSA_DESC:           VECTOR[2],           ! Descriptor for RSA
     6                     FILENAME:           VECTOR[20,BYTE];! Filename buffer
     6
     6
     6                 ! Access index file on current RVN.
     6                 !
     6                 CH$FILL (0, FIB$C_LENGTH, FIB);
     6                 FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
     6                 FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
     6                 FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
     6                 FIB[FIB$W_FID_RVN] = .RVN;
P    6                 STATUS = $QIOW(
P    6                     FUNC=IO$_ACCESS OR IO$M_ACCESS,
P    6                     CHAN=.INPUT_CHAN,
P    6                     IOSB=IOSB,
     6                     P1=FIB_DESC);
     6                 IF .STATUS THEN STATUS = .IOSB[0];
     6                 IF NOT .STATUS
     6                 THEN
     6                     SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
     6
     6
     6                 ! Read file header.
     6                 !
P    6                 STATUS = $QIOW(
P    6                     FUNC=IO$_READVBLK,
P    6                     CHAN=.INPUT_CHAN,
P    6                     IOSB=IOSB,
P    6                     P1=.FAST_BUFFER,
P    6                     P2=512,
     6                     P3=.FAST_HDR_OFFSET[.RVN-1] + .FILE_NUMBER);
     6                 IF .STATUS THEN STATUS = .IOSB[0];
     6                 IF NOT .STATUS
     6                 THEN
     6                     SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .STATUS);
     6
     6
     6                 ! Deaccess index file on current RVN.
     6                 !
P    6                 $QIOW(
P    6                     FUNC=IO$_DEACCESS,
                    CHAN=.INPUT_CHAN);


        ! Set up file ID.
        !
        INPUT_NAM[NAM$W_FID_NUM] = .FILE_NUMBER;
        INPUT_NAM[NAM$B_FID_NMX] = .FILE_NUMBER<16,8>;
        INPUT_NAM[NAM$B_FID_RVN] = .RVN;
        INPUT_NAM[NAM$W_DID_NUM] = 0;
        INPUT_NAM[NAM$W_DID_SEQ] = 0;
        INPUT_NAM[NAM$W_DID_RVN] = 0;


        ! Generate filename string from file header ident area.
        ! Also get file sequence number.
        !
        IF .FAST_STRUCLEV EQL 2
        THEN
            BEGIN
            LOCAL
                P;

            INPUT_NAM[NAM$W_FID_SEQ] = .FAST_BUFFER[FH2$W_FID_SEQ];
            NAME_ADDRESS =
                BBLOCK[.FAST_BUFFER + .FAST_BUFFER[FH2$B_IDOFFSET]*2,
                    FI2$T_FILENAME];
            NAME_LENGTH = FI2$S_FILENAME;
            P = CH$FIND_CH(FI2$S_FILENAME, .NAME_ADDRESS, %C' ');
            IF .P NEQ 0 THEN NAME_LENGTH = .P - .NAME_ADDRESS;
            END
        ELSE
            BEGIN
            INPUT_NAM[NAM$W_FID_SEQ] = .FAST_BUFFER[FH1$W_FID_SEQ];
            NAME_ADDRESS = FILENAME;
            NAME_LENGTH = MAKE_STRING(
            BBLOCK[.FAST_BUFFER + .FAST_BUFFER[FH1$B_IDOFFSET]*2,
                FI1$W_FILENAME] - $BYTEOFFSET(NMB$W_NAME),
                FILENAME) AND 65535;
            END;
        RSA_DESC[0] = NAM$C_MAXRSS;
        RSA_DESC[1] = .INPUT_NAM[NAM$L_RSA];
        $FAO(
            $DESCRIPTOR('!AS[]!AD'),
            RSA_DESC,
            RSA_DESC,
            INPUT_QUAL[QUAL_DEV_DESC],
            .NAME_LENGTH, .NAME_ADDRESS);
        INPUT_NAM[NAM$B_RSL] = .RSA_DESC[0];
        INIT_NAMEBLOCK(.INPUT_NAM);


        ! Call the routine to process the file.
        !
        SAVE_ONE_FILE();
        END;
    END;
END;
```

```
2       END;
2
2
2 ! Free the file header buffer.
2 !
2 FREE_VM(.FAST_BUFFER_SIZE, .FAST_BUFFER);
2 FAST_BUFFER = FAST_BUFFER_SIZE = 0;
2
2
2 ! Free memory for index file bitmaps.
2 !
2 INCR RVN FROM 1 TO .COM_I_SETCOUNT DO
3     BEGIN
3     FREE_VM(.FAST_IMAP_SIZE[.RVN-1] * 512, .FAST_IMAP[.RVN-1]);
3     FAST_IMAP[.RVN-1] = 0;
2     END;
2
2
2 ! Free the per-volume data.
2 !
2 INCRA A FROM FAST_VOL_BEG TO FAST_VOL_END-%UPVAL BY %UPVAL DO
3     BEGIN
3     FREE_VM(.COM_I_SETCOUNT*%UPVAL, ..A);
3     .A = 0;
2     END;
1 END;
```

```
                                        .TITLE  FASTSCAN Fast file scan
                                        .IDENT  \V04-000\

                                        .PSECT  COMMON,NOEXE,  OVR,2

                    00000 GLOBAL_BASE:
                                        .BLKB   0
                    00000 FREE_LIST:
                                        .BLKB   8
                    00008 INPUT_WAIT:
                                        .BLKB   8
                    00010 REREAD_WAIT:
                                        .BLKB   8
                    00018 OUTPUT_WAIT:
                                        .BLKB   8
                    00020 JPI_UIC:      .BLKB   4
                    00024 JPI_USERNAME:
                                        .BLKB   12
                    00030 JPI_DATE:
                                        .BLKB   8
                    00038 JPI_NODE_DESC:
                                        .BLKB   8
                    00040 JPI_CURPRIV:
                                        .BLKB   8
                    00048 SYI_VERSION:
                                        .BLKB   4
                    0004C SYI_SID:      .BLKB   4
                    00050 RWSV_HOLD_LIST:
                                        .BLKB   8
```

```
00058 RWSV_CRC16:
                .BLKB   64
00098 RWSV_AUTODIN:
                .BLKB   64
000D8 RWSV_FILESET_ID:
                .BLKB   8
000E0 RWSV_VOLUME_ID:
                .BLKB   12
000EC RWSV_VOL_NUMBER:
                .BLKB   2
000EE RWSV_SEG_NUMBER:
                .BLKB   2
000F0 RWSV_FILE_NUMBER:
                .BLKB   4
000F4 RWSV_SAVE_QUAL:
                .BLKB   4
000F8 RWSV_SAVE_FAB:
                .BLKB   4
000FC RWSV_CHAN:
                .BLKB   4
00100 RWSV_XOR_BCB:
                .BLKB   4
00104 RWSV_IN_SEQ:
                .BLKB   4
00108 RWSV_IN_SEQ_0:
                .BLKB   4
0010C RWSV_IN_XOR_SEQ:
                .BLKB   4
00110 RWSV_IN_XOR_RFA:
                .BLKB   6
00116 RWSV_LOOKAHEAD:
                .BLKB   1
00117 RWSV_XORSIZE:
                .BLKB   1
00118 RWSV_IN_GROUP_SIZE:
                .BLKB   4
0011C RWSV_IN_ERRORS:
                .BLKB   2
0011E RWSV_IN_XORUSE:
                .BLKB   2
00120 RWSV_IN_ORGERR:
                .BLKB   8
00128 RWSV_IN_VBN:
                .BLKB   4
0012C RWSV_IN_VBN_0:
                .BLKB   4
00130 RWSV_ALLOC:
                .BLKB   4
00134 RWSV_EOF:
                .BLKB   4
00138 RWSV_OUT_SEQ:
                .BLKB   4
0013C RWSV_OUT_VBN:
                .BLKB   4
00140 RWSV_OUT_BLOCK_COUNT:
                .BLKB   4
00144 RWSV_OUT_ERRORS:
```

```
                .BLKB   2
00146 RWSV_SEQ_ERRORS:
                .BLKB   2
00148 RWSV_OUT_GROUP_COUNT:
                .BLKB   1
00149 RWSV_PADDING:
                .BLKB   3
0014C QUAL:     .BLKB   112
001BC COM_SSNAME:
                .BLKB   8
001C4 COM_VALID_TYPES:
                .BLKB   2
001C6 COM_FLAGS:
                .BLKB   2
001C8 COM_PADDING:
                .BLKB   1
001C9 COM_BUFF_COUNT:
                .BLKB   1
001CA COM_I_SETCOUNT:
                .BLKB   1
001CB COM_O_SETCOUNT:
                .BLKB   1
001CC COM_I_STRUCNAME:
                .BLKB   12
001D8 COM_O_STRUCNAME:
                .BLKB   12
001E4 COM_O_BSRDATE:
                .BLKB   8
001EC ALT_SSNAME:
                .BLKB   32
0020C INPUT_FUNC:
                .BLKB   1
0020D INPUT_RTYPE:
                .BLKB   1
0020E OUTPUT_FUNC:
                .BLKB   1
0020F FAST_STRUCLEV:
                .BLKB   1
00210 INPUT_BEG:
                .BLKB   0
00210 INPUT_CHAN:
                .BLKB   4
00214 INPUT_FLAGS:
                .BLKB   2
00216 INPUT_PADDING:
                .BLKB   2
00218 INPUT_FAB:
                .BLKB   4
0021C INPUT_NAM:
                .BLKB   4
00220 INPUT_BCB:
                .BLKB   4
00224 INPUT_QUAL:
                .BLKB   4
00228 INPUT_BAD:
                .BLKB   4
0022C INPUT_BLOCK:
```

```
                    .BLKB   4
00230 INPUT_MAXBLOCK:
                    .BLKB   4
00234 INPUT_MEDIA_ID:
                    .BLKB   4
00238 INPUT_NAMEDESC:
                    .BLKB   8
00240 INPUT_STATBLK:
                    .BLKB   8
00248 INPUT_HDR_BEG:
                    .BLKB   0
00248 INPUT_CREDATE:
                    .BLKB   8
00250 INPUT_REVDATE:
                    .BLKB   8
00258 INPUT_EXPDATE:
                    .BLKB   8
00260 INPUT_BAKDATE:
                    .BLKB   8
00268 INPUT_FILEOWNER:
                    .BLKB   4
0026C INPUT_FILECHAR:
                    .BLKB   4
00270 INPUT_RECATTR:
                    .BLKB   32
00290 INPUT_HDR_END:
                    .BLKB   0
00290 INPUT_END:
                    .BLKB   0
00290 INPUT_PROC_LIST:
                    .BLKB   4
00294 INPUT_PLACEMENT:
                    .BLKB   8
0029C INPUT_VBN_LIST:
                    .BLKB   8
002A4 INPUT_PLACE_LEN:
                    .BLKB   2
002A6 INPUT_PADDING_2:
                    .BLKB   2
002A8 OUTPUT_BEG:
                    .BLKB   0
002A8 OUTPUT_CHAN:
                    .BLKB   4
002AC OUTPUT_FLAGS:
                    .BLKB   2
002AE OUTPUT_PADDING:
                    .BLKB   2
002B0 OUTPUT_FAB:
                    .BLKB   4
002B4 OUTPUT_NAM:
                    .BLKB   4
002B8 OUTPUT_BCB:
                    .BLKB   4
002BC OUTPUT_QUAL:
                    .BLKB   4
002C0 OUTPUT_BAD:
                    .BLKB   4
```

```
002C4 OUTPUT_BLOCK:
              .BLKB   4
002C8 OUTPUT_MAXBLOCK:
              .BLKB   4
002CC OUTPUT_DEVGEOM:
              .BLKB   8
002D4 OUTPUT_ATTBUF:
              .BLKB   144
00364 OUTPUT_END:
              .BLKB   0
00364 LIST_TOTFILES:
              .BLKB   4
00368 LIST_TOTSIZE:
              .BLKB   4
0036C VERIFY_FAB:
              .BLKB   4
00370 VERIFY_USE_COUNT:
              .BLKB   4
00374 VERIFY_QUAL:
              .BLKB   4
00378 COMPARE_BCB:
              .BLKB   4
0037C FAST_BUFFER:
              .BLKB   4
00380 FAST_BUFFER_SIZE:
              .BLKB   4
00384 FAST_RVN:
              .BLKB   1
00385 FAST_PADDING:
              .BLKB   1
00386 DIR_VERLIMIT:
              .BLKB   2
00388 FAST_VOL_BEG:
              .BLKB   0
00388 FAST_IMAP_SIZE:
              .BLKB   4
0038C FAST_IMAP:
              .BLKB   4
00390 FAST_HDR_OFFSET:
              .BLKB   4
00394 FAST_BOOT_LBN:
              .BLKB   4
00398 FAST_VOL_END:
              .BLKB   0
00398 JOUR_BUFFER:
              .BLKB   4
0039C JOUR_DIR:
              .BLKB   4
003A0 JOUR_HIBLK:
              .BLKB   4
003A4 JOUR_EFBLK:
              .BLKB   4
003A8 JOUR_INBLK:
              .BLKB   4
003AC JOUR_FFBYTE:
              .BLKB   2
003AE JOUR_INBYTE:
```

```
                      .BLKB   2
00380 JOUR_STRUCT_LEV:
                      .BLKB   2
003B2 JOUR_COUNT:
                      .BLKB   1
003B3 JOUR_REVERSE:
                      .BLKB   1
003B4 JOUR_EXSZ:
                      .BLKB   2
003B6 JOUR_PADDING:
                      .BLKB   2
003B8 CHKPT_HIGH_SP:
                      .BLKB   4
003BC CHKPT_LOW_SP:
                      .BLKB   4
003C0 CHKPT_STACK:
                      .BLKB   4
003C4 CHKPT_VARS:
                      .BLKB   4
003C8 CHKPT_STATUS:
                      .BLKB   4
003CC DIR_BEG:.BLKB   0
003CC DIR_CHAN:
                      .BLKB   4
003D0 DIR_NAM:.BLKB   4
003D4 DIR_DEV_DESC:
                      .BLKB   4
003D8 DIR_SEL_DIR:
                      .BLKB   8
003E0 DIR_SEL_NTV:
                      .BLKB   8
003E8 DIR_STRUCLEV:
                      .BLKB   1
003E9 DIR_LEVELS:
                      .BLKB   1
003EA DIR_FLAGS:
                      .BLKB   1
003EB DIR_STATUS:
                      .BLKB   1
003EC DIR_STRING:
                      .BLKB   320
0052C DIR_STACK:
                      .BLKB   612
00790 DIR_SP: .BLKB   4
00794 DIR_SEL_LATEST:
                      .BLKB   4
00798 DIR_END:.BLKB   0
00798 DIR_SCANLIMIT:
                      .BLKB   36
007BC INPUT_MTL:
                      .BLKB   4
007C0 OUTPUT_MTL:
                      .BLKB   4
007C4 CURRENT_MTL:
                      .BLKB   4
007C8 CURRENT_VCB:
                      .BLKB   4
```

```
                                           007CC CURRENT_WCB:
                                                         .BLKB    4
                                           007D0 ACL_FIB_DESCR:
                                                         .BLKB    8
                                           007D8 ACL_FIB:.BLKB    64
                                           00818 ACL_LENGTH:
                                                         .BLKB    4
                                           0081C ACL_BUFFER:
                                                         .BLKB    4
                                           00820 CRYP_IN_CONTEXT:
                                                         .BLKB    4
                                           00824 CRYP_OU_CONTEXT:
                                                         .BLKB    4
                                           00828 CRYP_DA_CONTEXT:
                                                         .BLKB    4
                                           0082C CRYP_DATA_ENCIV:
                                                         .BLKB    8
                                           00834 CRYP_DATA_CODE:
                                                         .BLKB    4
                                           00838 CRYP_DATA_KEY:
                                                         .BLKB    8
                                           00840 CRYP_DATA_IV:
                                                         .BLKB    8
                                           00848 CRYP_DATA_CKSM:
                                                         .BLKB    4

                                                         .PSECT   CODE,NOWRT,2

      30  30  30  30  30  30  06  00000 P.AAA:   .ASCII   <6>\000000\                        ;
  44  41  21  5D  5B  53  41  21  00007 P.AAC:   .ASCII   \!AS[]!AD\                         ;
                                  0000F          .BLKB    1
                       00000008  00010 P.AAB:   .LONG    8                                   ;
                       00000000' 00014          .ADDRESS P.AAC                               ;

                                       MFD=               P.AAA
                                                 .EXTRN   LIB$EXTRACT_CONCEALED
                                                 .EXTRN   CHECKSUM, CHECKSUM2
                                                 .EXTRN   GEN_FID_RECORD, INIT_NAMEBLOCK
                                                 .EXTRN   VOLUME_ATTRIBUTES
                                                 .EXTRN   SAVE_ONE_FILE, FILE_ERROR
                                                 .EXTRN   INIT_ATTR, LEFT_ONE
                                                 .EXTRN   MAKE_STRING, INIT_SEL_INFO
                                                 .EXTRN   MATCH_DIRECTORY
                                                 .EXTRN   MATCH_FILENAME, SELECT_INPUT_FILE
                                                 .EXTRN   TERMINATE_SCAN, FREE_VM
                                                 .EXTRN   GET_VM, GET_ZERO_VM
                                                 .EXTRN   LIB$SIGNAL, BACKUP$_MAXVOLS
                                                 .EXTRN   BACKUP$_PROCINDEX
                                                 .EXTRN   BACKUP$_OPENDIR
                                                 .EXTRN   BACKUP$_OPENIN, BACKUP$_READDIR
                                                 .EXTRN   BACKUP$_BADDIR, BACKUP$_NOSUCHRVN
                                                 .EXTRN   LIB$_INVFILSPE, SYS$QIOW
                                                 .EXTRN   SYS$FAO

                            0FFC 00000          .ENTRY   FAST_FILE_SCAN, Save R2,R3,R4,R5,R6,R7,R8,- ; 1330
                                                          R9,R10,R11                                 ;
              5E      90  AE  9E 00002          MOVAB    -112(SP), SP                                ;
```

```
                                                                    H  4
FASTSCAN        Fast file scan                                      15-Sep-1984 23:56:53     VAX-11 Bliss-32 V4.0-742              Page  24
V04-000         FAST_FILE_SCAN - fast file scan main routine        14-Sep-1984 11:53:52     [BACKUP.SRC]FASTSCAN.B32;1                (3)

                         28   AE         40   8F   9A 00006            MOVZBL  #64, FIB_DESC                                 ; 1369
                         2C   AE         30   AE   9E 0000B            MOVAB   FIB, FIB_DESC+4                               ; 1370
                  00000000'   EF         40   8F   88 00010            BISB2   #64, COM_FLAGS                                ; 1375
                  00000000'   EF       8000   8F   3C 00018            MOVZWL  #32768, FAST_BUFFER_SIZE                      ; 1376
                         7E            8000   8F   3C 00021            MOVZWL  #32768, -(SP)                                 ; 1377
                  00000000G   00              01   FB 00026            CALLS   #1, GET_VM                                    ;
                  00000000'   EF              50   D0 0002D            MOVL    R0, FAST_BUFFER                               ;
                              58              01   D0 00034            MOVL    #1, RVN                                       ; 1382
     0040   8F           00   6E              00   2C 00037  1$:       MOVC5   #0, (SP), #0, #64, FIB                        ; 1397
                                         30   AE      0003E                                                                  ;
                         56 00200100          8F   D0 00040            MOVL    #2097408, ACCTL                               ; 1398
                         30   AE              56   D0 00047            MOVL    ACCTL, FIB                                    ; 1399
                         34   AE 00010001     8F   D0 0004B            MOVL    #65537, FIB+4                                 ; 1400
                         38   AE              58   B0 00053            MOVW    RVN, FIB+8                                    ; 1402
                         14   AE 00090008     8F   D0 00057            MOVL    #589832, ATR_DESC                             ; 1403
                         18   AE         0C   AE   9E 0005F            MOVAB   STATBLK, ATR_DESC+4                           ; 1405
                                         1C   AE   D4 00064            CLRL    ATR_DESC+8                                    ; 1406
                                              7E   D4 00067            CLRL    -(SP)                                         ; 1412
                                         18   AE   9F 00069            PUSHAB  ATR_DESC                                      ;
                                              7E   7C 0006C            CLRQ    -(SP)                                         ;
                                              7E   D4 0006E            CLRL    -(SP)                                         ;
                                         3C   AE   9F 00070            PUSHAB  FIB_DESC                                      ;
                                              7E   7C 00073            CLRQ    -(SP)                                         ;
                                         40   AE   9F 00075            PUSHAB  IOSB                                          ;
                              7E         72   8F   9A 00078            MOVZBL  #114, -(SP)                                   ;
                                  00000000'   EF   DD 0007C            PUSHL   INPUT_CHAN                                    ;
                                              7E   D4 00082            CLRL    -(SP)                                         ;
                  00000000G   00              0C   FB 00084            CALLS   #12, SYS$QIOW                                 ;
                              5A              50   D0 0008B            MOVL    R0, STATUS                                    ;
                              04              5A   E9 0008E            BLBC    STATUS, 2$                                    ; 1413
                              5A         20   AE   3C 00091            MOVZWL  IOSB, STATUS                                  ;
                  0000025C    8F              5A   D1 00095  2$:       CMPL    STATUS, #604                                  ; 1414
                              52              12      0009C            BNEQ    3$                                            ;
     0040   8F           00   6E              00   2C 0009E            MOVC5   #0, (SP), #0, #64, FIB                        ; 1417
                                         30   AE      000A5                                                                  ;
                         56 00200000          8F   D0 000A7            MOVL    #2097152, ACCTL                               ; 1418
                         30   AE 00200000     8F   D0 000AE            MOVL    #2097152, FIB                                 ; 1419
                         34   AE 00010001     8F   D0 000B6            MOVL    #65537, FIB+4                                 ; 1420
                         38   AE              58   B0 000BE            MOVW    RVN, FIB+8                                    ; 1422
                                              7E   D4 000C2            CLRL    -(SP)                                         ; 1428
                                         18   AE   9F 000C4            PUSHAB  ATR_DESC                                      ;
                                              7E   7C 000C7            CLRQ    -(SP)                                         ;
                                              7E   D4 000C9            CLRL    -(SP)                                         ;
                                         3C   AE   9F 000CB            PUSHAB  FIB_DESC                                      ;
                                              7E   7C 000CE            CLRQ    -(SP)                                         ;
                                         40   AE   9F 000D0            PUSHAB  IOSB                                          ;
                              7E         72   8F   9A 000D3            MOVZBL  #114, -(SP)                                   ;
                                  00000000'   EF   DD 000D7            PUSHL   INPUT_CHAN                                    ;
                                              7E   D4 000DD            CLRL    -(SP)                                         ;
                  00000000G   00              0C   FB 000DF            CALLS   #12, SYS$QIOW                                 ;
                              5A              50   D0 000E6            MOVL    R0, STATUS                                    ;
                              07              5A   E9 000E9            BLBC    STATUS, 4$                                    ; 1429
                              5A         20   AE   3C 000EC            MOVZWL  IOSB, STATUS                                  ;
                              1B              5A   E8 000F0  3$:       BLBS    STATUS, 5$                                    ; 1431
                                       0500   8F   BB 000F3  4$:       PUSHR   #^M<R8,R10>                                   ; 1433
               7E 00000000'   EF              10   C1 000F7            ADDL3   #16, INPUT_QUAL, -(SP)                        ;
                                              02   DD 000FF            PUSHL   #2                                            ;
```

```
                      00000000G   8F  DD 00101       PUSHL   #BACKUP$_PROCINDEX                      ;
         00000000G  00            05  FB 00107       CALLS   #5, LIB$SIGNAL                          ;
                      00000000'   EF  DD 0010E 5$:   PUSHL   FAST_BUFFER                             ; 1439
                                  58  DD 00114       PUSHL   RVN                                     ;
             0000V  CF            02  FB 00116       CALLS   #2, READ_HOMEBLOCK                      ;
                    01            58  D1 0011B       CMPL    RVN, #1                                 ; 1446
                                  2A  12 0011E       BNEQ    8$                                      ;
                   52 00000000'   EF  9E 00120       MOVAB   FAST_VOL_BEG, R2                        ; 1449
                   53 00000000'   EF  9E 00127       MOVAB   FAST_VOL_END-4, R3                      ;
                                  15  11 0012E       BRB     7$                                      ;
                   50 00000000'   EF  9A 00130 6$:   MOVZBL  COM_I_SETCOUNT, R0                      ; 1451
      7E            50            02  78 00137       ASHL    #2, R0, -(SP)                           ;
         00000000G  00            01  FB 0013B       CALLS   #1, GET_ZERO_VM                         ;
                    82            50  D0 00142       MOVL    R0, (A)+                                ;
                    53            52  D1 00145 7$:   CMPL    A, R3                                   ; 1449
                                  E6  1B 00148       BLEQU   6$                                      ;
                   54 00000000'   EF  D0 0014A 8$:   MOVL    FAST_BUFFER, R4                         ; 1461
                   53 00000000'FF48   DE 00151       MOVAL   @FAST_IMAP_SIZE[RVN], R3                ; 1462
                   51 00000000'FF48   DE 00159       MOVAL   @FAST_HDR_OFFSET[RVN], R1               ; 1464
                   02 00000000'   EF  91 00161       CMPB    FAST_STRUCLEV, #2                       ; 1458
                                  19  12 00168       BNEQ    9$                                      ;
                   50         0E  A4  3C 0016A       MOVZWL  14(R4), CLUSTER                         ; 1461
                   55         20  A4  3C 0016E       MOVZWL  32(R4), R5                              ; 1462
              FC    A3            55  D0 00172       MOVL    R5, -4(R3)                              ;
      52            50            02  78 00176       ASHL    #2, CLUSTER, BITMAP_OFFSET              ; 1463
                                  52  D6 0017A       INCL    BITMAP_OFFSET                           ;
              FC    A1          6540  DE 0017C       MOVAL   (R5)[CLUSTER], -4(R1)                   ; 1464
                                  12  11 00181       BRB     10$                                     ; 1458
                    50            01  D0 00183 9$:   MOVL    #1, CLUSTER                             ; 1468
              FC    A3            64  3C 00186       MOVZWL  (R4), -4(R3)                            ; 1469
                    52            03  D0 0018A       MOVL    #3, BITMAP_OFFSET                       ; 1470
              FC    A1            64  3C 0018D       MOVZWL  (R4), -4(R1)                            ; 1471
              FC    A1            02  C0 00191       ADDL2   #2, -4(R1)                              ;
                   54 00000000'FF48   DE 00195 10$:  MOVAL   @FAST_IMAP[RVN], R4                     ; 1477
      7E      FC    A3            09  78 0019D       ASHL    #9, -4(R3), -(SP)                       ;
         00000000G  00            01  FB 001A2       CALLS   #1, GET_VM                              ;
              FC    A4            50  D0 001A9       MOVL    R0, -4(R4)                              ;
                                  7E  7C 001AD       CLRQ    -(SP)                                   ; 1484
                                  7E  D4 001AF       CLRL    -(SP)                                   ;
                                  52  DD 001B1       PUSHL   BITMAP_OFFSET                           ;
                   50 00000000'FF48   DE 001B3       MOVAL   @FAST_IMAP_SIZE[RVN], R0                ;
      7E      FC    A0            09  78 001BB       ASHL    #9, -4(R0), -(SP)                       ;
                   50 00000000'FF48   DE 001C0       MOVAL   @FAST_IMAP[RVN], R0                     ;
                              FC  A0  DD 001C8       PUSHL   -4(R0)                                  ;
                                  7E  7C 001CB       CLRQ    -(SP)                                   ;
                              40  AE  9F 001CD       PUSHAB  IOSB                                    ;
                                  31  DD 001D0       PUSHL   #49                                     ;
                      00000000'   EF  DD 001D2       PUSHL   INPUT_CHAN                              ;
                                  7E  D4 001D8       CLRL    -(SP)                                   ;
         00000000G  00            0C  FB 001DA       CALLS   #12, SYS$QIOW                           ;
                    5A            50  D0 001E1       MOVL    R0, STATUS                              ;
                    07            5A  E9 001E4       BLBC    STATUS, 11$                             ; 1485
                    5A        20  AE  3C 001E7       MOVZWL  IOSB, STATUS                            ;
                    1B            5A  E8 001EB       BLBS    STATUS, 12$                             ; 1486
                            0500  8F  BB 001EE 11$:  PUSHR   #^M<R8,R10>                             ; 1488
      7E 00000000'  EF            10  C1 001F2       ADDL3   #16, INPUT_QUAL, -(SP)                  ;
                                  02  DD 001FA       PUSHL   #2                                      ;
```

```
               00000000G 8F   DD 001FC          PUSHL    #BACKUP$_PROCINDEX                ;
   00000000G 00          05   FB 00202          CALLS    #5, LIB$SIGNAL                    ;
                         7E   7C 00209 12$:     CLRQ     -(SP)                             ; 1499
                         7E   D4 0020B          CLRL     -(SP)                             ;
             50 00000000'FF48 DE 0020D          MOVAL    @FAST_HDR_OFFSET[RVN], R0         ;
7E        FC A0          01   C1 00215          ADDL3    #1, -4(R0), -(SP)                 ;
             7E     0200 8F   3C 0021A          MOVZWL   #512, -(SP)                       ;
7E 00000000' EF 00000200 8F   C1 0021F          ADDL3    #512, FAST_BUFFER, -(SP)          ;
                         7E   7C 0022B          CLRQ     -(SP)                             ;
                      40 AE   9F 0022D          PUSHAB   IOSB                              ;
                         31   DD 00230          PUSHL    #49                               ;
               00000000' EF   DD 00232          PUSHL    INPUT_CHAN                        ;
                         7E   D4 00238          CLRL     -(SP)                             ;
   00000000G 00          0C   FB 0023A          CALLS    #12, SYS$QIOW                     ;
             5A          50   D0 00241          MOVL     R0, STATUS                        ;
             07          5A   E9 00244          BLBC     STATUS, 13$                       ; 1500
             5A       20 AE   3C 00247          MOVZWL   IOSB, STATUS                      ;
             1B          5A   E8 0024B          BLBS     STATUS, 14$                       ; 1501
                    0500 8F   BB 0024E 13$:     PUSHR    #^M<R8,R10>                       ; 1503
7E 00000000' EF          10   C1 00252          ADDL3    #16, INPUT_QUAL, -(SP)            ;
                         02   DD 0025A          PUSHL    #2                                ;
               00000000G 8F   DD 0025C          PUSHL    #BACKUP$_PROCINDEX                ;
   00000000G 00          05   FB 00262          CALLS    #5, LIB$SIGNAL                    ;
                         57   D4 00269 14$:     CLRL     EOF                               ; 1506
             50 00000000' EF  D0 0026B          MOVL     FAST_BUFFER, R0                   ; 1508
             02 00000000' EF  91 00272          CMPB     FAST_STRUCLEV, #2                 ; 1507
                         0C   12 00279          BNEQ     15$                               ;
50      021C C0          10   9C 0027B          ROTL     #16, 540(R0), R0                  ; 1508
             57       FF A0   9E 00281          MOVAB    -1(R0), EOF                       ;
                         0B   11 00285          BRB      16$                               ;
             02       0C A0   91 00287 15$:     CMPB     12(R0), #2                        ; 1509
                         05   12 0028B          BNEQ     16$                               ;
57        10 AE          10   9C 0028D          ROTL     #16, STATBLK+4, EOF               ; 1510
             53 00000000'FF48 DE 00292 16$:     MOVAL    @FAST_IMAP_SIZE[RVN], R3          ; 1517
53        FC A3          07   78 0029A          ASHL     #7, -4(R3), R3                    ;
             52          53   D0 0029F          MOVL     R3, J                             ;
                         39   11 002A2          BRB      19$                               ;
             50 00000000'FF48 DE 002A4 17$:     MOVAL    @FAST_IMAP[RVN], R0               ; 1519
             50       FC B042 D0 002AC          MOVL     @-4(R0)[J], R0                    ;
                         2A   13 002B1          BEQL     19$                               ;
53           52          05   78 002B3          ASHL     #5, J, R3                         ; 1523
                         50   DD 002B7          PUSHL    R0                                ; 1524
   00000000G 00          01   FB 002B9          CALLS    #1, LEFT_ONE                      ;
51           53          50   C1 002C0          ADDL3    R0, R3, R1                        ; 1523
             50 00000000'FF48 DE 002C4          MOVAL    @FAST_HDR_OFFSET[RVN], R0         ; 1525
             51       FC A0   C0 002CC          ADDL2    -4(R0), R1                        ;
             57          51   D1 002D0          CMPL     R1, EOF                           ; 1526
                         03   1E 002D3          BGEQU    18$                               ;
             51          57   D0 002D5          MOVL     EOF, R1                           ;
             57          51   D0 002D8 18$:     MOVL     R1, EOF                           ; 1522
                         03   11 002DB          BRB      20$                               ; 1521
             C4          52   F4 002DD 19$:     SOBGEQ   J, 17$                            ; 1517
74 00000000' EF          03   E1 002E0 20$:     BBC      #3, QUAL+10, 23$                  ; 1532
                         7E   7C 002E8          CLRQ     -(SP)                             ; 1544
             7E          01   7D 002EA          MOVQ     #1, -(SP)                         ;
             7E     0200 8F   3C 002ED          MOVZWL   #512, -(SP)                       ;
7E 00000000' EF 00000200 8F   C1 002F2          ADDL3    #512, FAST_BUFFER, -(SP)          ;
```

```
                                                                          K  4
FASTSCAN        Fast file scan                                      15-Sep-1984 23:56:53    VAX-11 Bliss-32 V4.0-742             Page  27
V04-000         FAST_FILE_SCAN - fast file scan main routine        14-Sep-1984 11:53:52    [BACKUP.SRC]FASTSCAN.B32;1                 (3)

                                                 7E  7C 002FE         CLRQ     -(SP)                                          ;
                                       40        AE  9F 00300         PUSHAB   IOSB                                           ;
                                                 31  DD 00303         PUSHL    #49                                            ;
                                00000000'        EF  DD 00305         PUSHL    INPUT_CHAN                                     ;
                                                 7E  D4 0030B         CLRL     -(SP)                                          ;
                         00000000G  00           0C  FB 0030D         CALLS    #12, SYS$QIOW                                  ;
                                    5A           50  D0 00314         MOVL     R0, STATUS                                     ;
                                    07           5A  E9 00317         BLBC     STATUS, 21$                                    ; 1545
                                    5A     20    AE  3C 0031A         MOVZWL   IOSB, STATUS                                   ;
                                    1B           5A  E8 0031E         BLBS     STATUS, 22$                                    ; 1546
                                         0500    8F  BB 00321 21$:    PUSHR    #^M<R8,R10>                                    ; 1548
                   7E 00000000'  EF              10  C1 00325         ADDL3    #16, INPUT_QUAL, -(SP)                         ;
                                                 02  DD 0032D         PUSHL    #2                                             ;
                                00000000G        8F  DD 0032F         PUSHL    #BACKUP$_PROCINDEX                             ;
                         00000000G  00           05  FB 00335         CALLS    #5, LIB$SIGNAL                                 ;
                                    50 00000000'FF48  DE 0033C 22$:   MOVAL    @FAST_BOOT_LBN[RVN], R0                        ; 1553
                              FC    A0           01  CE 00344         MNEGL    #1, -4(R0)                                     ;
                                    51 00000000'  EF  D0 00348         MOVL     FAST_BUFFER, R1                                ; 1554
                                         0205    C1  95 0034F         TSTB     517(R1)                                        ;
                                                 07  12 00353         BNEQ     23$                                            ;
                 FC    A0     0204   C1          10  9C 00355         ROTL     #16, 516(R1), -4(R0)                           ; 1556
                                                 7E  7C 0035C 23$:    CLRQ     -(SP)                                          ; 1564
                                                 7E  7C 0035E         CLRQ     -(SP)                                          ;
                                                 7E  7C 00360         CLRQ     -(SP)                                          ;
                                                 7E  7C 00362         CLRQ     -(SP)                                          ;
                                    7E           34  7D 00364         MOVQ     #52, -(SP)                                     ;
                                00000000'        EF  DD 00367         PUSHL    INPUT_CHAN                                     ;
                                                 7E  D4 0036D         CLRL     -(SP)                                          ;
                         00000000G  00           0C  FB 0036F         CALLS    #12, SYS$QIOW                                  ;
                      03 00000000'  EF           03  E0 00376         BBS      #3, QUAL+10, 24$                               ; 1567
                                            0107     31 0037E         BRW      30$                                            ;
     0040    8F             00      6E           00  2C 00381 24$:    MOVC5    #0, (SP), #0, #64, FIB                         ; 1573
                                       30        AE     00388                                                                 ;
                              30    AE           56  D0 0038A         MOVL     ACCTL, FIB                                     ; 1574
                              34    AE 00020002  8F  D0 0038E         MOVL     #131074, FIB+4                                 ; 1575
                              38    AE           58  B0 00396         MOVW     RVN, FIB+8                                     ; 1577
                                                 7E  7C 0039A         CLRQ     -(SP)                                          ; 1582
                                                 7E  7C 0039C         CLRQ     -(SP)                                          ;
                                                 7E  D4 0039E         CLRL     -(SP)                                          ;
                                       3C        AE  9F 003A0         PUSHAB   FIB_DESC                                       ;
                                                 7E  7C 003A3         CLRQ     -(SP)                                          ;
                                       40        AE  9F 003A5         PUSHAB   IOSB                                           ;
                                    7E     72    8F  9A 003A8         MOVZBL   #114, -(SP)                                    ;
                                00000000'        EF  DD 003AC         PUSHL    INPUT_CHAN                                     ;
                                                 7E  D4 003B2         CLRL     -(SP)                                          ;
                         00000000G  00           0C  FB 003B4         CALLS    #12, SYS$QIOW                                  ;
                                    5A           50  D0 003BB         MOVL     R0, STATUS                                     ;
                                    07           5A  E9 003BE         BLBC     STATUS, 25$                                    ; 1583
                                    5A     20    AE  3C 003C1         MOVZWL   IOSB, STATUS                                   ;
                                    1B           5A  E8 003C5         BLBS     STATUS, 26$                                    ; 1584
                                         0500    8F  BB 003C8 25$:    PUSHR    #^M<R8,R10>                                    ; 1586
                   7E 00000000'  EF              10  C1 003CC         ADDL3    #16, INPUT_QUAL, -(SP)                         ;
                                                 02  DD 003D4         PUSHL    #2                                             ;
                                00000000G        8F  DD 003D6         PUSHL    #BACKUP$_PROCINDEX                             ;
                         00000000G  00           05  FB 003DC         CALLS    #5, LIB$SIGNAL                                 ;
                                                 7E  7C 003E3 26$:    CLRQ     -(SP)                                          ; 1597
                                    7E           01  7D 003E5         MOVQ     #1, -(SP)                                      ;
```

```
                                                                     L 4
FASTSCAN        Fast file scan                                 15-Sep-1984 23:56:53    VAX-11 Bliss-32 V4.0-742                 Page  28
V04-000         FAST_FILE_SCAN - fast file scan main routine   14-Sep-1984 11:53:52    [BACKUP.SRC]FASTSCAN.B32;1                     (3)

                        7E        0200   8F  3C 003E8          MOVZWL  #512, -(SP)                                   ;
           7E 00000000'   EF  00000400   8F  C1 003ED          ADDL3   #1024, FAST_BUFFER, -(SP)                     ;
                                         7E  7C 003F9          CLRQ    -(SP)                                         ;
                                  40     AE  9F 003FB          PUSHAB  IOSB                                          ;
                                         31  DD 003FE          PUSHL   #49                                           ;
                           00000000'     EF  DD 00400          PUSHL   INPUT_CHAN                                    ;
                                         7E  D4 00406          CLRL    -(SP)                                         ;
                   00000000G  00         0C  FB 00408          CALLS   #12, SYS$QIOW                                 ;
                              5A         50  D0 0040F          MOVL    R0, STATUS                                    ;
                              07         5A  E9 00412          BLBC    STATUS, 27$                                   ; 1598
                              5A    20   AE  3C 00415          MOVZWL  IOSB, STATUS                                  ;
                              1B         5A  E8 00419          BLBS    STATUS, 28$                                   ; 1599
                                  0500   8F  BB 0041C  27$:    PUSHR   #^M<R8,R10>                                   ; 1601
           7E 00000000'   EF             10  C1 00420          ADDL3   #16, INPUT_QUAL, -(SP)                        ;
                                         02  DD 00428          PUSHL   #2                                            ;
                           00000000G     8F  DD 0042A          PUSHL   #BACKUP$_PROCINDEX                            ;
                   00000000G  00         05  FB 00430          CALLS   #5, LIB$SIGNAL                                ;
                                         7E  7C 00437  28$:    CLRQ    -(SP)                                         ; 1608
                                         7E  7C 00439          CLRQ    -(SP)                                         ;
                                         7E  7C 0043B          CLRQ    -(SP)                                         ;
                                         7E  7C 0043D          CLRQ    -(SP)                                         ;
                              7E         34  7D 0043F          MOVQ    #52, -(SP)                                    ;
                           00000000'     EF  DD 00442          PUSHL   INPUT_CHAN                                    ;
                                         7E  D4 00448          CLRL    -(SP)                                         ;
                   00000000G  00         0C  FB 0044A          CALLS   #12, SYS$QIOW                                 ;
                              0B 00000000'  EF  E9 00451       BLBC    QUAL+14, 29$                                  ; 1613
        58 00000000'   EF     08         00  ED 00458          CMPZV   #0, #8, QUAL+79, RVN                          ;
                                         25  12 00461          BNEQ    30$                                           ;
                              50 00000000'FF48  DE 00463  29$: MOVAL   @FAST_HDR_OFFSET[RVN], R0                     ; 1619
                        7E    57    FC   A0  C3 0046B          SUBL3   -4(R0), EOF, -(SP)                            ;
                              50 00000000'  EF  D0 00470       MOVL    FAST_BUFFER, R0                               ; 1618
                                  0400   C0  9F 00477          PUSHAB  1024(R0)                                      ;
                                  0200   C0  9F 0047B          PUSHAB  512(R0)                                       ; 1617
                                         50  DD 0047F          PUSHL   R0                                            ; 1616
                   00000000G  00         04  FB 00481          CALLS   #4, VOLUME_ATTRIBUTES                         ;
    0040    8F                00   6E    00  2C 00488  30$:    MOVC5   #0, (SP), #0, #64, FIB                        ; 1625
                                  30     AE     0048F                                                               ;
                        30    AE         56  D0 00491          MOVL    ACCTL, FIB                                    ; 1626
                        34    AE  00010001   8F  D0 00495      MOVL    #65537, FIB+4                                 ; 1627
                        38    AE         58  B0 0049D          MOVW    RVN, FIB+8                                    ; 1629
                                         7E  7C 004A1          CLRQ    -(SP)                                         ; 1634
                                         7E  7C 004A3          CLRQ    -(SP)                                         ;
                                         7E  D4 004A5          CLRL    -(SP)                                         ;
                                  3C     AE  9F 004A7          PUSHAB  FIB_DESC                                      ;
                                         7E  7C 004AA          CLRQ    -(SP)                                         ;
                                  40     AE  9F 004AC          PUSHAB  IOSB                                          ;
                              7E  72     8F  9A 004AF          MOVZBL  #114, -(SP)                                   ;
                           00000000'     EF  DD 004B3          PUSHL   INPUT_CHAN                                    ;
                                         7E  D4 004B9          CLRL    -(SP)                                         ;
                   00000000G  00         0C  FB 004BB          CALLS   #12, SYS$QIOW                                 ;
                              5A         50  D0 004C2          MOVL    R0, STATUS                                    ;
                              07         5A  E9 004C5          BLBC    STATUS, 31$                                   ; 1635
                              5A    20   AE  3C 004C8          MOVZWL  IOSB, STATUS                                  ;
                              1B         5A  E8 004CC          BLBS    STATUS, 32$                                   ; 1636
                                  0500   8F  BB 004CF  31$:    PUSHR   #^M<R8,R10>                                   ; 1638
           7E 00000000'   EF             10  C1 004D3          ADDL3   #16, INPUT_QUAL, -(SP)                        ;
                                         02  DD 004DB          PUSHL   #2                                            ;
```

```
                             00000000G  8F  DD 004DD         PUSHL   #BACKUP$_PROCINDEX                        :
                 00000000G  00          05  FB 004E3         CALLS   #5, LIB$SIGNAL                            :
                            00000000'   EF  94 004EA  32$:   CLRB    DIR_STATUS                                : 1644
                        50  00000000'FF48   DE 004F0         MOVAL   @FAST_HDR_OFFSET[RVN], R0                 : 1645
          59        FC  A0              01  C1 004F8         ADDL3   #1, -4(R0), VBN                           :
                        6E        01    A7  9E 004FD         MOVAB   1(R7), (SP)                               : 1655
                        57              59  D1 00501  33$:   CMPL    VBN, EOF                                  : 1646
                                        03  1B 00504         BLEQU   34$                                       :
                                      01B4  31 00506         BRW     58$                                       :
          50            6E              59  C3 00509  34$:   SUBL3   VBN, (SP), R0                             : 1655
                 00000040  8F           50  D1 0050D         CMPL    R0, #64                                   :
                                        04  1B 00514         BLEQU   35$                                       :
                        50        40    8F  9A 00516         MOVZBL  #64, R0                                   :
                        5B              50  D0 0051A  35$:   MOVL    R0, READ_COUNT                            :
                                        7E  7C 0051D         CLRQ    -(SP)                                     : 1662
                                        7E  D4 0051F         CLRL    -(SP)                                     :
                                        59  DD 00521         PUSHL   VBN                                       :
          7E            5B              09  78 00523         ASHL    #9, READ_COUNT, -(SP)                     :
                            00000000'   EF  DD 00527         PUSHL   FAST_BUFFER                               :
                                        7E  7C 0052D         CLRQ    -(SP)                                     :
                                  40    AE  9F 0052F         PUSHAB  IOSB                                      :
                                        31  DD 00532         PUSHL   #49                                       :
                            00000000'   EF  DD 00534         PUSHL   INPUT_CHAN                                :
                                        7E  D4 0053A         CLRL    -(SP)                                     :
                 00000000G  00          0C  FB 0053C         CALLS   #12, SYS$QIOW                             :
                        5A              50  D0 00543         MOVL    R0, STATUS                                :
                        07              5A  E9 00546         BLBC    STATUS, 36$                               : 1663
                        5A        20    AE  3C 00549         MOVZWL  IOSB, STATUS                              :
                        4E              5A  E8 0054D         BLBS    STATUS, 40$                               : 1669
                        56              01  CE 00550  36$:   MNEGL   #1, XVBN                                  : 1672
                                        45  11 00553         BRB     39$                                       :
          50            56              09  78 00555  37$:   ASHL    #9, XVBN, R0                              : 1678
          52            50  00000000'   EF  C1 00559         ADDL3   FAST_BUFFER, R0, HEADER                   :
                                        7E  7C 00561         CLRQ    -(SP)                                     : 1685
                                        7E  D4 00563         CLRL    -(SP)                                     :
                                      6649  9F 00565         PUSHAB  (XVBN)[VBN]                               :
                        7E      0200    8F  3C 00568         MOVZWL  #512, -(SP)                               :
                                        52  DD 0056D         PUSHL   HEADER                                    :
                                        7E  7C 0056F         CLRQ    -(SP)                                     :
                                  40    AE  9F 00571         PUSHAB  IOSB                                      :
                                        31  DD 00574         PUSHL   #49                                       :
                            00000000'   EF  DD 00576         PUSHL   INPUT_CHAN                                :
                                        7E  D4 0057C         CLRL    -(SP)                                     :
                 00000000G  00          0C  FB 0057E         CALLS   #12, SYS$QIOW                             :
                        5A              50  D0 00585         MOVL    R0, STATUS                                :
                        07              5A  E9 00588         BLBC    STATUS, 38$                               : 1686
                        5A        20    AE  3C 0058B         MOVZWL  IOSB, STATUS                              :
                        08              5A  E8 0058F         BLBS    STATUS, 39$                               : 1687
 0200   8F        00    6E              00  2C 00592  38$:   MOVC5   #0, (SP), #0, #512, (HEADER)              : 1689
                                        62     00599                                                           :
          B7            56              5B  F2 0059A  39$:   AOBLSS  READ_COUNT, XVBN, 37$                     : 1672
                        52  00000000'   EF  D0 0059E  40$:   MOVL    FAST_BUFFER, HEADER                       : 1697
                        54              01  CE 005A5         MNEGL   #1, XVBN                                  : 1698
                                      0102  31 005A8         BRW     55$                                       :
          51            59              54  C1 005AB  41$:   ADDL3   XVBN, VBN, R1                             : 1708
                        50  00000000'FF48   DE 005AF         MOVAL   @FAST_HDR_OFFSET[RVN], R0                 :
          53            51        FC    A0  C3 005B7         SUBL3   -4(R0), R1, FILE_NUMBER                   :
```

```
                     04  AE            53  B0 005BC          MOVW    FILE_NUMBER, FILE_ID                       ; 1709
   50                53  08            10  EF 005C0          EXTZV   #16, #8, FILE_NUMBER, R0                   ; 1710
                     09  AE            50  90 005C5          MOVB    R0, FILE_ID+5                              ;
                     02 00000000'  EF  91 005C9              CMPB    FAST_STRUCLEV, #2                          ; 1711
                                   07  12 005D0              BNEQ    42$                                        ;
                     06  AE      0A    A2  B0 005D2          MOVW    10(HEADER), FILE_ID+2                      ; 1712
                                   05  11 005D7              BRB     43$                                        ;
                     06  AE      04    A2  B0 005D9 42$:     MOVW    4(HEADER), FILE_ID+2                       ; 1713
                     08  AE            58  90 005DE 43$:     MOVB    RVN, FILE_ID+4                             ; 1714
                  25 00000000'   EF    03  E1 005E2          BBC     #3, QUAL+10, 45$                           ; 1720
                                       54  D5 005EA          TSTL    XVBN                                       ; 1721
                                       21  12 005EC          BNEQ    45$                                        ;
                  0B 00000000'   EF    E9 005EE              BLBC    QUAL+14, 44$                               ; 1722
   58 00000000'   EF             08    00  ED 005F5          CMPZV   #0, #8, QUAL+79, RVN                       ;
                                       0F  12 005FE          BNEQ    45$                                        ;
                             0108  8F  BB 00600 44$:         PUSHR   #^M<R3,R8>                                 ; 1724
                             0804  8F  BB 00604              PUSHR   #^M<R2,R11>                                ;
                     00000000G  00     04  FB 00608          CALLS   #4, GEN_FID_RECORD                         ;
                             04  AE    9F 0060F 45$:         PUSHAB  FILE_ID                                    ; 1732
                                       52  DD 00612          PUSHL   HEADER                                     ;
                         0000V  CF     02  FB 00614          CALLS   #2, VERIFY_HEADER                          ;
                                 5A    50  D0 00619          MOVL    R0, STATUS                                 ;
                     01 00000000'  EF  91 0061C              CMPB    FAST_STRUCLEV, #1                          ; 1734
                                   13  12 00623              BNEQ    46$                                        ;
                     50 00000000'FF48  DE 00625              MOVAL   @FAST_IMAP[RVN], R0                        ; 1735
                     51          FF    A3  9E 0062D          MOVAB   -1(R3), R1                                 ;
                     02          FC    B0  51  E0 00631      BBS     R1, @-4(R0), 46$                           ;
                                       5A  D4 00636          CLRL    STATUS                                     ; 1737
                     50 00000000'FF48  DE 00638 46$:         MOVAL   @FAST_IMAP[RVN], R0                        ; 1743
                                       53  D7 00640          DECL    R3                                         ;
                     00          FC    B0  53  E5 00642      BBCC    R3, @-4(R0), 47$                           ;
                                 5E    5A  E9 00647 47$:     BLBC    STATUS, 54$                                ; 1746
                                 51  01  A2  9A 0064A        MOVZBL  1(HEADER), R1                              ; 1753
                                 51  6241  3E 0064E          MOVAW   (HEADER)[R1], MAP_AREA                     ;
                                       55  D4 00652          CLRL    R5                                         ; 1756
                     02 00000000'  EF  91 00654              CMPB    FAST_STRUCLEV, #2                          ;
                                   07  12 0065B              BNEQ    48$                                        ;
                                       55  D6 0065D          INCL    R5                                         ;
                                 04    A2  B5 0065F          TSTW    4(HEADER)                                  ; 1757
                                       02  11 00662          BRB     49$                                        ;
                                       61  95 00664 48$:     TSTB    (MAP_AREA)                                 ; 1758
                                       40  12 00666 49$:     BNEQ    54$                                        ;
                     00          FC    B0  53  E2 00668      BBSS    R3, @-4(R0), 50$                           ; 1769
                                 06    55  E9 0066D 50$:     BLBC    R5, 51$                                    ; 1774
                                 50  34  A2  9E 00670        MOVAB   52(R2), FCH                                ; 1775
                                       04  11 00674          BRB     52$                                        ;
                                 50  0C  A2  9E 00676 51$:   MOVAB   12(R2), FCH                                ; 1776
                                       60  B5 0067A 52$:     TSTW    (FCH)                                      ; 1785
                                       1D  19 0067C          BLSS    53$                                        ;
                                       52  DD 0067E          PUSHL   HEADER                                     ; 1790
                     00000000G  00     01  FB 00680          CALLS   #1, INIT_ATTR                              ;
                  19 00000000'   EF    03  E0 00687          BBS     #3, QUAL+10, 54$                           ; 1791
                                       01  DD 0068F          PUSHL   #1                                         ; 1793
                     00000000G  00     01  FB 00691          CALLS   #1, SELECT_INPUT_FILE                      ;
                                 0D    50  E8 00698          BLBS    R0, 54$                                    ;
                     50 00000000'FF48  DE 0069B 53$:         MOVAL   @FAST_IMAP[RVN], R0                        ; 1797
                     00          FC    B0  53  E4 006A3      BBSC    R3, @-4(R0), 54$                           ;
```

```
                            52       0200    C2  9E 006A8 54$:    MOVAB   512(R2), HEADER                                 ; 1802
                 02         54                5B  F2 006AD 55$:    AOBLSS  READ_COUNT, XVBN, 56$                           ; 1698
                                              03  11 006B1         BRB     57$                                             ;
                                            FEF5  31 006B3 56$:    BRW     41$                                             ;
                            59         40    A9  9E 006B6 57$:    MOVAB   64(R9), VBN                                     ; 1806
                                            FE44  31 006BA         BRW     33$                                             ; 1646
                                              7E  7C 006BD 58$:    CLRQ    -(SP)                                           ; 1814
                                              7E  7C 006BF         CLRQ    -(SP)                                           ;
                                              7E  7C 006C1         CLRQ    -(SP)                                           ;
                                              7E  7C 006C3         CLRQ    -(SP)                                           ;
                            7E                34  7D 006C5         MOVQ    #52, -(SP)                                      ;
                                00000000'     EF  DD 006C8         PUSHL   INPUT_CHAN                                      ;
                                              7E  D4 006CE         CLRL    -(SP)                                           ;
                   00000000G  00              0C  FB 006D0         CALLS   #12, SYS$QIOW                                   ;
                                              58  D6 006D7         INCL    RVN                                             ; 1817
         58 00000000'  EF       08            00  ED 006D9         CMPZV   #0, #8, COM_I_SETCOUNT, RVN                     ; 1819
                                              03  1F 006E2         BLSSU   59$                                             ;
                                            F950  31 006E4         BRW     1$                                              ;
                                00000000'     EF  DD 006E7 59$:    PUSHL   FAST_BUFFER                                     ; 1824
                                00000000'     EF  DD 006ED         PUSHL   FAST_BUFFER_SIZE                                ;
                   00000000G  00              02  FB 006F3         CALLS   #2, FREE_VM                                     ;
                                00000000'     EF  7C 006FA         CLRQ    FAST_BUFFER                                     ; 1825
                   00000000'  EF       40    8F  8A 00700         BICB2   #64, COM_FLAGS                                  ; 1826
                            53 00000000'     EF  9A 00708         MOVZBL  COM_I_SETCOUNT, R3                              ; 1831
                                              52  D4 0070F         CLRL    RVN                                             ;
                                              07  11 00711         BRB     61$                                             ;
                                              52  DD 00713 60$:    PUSHL   RVN                                             ;
                       0000V  CF              01  FB 00715         CALLS   #1, DIR_SCAN                                    ;
                 F5         52                53  F3 0071A 61$:    AOBLEQ  R3, RVN, 60$                                    ;
                   00000000'  EF     0200    8F  3C 0071E         MOVZWL  #512, FAST_BUFFER_SIZE                          ; 1836
                              7E     0200    8F  3C 00727         MOVZWL  #512, -(SP)                                     ; 1837
                   00000000G  00              01  FB 0072C         CALLS   #1, GET_VM                                      ;
                   00000000'  EF              50  D0 00733         MOVL    R0, FAST_BUFFER                                 ;
                03 00000000'  EF              03  E0 0073A         BBS     #3, QUAL+10, 62$                                ; 1842
                                            01CA  31 00742         BRW     75$                                             ;
                            59 00000000'     EF  9A 00745 62$:    MOVZBL  COM_I_SETCOUNT, R9                              ; 1845
                                              56  D4 0074C         CLRL    RVN                                             ;
                                            01B8  31 0074E         BRW     74$                                             ;
                            50 00000000'FF46  DE 00751 63$:    MOVAL   @FAST_IMAP_SIZE[RVN], R0                        ; 1850
                 57              FC    A0     0C  78 00759         ASHL    #12, -4(R0), R7                                 ;
                                              58  D4 0075E         CLRL    FILE_NUMBER                                     ;
                                            01A0  31 00760 64$:    BRW     73$                                             ;
                            50 00000000'FF46  DE 00763 65$:    MOVAL   @FAST_IMAP[RVN], R0                             ; 1852
                            51         FF    A8  9E 0076B         MOVAB   -1(R8), R1                                      ;
                 EC              FC    B0     51  E1 0076F         BBC     R1, @-4(R0), 64$                                ;
     0040    8F  00                    6E     00  2C 00774         MOVC5   #0, (SP), #0, #64, FIB                          ; 1864
                                       30    AE      0077B                                                             ;
                 30         AE 00200000    8F  D0 0077D         MOVL    #2097152, FIB                                   ; 1865
                 34         AE 00010001    8F  D0 00785         MOVL    #65537, FIB+4                                   ; 1866
                 38         AE                56  B0 0078D         MOVW    RVN, FIB+8                                      ; 1868
                                              7E  7C 00791         CLRQ    -(SP)                                           ; 1873
                                              7E  7C 00793         CLRQ    -(SP)                                           ;
                                              7E  D4 00795         CLRL    -(SP)                                           ;
                                       3C    AE  9F 00797         PUSHAB  FIB_DESC                                        ;
                                              7E  7C 0079A         CLRQ    -(SP)                                           ;
                                       40    AE  9F 0079C         PUSHAB  IOSB                                            ;
                            7E         72    8F  9A 0079F         MOVZBL  #114, -(SP)                                     ;
```

```
                                    00000000'  EF  DD 007A3          PUSHL   INPUT_CHAN                              :
                                               7E  D4 007A9          CLRL    -(SP)                                   :
                      00000000G  00            0C  FB 007AB          CALLS   #12, SYS$QIOW                           :
                                 5A            50  D0 007B2          MOVL    R0, STATUS                              :
                                 07            5A  E9 007B5          BLBC    STATUS, 66$                             : 1874
                                 5A        20  AE  3C 007B8          MOVZWL  IOSB, STATUS                            :
                                 1B            5A  E8 007BC          BLBS    STATUS, 67$                             : 1875
                                         0440  8F  BB 007BF 66$:     PUSHR   #^M<R6,R10>                             : 1877
                   7E 00000000'  EF            10  C1 007C3          ADDL3   #16, INPUT_QUAL, -(SP)                  :
                                               02  DD 007CB          PUSHL   #2                                      :
                                    00000000G  8F  DD 007CD          PUSHL   #BACKUP$_PROCINDEX                      :
                      00000000G  00            05  FB 007D3          CALLS   #5, LIB$SIGNAL                          :
                                               7E  7C 007DA 67$:     CLRQ    -(SP)                                   : 1888
                                               7E  D4 007DC          CLRL    -(SP)                                   :
                                 50 00000000'FF46  DE 007DE          MOVAL   @FAST_HDR_OFFSET[RVN], R0               :
                                          FC B048  9F 007E6          PUSHAB  @-4(R0)[FILE_NUMBER]                    :
                                 7E      0200  8F  3C 007EA          MOVZWL  #512, -(SP)                             :
                                    00000000'  EF  DD 007EF          PUSHL   FAST_BUFFER                             :
                                               7E  7C 007F5          CLRQ    -(SP)                                   :
                                           40  AE  9F 007F7          PUSHAB  IOSB                                    :
                                               31  DD 007FA          PUSHL   #49                                     :
                                    00000000'  EF  DD 007FC          PUSHL   INPUT_CHAN                              :
                                               7E  D4 00802          CLRL    -(SP)                                   :
                      00000000G  00            0C  FB 00804          CALLS   #12, SYS$QIOW                           :
                                 5A            50  D0 0080B          MOVL    R0, STATUS                              :
                                 07            5A  E9 0080E          BLBC    STATUS, 68$                             : 1889
                                 5A        20  AE  3C 00811          MOVZWL  IOSB, STATUS                            :
                                 1B            5A  E8 00815          BLBS    STATUS, 69$                             : 1890
                                         0440  8F  BB 00818 68$:     PUSHR   #^M<R6,R10>                             : 1892
                   7E 00000000'  EF            10  C1 0081C          ADDL3   #16, INPUT_QUAL, -(SP)                  :
                                               02  DD 00824          PUSHL   #2                                      :
                                    00000000G  8F  DD 00826          PUSHL   #BACKUP$_PROCINDEX                      :
                      00000000G  00            05  FB 0082C          CALLS   #5, LIB$SIGNAL                          :
                                               7E  7C 00833 69$:     CLRQ    -(SP)                                   : 1899
                                               7E  7C 00835          CLRQ    -(SP)                                   :
                                               7E  7C 00837          CLRQ    -(SP)                                   :
                                               7E  7C 00839          CLRQ    -(SP)                                   :
                                 7E            34  7D 0083B          MOVQ    #52, -(SP)                              :
                                    00000000'  EF  DD 0083E          PUSHL   INPUT_CHAN                              :
                                               7E  D4 00844          CLRL    -(SP)                                   :
                      00000000G  00            0C  FB 00846          CALLS   #12, SYS$QIOW                           :
                                 50 00000000'  EF  D0 0084D          MOVL    INPUT_NAM, R0                           : 1904
                           24    A0            58  B0 00854          MOVW    FILE_NUMBER, 36(R0)                     :
      51                   58    08            10  EF 00858          EXTZV   #16, #8, FILE_NUMBER, R1                : 1905
                           29    A0            51  90 0085D          MOVB    R1, 41(R0)                              :
                           28    A0            56  90 00861          MOVB    RVN, 40(R0)                             : 1906
                                           2A  A0  D4 00865          CLRL    42(R0)                                  : 1907
                                           2E  A0  B4 00868          CLRW    46(R0)                                  : 1909
                                 51 00000000'  EF  D0 0086B          MOVL    FAST_BUFFER, R1                         : 1921
                                 02 00000000'  EF  91 00872          CMPB    FAST_STRUCLEV, #2                       : 1915
                                               21  12 00879          BNEQ    71$                                     :
                           26    A0        0A  A1  B0 0087B          MOVW    10(R1), 38(R0)                          : 1921
                                 50            61  9A 00880          MOVZBL  (R1), R0                                : 1923
                                 52          6140  3E 00883          MOVAW   (R1)[R0], NAME_ADDRESS                  : 1924
                                 53            14  D0 00887          MOVL    #20, NAME_LENGTH                        : 1925
                           62    14            20  3A 0088A          LOCC    #32, #20, (NAME_ADDRESS)                : 1926
                                 02            12  12 0088E          BNEQ    70$                                     :
```

```
                                                                    D  5
FASTSCAN        Fast file scan                                    15-Sep-1984 23:56:53    VAX-11 Bliss-32 V4.0-742               Page  33
V04-000         FAST_FILE_SCAN - fast file scan main routine      14-Sep-1984 11:53:52    [BACKUP.SRC]FASTSCAN.B32;1                (3)

                                                       51 D4 00890          CLRL    R1                                            ;
                                                       51 D5 00892 70$:     TSTL    P                                             ; 1927
                                                       23 13 00894          BEQL    72$                                           ;
                          53             51            52 C3 00896          SUBL3   NAME_ADDRESS, P, NAME_LENGTH                  ;
                                                       1D 11 0089A          BRB     72$                                           ; 1915
                          26             A0       04   A1 B0 0089C 71$:     MOVW    4(R1), 38(R0)                                 ; 1931
                                         52       04   AE 9E 008A1          MOVAB   FILENAME, NAME_ADDRESS                        ; 1932
                                                  04   AE 9F 008A5          PUSHAB  FILENAME                                      ; 1933
                                         50            61 9A 008A8          MOVZBL  (R1), R0                                      ; 1934
                                                  FA A140 3F 008AB          PUSHAW  -6(R1)[R0]                                    ; 1935
                             00000000G   00            02 FB 008AF          CALLS   #2, MAKE_STRING                               ;
                                         53            50 3C 008B6          MOVZWL  R0, NAME_LENGTH                               ; 1936
                          18             AE       FF   8F 9A 008B9 72$:     MOVZBL  #255, RSA_DESC                                ; 1938
                                         50 00000000'  EF D0 008BE          MOVL    INPUT_NAM, R0                                 ; 1939
                          1C             AE       04   A0 D0 008C5          MOVL    4(R0), RSA_DESC+4                             ;
                                                       52 DD 008CA          PUSHL   NAME_ADDRESS                                  ; 1945
                                                       53 DD 008CC          PUSHL   NAME_LENGTH                                   ;
                          7E 00000000'   EF            10 C1 008CE          ADDL3   #16, INPUT_QUAL, -(SP)                        ;
                                                  24   AE 9F 008D6          PUSHAB  RSA_DESC                                      ;
                                                  28   AE 9F 008D9          PUSHAB  RSA_DESC                                      ;
                                                F718   CF 9F 008DC          PUSHAB  P.AAB                                         ;
                             00000000G   00            06 FB 008E0          CALLS   #6, SYS$FAO                                   ;
                                         50 00000000'  EF D0 008E7          MOVL    INPUT_NAM, R0                                 ; 1946
                          03             A0       18   AE 90 008EE          MOVB    RSA_DESC, 3(R0)                               ;
                                                       50 DD 008F3          PUSHL   R0                                            ; 1947
                             00000000G   00            01 FB 008F5          CALLS   #1, INIT_NAMEBLOCK                            ;
                             00000000G   00            00 FB 008FC          CALLS   #0, SAVE_ONE_FILE                             ; 1952
         FE5A             58             01            57 F1 00903 73$:     ACBL    R7, #1, FILE_NUMBER, 65$                      ; 1850
         FE42             56             01            59 F1 00909 74$:     ACBL    R9, #1, RVN, 63$                              ; 1845
                                            00000000'  EF DD 0090F 75$:     PUSHL   FAST_BUFFER                                   ; 1961
                                            00000000'  EF DD 00915          PUSHL   FAST_BUFFER_SIZE                              ;
                             00000000G   00            02 FB 0091B          CALLS   #2, FREE_VM                                   ;
                                            00000000'  EF 7C 00922          CLRQ    FAST_BUFFER                                   ; 1962
                                         54 00000000'  EF 9A 00928          MOVZBL  COM_I_SETCOUNT, R4                            ; 1967
                                         53 00000000'  EF D0 0092F          MOVL    FAST_IMAP, R3                                 ; 1969
                                                       52 D4 00936          CLRL    RVN                                           ;
                                                       23 11 00938          BRB     77$                                           ;
                                                  FC A342 DD 0093A 76$:     PUSHL   -4(R3)[RVN]                                   ;
                                         50 00000000'FF42 DE 0093E          MOVAL   @FAST_IMAP_SIZE[RVN], R0                      ;
                          7E       FC    A0            09 78 00946          ASHL    #9, -4(R0), -(SP)                             ;
                             00000000G   00            02 FB 0094B          CALLS   #2, FREE_VM                                   ;
                                         53 00000000'  EF D0 00952          MOVL    FAST_IMAP, R3                                 ; 1970
                                                  FC A342 D4 00959          CLRL    -4(R3)[RVN]                                   ;
                          D9             52            54 F3 0095D 77$:     AOBLEQ  R4, RVN, 76$                                  ; 1967
                                         52 00000000'  EF 9E 00961          MOVAB   FAST_VOL_BEG, R2                              ; 1976
                                         53 00000000'  EF 9E 00968          MOVAB   FAST_VOL_END-4, R3                            ;
                                                       16 11 0096F          BRB     79$                                           ;
                                                       62 DD 00971 78$:     PUSHL   (A)                                           ; 1978
                                         50 00000000'  EF 9A 00973          MOVZBL  COM_I_SETCOUNT, R0                            ;
                          7E             50            02 78 0097A          ASHL    #2, R0, -(SP)                                 ;
                             00000000G   00            02 FB 0097E          CALLS   #2, FREE_VM                                   ;
                                                       82 D4 00985          CLRL    (A)+                                          ; 1979
                                         53            52 D1 00987 79$:     CMPL    A, R3                                         ; 1976
                                                       E5 1B 0098A          BLEQU   78$                                           ;
                                                          04 0098C          RET                                                   ; 1981

; Routine Size:  2445 bytes,    Routine Base:  CODE + 0018
```

```
;   880         1982  1 %SBTTL 'SLOW_FILE_SCAN - slow file scan main routine'
;   881         1983  1 GLOBAL ROUTINE SLOW_FILE_SCAN: NOVALUE=
;   882         1984  1
;   883         1985  1 !++
;   884         1986  1 !
;   885         1987  1 ! FUNCTIONAL DESCRIPTION:
;   886         1988  1 !       This routine is the driver for the slow file scan.
;   887         1989  1 !
;   888         1990  1 ! INPUT PARAMETERS:
;   889         1991  1 !       NONE
;   890         1992  1 !
;   891         1993  1 ! IMPLICIT INPUTS:
;   892         1994  1 !       NONE
;   893         1995  1 !
;   894         1996  1 ! OUTPUT PARAMETERS:
;   895         1997  1 !       NONE
;   896         1998  1 !
;   897         1999  1 ! IMPLICIT OUTPUTS:
;   898         2000  1 !       NONE
;   899         2001  1 !
;   900         2002  1 ! ROUTINE VALUE:
;   901         2003  1 !       NONE
;   902         2004  1 !
;   903         2005  1 ! SIDE EFFECTS:
;   904         2006  1 !       NONE
;   905         2007  1 !
;   906         2008  1 !--
;   907         2009  1
;   908         2010  2 BEGIN
;   909         2011  2 DIR_SCAN(1);
;   910         2012  1 END;



                                         0000 00000        .ENTRY  SLOW_FILE_SCAN, Save nothing                      ; 1983
                                      01  DD 00002         PUSHL   #1                                                ; 2011
                   0000V  CF          01  FB 00004         CALLS   #1, DIR_SCAN                                      ;
                                          04 00009         RET                                                       ; 2012

; Routine Size:  10 bytes,     Routine Base:  CODE + 09A5
```

```
    1 %SBTTL 'READ_HOMEBLOCK - read home block from index file'
    1 ROUTINE READ_HOMEBLOCK(RVN,BUFFER): NOVALUE=
    1
    1 !++
    1 !
    1 ! FUNCTIONAL DESCRIPTION:
    1 !       This routine reads the first good home block of the currently open
    1 !       index file into the buffer supplied.
    1 !
    1 ! INPUT PARAMETERS:
    1 !       RVN             - Relative volume number.
    1 !       BUFFER          - Pointer to buffer.
    1 !
    1 ! IMPLICIT INPUTS:
    1 !       NONE
    1 !
    1 ! OUTPUT PARAMETERS:
    1 !       NONE
    1 !
    1 ! IMPLICIT OUTPUTS:
    1 !       BUFFER          - Contains a valid home block.
    1 !       COM_I_SETCOUNT  - Count of volumes in volume set.
    1 !       FAST_STRUCLEV   - Structure level (1 or 2) of the volume set.
    1 !
    1 ! ROUTINE VALUE:
    1 !       NONE
    1 !
    1 ! SIDE EFFECTS:
    1 !       NONE
    1 !
    1 !--
    1
    2 BEGIN
    2 MAP
    2       BUFFER:         REF BBLOCK;     ! Pointer to buffer
    2 LOCAL
    2       STATUS,                         ! General status value
    2       IOSB:           VECTOR[4,WORD], ! I/O status block
    2       OLD_STATUS;                     ! Save status for error message
    2
    2
    2 ! We keep reading until we get a block that reads without errors and looks
    2 ! like a home block. Track any error status for the eventual error message.
    2 !
    2 OLD_STATUS = SS$_ABORT;
    2 INCR VBN FROM 2 TO 100 DO
    3     BEGIN
P   3     STATUS = $QIOW(
P   3         FUNC=IO$_READVBLK,
P   3         CHAN=.INPUT_CHAN,
P   3         IOSB=IOSB,
P   3         P1=.BUFFER,
P   3         P2=512,
    3         P3=.VBN);
    3     IF .STATUS THEN STATUS = .IOSB[0];
    3
    3     IF NOT .STATUS
```

```
3       THEN
3           OLD_STATUS = .STATUS
3       ELSE
3           IF
3               .BUFFER[HM2$B_STRUCLEV] EQL 2 AND
3               .BUFFER[HM2$W_HOMEVBN] EQL .VBN AND
3               .BUFFER[HM2$L_ALTIDXLBN] NEQ 0 AND
3               .BUFFER[HM2$W_CLUSTER] NEQ 0 AND
3               .BUFFER[HM2$W_HOMEVBN] NEQ 0 AND
3               .BUFFER[HM2$W_ALHOMEVBN] NEQ 0 AND
3               .BUFFER[HM2$W_ALTIDXVBN] NEQ 0 AND
3               .BUFFER[HM2$W_IBMAPVBN] NEQ 0 AND
3               .BUFFER[HM2$L_IBMAPLBN] NEQ 0 AND
3               .BUFFER[HM2$L_MAXFILES] NEQ 0 AND
3               .BUFFER[HM2$W_IBMAPSIZE] NEQ 0 AND
3               .BUFFER[HM2$W_RESFILES] NEQ 0 AND
3               CHECKSUM2(.BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM1)) AND
3               CHECKSUM2(.BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM2))
3           THEN
4               BEGIN
4               IF .RVN EQL 1
4               THEN
5                   BEGIN
5                   FAST_STRUCLEV = 2;
5                   IF .BUFFER[HM2$W_SETCOUNT] GTRU MAX_VOLUMES
5                       THEN SIGNAL(BACKUP$_MAXVOLS, 1, INPUT_QUAL[QUAL_DEV_DESC]);
5                   COM_I_SETCOUNT = .BUFFER[HM2$W_SETCOUNT];
5                   IF .COM_I_SETCOUNT EQL 0 THEN COM_I_SETCOUNT = 1;
5                   IF
5                       .QUAL[QUAL_VOLU] AND
5                       .QUAL[QUAL_VOLU_VALUE] GTRU .COM_I_SETCOUNT
5                   THEN
5                       SIGNAL(
5                           BACKUP$_NOSUCHRVN,
5                           2, .QUAL[QUAL_VOLU_VALUE], .COM_I_SETCOUNT);
5                   CH$MOVE(
5                       HM2$S_STRUCNAME,
5                       BUFFER[HM2$T_STRUCNAME],
5                       COM_I_STRUCNAME);
4                   END;
4               RETURN;
4               END
3           ELSE IF
3               .RVN EQL 1 AND
3               .BUFFER[HM2$B_STRUCLEV] EQL 1 AND
3               .BUFFER[HM1$W_CLUSTER] EQL 1 AND
3               .BUFFER[HM1$L_IBMAPLBN] NEQ 0 AND
3               .BUFFER[HM1$W_MAXFILES] NEQ 0 AND
3               .BUFFER[HM1$W_IBMAPSIZE] NEQ 0 AND
3               CHECKSUM2(.BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM1)) AND
3               CHECKSUM2(.BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM2))
3           THEN
4               BEGIN
4               FAST_STRUCLEV = 1;
4               COM_I_SETCOUNT = 1;
4               IF
4                   .QUAL[QUAL_VOLU] AND
```

```
; 1026          2127  4                     .QUAL[QUAL_VOLU_VALUE] GTRU 1
; 1027          2128  4                 THEN
; 1028          2129  4                     SIGNAL(
; 1029          2130  4                         BACKUP$_NOSUCHRVN,
; 1030          2131  4                         2, .QUAL[QUAL_VOLU_VALUE], .COM_I_SETCOUNT);
; 1031          2132  4                 RETURN;
; 1032          2133  3                 END;
; 1033          2134  2         END;
; 1034          2135  2
; 1035          2136  2
; 1036          2137  2 ! No good home block found.  Report failure.
; 1037          2138  2 !
; 1038          2139  2 SIGNAL(BACKUP$_PROCINDEX, 2, INPUT_QUAL[QUAL_DEV_DESC], .RVN, .OLD_STATUS);
; 1039          2140  1 END;



                                        0FFC 00000 READ_HOMEBLOCK:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11             ; 2014
                      5B 00000000G  00   9E 00002          MOVAB   CHECKSUM2, R11                                  ;
                      5A 00000000'  EF   9E 00009          MOVAB   COM_I_SETCOUNT, R10                             ;
                      5E            08   C2 00010          SUBL2   #8, SP                                          ;
                      59            2C   D0 00013          MOVL    #44, OLD_STATUS                                 ; 2057
                      56       08   AC   D0 00016          MOVL    BUFFER, R6                                      ; 2066
                      57            02   D0 0001A          MOVL    #2, VBN                                         ;
                                    7E   7C 0001D 1$:      CLRQ    -(SP)                                           ;
                                    7E   D4 0001F          CLRL    -(SP)                                           ;
                                    57   DD 00021          PUSHL   VBN                                             ;
                      7E     0200   8F   3C 00023          MOVZWL  #512, -(SP)                                     ;
                                    56   DD 00028          PUSHL   R6                                              ;
                                    7E   7C 0002A          CLRQ    -(SP)                                           ;
                               20   AE   9F 0002C          PUSHAB  IOSB                                            ;
                                    31   DD 0002F          PUSHL   #49                                             ;
                               46   AA   DD 00031          PUSHL   INPUT_CHAN                                      ;
                                    7E   D4 00034          CLRL    -(SP)                                           ;
                   00000000G  00    0C   FB 00036          CALLS   #12, SYS$QIOW                                   ;
                      58            50   D0 0003D          MOVL    R0, STATUS                                      ;
                      06            58   E9 00040          BLBC    STATUS, 2$                                      ; 2067
                      58            6E   3C 00043          MOVZWL  IOSB, STATUS                                    ;
                      06            58   E8 00046          BLBS    STATUS, 3$                                      ; 2069
                      59            58   D0 00049 2$:      MOVL    STATUS, OLD_STATUS                              ; 2071
                                 0111    31 0004C          BRW     12$                                             ;
                      02       0D   A6   91 0004F 3$:      CMPB    13(R6), #2                                      ; 2074
                                    06   12 00053          BNEQ    4$                                              ;
  57        10   A6   10            00   ED 00055          CMPZV   #0, #16, 16(R6), VBN                            ; 2075
                                    03   13 0005B 4$:      BEQL    6$                                              ;
                                 00A1    31 0005D 5$:      BRW     11$                                             ;
                               08   A6   D5 00060 6$:      TSTL    8(R6)                                           ; 2076
                                    F8   13 00063          BEQL    5$                                              ;
                               0E   A6   B5 00065          TSTW    14(R6)                                          ; 2077
                                    F3   13 00068          BEQL    5$                                              ;
                               10   A6   B5 0006A          TSTW    16(R6)                                          ; 2078
                                    EE   13 0006D          BEQL    5$                                              ;
                               12   A6   B5 0006F          TSTW    18(R6)                                          ; 2079
                                    E9   13 00072          BEQL    5$                                              ;
```

```
                                   14   A6  B5 00074           TSTW    20(R6)                                          ; 2080
                                        E4  13 00077           BEQL    5$                                              ;
                                   16   A6  B5 00079           TSTW    22(R6)                                          ; 2081
                                        DF  13 0007C           BEQL    5$                                              ;
                                   18   A6  D5 0007E           TSTL    24(R6)                                          ; 2082
                                        7E  13 00081           BEQL    11$                                             ;
                                   1C   A6  D5 00083           TSTL    28(R6)                                          ; 2083
                                        79  13 00086           BEQL    11$                                             ;
                                   20   A6  B5 00088           TSTW    32(R6)                                          ; 2084
                                        74  13 0008B           BEQL    11$                                             ;
                                   22   A6  B5 0008D           TSTW    34(R6)                                          ; 2085
                                        6F  13 00090           BEQL    11$                                             ;
                                        3A  DD 00092           PUSHL   #58                                             ; 2086
                                        56  DD 00094           PUSHL   R6                                              ;
                          6B            02  FB 00096           CALLS   #2, CHECKSUM2                                   ;
                          65            50  E9 00099           BLBC    R0, 11$                                         ;
                          7E      01FE  8F  3C 0009C           MOVZWL  #510, -(SP)                                     ; 2087
                                        56  DD 000A1           PUSHL   R6                                              ;
                          6B            02  FB 000A3           CALLS   #2, CHECKSUM2                                   ;
                          58            50  E9 000A6           BLBC    R0, 11$                                         ;
                          01       04   AC  D1 000A9           CMPL    RVN, #1                                         ; 2090
                                        01  13 000AD           BEQL    7$                                              ;
                                            04 000AF           RET                                                     ;
                    45    AA            02  90 000B0 7$:       MOVB    #2, FAST_STRUCLEV                               ; 2093
                  00FF    8F       28   A6  B1 000B4           CMPW    40(R6), #255                                    ; 2094
                                        14  1B 000BA           BLEQU   8$                                              ;
           7E       5A    AA            10  C1 000BC           ADDL3   #16, INPUT_QUAL, -(SP)                          ; 2095
                                        01  DD 000C1           PUSHL   #1                                              ;
                              00000000G 8F  DD 000C3           PUSHL   #BACKUP$_MAXVOLS                                ;
             00000000G    00            03  FB 000C9           CALLS   #3, LIB$SIGNAL                                  ;
                          6A       28   A6  90 000D0 8$:       MOVB    40(R6), COM_I_SETCOUNT                          ; 2096
                                        03  12 000D4           BNEQ    9$                                              ; 2097
                          6A            01  90 000D6           MOVB    #1, COM_I_SETCOUNT                              ;
                          1C       90   AA  E9 000D9 9$:       BLBC    QUAL+14, 10$                                    ; 2099
                          6A       D1   AA  91 000DD           CMPB    QUAL+79, COM_I_SETCOUNT                         ; 2100
                                        16  1B 000E1           BLEQU   10$                                             ;
                          7E            6A  9A 000E3           MOVZBL  COM_I_SETCOUNT, -(SP)                           ; 2104
                          7E       D1   AA  9A 000E6           MOVZBL  QUAL+79, -(SP)                                  ;
                                        02  DD 000EA           PUSHL   #2                                              ; 2102
                              00000000G 8F  DD 000EC           PUSHL   #BACKUP$_NOSUCHRVN                              ;
             00000000G    00            04  FB 000F2           CALLS   #4, LIB$SIGNAL                                  ;
     02    AA       01CC  C6            0C  28 000F9 10$:      MOVC3   #12, 460(R6), COM_I_STRUCNAME                   ; 2107
                                            04 00100           RET                                                     ; 2089
                          01       04   AC  D1 00101 11$:      CMPL    RVN, #1                                         ; 2113
                                        59  12 00105           BNEQ    12$                                             ;
                          01       0D   A6  91 00107           CMPB    13(R6), #1                                      ; 2114
                                        53  12 0010B           BNEQ    12$                                             ;
                          01       08   A6  B1 0010D           CMPW    8(R6), #1                                       ; 2115
                                        4D  12 00111           BNEQ    12$                                             ;
                                   02   A6  D5 00113           TSTL    2(R6)                                           ; 2116
                                        48  13 00116           BEQL    12$                                             ;
                                   06   A6  B5 00118           TSTW    6(R6)                                           ; 2117
                                        43  13 0011B           BEQL    12$                                             ;
                                        66  B5 0011D           TSTW    (R6)                                            ; 2118
                                        3F  13 0011F           BEQL    12$                                             ;
                                        3A  DD 00121           PUSHL   #58                                             ; 2119
                                        56  DD 00123           PUSHL   R6                                              ;
```

```
                                6B              02  FB 00125            CALLS   #2, CHECKSUM2                              ;
                                35              50  E9 00128            BLBC    R0, 12$                                    ;
                                7E        01FE  8F  3C 0012B            MOVZWL  #510, -(SP)                                ; 2120
                                                56  DD 00130            PUSHL   R6                                         ;
                                6B              02  FB 00132            CALLS   #2, CHECKSUM2                              ;
                                28              50  E9 00135            BLBC    R0, 12$                                    ;
                          45    AA              01  90 00138            MOVB    #1, FAST_STRUCLEV                          ; 2123
                                6A              01  90 0013C            MOVB    #1, COM_I_SETCOUNT                         ; 2124
                                40          90  AA  E9 0013F            BLBC    QUAL+14, 13$                               ; 2126
                                01          D1  AA  91 00143            CMPB    QUAL+79, #1                                ; 2127
                                                3A  1B 00147            BLEQU   13$                                        ;
                                7E              6A  9A 00149            MOVZBL  COM_I_SETCOUNT, -(SP)                      ; 2131
                                7E          D1  AA  9A 0014C            MOVZBL  QUAL+79, -(SP)                             ;
                                                02  DD 00150            PUSHL   #2                                         ; 2129
                                     00000000G  8F  DD 00152            PUSHL   #BACKUP$_NOSUCHRVN                         ;
                      00000000G  00             04  FB 00158            CALLS   #4, LIB$SIGNAL                             ;
                                                    04 0015F            RET                                                ; 2122
         FEB3            57              01 00000064  8F  F1 00160 12$:  ACBL    #100, #1, VBN, 1$                          ; 2058
                                                59  DD 0016A            PUSHL   OLD_STATUS                                 ; 2139
                                            04  AC  DD 0016C            PUSHL   RVN                                        ;
                         7E          5A     AA  10  C1 0016F            ADDL3   #16, INPUT_QUAL, -(SP)                     ;
                                                02  DD 00174            PUSHL   #2                                         ;
                                     00000000G  8F  DD 00176            PUSHL   #BACKUP$_PROCINDEX                         ;
                      00000000G  00             05  FB 0017C            CALLS   #5, LIB$SIGNAL                             ;
                                                    04 00183 13$:       RET                                                ; 2140

; Routine Size:  388 bytes,    Routine Base:  CODE + 09AF
```

```
1 %SBTTL 'VERIFY_HEADER - verify file header'
1 ROUTINE VERIFY_HEADER(HEADER,FILE_ID)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine determines if the block given it is a valid file header.
1 !
1 ! INPUT PARAMETERS:
1 !       HEADER          - Pointer to header.
1 !       FILE_ID         - Purported file ID.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       0 if invalid file header
1 !       1 if valid file header
1 !       2 if deleted file header
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 MAP
2       HEADER:         REF BBLOCK,     ! file header arg
2       FILE_ID:        REF BBLOCK;     ! file ID arg
2
2
2 ! First check the structure level.
2 !
2 IF .HEADER[FH2$B_STRUCLEV] NEQ .FAST_STRUCLEV
2 THEN
2     RETURN 0;
2
2
2 IF .FAST_STRUCLEV EQL 2
2 THEN
3     BEGIN
3
3     ! Check the area offsets and the retrieval pointer use counts for
3     ! consistency.
3     !
3     IF
3         .HEADER[FH2$B_IDOFFSET] LSSU $BYTEOFFSET (FH2$L_HIGHWATER)/2 OR
3         .HEADER[FH2$B_MPOFFSET] LSSU .HEADER[FH2$B_IDOFFSET] OR
3         .HEADER[FH2$B_ACOFFSET] LSSU .HEADER[FH2$B_MPOFFSET] OR
3         .HEADER[FH2$B_RSOFFSET] LSSU .HEADER[FH2$B_ACOFFSET] OR
3         .HEADER[FH2$B_MAP_INUSE] GTRU .HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET]
```

```
3           THEN
3               RETURN 0;
3
3
3           ! At this point, we have verified that the block at least once was a
3           ! valid file header.
3           !
3           ! Look at the file number in the header. If zero, this is a
3           ! deleted header.
3           !
3           IF
3               .HEADER[FH2$W_FID_NUM] EQL 0 AND
3               .HEADER[FH2$B_FID_NMX] EQL 0
3           THEN
3               RETURN 2;
3
3
3           ! Now compute the header checksum.
3           !
3           IF NOT CHECKSUM(.HEADER)
3           THEN
3               RETURN 2;
3
3
3           ! Check file number and file sequence number.
3           !
3           IF
3               .HEADER[FH2$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
3               .HEADER[FH2$B_FID_NMX] NEQ .FILE_ID[FID$B_NMX] OR
3               .HEADER[FH2$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
3           THEN
3               RETURN 2;
3           END
2       ELSE
3           BEGIN
3           LOCAL
3               MAP_AREA:       REF BBLOCK;
3
3
3           ! Now point to the map area and make sure that the extension
3           ! RVN is zero.  Also check the retrieval pointer format data.
3           !
3           MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
3           IF
3               .MAP_AREA[FM1$B_EX_RVN] NEQ 0 OR
3               .MAP_AREA[FM1$B_COUNTSIZE] NEQ 1 OR
3               .MAP_AREA[FM1$B_LBNSIZE] NEQ 3
3           THEN
3               RETURN 0;
3
3
3           ! Check the retrieval pointer counts for consistency with the
3           ! available space.
3           !
3           IF
3               .MAP_AREA[FM1$B_INUSE] GTRU .MAP_AREA[FM1$B_AVAIL] OR
3               .MAP_AREA[FM1$B_AVAIL] GTRU 255 - (.MAP_AREA + FM1$C_POINTERS - .HEADER) / 2
```

```
; 1155          2255 3          THEN
; 1156          2256 3              RETURN 0;
; 1157          2257 3
; 1158          2258 3
; 1159          2259 3          ! At this point, we have verified that the block at least once was a
; 1160          2260 3          ! valid file header.
; 1161          2261 3          !
; 1162          2262 3          ! Look at the file number in the header. If zero, this is a
; 1163          2263 3          ! deleted header.
; 1164          2264 3          !
; 1165          2265 3          IF .HEADER[FH1$W_FID_NUM] EQL 0
; 1166          2266 3          THEN
; 1167          2267 3              RETURN 2;
; 1168          2268 3
; 1169          2269 3
; 1170          2270 3          ! Now compute the header checksum.
; 1171          2271 3          !
; 1172          2272 3          IF NOT CHECKSUM(.HEADER)
; 1173          2273 3          THEN
; 1174          2274 3              RETURN 2;
; 1175          2275 3
; 1176          2276 3
; 1177          2277 3          ! Check file number and file sequence number.
; 1178          2278 3          !
; 1179          2279 3          IF
; 1180          2280 3              .HEADER[FH1$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
; 1181          2281 3              .HEADER[FH1$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
; 1182          2282 3          THEN
; 1183          2283 3              RETURN 2;
; 1184          2284 2          END;
; 1185          2285 2
; 1186          2286 2
; 1187          2287 2      ! Header is OK.
; 1188          2288 2      !
; 1189          2289 2      RETURN 1;
; 1190          2290 1      END;



                                        003C 00000 VERIFY_HEADER:
                                                                .WORD   Save R2,R3,R4,R5                        ; 2142
                        55 00000000'  EF  9E 00002              MOVAB   FAST_STRUCLEV, R5                       ;
                        54 00000000G  00  9E 00009              MOVAB   CHECKSUM, R4                            ;
                        52        04  AC  D0 00010              MOVL    HEADER, R2                              ; 2180
                        65        07  A2  91 00014              CMPB    7(R2), FAST_STRUCLEV                    ;
                                      75  12 00018              BNEQ    5$                                      ;
                        53        08  AC  D0 0001A              MOVL    FILE_ID, R3                             ; 2225
                        02            65  91 0001E              CMPB    FAST_STRUCLEV, #2                       ; 2185
                                      55  12 00021              BNEQ    4$                                      ;
                        26            62  91 00023              CMPB    (R2), #38                               ; 2193
                                      12  1F 00026              BLSSU   1$                                      ;
                        62        01  A2  91 00028              CMPB    1(R2), (R2)                             ; 2194
                                      0C  1F 0002C              BLSSU   1$                                      ;
                01      A2        02  A2  91 0002E              CMPB    2(R2), 1(R2)                            ; 2195
                                      05  1F 00033              BLSSU   1$                                      ;
```

```
                         02   A2     03   A2  91 00035      CMPB    3(R2), 2(R2)                    ; 2196
                                          03  1E 0003A 1$:  BGEQU   2$                              ;
                                        0092  31 0003C      BRW     9$                              ;
                              51     02   A2  9A 0003F 2$:  MOVZBL  2(R2), R1                       ; 2197
                              50     01   A2  9A 00043      MOVZBL  1(R2), R0                       ;
                              51          50  C2 00047      SUBL2   R0, R1                          ;
      51       3A    A2       08          00  ED 0004A      CMPZV   #0, #8, 58(R2), R1              ;
                                          7F  1A 00050      BGTRU   9$                              ;
                                     08   A2  B5 00052      TSTW    8(R2)                           ; 2209
                                          05  12 00055      BNEQ    3$                              ;
                                     0D   A2  95 00057      TSTB    13(R2)                          ; 2210
                                          6D  13 0005A      BEQL    7$                              ;
                                          52  DD 0005C 3$:  PUSHL   R2                              ; 2217
                              64          01  FB 0005E      CALLS   #1, CHECKSUM                    ;
                              65          50  E9 00061      BLBC    R0, 7$                          ;
                              63     08   A2  B1 00064      CMPW    8(R2), (R3)                     ; 2225
                                          5F  12 00068      BNEQ    7$                              ;
                         05   A3     0D   A2  91 0006A      CMPB    13(R2), 5(R3)                   ; 2226
                                          58  12 0006F      BNEQ    7$                              ;
                         02   A3     0A   A2  B1 00071      CMPW    10(R2), 2(R3)                   ; 2227
                                          4F  11 00076      BRB     6$                              ;
                              50     01   A2  9A 00078 4$:  MOVZBL  1(R2), R0                       ; 2240
                              50        6240  3E 0007C      MOVAW   (R2)[R0], MAP_AREA              ;
                                     01   A0  95 00080      TSTB    1(MAP_AREA)                     ; 2242
                                          4C  12 00083      BNEQ    9$                              ;
                              01     06   A0  91 00085      CMPB    6(MAP_AREA), #1                 ; 2243
                                          46  12 00089      BNEQ    9$                              ;
                              03     07   A0  91 0008B      CMPB    7(MAP_AREA), #3                 ; 2244
                                          40  12 0008F 5$:  BNEQ    9$                              ;
                         09   A0     08   A0  91 00091      CMPB    8(MAP_AREA), 9(MAP_AREA)        ; 2253
                                          39  1A 00096      BGTRU   9$                              ;
                    51        52          50  C3 00098      SUBL3   MAP_AREA, R2, R1                ; 2254
                              51          0A  C2 0009C      SUBL2   #10, R1                         ;
                              51          02  C6 0009F      DIVL2   #2, R1                          ;
                              51   00FF   C1  9E 000A2      MOVAB   255(R1), R1                     ;
      51       09    A0       08          00  ED 000A7      CMPZV   #0, #8, 9(MAP_AREA), R1         ;
                                          22  1A 000AD      BGTRU   9$                              ;
                                     02   A2  B5 000AF      TSTW    2(R2)                           ; 2265
                                          15  13 000B2      BEQL    7$                              ;
                                          52  DD 000B4      PUSHL   R2                              ; 2272
                              64          01  FB 000B6      CALLS   #1, CHECKSUM                    ;
                              0D          50  E9 000B9      BLBC    R0, 7$                          ;
                              63     02   A2  B1 000BC      CMPW    2(R2), (R3)                     ; 2280
                                          07  12 000C0      BNEQ    7$                              ;
                         02   A3     04   A2  B1 000C2      CMPW    4(R2), 2(R3)                    ; 2281
                                          04  13 000C7 6$:  BEQL    8$                              ;
                              50          02  D0 000C9 7$:  MOVL    #2, R0                          ; 2283
                                              04 000CC      RET                                     ;
                              50          01  D0 000CD 8$:  MOVL    #1, R0                          ; 2289
                                              04 000D0      RET                                     ;
                                          50  D4 000D1 9$:  CLRL    R0                              ; 2290
                                              04 000D3      RET                                     ;

; Routine Size:  212 bytes,    Routine Base:  CODE + 0B33
```

```
1 %SBTTL 'PROCESS_FILE - process selected file'
1 ROUTINE PROCESS_FILE: NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called when the next file that matches the selection
1 !       file specification has been located.  It completes the tests that
1 !       select files to be processed, and if these are passed, processes the
1 !       file.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       INPUT_NAM         - Contains resultant string and file ID.
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       File processed if appropriate.
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       FILE_NUMBER,                                    ! Clean file number
2       RVN;                                            ! Clean RVN
2
2
2 ! Get clean file ID.
2 !
2 FILE_NUMBER = .INPUT_NAM[NAM$W_FID_NUM];
2 FILE_NUMBER<16,8> = .INPUT_NAM[NAM$B_FID_NMX];
2 RVN = .INPUT_NAM[NAM$B_FID_RVN];
2
2
2 IF .QUAL[QUAL_FAST]
2 THEN
3     BEGIN
3
3     ! First, make sure the RVN is in range.  Then, make sure the file number
3     ! is in range.
3     !
3     IF .RVN GTRU .COM_I_SETCOUNT THEN RETURN;
3     IF .FILE_NUMBER GTRU .FAST_IMAP_SIZE[.RVN-1]*4096 THEN RETURN;
3
3
3     ! See if file is selected.
3     !
```

```
; 1249          2348 3       IF .QUAL[QUAL_IMAG]
; 1250          2349 3       THEN
; 1251          2350 4           BEGIN
; 1252          2351 4           IF NOT .BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1]
; 1253          2352 4           THEN
; 1254          2353 4               RETURN;
; 1255          2354 4           END
; 1256          2355 3       ELSE
; 1257          2356 4           BEGIN
; 1258          2357 4           IF
; 1259          2358 4               NOT .BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1] AND
; 1260          2359 5               NOT (.DIR_STATUS[D_STAT_SCANNED] AND .QUAL[QUAL_OSAV])
; 1261          2360 4           THEN
; 1262          2361 4               RETURN;
; 1263          2362 3           END;
; 1264          2363 2       END;
; 1265          2364 2
; 1266          2365 2
; 1267          2366 2   ! Finish evaluating selection criteria.
; 1268          2367 2   !
; 1269          2368 2   IF
; 1270          2369 2       NOT .QUAL[QUAL_IMAG] AND
; 1271          2370 3       NOT (.DIR_STATUS[D_STAT_SCANNED] AND .QUAL[QUAL_OSAV])
; 1272          2371 2   THEN
; 1273          2372 2       IF NOT SELECT_INPUT_FILE(%B'010') THEN RETURN;
; 1274          2373 2
; 1275          2374 2
; 1276          2375 2   ! File is selected.  Process it.  If successfully processed in image mode,
; 1277          2376 2   ! clear the bitmap bit to avoid processing its synonyms.
; 1278          2377 2   !
; 1279          2378 2   IF SAVE_ONE_FILE()
; 1280          2379 2   THEN
; 1281          2380 2       IF .QUAL[QUAL_IMAG]
; 1282          2381 2       THEN
; 1283          2382 2           BITVECTOR[.FAST_IMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
; 1284          2383 1   END;



                                                  001C 00000 PROCESS_FILE:
                                                                   .WORD    Save R2,R3,R4                          ; 2292
                                54 00000000'   EF   9E 00002       MOVAB    QUAL+8, R4                             ;
                                50     00C8    C4   D0 00009       MOVL     INPUT_NAM, R0                          ; 2330
                                52       24    A0   3C 0000E       MOVZWL   36(R0), FILE_NUMBER                    ;
      52            08          10       29    A0   F0 00012       INSV     41(R0), #16, #8, FILE_NUMBER           ; 2331
                                53       28    A0   9A 00018       MOVZBL   40(R0), RVN                            ; 2332
                    3D     01   A4             06   E1 0001C       BBC      #6, QUAL+9, 2$                         ; 2335
      53     76     A4          08             00   ED 00021       CMPZV    #0, #8, COM_I_SETCOUNT, RVN            ; 2342
                                               6D   1F 00027       BLSSU    5$                                     ;
                                50     0234  D443   DE 00029       MOVAL    @FAST_IMAP_SIZE[RVN], R0               ; 2343
                    50     FC   A0             0C   78 0002F       ASHL     #12, -4(R0), R0                        ;
                                50             52   D1 00034       CMPL     FILE_NUMBER, R0                        ;
                                               5D   1A 00037       BGTRU    5$                                     ;
                                50     0238  D443   DE 00039       MOVAL    @FAST_IMAP[RVN], R0                    ; 2351
                                51       FF    A2   9E 0003F       MOVAB    -1(R2), R1                             ;
                    06          02   A4          03  E1 00043         BBC     #3, QUAL+10, 1$                       ; 2348
                    11          FC   B0          51  E0 00048         BBS     R1, @-4(R0), 2$                       ; 2351
                                                     04 0004D         RET                                           ; 2353
                    0B          FC   B0          51  E0 0004E 1$:     BBS     R1, @-4(R0), 2$                       ; 2358
                    3D        0297   C4          02  E1 00053         BBC     #2, DIR_STATUS, 5$                    ; 2359
                                           07    A4  95 00059         TSTB    QUAL+15                               ;
                                                 38  18 0005C         BGEQ    5$                                    ;
                    17          02   A4          03  E0 0005E 2$:     BBS     #3, QUAL+10, 4$                       ; 2369
                    05        0297   C4          02  E1 00063         BBC     #2, DIR_STATUS, 3$                    ; 2370
                                           07    A4  95 00069         TSTB    QUAL+15                               ;
                                                 0C  19 0006C         BLSS    4$                                    ;
                                                 02  DD 0006E 3$:     PUSHL   #2                                    ; 2372
                         00000000G   00          01  FB 00070         CALLS   #1, SELECT_INPUT_FILE                 ;
                                     1C          50  E9 00077         BLBC    R0, 5$                                ;
                         00000000G   00          00  FB 0007A 4$:     CALLS   #0, SAVE_ONE_FILE                     ; 2378
                                     12          50  E9 00081         BLBC    R0, 5$                                ;
                    0D          02   A4          03  E1 00084         BBC     #3, QUAL+10, 5$                       ; 2380
                                     50    0238 D443 DE 00089         MOVAL   @FAST_IMAP[RVN], R0                   ; 2382
                                                 52  D7 0008F         DECL    R2                                    ;
                    00          FC   B0          52  E5 00091         BBCC    R2, @-4(R0), 5$                       ;
                                                     04 00096 5$:     RET                                           ; 2383

; Routine Size:  151 bytes,    Routine Base:  CODE + 0C07
; 1286          2384  1 %SBTTL 'DIR_SCAN - scan a directory'
; 1287          2385  1 ROUTINE DIR_SCAN(RVN): NOVALUE=
; 1288          2386  1
; 1289          2387  1 !++
; 1290          2388  1 !
; 1291          2389  1 ! FUNCTIONAL DESCRIPTION:
; 1292          2390  1 !       This routine is the driver for a directory scan.
; 1293          2391  1 !
; 1294          2392  1 ! INPUT PARAMETERS:
; 1295          2393  1 !       RVN             - Relative volume number.
; 1296          2394  1 !
; 1297          2395  1 ! IMPLICIT INPUTS:
; 1298          2396  1 !       NONE
; 1299          2397  1 !
; 1300          2398  1 ! OUTPUT PARAMETERS:
; 1301          2399  1 !       NONE
; 1302          2400  1 !
; 1303          2401  1 ! IMPLICIT OUTPUTS:
; 1304          2402  1 !       NONE
; 1305          2403  1 !
; 1306          2404  1 ! ROUTINE VALUE:
; 1307          2405  1 !       NONE
; 1308          2406  1 !
; 1309          2407  1 ! SIDE EFFECTS:
; 1310          2408  1 !       NONE
; 1311          2409  1 !
; 1312          2410  1 !--
; 1313          2411  1
; 1314          2412  2 BEGIN
; 1315          2413  2 FAST_RVN = .RVN;
; 1316          2414  2 INIT_DIR_SCAN(
; 1317          2415  2     .INPUT_CHAN,
; 1318          2416  2     .INPUT_NAM,
; 1319          2417  2     INPUT_QUAL[QUAL_DEV_DESC],
; 1320          2418  2     INPUT_QUAL[QUAL_EXP_DESC],
; 1321          2419  2     .QUAL[QUAL_IMAG] + .QUAL[QUAL_OSAV]^2,
; 1322          2420  2     .RVN,
; 1323          2421  2     0);
; 1324          2422  2 WHILE FIND_NEXT() DO PROCESS_FILE();
; 1325          2423  2 FREE_DIR_DATA();
; 1326          2424  1 END;



                                                0004 00000 DIR_SCAN:
                                                                .WORD   Save R2                                 ; 2385
                        52 00000000'    EF   9E 00002           MOVAB   INPUT_QUAL, R2                          ;
              0160  C2           04     AC   90 00009           MOVB    RVN, FAST_RVN                           ; 2413
                                        7E   D4 0000F           CLRL    -(SP)                                   ; 2418
                                 04     AC   DD 00011           PUSHL   RVN                                     ; 2420
   51      FF32     C2          01      03   EF 00014           EXTZV   #3, #1, QUAL+10, R1                     ; 2419
   50      FF37     C2          01      07   EF 0001B           EXTZV   #7, #1, QUAL+15, R0                     ;
                                      6140   DF 00022           PUSHAL  (R1)[R0]                                ;
                    7E          62      08   C1 00025           ADDL3   #8, INPUT_QUAL, -(SP)                   ; 2418
                    7E          62      10   C1 00029           ADDL3   #16, INPUT_QUAL, -(SP)                  ; 2417
```

```
                                      F8   A2  DD 0002D          PUSHL   INPUT_NAM                         ; 2418
                                      EC   A2  DD 00030          PUSHL   INPUT_CHAN                        ;
                      0000V  CF            07  FB 00033          CALLS   #7, INIT_DIR_SCAN                 ;
                      0000V  CF            00  FB 00038 1$:      CALLS   #0, FIND_NEXT                     ; 2422
                             07            50  E9 0003D          BLBC    R0, 2$                            ;
                      FF24   CF            00  FB 00040          CALLS   #0, PROCESS_FILE                  ;
                                           F1  11 00045          BRB     1$                                ;
                      0000V  CF            00  FB 00047 2$:      CALLS   #0, FREE_DIR_DATA                 ; 2423
                                               04 0004C          RET                                       ; 2424

; Routine Size:  77 bytes,    Routine Base:  CODE + 0C9E
```

```
1 %SBTTL 'INIT_DIR_SCAN - initialize directory scan'
1 GLOBAL ROUTINE INIT_DIR_SCAN(CHAN,NAM,DEV_DESC,SEL_DESC,FLAGS,RVN,LIMIT): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine initializes context for a directory scan.
1 !
1 ! INPUT PARAMETERS:
1 !       CHANNEL         - Channel assigned to volume set.
1 !       NAM             - Pointer to name block.
1 !       DEV_DESC        - Pointer to device name descriptor.
1 !       SEL_DESC        - Pointer to selection filespec descriptor.
1 !       FLAGS           - Bit 0 true to request an image scan.
1 !                         Bit 1 true to request immediate return on terminator.
1 !                         Bit 2 true to request return of scanned directories.
1 !       RVN             - Relative volume number.
1 !       LIMIT           - Pointer to vector of ODS-1 scan limits or 0.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 MAP
2       DEV_DESC        : REF VECTOR,   ! Device name descriptor
2       NAM             : REF BBLOCK;   ! Pointer to name block
2 LOCAL
2       STATUS,                         ! General status value
2       LOCAL_FAB       : $FAB_DECL,    ! FAB for $PARSE
2       LOCAL_NAM       : $NAM_DECL;    ! NAM for $PARSE
2
2
2 ! Initialize the impure area.
2 !
2 CH$FILL(0, DIR_END-DIR_BEG, DIR_BEG);
2 DIR_FLAGS[D_INITIAL] = TRUE;
2
2
2 ! Save the parameters.
2 !
2 DIR_CHAN = .CHAN;
2 DIR_NAM = .NAM;
2 DIR_NAM[NAM$W_DID_NUM] = FID$C_MFD;
```

```
       2 DIR_NAM[NAM$W_DID_SEQ] = FID$C_MFD;
       2 DIR_NAM[NAM$W_DID_RVN] = .RVN;
       2 DIR_DEV_DESC = .DEV_DESC;
       2
       2 ! Determine if a rooted directory is being used. If so, do another
       2 ! parse to get the root directory ID.
       2 !
       2
       2 IF .NAM[NAM$V_ROOT_DIR]
       2 THEN
       3     BEGIN
     P 3     $FAB_INIT (FAB = LOCAL_FAB,
     P 3                NAM = LOCAL_NAM,
     P 3                FNS = .DEV_DESC[0],
     P 3                FNA = .DEV_DESC[1],
     P 3                DNM = '[000000]'
       3                );
     P 3     $NAM_INIT (NAM = LOCAL_NAM,
     P 3                NOP = NOCONCEAL
       3                );
       3     STATUS = $PARSE (FAB = LOCAL_FAB);
       3     IF NOT .STATUS
       3     THEN SIGNAL (BACKUP$_OPENIN+STS$K_SEVERE, 1, .DEV_DESC, .STATUS, .LOCAL_FAB[FAB$L_STV]);
       3
       3     DIR_NAM[NAM$W_FID_NUM] = .LOCAL_NAM[NAM$W_DID_NUM];
       3     DIR_NAM[NAM$W_FID_SEQ] = .LOCAL_NAM[NAM$W_DID_SEQ];
       3     DIR_NAM[NAM$W_FID_RVN] = .LOCAL_NAM[NAM$W_DID_RVN];
       3     IF .DIR_NAM[NAM$B_FID_RVN] EQL 0 THEN DIR_NAM[NAM$B_FID_RVN] = .RVN;
       3     END
       3
       3
       3 ! If no root directory is used, establish the MFD as the root.
       3 !
       2 ELSE
       3     BEGIN
       3     DIR_NAM[NAM$W_FID_NUM] = FID$C_MFD;
       3     DIR_NAM[NAM$W_FID_SEQ] = FID$C_MFD;
       3     DIR_NAM[NAM$W_FID_RVN] = .RVN;
       2     END;
       2
       2 ! Initialize the result string in the name block with the name of
       2 ! the MFD. Note that in the case of a rooted directory, we are lying
       2 ! about its name. However, this is consistent with RMS behavior and
       2 ! is generally all to the best.
       2 !
       2
       2 CH$COPY (.DEV_DESC[0], .DEV_DESC[1],
       2           %CHARCOUNT ('[000000]000000.DIR;1'),
       2           UPLIT BYTE ('[000000]000000.DIR;1'),
       2           ' ',
       2           .NAM[NAM$B_RSS],
       2           .NAM[NAM$L_RSA]
       2           );
       2 NAM[NAM$B_RSL] = MINU (.NAM[NAM$B_RSS], .DEV_DESC[0] + %CHARCOUNT ('[000000]000000.DIR;1'));
       2 INIT_NAMEBLOCK (.NAM);
       2
       2 ! Initialize the level stack with the root of the selected RVN.
```

```
; 1442          2539  2 !
; 1443          2540  2 DIR_SP = DIR_STACK;
; 1444          2541  2 DIR_LEVELS = 1;
; 1445          2542  2 BBLOCK[DIR_STACK[D_FID], FID$W_NUM] = .DIR_NAM[NAM$W_FID_NUM];
; 1446          2543  2 BBLOCK[DIR_STACK[D_FID], FID$W_SEQ] = .DIR_NAM[NAM$W_FID_SEQ];
; 1447          2544  2 BBLOCK[DIR_STACK[D_FID], FID$W_RVN] = .DIR_NAM[NAM$W_FID_RVN];
; 1448          2545  2 DIR_STACK[D_VBN] = 1;
; 1449          2546  2
; 1450          2547  2 ! Establish the selection file specification.  This will also set the
; 1451          2548  2 ! terminator and status bits for the root level.
; 1452          2549  2 !
; 1453          2550  2 RESET_DIR_SPEC(.SEL_DESC, .FLAGS);
; 1454          2551  2 IF .LIMIT NEQ 0 THEN CH$MOVE(D_K_NLEVELS*%UPVAL, .LIMIT, DIR_SCANLIMIT);
; 1455          2552  2
; 1456          2553  2 ! Special cases for root directories:
; 1457          2554  2 !       the root directory is not saved initially.
; 1458          2555  2 ! If the directory pointed to by DIR_SEL_DIR is "000000" then
; 1459          2556  2 ! we want to save the MFD only.
; 1460          2557  2 !
; 1461          2558  2
; 1462          2559  2 IF .NAM[NAM$V_ROOT_DIR]
; 1463          2560  2 THEN
; 1464          2561  3     BEGIN
; 1465          2562  3     IF
; 1466          2563  3     CH$EQL ( %CHARCOUNT ('000000'),
; 1467          2564  3              UPLIT BYTE ('000000'),
; 1468          2565  3              .DIR_SEL_DIR [DSC$W_LENGTH],
; 1469          2566  3              .DIR_SEL_DIR [DSC$A_POINTER],
; 1470          2567  3             %C' ')
; 1471          2568  3         THEN DIR_FLAGS [D_ROOT_MFD] = TRUE ;
; 1472          2569  3     DIR_FLAGS[D_INITIAL] = FALSE;
; 1473          2570  2     END;
; 1474          2571  2
; 1475          2572  1 END;


                            5D  30  30  30  30  30  30  5B   00CEB P.AAD:   .ASCII  \[000000]\                              ;
2E  30  30  30  30  30  30  5D  30  30  30  30  30  30  5B   00CF3 P.AAE:   .ASCII  \[000000]000000.DIR;1\                  ;
                                        31  3B  52  49  44   00D02                                                          ;
                                30  30  30  30  30  30       00D07 P.AAF:   .ASCII  \000000\                                ;

                                                                            .EXTRN  SYS$PARSE

                                                        07FC 00000          .ENTRY  INIT_DIR_SCAN, Save R2,R3,R4,R5,R6,R7,R8,-  ; 2426
                                                                                    R9,R10                                      ;
                            5A 00000000'      EF    9E 00002                MOVAB   DIR_NAM, R10                                ;
                            5E        FF50    CE    9E 00009                MOVAB   -176(SP), SP                                ;
   03CC    8F          00   6E                00    2C 0000E                MOVC5   #0, (SP), #0, #972, DIR_BEG                 ; 2473
                                        FC    AA       00015                                                                ;
                        1A  AA                10    88 00017                BISB2   #16, DIR_FLAGS                              ; 2474
                        FC  AA            04  AC    D0 0001B                MOVL    CHAN, DIR_CHAN                              ; 2479
                            58            08  AC    D0 00020                MOVL    NAM, R8                                     ; 2480
                            6A                58    D0 00024                MOVL    R8, DIR_NAM                                 ;
                            56                6A    D0 00027                MOVL    DIR_NAM, R6                                 ; 2481
                        2A  A6 00040004       8F    D0 0002A                MOVL    #262148, 42(R6)                             ;
                        2E  A6            18  AC    B0 00032                MOVW    RVN, 46(R6)                                 ; 2483
```

```
                                57        0C   AC   D0 00037        MOVL    DEV_DESC, R7                                 ; 2484
                          04    AA             57   D0 0003B        MOVL    R7, DIR_DEV_DESC                             ;
                    7B    35    A8             05   E1 0003F        BBC     #5, 53(R8), 2$                               ; 2490
 0050   8F          00          6E             00   2C 00044        MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 2498
                                          60   AE      0004B                                                         ;
                          60    AE      5003   8F   B0 0004D        MOVW    #20483, $RMS_PTR                             ;
                          76    AE             02   90 00053        MOVB    #2, $RMS_PTR+22                              ;
                          7F    AE             02   90 00057        MOVB    #2, $RMS_PTR+31                              ;
                          D8    AD             6E   9E 0005B        MOVAB   LOCAL_NAM, $RMS_PTR+40                       ;
                          DC    AD        04   A7   D0 0005F        MOVL    4(R7), $RMS_PTR+44                           ;
                          E0    AD      FF76   CF   9E 00064        MOVAB   P.AAD, $RMS_PTR+48                           ;
                          E4    AD             67   90 0006A        MOVB    (R7), $RMS_PTR+52                            ;
                          E5    AD             08   90 0006E        MOVB    #8, $RMS_PTR+53                              ;
 0060   8F          00          6E             00   2C 00072        MOVC5   #0, (SP), #0, #96, $RMS_PTR                  ; 2501
                                               6E      00079                                                         ;
                          6E          6002     8F   B0 0007A        MOVW    #24578, $RMS_PTR                             ;
                          08    AE             10   90 0007F        MOVB    #16, $RMS_PTR+8                              ;
                                          60   AE   9F 00083        PUSHAB  LOCAL_FAB                                    ; 2502
                   00000000G    00             01   FB 00086        CALLS   #1, SYS$PARSE                                ;
                                16             50   E8 0008D        BLBS    STATUS, 1$                                   ; 2503
                                          6C   AE   DD 00090        PUSHL   LOCAL_FAB+12                                 ; 2504
                                               50   DD 00093        PUSHL   STATUS                                       ;
                                               57   DD 00095        PUSHL   R7                                           ;
                                               01   DD 00097        PUSHL   #1                                           ;
                                   00000000G   8F   DD 00099        PUSHL   #BACKUP$_OPENIN+4                            ;
                   00000000G    00             05   FB 0009F        CALLS   #5, LIB$SIGNAL                               ;
                                50             6A   D0 000A6 1$:    MOVL    DIR_NAM, R0                                  ; 2506
                          24    A0        2A   AE   D0 000A9        MOVL    LOCAL_NAM+42, 36(R0)                         ;
                          28    A0        2E   AE   B0 000AE        MOVW    LOCAL_NAM+46, 40(R0)                         ; 2508
                                          28   A0   95 000B3        TSTB    40(R0)                                       ; 2509
                                               14   12 000B6        BNEQ    3$                                           ;
                          28    A0        18   AC   90 000B8        MOVB    RVN, 40(R0)                                  ;
                                               0D   11 000BD        BRB     3$                                           ; 2490
                          24    A6  00040004   8F   D0 000BF 2$:    MOVL    #262148, 36(R6)                              ; 2517
                          28    A6        18   AC   B0 000C7        MOVW    RVN, 40(R6)                                  ; 2519
                                59        02   A8   9A 000CC 3$:    MOVZBL  2(R8), R9                                    ; 2532
                                56        04   A8   D0 000D0        MOVL    4(R8), R6                                    ; 2533
        59          20    04    B7             67   2C 000D4        MOVC5   (R7), @4(R7), #32, R9, (R6)                  ;
                                               66      000DA                                                         ;
                                               0E   18 000DB        BGEQ    4$                                           ;
                                56             67   C0 000DD        ADDL2   (R7), R6                                     ;
                                59             67   C2 000E0        SUBL2   (R7), R9                                     ;
        59          20  FEFE    CF             14   2C 000E3        MOVC5   #20, P.AAE, #32, R9, (R6)                    ;
                                               66      000EA                                                         ;
                    51          67             14   C1 000EB 4$:    ADDL3   #20, (R7), R1                                ; 2535
                                50        02   A8   9A 000EF        MOVZBL  2(R8), R0                                    ;
                                51             50   D1 000F3        CMPL    R0, R1                                       ;
                                               03   1B 000F6        BLEQU   5$                                           ;
                                50             51   D0 000F8        MOVL    R1, R0                                       ;
                          03    A8             50   90 000FB 5$:    MOVB    R0, 3(R8)                                    ;
                                               58   DD 000FF        PUSHL   R8                                           ; 2536
                   00000000G    00             01   FB 00101        CALLS   #1, INIT_NAMEBLOCK                           ;
                        03C0    CA      015C   CA   9E 00108        MOVAB   DIR_STACK, DIR_SP                            ; 2540
                          19    AA             01   90 0010F        MOVB    #1, DIR_LEVELS                               ; 2541
                                50             6A   D0 00113        MOVL    DIR_NAM, R0                                  ; 2542
                        0178    CA        24   A0   D0 00116        MOVL    36(R0), DIR_STACK+28                         ;
                        017C    CA        28   A0   B0 0011C        MOVW    40(R0), DIR_STACK+32                         ; 2544
                    0170  CA         01  D0 00122          MOVL    #1, DIR_STACK+20                          ; 2545
                          7E    10   AC  7D 00127          MOVQ    SEL_DESC, -(SP)                           ; 2550
                    0000V CF         02  FB 0012B          CALLS   #2, RESET_DIR_SPEC                        ;
                                1C   AC  D5 00130          TSTL    LIMIT                                     ; 2551
                                     07  13 00133          BEQL    6$                                        ;
           03C8  CA     1C  BC       24  28 00135          MOVC3   #36, @LIMIT, DIR_SCANLIMIT                ;
                 14     35  A8       05  E1 0013C 6$:      BBC     #5, 53(R8), 8$                            ; 2559
    08   AA      20   FEB4  CF       06  2D 00141          CMPC5   #6, P.AAF, #32, DIR_SEL_DIR, @DIR_SEL_DIR+4 ; 2563
                                0C   BA     00149                                                    ;
                                     04  12 0014B          BNEQ    7$                                        ;
                        1A  AA       20  88 0014D          BISB2   #32, DIR_FLAGS                            ; 2568
                        1A  AA       10  8A 00151 7$:      BICB2   #16, DIR_FLAGS                            ; 2569
                                         04 00155 8$:      RET                                               ; 2572

; Routine Size:  342 bytes,    Routine Base:  CODE + 0D0D
```

```
1 %SBTTL 'RESET_DIR_SPEC - reinitialize directory context'
1 GLOBAL ROUTINE RESET_DIR_SPEC(SEL_DESC,FLAGS): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine reinitializes context to change the selection file spec.
1 !
1 ! INPUT PARAMETERS:
1 !       SEL_DESC        - Pointer to selection filespec descriptor.
1 !       FLAGS           - Bit 0 true to request an image scan.
1 !                         Bit 1 true to request immediate return on terminator.
1 !                         Bit 2 true to request return of scanned directories.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 INIT_SEL_INFO(.SEL_DESC, DIR_SEL_DIR, DIR_SEL_NTV, DIR_SEL_LATEST);
2 DIR_FLAGS[D_IMAGE_SCAN] = .FLAGS<0,1>;
2 DIR_FLAGS[D_HARD_STOP] = .FLAGS<1,1>;
2
2 DIR_FLAGS[D_SCANNED_DIRS] = FALSE;
2 IF
2     NOT .QUAL[QUAL_INTE] AND
2     .DIR_SEL_NTV[DSC$W_LENGTH] EQL 5 AND
2     CH$EQL(5, .DIR_SEL_NTV[DSC$A_POINTER], 5, UPLIT BYTE ('*.*;*'))
2 THEN
2     DIR_FLAGS[D_SCANNED_DIRS] = .FLAGS<2,1>;
2
2 IF .DIR_FLAGS[D_IMAGE_SCAN] THEN DIR_SEL_LATEST = +1;
2 CH$FILL(-1, D_K_NLEVELS*%UPVAL, DIR_SCANLIMIT);
2
2
2 INCRA D FROM DIR_STACK TO .DIR_SP BY D_S_ENTRY DO
3     BEGIN
3     MAP
3         D:              REF BBLOCK;      ! Pointer to stack entry
3     LOCAL
3         STATUS,                          ! Status return
3         DESC:           VECTOR[2];       ! Descriptor for directory string
3
```

```
; 1534          2630  3        ! Establish the descriptor for the directory string at the current level.
; 1535          2631  3        !
; 1536          2632  3        IF .D EQLA DIR_STACK
; 1537          2633  3        THEN
; 1538          2634  4            BEGIN
; 1539          2635  4            DESC[0] = %CHARCOUNT('000000');
; 1540          2636  4            DESC[1] = UPLIT BYTE('000000');
; 1541          2637  4            END
; 1542          2638  3        ELSE
; 1543          2639  4            BEGIN
; 1544          2640  4            IF .D EQLA .DIR_SP
; 1545          2641  4                THEN DESC[0] = .DIR_STRING[0]
; 1546          2642  4                ELSE DESC[0] = .BBLOCK[.D + D_S_ENTRY, D_SAV_LEN];
; 1547          2643  4            DESC[1] = DIR_STRING[1];
; 1548          2644  3            END;
; 1549          2645  3
; 1550          2646  3
; 1551          2647  3        ! Allocate the dynamic areas for this level.
; 1552          2648  3        !
; 1553          2649  3        BBLOCK[D[D_TERM_DESC], DSC$A_POINTER] = GET_VM(DIR$S_NAME);
; 1554          2650  3        BBLOCK[D[D_NAME_DESC], DSC$A_POINTER] = GET_VM(DIR$S_NAME);
; 1555          2651  3
; 1556          2652  3
; 1557          2653  3        ! Establish the match bit and the terminator specification for this level.
; 1558          2654  3        !
; 1559          2655  3        STATUS = MATCH_DIRECTORY(
; 1560          2656  3            DESC,
; 1561          2657  3            DIR_SEL_DIR,
; 1562          2658  3            D[D_TERM_DESC],
; 1563          2659  3            D[D_TERM_VER],
; 1564          2660  3            DIR_SEL_NTV);
; 1565          2661  3        IF .STATUS<0,1> THEN D[D_DIR_MATCHES] = TRUE;
; 1566          2662  3        IF .STATUS<2,1> THEN D[D_WILD_TERM] = TRUE;
; 1567          2663  3        IF .STATUS<3,1> THEN D[D_NON_TERM] = TRUE;
; 1568          2664  3
; 1569          2665  2        END;
; 1570          2666  1 END;


                              2A  3B  2A  2E  2A  00E63 P.AAG:  .ASCII   \*.*;*\                                       ;
                          30  30  30  30  30  30  00E68 P.AAH:  .ASCII   \000000\                                      ;


                                           00FC 00000           .ENTRY   RESET_DIR_SPEC, Save R2,R3,R4,R5,R6,R7        ; 2574
                          57 00000000G  00   9E 00002           MOVAB    GET_VM, R7                                    ;
                          56 00000000'  EF   9E 00009           MOVAB    DIR_FLAGS, R6                                 ;
                          5E            08   C2 00010           SUBL2    #8, SP                                        ;
                                  03AA  C6   9F 00013           PUSHAB   DIR_SEL_LATEST                                ; 2605
                                    F6  A6   9F 00017           PUSHAB   DIR_SEL_NTV                                   ;
                                    EE  A6   9F 0001A           PUSHAB   DIR_SEL_DIR                                   ;
                                    04  AC   DD 0001D           PUSHL    SEL_DESC                                      ;
                    00000000G  00       04   FB 00020           CALLS    #4, INIT_SEL_INFO                             ;
    66          01             00   08  AC   F0 00027           INSV     FLAGS, #0, #1, DIR_FLAGS                      ; 2606
    50     08   AC             01       01   EF 0002D           EXTZV    #1, #1, FLAGS, R0                             ; 2607
    66          01             02       50   F0 00033           INSV     R0, #2, #1, DIR_FLAGS                         ;
```

```
                                        66            08 8A 00038         BICB2   #8, DIR_FLAGS                       ; 2609
                          19      FD70  C6            01 E0 0003B         BBS     #1, QUAL+14, 1$                     ; 2611
                                        05        F6  A6 B1 00041         CMPW    DIR_SEL_NTV, #5                     ; 2612
                                                      13 12 00045         BNEQ    1$
                    A8    AF        FA  B6            05 29 00047         CMPC3   #5, @DIR_SEL_NTV+4, P.AAG           ; 2613
                                                      0B 12 0004D         BNEQ    1$
            50      08    AC            01            02 EF 0004F         EXTZV   #2, #1, FLAGS, R0                   ; 2615
            66            01            03            50 F0 00055         INSV    R0, #3, #1, DIR_FLAGS
                                        05            66 E9 0005A 1$:     BLBC    DIR_FLAGS, 2$                       ; 2617
                                  03AA  C6            01 D0 0005D         MOVL    #1, DIR_SEL_LATEST
            24      FF    8F            6E            00 2C 00062 2$:     MOVC5   #0, (SP), #-1, #36, DIR_SCANLIMIT   ; 2618
                                                03AE  C6    00068
                                        52      0142  C6 9E 0006B         MOVAB   DIR_STACK, R2                       ; 2621
                                        53      03A6  C6 D0 00070         MOVL    DIR_SP, R3
                                                      72 11 00075         BRB     11$
                                        50      0142  C6 9E 00077 3$:     MOVAB   DIR_STACK, R0                       ; 2632
                                        50            52 D1 0007C         CMPL    D, R0
                                                      0B 12 0007F         BNEQ    4$
                                        6E            06 D0 00081         MOVL    #6, DESC                            ; 2635
                                    04  AE      FF72  CF 9E 00084         MOVAB   P.AAH, DESC+4                       ; 2636
                                                      16 11 0008A         BRB     7$                                  ; 2632
                                  03A6  C6            52 D1 0008C 4$:     CMPL    D, DIR_SP                           ; 2640
                                                      06 12 00091         BNEQ    5$
                                        6E        02  A6 9A 00093         MOVZBL  DIR_STRING, DESC                    ; 2641
                                                      04 11 00097         BRB     6$
                                        6E        66  A2 9A 00099 5$:     MOVZBL  102(D), DESC                        ; 2642
                                    04  AE        03  A6 9E 0009D 6$:     MOVAB   DIR_STRING+1, DESC+4                ; 2643
                                        7E        50  8F 9A 000A2 7$:     MOVZBL  #80, -(SP)                          ; 2649
                                        67            01 FB 000A6         CALLS   #1, GET_VM
                                    28  A2            50 D0 000A9         MOVL    R0, 40(D)
                                        7E        50  8F 9A 000AD         MOVZBL  #80, -(SP)                          ; 2650
                                        67            01 FB 000B1         CALLS   #1, GET_VM
                                    30  A2            50 D0 000B4         MOVL    R0, 48(D)
                                                  F6  A6 9F 000B8         PUSHAB  DIR_SEL_NTV                         ; 2659
                                                  38  A2 9F 000BB         PUSHAB  56(D)
                                                  24  A2 9F 000BE         PUSHAB  36(R2)                              ; 2658
                                                  EE  A6 9F 000C1         PUSHAB  DIR_SEL_DIR                         ; 2655
                                                  10  AE 9F 000C4         PUSHAB  DESC
                             00000000G  00            05 FB 000C7         CALLS   #5, MATCH_DIRECTORY                 ; 2659
                                        04            50 E9 000CE         BLBC    STATUS, 8$                          ; 2661
                                    23  A2            02 88 000D1         BISB2   #2, 35(D)
                          04            50            02 E1 000D5 8$:     BBC     #2, STATUS, 9$                      ; 2662
                                    23  A2            04 88 000D9         BISB2   #4, 35(D)
                          04            50            03 E1 000DD 9$:     BBC     #3, STATUS, 10$                     ; 2663
                                    23  A2            08 88 000E1         BISB2   #8, 35(D)
                                        52        44  A2 9E 000E5 10$:    MOVAB   68(R2), D                           ; 2621
                                        53            52 D1 000E9 11$:    CMPL    D, R3
                                                      89 1B 000EC         BLEQU   3$
                                                         04 000EE         RET                                         ; 2666

; Routine Size:  239 bytes,    Routine Base:  CODE + 0E6E
```

```
1 %SBTTL 'FIND_NEXT - find next file'
1 GLOBAL ROUTINE FIND_NEXT=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine searches for the next file matching the specified
1 !       selection filespec.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       Directory scan context.
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       Directory scan context.  If a file was found, the name block contains
1 !       the resultant string and file ID.
1 !
1 ! ROUTINE VALUE:
1 !       True if a file was found, false otherwise.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       N:                      REF BBLOCK,     ! Local copy of DIR_NAM
2       D:                      REF BBLOCK;     ! Local copy of DIR_SP
2
2
2 N = .DIR_NAM;
2 D = .DIR_SP;
2 DIR_STATUS = DIR_VERLIMIT = 0;
2
2
2 ! If the directory stack is now empty, we have completed the MFD.
2 !
2 IF .DIR_LEVELS LEQ 0
2 THEN
3     BEGIN
3     N[NAM$B_RSL] = 0;
3     N[NAM$W_FID_NUM] = N[NAM$W_FID_SEQ] = N[NAM$W_FID_RVN] = 0;
3     N[NAM$W_DID_NUM] = N[NAM$W_DID_SEQ] = N[NAM$W_DID_RVN] = 0;
3     RETURN FALSE;
3     END;
2
2
2 ! Return the MFD if requested.
2 !
2 IF TESTBITSC(DIR_FLAGS[D_INITIAL])
2 THEN
```

```
2       IF .DIR_FLAGS[D_SCANNED_DIRS] AND .D[D_NON_TERM]
2       THEN
3           BEGIN
3           DIR_STATUS[D_STAT_VALID] = DIR_STATUS[D_STAT_SCANNED] = TRUE;
3           RETURN TRUE;
2           END;
2
2
2   ! Loop until we find something or traverse past a possible match.
2   !
2   WHILE TRUE DO
3       BEGIN
3       LOCAL
3           R:                  REF BBLOCK,     ! Local copy of D_REC
3           V:                  REF BBLOCK;     ! Local copy of D_VER
3
3
3       ! Push down to a lower directory if necessary.  This logic depends upon the
3       ! resultant string in the name block being unmodified from the previous
3       ! call.
3       !
4       IF NOT (.DIR_FLAGS[D_SCAN_FAILED] OR .DIR_FLAGS[D_ROOT_MFD])
3       THEN IF TESTBITSC(D[D_DIR_SCAN])
3           THEN
4               BEGIN
4
4               ! Push down the directory stack.
4               !
4               DIR_LEVELS = .DIR_LEVELS + 1;
4               DIR_SP = D = .D + D_S_ENTRY;
4               BBLOCK[D[D_FID], FID$W_NUM] = .N[NAM$W_FID_NUM];
4               BBLOCK[D[D_FID], FID$W_SEQ] = .N[NAM$W_FID_SEQ];
4               BBLOCK[D[D_FID], FID$W_RVN] = .N[NAM$W_FID_RVN];
4               D[D_SAV_LEN] = .DIR_STRING[0];
4               D[D_VBN] = 1;
4
4
4               ! Generate the new directory string.  If this is not the top level,
4               ! append a dot, and then append the directory name.
4               !
4               IF .DIR_STRING[0] NEQ 0 THEN DIR_STRING[0] = .DIR_STRING[0] + 1;
4               DIR_STRING[0] = .DIR_STRING[0] + .N[NAM$B_NAME];
3               END;
3
3
3       ! Get a new chunk of directory if necessary.
3       !
3       IF
3           .D[D_REC] GEQA .D[D_BUF_LIM] AND
4           (.D[D_VBN] EQL 1 OR .D[D_VBN] LEQU .D[D_DIR_LEN])
3       THEN
4           BEGIN
4           LITERAL
4               ATR_LENGTH= MAXU(
4                           $BYTEOFFSET(FH1$W_RECATTR)+32,
4                           $BYTEOFFSET(FH2$W_FILEPROT)+2);
4           LOCAL
```

```
            FIB:        BBLOCK[FIB$C_LENGTH],   ! FIB
            FIB_DESC:   VECTOR[2],              ! Descriptor for FIB
            STATUS,                             ! Status variable
            IOSB:       VECTOR[4,WORD],         ! I/O status block
            HEADER:     BBLOCK[ATR_LENGTH],     ! Beginning of file header
            ATR_DESC:   BBLOCK[12];             ! ACP attributes list


        ! Access the current directory file.
        !
        FIB_DESC[0] = FIB$C_LENGTH;
        FIB_DESC[1] = FIB;
        CH$FILL (0, FIB$C_LENGTH, FIB);
        FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
        IF .QUAL[QUAL_IGNO_INTE] THEN FIB[FIB$L_ACCTL] = FIB$M_NOLOCK OR FIB$M_NORECORD;
        FIB[FIB$W_FID_NUM] = .BBLOCK[D[D_FID], FID$W_NUM];
        FIB[FIB$W_FID_SEQ] = .BBLOCK[D[D_FID], FID$W_SEQ];
        FIB[FIB$W_FID_RVN] = .BBLOCK[D[D_FID], FID$W_RVN];
        ATR_DESC[0,0,16,0] = ATR_LENGTH;
        ATR_DESC[2,0,16,0] = ATR$C_HEADER;
        ATR_DESC[4,0,32,0] = HEADER;
        ATR_DESC[8,0,32,0] = 0;
        STATUS = $QIOW(
            FUNC=IO$_ACCESS OR IO$M_ACCESS,
            CHAN=.DIR_CHAN,
            IOSB=IOSB,
            P1=FIB_DESC,
            P5=ATR_DESC);
        IF .STATUS THEN STATUS = .IOSB[0];


        ! If a privilege violation occurred and the file specification is
        ! nonwild, try an ACP call in case the problem is an execute-only
        ! directory.
        !
        IF
            .STATUS<0,16> EQL SS$_NOPRIV AND
            NOT .COM_FLAGS[COM_STANDALONE] AND
            NOT .D[D_WILD_TERM] AND
            .D[D_BUF_ADDR] EQL 0
        THEN
            BEGIN
            LOCAL
                FIB_DESC:       VECTOR[2],              ! Descriptor for FIB
                FIB:            BBLOCK[FIB$C_LENGTH],   ! FIB
                FNA_DESC:       VECTOR[2],              ! Descriptor for FNA
                FNA:            VECTOR[86,BYTE];        ! Buffer for 'n.t;v'


            D[D_BUF_LEN] = 512;
            D[D_BUF_ADDR] = GET_VM(512);


            ! Initialize the FIB.
            !
            FIB_DESC[0] = FIB$C_LENGTH;
            FIB_DESC[1] = FIB;
```

```
   5          CH$FILL(0, FIB$C_LENGTH, FIB);
   5          FIB[FIB$W_DID_NUM] = .BBLOCK[D[D_FID], FID$W_NUM];
   5          FIB[FIB$W_DID_SEQ] = .BBLOCK[D[D_FID], FID$W_SEQ];
   5          FIB[FIB$W_DID_RVN] = .BBLOCK[D[D_FID], FID$W_RVN];
   5
   5
   5          ! Initialize the filename.
   5          !
   5          FNA_DESC[0] = %ALLOCATION(FNA);
   5          FNA_DESC[1] = FNA;
P  5          $FAO(
P  5              $DESCRIPTOR('!AS;!UW'),
P  5              FNA_DESC,
P  5              FNA_DESC,
   5              D[D_TERM_DESC], .D[D_TERM_VER]);
   5
   5
   5          ! Execute the lookup.
   5          !
P  5          STATUS = $QIOW(
P  5              FUNC=IO$_ACCESS,
P  5              CHAN=.DIR_CHAN,
P  5              IOSB=IOSB,
P  5              P1=FIB_DESC,
   5              P2=FNA_DESC);
   5          IF .STATUS THEN STATUS = .IOSB[0];
   5
   5
   5          ! If a no such file error occurred, simulate an empty directory.
   5          ! Otherwise, simulate a single block directory that contains the
   5          ! desired entry.
   5          !
   5          IF .STATUS<0,16> EQL SS$_NOSUCHFILE
   5          THEN
   6              BEGIN
   6              D[D_DIR_LEN] = 0;
   6              STATUS = SS$_NORMAL;
   6              END
   5          ELSE
   6              BEGIN
   6              LOCAL
   6                  L,                                    ! Length of name and type
   6                  P:              REF BBLOCK;           ! Pointer to entry
   6
   6
   6                  ! Initialize directory stack context.
   6                  !
   6                  DIR_STRUCLEV = 2;
   6                  D[D_VBN] = 2;
   6                  D[D_DIR_LEN] = 1;
   6
   6
   6                  ! Initialize simulated directory entry.
   6                  !
   6                  L = (.BBLOCK[D[D_TERM_DESC], DSC$W_LENGTH] + 1) AND NOT 1;
   6                  P = .D[D_BUF_ADDR];
   6                  P[DIR$W_SIZE] = .L + DIR$C_LENGTH - 2 + DIR$C_VERSION;
```

```
                P[DIR$W_VERLIMIT] = .FIB[FIB$W_VERLIMIT];
                P[DIR$B_FLAGS] = 0;
                P[DIR$B_NAMECOUNT] = .BBLOCK[D[D_TERM_DESC], DSC$W_LENGTH];
                P = CH$COPY(
                    .BBLOCK[D[D_TERM_DESC], DSC$W_LENGTH],
                    .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER],
                    0,
                    .L, P[DIR$T_NAME]);
                P[DIR$W_VERSION] = .D[D_TERM_VER];
                P[DIR$W_FID_NUM] = .FIB[FIB$W_FID_NUM];
                P[DIR$W_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
                P[DIR$W_FID_RVN] = .FIB[FIB$W_FID_RVN];


                D[D_BUF_LIM] = .P + DIR$C_VERSION;
                END;
            END;


        IF NOT .STATUS
        THEN
            BEGIN

            ! Report failure to access the directory.
            !
            SIGNAL(
                BACKUP$_OPENDIR,
                2,
                .DIR_DEV_DESC,
                (IF .DIR_STRING[0] EQL 0 THEN MFD ELSE DIR_STRING),
                .STATUS);

            ! Readjust context so that processing of the directory is
            ! aborted.
            !
            D[D_DIR_LEN] = 0;
            END
        ELSE
            BEGIN

            ! If there is not currently a buffer, this is the first chunk of
            ! this directory.  Determine whether the file is indeed a
            ! directory, determine its size, and allocate the buffer.
            !
            IF .D[D_BUF_ADDR] EQL 0
            THEN
                BEGIN

                ! Ensure that the file is, in fact, a directory.  At this point
                ! we know only that the filename is ".DIR;1".  Use the portion
                ! of the file header that was obtained during the access.
                ! Compute the file length if valid.  If invalid, leave the file
                ! length zero to avoid processing the file.
                !
                D[D_DIR_LEN] = 0;
                DIR_STRUCLEV = .HEADER[FH2$B_STRUCLEV];
```

```
6               IF .DIR_STRUCLEV EQL 2
6               THEN
7                   BEGIN
7                   BIND
7                       RECATTR=            HEADER[FH2$W_RECATTR]: BBLOCK;
7
7                   IF .HEADER[FH2$V_DIRECTORY]
7                   THEN
8                       BEGIN
8                       D[D_DIR_LEN] = ROT(.RECATTR[FAT$L_EFBLK], 16);
8                       IF .RECATTR[FAT$W_FFBYTE] EQL 0
8                           THEN D[D_DIR_LEN] = .D[D_DIR_LEN] - 1;
8                       D[D_FPRO] = .HEADER[FH2$W_FILEPROT];
8                       D[D_UIC] = .HEADER[FH2$L_FILEOWNER];
8                       D[D_VERLIM] = .BBLOCK[HEADER[FH2$W_RECATTR], FAT$W_VERSIONS];
7                       END;
7                   END
6               ELSE
7                   BEGIN
7                   BIND
7                       RECATTR=            HEADER[FH1$W_RECATTR]: BBLOCK;
7
7                   IF
7                       .RECATTR[FAT$B_RTYPE] EQL FAT$C_FIXED AND
7                       .RECATTR[FAT$W_RSIZE] EQL NMB$C_DIRENTRY
7                   THEN
8                       BEGIN
8                       D[D_DIR_LEN] = ROT(.RECATTR[FAT$L_EFBLK], 16);
8                       IF .RECATTR[FAT$W_FFBYTE] EQL 0
8                           THEN D[D_DIR_LEN] = .D[D_DIR_LEN] - 1;
8                       D[D_FPRO] = .HEADER[FH1$W_FILEPROT];
8                       D[D_UIC] = .HEADER[FH1$B_UICMEMBER];
8                       (D[D_UIC])<16,16> = .HEADER[FH1$B_UICGROUP];
8                       D[D_VERLIM] = .BBLOCK[HEADER[FH1$W_RECATTR], FAT$W_VERSIONS];
7                       END;
6                   END;
6
6
6               ! If the file looks like a directory and is not zero length,
6               ! allocate a buffer for it.
6               !
6               IF .D[D_DIR_LEN] NEQ 0
6               THEN
7                   BEGIN
7
7                   ! Compute buffer length.  Try to read the entire directory
7                   ! at one time, but no more than DIR_BUF_COUNT blocks.
7                   ! However, for an ODS-1 directory on which a latest-version
7                   ! scan is in progress, always read the entire directory.
7                   !
7                   D[D_BUF_LEN] = .D[D_DIR_LEN] * 512;
7                   IF
7                       .D[D_BUF_LEN] GTRU DIR_BUF_COUNT*512 AND
8                       NOT (.DIR_STRUCLEV EQL 1 AND .DIR_SEL_LATEST LEQ 0)
7                   THEN
7                       D[D_BUF_LEN] = DIR_BUF_COUNT*512;
7
```

```
                                !
                                ! Allocate memory for buffer.
                                !
                                D[D_BUF_ADDR] = GET_VM(.D[D_BUF_LEN]);
                                END;
                        END;

                ! If we are not yet beyond the end, size the portion of the
                ! directory to be read on this pass and read it.
                !
                IF .D[D_VBN] LEQU .D[D_DIR_LEN]
                THEN
                        BEGIN
                        LOCAL
                                READ_ADDRESS,           ! Address for current read QIO
                                PROC_LENGTH;            ! Bytes to process this time

                        ! Compute size of chunk to be processed on this iteration.
                        !
                        PROC_LENGTH = MINU(
                                .D[D_BUF_LEN],
                                (.D[D_DIR_LEN] - .D[D_VBN] + 1) * 512);
                        D[D_BUF_LIM] = .D[D_BUF_ADDR] + .PROC_LENGTH;
                        D[D_REC] = D[D_VER] = 0;


                        ! Loop to read the chunk of directory that will be processed on
                        ! this iteration.  Because the chunk might exceed 65535 bytes,
                        ! we must have a loop, though it is unlikely to be executed
                        ! more than once.
                        !
                        READ_ADDRESS = .D[D_BUF_ADDR];
                        WHILE .READ_ADDRESS LSSA .D[D_BUF_LIM] DO
                                BEGIN
                                LOCAL
                                        READ_LENGTH;    ! Size of current transfer in bytes


                                ! Compute size of this transfer.
                                !
                                READ_LENGTH = MINU(
                                        127 * 512,
                                        .D[D_BUF_LIM] - .READ_ADDRESS);


                                ! Read in the blocks.
                                !
                                STATUS = $QIOW(
                                        FUNC=IO$_READVBLK,
                                        CHAN=.DIR_CHAN,
                                        IOSB=IOSB,
                                        P1=.READ_ADDRESS,
                                        P2=.READ_LENGTH,
                                        P3=.D[D_VBN]);
                                READ_ADDRESS = .READ_ADDRESS + .READ_LENGTH;
```

```
  7                  D[D_VBN] = .D[D_VBN] + .READ_LENGTH/512;
  7                  IF .STATUS THEN STATUS = .IOSB[0];
  7                  IF NOT .STATUS
  7                  THEN
  8                      BEGIN
  8                      SIGNAL(
  8                          BACKUP$_READDIR,
  8                          2,
  8                          .DIR_DEV_DESC,
  8                          (IF .DIR_STRING[0] EQL 0 THEN MFD ELSE DIR_STRING),
  8                          .STATUS);
  8
  8
  8                      ! Readjust context so that processing of the directory
  8                      ! is aborted.
  8                      !
  8                      D[D_DIR_LEN] = 0;
  8                      D[D_BUF_LIM] = 0;
  8                      EXITLOOP;
  7                      END;
  6                  END;
  5              END;
  5
  5
  5          ! Deaccess the directory file.
  5          !
P 5          C$QIOW(
P 5              FUNC=IO$_DEACCESS,
  5              CHAN=.DIR_CHAN);
  4          END;
  3      END;
  3
  3
  3  ! If no more directory is available, pop up to the higher directory.
  3  !
  3  IF .D[D_REC] GEQA .D[D_BUF_LIM]
  3  THEN
  4      BEGIN
  4
  4      ! Deallocate dynamic memory.
  4      !
  4      IF .D[D_BUF_ADDR] NEQ 0
  4      THEN
  4          FREE_VM(.D[D_BUF_LEN], .D[D_BUF_ADDR]);
  4      IF .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER] NEQ 0
  4      THEN
  4          FREE_VM(DIR$S_NAME, .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER]);
  4      IF .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER] NEQ 0
  4      THEN
  4          FREE_VM(DIR$S_NAME, .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER]);
  4
  4
  4      ! Pop the directory stack.
  4      !
  4      DIR_STRING[0] = .D[D_SAV_LEN];
  4      CH$FILL(0, D_S_ENTRY, .D);
  4      DIR_SP = D = .D - D_S_ENTRY;
```

```
4               DIR_LEVELS = .DIR_LEVELS - 1;
4
4
4               ! If the directory stack is now empty, we have completed the MFD.
4               !
4               IF .DIR_LEVELS LEQ 0
4               THEN
5                   BEGIN
5                   N[NAM$B_RSL] = 0;
5                   N[NAM$W_FID_NUM] = N[NAM$W_FID_SEQ] = N[NAM$W_FID_RVN] = 0;
5                   N[NAM$W_DID_NUM] = N[NAM$W_DID_SEQ] = N[NAM$W_DID_RVN] = 0;
5                   RETURN FALSE;
4                   END;
3               END;
3
3
3           ! Adjust pointers to the next entry in the directory.
3           !
3           R = .D[D_REC];
3           V = .D[D_VER];
3           IF TESTBITCC(DIR_FLAGS[D_SCAN_FAILED])
3           THEN
4               BEGIN
4               IF .DIR_STRUCLEV EQL 1
4               THEN
5                   BEGIN
5                   WHILE TRUE DO
6                       BEGIN
6
6                       ! Advance the record pointer to the next (or first) entry.
6                       !
6                       IF .R EQL 0
6                       THEN
6                           R = .D[D_BUF_ADDR]
6                       ELSE
7                           BEGIN
7                           R = .R + NMB$C_DIRENTRY;
7                           V = .V + 1;
6                           END;
6
6
6                       ! If there are no more entries, exit the loop.
6                       !
6                       IF .R GEQA .D[D_BUF_LIM] THEN EXITLOOP;
6
6
6                       ! If the entry contains a non-zero file number, it is in use.
6                       !
6                       IF .R[NMB$W_FID_NUM] NEQ 0
6                       THEN
7                           BEGIN
7
7                           ! If the selection filespec selects latest version,
7                           ! determine if this is the latest version, by scanning
7                           ! the entire directory for a higher version.
7                           !
7                           D[D_VER_COUNT] = 0;
```

```
7                       IF .DIR_SEL_LATEST LEQ 0
7                       THEN
7                           INCRA P
7                           FROM .D[D_BUF_ADDR]
7                           TO .D[D_BUF_ADDR] + .D[D_BUF_LEN] - NMB$C_DIRENTRY
7                           BY NMB$C_DIRENTRY DO
8                               BEGIN
8                               MAP
8                                   P:              REF BBLOCK;
8
8                               IF
8                                   .(P[NMB$W_NAME])   EQL .(R[NMB$W_NAME])   AND
8                                   .(P[NMB$W_NAME]+4) EQL .(R[NMB$W_NAME]+4) AND
8                                   .P[NMB$W_VERSION] GTRU .R[NMB$W_VERSION]
8                               THEN
9                                   BEGIN
9                                   D[D_VER_COUNT] = -32768;
9                                   EXITLOOP;
8                                   END;
7                               END;
7
7
7                       ! Exit the loop with a valid entry, and D_VER_COUNT set.
7                       !
7                       EXITLOOP;
6                       END;
5                   END;
5               END
4           ELSE
5               BEGIN
5               WHILE TRUE DO
6                   BEGIN
6                   LOCAL
6                       NEXT_RECORD;                ! Pointer to next record
6
6
6                   ! Advance the record and version pointers to the
6                   ! next (or first) entry.
6                   !
6                   IF .R EQL 0
6                   THEN
6                       R = .D[D_BUF_ADDR]
6                   ELSE
6                       IF .V NEQ 0
6                       THEN
7                           BEGIN
7                           NEXT_RECORD = .R[DIR$W_SIZE] + .R + 2;
7                           V = .V + DIR$C_VERSION;
7                           IF .V LEQA .NEXT_RECORD - DIR$C_VERSION THEN EXITLOOP;
7                           R = .NEXT_RECORD;
6                           END;
6
6
6                   ! If there are no more entries, exit the loop.
6                   !
6                   IF .R GEQA .D[D_BUF_LIM] THEN EXITLOOP;
6
```

```
6
6       IF .R[DIR$W_SIZE] EQL 65535
6       THEN
7           BEGIN
7
7           ! End of this block.  Advance to next.
7           !
7           R = ((.R - .D[D_BUF_ADDR]) AND NOT 511) + .D[D_BUF_ADDR] + 512;
7           V = 0;
7           END
6       ELSE
7           BEGIN
7
7           ! Point to where next record should start.  Make some
7           ! validity tests on the entry we are looking at.
7           !
7           NEXT_RECORD = .R[DIR$W_SIZE] + .R + 2;
7           IF
8               BEGIN
8               IF
8                   .NEXT_RECORD GEQA ((.R - .D[D_BUF_ADDR]) AND NOT 511) + .D[D_BUF_ADDR] + 512 OR
8                                                           ! Entry within block?
8                   .R[DIR$W_SIZE] OR                       ! Length even?
8                   .R[DIR$W_SIZE] LSSU DIR$C_LENGTH + DIR$C_VERSION
8                                                           ! Long enough?
8               THEN
8                   TRUE
8               ELSE
9                   BEGIN
9                   V = (.R + DIR$C_LENGTH + .R[DIR$B_NAMECOUNT] + 1) AND NOT 1;
9                   .R[DIR$V_TYPE] NEQ DIR$C_FID OR         ! Proper type code?
9                   .V GEQA ((.R - .D[D_BUF_ADDR]) AND NOT 511) + .D[D_BUF_ADDR] + 512 - DIR$C_VERSI
9                                                           ! Version entry within block?
9                   END
8               END
7           THEN
8               BEGIN
8
8               ! Directory format error.  Report it and quit.
8               !
8               SIGNAL(
8                   BACKUP$_BADDIR,
8                   2,
8                   .DIR_DEV_DESC,
8                   (IF .DIR_STRING[0] EQL 0 THEN MFD ELSE DIR_STRING));
8
8
8               ! Adjust context to avoid processing this directory.
8               !
8               R = .D[D_BUF_LIM];
8               D[D_DIR_LEN] = 0;
7               END;
7
7
7           ! Found a valid entry.  Exit the loop.
7           !
7           EXITLOOP;
```

```
6                               END;
5                           END;
4                       END;
4                   D[D_REC] = .R;
4                   D[D_VER] = .V;
3                   END;
3
3
3           ! If an entry was found, finish processing it.
3           !
3           IF .R LSSA .D[D_BUF_LIM]
3           THEN
4               BEGIN
4               LOCAL
4                   RSA_DESC:   VECTOR[2],              ! Descriptor for RSA area
4                   NAME:       REF VECTOR[,BYTE],      ! Pointer to ASCIC name.typ
4                   VERSION,                            ! Binary version number
4                   FILE_NAME:  VECTOR[20,BYTE];        ! Area to build ODS-1 filename
4
4
4               ! Get a pointer to the name.typ string and the version number.
4               ! Initialize the file ID in the name block.
4               !
4               IF .DIR_STRUCLEV EQL 2
4               THEN
5                   BEGIN
5                   D[D_VER_COUNT] = .D[D_VER_COUNT] - 1;
5                   IF
5                       .R[DIR$B_NAMECOUNT] NEQ .BBLOCK[D[D_NAME_DESC], DSC$W_LENGTH] OR
5                       CH$NEQ(
5                           .R[DIR$B_NAMECOUNT],
5                           R[DIR$T_NAME],
5                           .R[DIR$B_NAMECOUNT],
5                           .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER])
5                   THEN
6                       BEGIN
6                       D[D_VER_COUNT] = 0;
6                       BBLOCK[D[D_NAME_DESC], DSC$W_LENGTH] = .R[DIR$B_NAMECOUNT];
6                       CH$MOVE(
6                           .R[DIR$B_NAMECOUNT],
6                           R[DIR$T_NAME],
6                           .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER]);
5                       END;
5                   NAME = R[DIR$B_NAMECOUNT];
5                   VERSION = .V[DIR$W_VERSION];
5                   DIR_VERLIMIT = .R[DIR$W_VERLIMIT];
5                   N[NAM$W_FID_NUM] = .V[DIR$W_FID_NUM];
5                   N[NAM$W_FID_SEQ] = .V[DIR$W_FID_SEQ];
5                   N[NAM$W_FID_RVN] = .V[DIR$W_FID_RVN];
5                   IF .N[NAM$B_FID_RVN] EQL 0
5                       THEN N[NAM$B_FID_RVN] = .BBLOCK[D[D_FID], FID$B_RVN];
5                   END
4               ELSE
5                   BEGIN LOCAL T;
5                   T = MAKE_STRING(.R, FILE_NAME[1]);
5                   FILE_NAME[0] = CH$FIND_CH(.T, FILE_NAME[1], %C';') - FILE_NAME[1];
5                   NAME = FILE_NAME;
```

```
   5                    VERSION = .R[NMB$W_VERSION];
   5                    N[NAM$W_FID_NUM] = .R[NMB$W_FID_NUM];
   5                    N[NAM$W_FID_SEQ] = .R[NMB$W_FID_SEQ];
   5                    N[NAM$W_FID_RVN] = 1;
   4                    END;
   4
   4
   4            ! Initialize the resultant string in the name block.
   4            !
   4            RSA_DESC[0] = .N[NAM$B_RSS];
   4            RSA_DESC[1] = .N[NAM$L_RSA];
P  4            $FAO(
P  4                $DESCRIPTOR('!AS[!AC]!AC;!UW'),
P  4                RSA_DESC,
P  4                RSA_DESC,
P  4                .DIR_DEV_DESC,
P  4                (IF .DIR_STRING[0] EQL 0 THEN MFD ELSE DIR_STRING),
P  4                .NAME,
   4                .VERSION);
   4            N[NAM$B_RSL] = .RSA_DESC[0];
   4            INIT_NAMEBLOCK(.N);
   4
   4
   4            ! Initialize the directory ID in the name block.
   4            !
   4            N[NAM$W_DID_NUM] = .BBLOCK[D[D_FID], FID$W_NUM];
   4            N[NAM$W_DID_SEQ] = .BBLOCK[D[D_FID], FID$W_SEQ];
   4            N[NAM$W_DID_RVN] = .BBLOCK[D[D_FID], FID$W_RVN];
   4
   4
   4            ! Set the directory scan bits for the next iteration.
   4            !
   4            DIR_STATUS = 0;
   4            IF
   4                .DIR_LEVELS LEQ D_K_NLEVELS-1 AND
   5                (IF .N[NAM$B_TYPE] EQL 4
   5                    THEN ..N[NAM$L_TYPE] EQL '.DIR'
   4                    ELSE FALSE) AND
   4                .VERSION EQL 1 AND
   5                (.N[NAM$W_FID_NUM] NEQ FID$C_MFD OR
   5                    .N[NAM$B_FID_NMX] NEQ 0)
   4            THEN
   5                BEGIN
   5                LOCAL
   5                    STATUS,                 ! Status variable
   5                    DIR_DESC: VECTOR[2];    ! Descriptor for new directory string
   5
   5
   5                ! Note that this is a directory.
   5                !
   5                DIR_STATUS[D_STAT_VALID] = TRUE;
   5
   5
   5                ! Generate the new directory string.  If this is not the top level,
   5                ! append a dot, and then append the directory name.
   5                !
   5                DIR_DESC[0] = .DIR_STRING[0];
```

```
5                       DIR_DESC[1] = DIR_STRING[1];
5                       IF .DIR_DESC[0] NEQ 0
5                       THEN
6                           BEGIN
6                           DIR_DESC[0] = .DIR_DESC[0] + 1;
6                           DIR_STRING[.DIR_DESC[0]] = %C'.';
5                           END;
5                       CH$MOVE(
5                           .N[NAM$B_NAME],
5                           .N[NAM$L_NAME],
5                           DIR_STRING[.DIR_DESC[0]+1]);
5                       DIR_DESC[0] = .DIR_DESC[0] + .N[NAM$B_NAME];
5
5
5                       ! Temporarily push down the directory stack.
5                       !
5                       D = .D + D_S_ENTRY;
5
5
5                       ! Allocate the dynamic areas for this level.
5                       !
5                       BBLOCK[D[D_TERM_DESC], DSC$A_POINTER] = GET_VM(DIR$S_NAME);
5                       BBLOCK[D[D_NAME_DESC], DSC$A_POINTER] = GET_VM(DIR$S_NAME);
5
5
5                       ! Match the directory specification against the pattern string to
5                       ! determine if this directory should be scanned.  If it should not,
5                       ! pop the directory stack.  Otherwise, finish initializing context.
5                       !
5                       IF .DIR_FLAGS[D_IMAGE_SCAN]
5                       THEN
5                           STATUS = %B'111'
5                       ELSE
5                           STATUS = MATCH_DIRECTORY(
5                               DIR_DESC,
5                               DIR_SEL_DIR,
5                               D[D_TERM_DESC],
5                               D[D_TERM_VER],
5                               DIR_SEL_NTV);
5
5
5                       IF .STATUS<1,1>
5                       THEN
6                           BEGIN
6                           DIR_SP[D_DIR_SCAN] = DIR_STATUS[D_STAT_SCANNED] = TRUE;
6                           IF .STATUS<0,1> THEN D[D_DIR_MATCHES] = DIR_STATUS[D_STAT_FILE_SEL] = TRUE;
6                           IF .STATUS<2,1> THEN D[D_WILD_TERM] = TRUE;
6                           IF .STATUS<3,1> THEN D[D_NON_TERM] = TRUE;
6                           END
5                       ELSE
6                           BEGIN
6                           FREE_VM(DIR$S_NAME, .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER]);
6                           FREE_VM(DIR$S_NAME, .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER]);
6                           CH$FILL(0, D_S_ENTRY, .D);
5                           END;
5
5
```

```
5                       ! Pop the directory stack.
5                       !
5                       D = .DIR_SP;
4                       END;
4
4
4               ! If this is image mode, the file always matches.
4               !
4               IF
4                   .DIR_FLAGS[D_IMAGE_SCAN] AND
5                   (NOT .DIR_FLAGS[D_SCANNED_DIRS] OR
5                       .N[NAM$W_FID_NUM] NEQ FID$C_MFD OR
5                       .N[NAM$B_FID_NMX] NEQ 0)
4               THEN
5                   BEGIN
5                   IF .DIR_STATUS[D_STAT_VALID]
5                       THEN DIR_STATUS[D_STAT_DIR_SEL] = TRUE;
5                   EXITLOOP;
4                   END;
4
4
4               ! Test for scan termination.
4               !
4               IF
5                   BEGIN
5                   IF .DIR_STRUCLEV EQL 1
5                   THEN
5                       .V GTRU .DIR_SCANLIMIT[.DIR_LEVELS-1]
5                   ELSE
5                       IF .BBLOCK[D[D_TERM_DESC], DSC$W_LENGTH] EQL 0
5                       THEN
5                           FALSE
5                       ELSE
5                           TERMINATE_SCAN(
5                               .NAME[0], NAME[1],
5                               .VERSION,
5                               .BBLOCK[D[D_TERM_DESC], DSC$W_LENGTH],
5                               .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER],
5                               .D[D_TERM_VER])
5                   END
4               THEN
5                   BEGIN
5                   IF .DIR_FLAGS[D_HARD_STOP]
5                   THEN
6                       BEGIN
6                       DIR_FLAGS[D_SCAN_FAILED] = TRUE;
6                       RETURN FALSE;
5                       END;
5
5
5                   ! Adjust context to skip rest of this directory.
5                   !
5                   D[D_REC] = .D[D_BUF_LIM];
5                   D[D_DIR_LEN] = 0;
4                   END;
4
4
```

```
4               ! Match the selection file specification with the resultant string.
4               !
4               IF
4                   .D[D_REC] LSSA .D[D_BUF_LIM] AND
4                   .D[D_DIR_MATCHES] AND
4                   (.DIR_SEL_LATEST GTR 0 OR .DIR_SEL_LATEST EQL .D[D_VER_COUNT]) AND
5                   (NOT .DIR_FLAGS[D_SCANNED_DIRS] OR
5                       .N[NAM$W_FID_NUM] NEQ FID$C_MFD OR
5                       .N[NAM$B_FID_NMX] NEQ 0)
4               THEN
5                   BEGIN
5                   LOCAL
5                       NTV_DESC:           VECTOR[2];          ! Descriptor for n.t;v
5
5                   ! Match the file specification.
5                   !
5                   NTV_DESC[0] = .N[NAM$L_RSA] + .N[NAM$B_RSL] - .N[NAM$L_NAME];
5                   NTV_DESC[1] = .N[NAM$L_NAME];
5                   IF MATCH_FILENAME(NTV_DESC, DIR_SEL_NTV)
5                   THEN
6                       BEGIN
6                       IF .DIR_STATUS[D_STAT_VALID]
6                           THEN DIR_STATUS[D_STAT_DIR_SEL] = TRUE;
6                       EXITLOOP;
5                       END;
4                   END;
4
4               ! If scanned directories are requested, and the directory was not
4               ! selected above, return it anyway.
4               !
4               IF .DIR_FLAGS[D_SCANNED_DIRS] AND .D[D_DIR_SCAN] AND .D[D_NON_TERM]
4               THEN
4                   EXITLOOP;
3               END;
2           END;
2
2
2       TRUE
1       END;


                                57  55  21  3B  53  41  21  00F5D P.AAJ:   .ASCII   \!AS;!UW\                          ;
                                             00000007  00F64 P.AAI:   .LONG    7                                  ;
                                             00000000' 00F68          .ADDRESS P.AAJ                              ;
57  55  21  3B  43  41  21  5D  43  41  21  5B  53  41  21  00F6C P.AAL:   .ASCII   \!AS[!AC]!AC;!UW\                  ;
                                                       00F7B          .BLKB    1
                                             0000000F  00F7C P.AAK:   .LONG    15                                 ;
                                             00000000' 00F80          .ADDRESS P.AAL                              ;

                                                                      .EXTRN   STA_QIOW

                                                  0FFC 00000          .ENTRY   FIND_NEXT, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 2668
                                                                               R10,R11                            ;
                                  5E      FEB4    CE  9E 00002          MOVAB    -332(SP), SP                       ;
                                  58 00000000'  EF D0 00007       MOVL    DIR_NAM, N                         ; 2703
                                  56 00000000'  EF D0 0000E       MOVL    DIR_SP, D                          ; 2704
                                     00000000'  EF B4 00015       CLRW    DIR_VERLIMIT                       ; 2705
                                     00000000'  EF 94 0001B       CLRB    DIR_STATUS
                                     00000000'  EF 95 00021       TSTB    DIR_LEVELS                         ; 2710
                                                03 12 00027       BNEQ    1$
                                              0490 31 00029       BRW     44$
                    17 00000000'  EF            04 E5 0002C 1$:   BBCC    #4, DIR_FLAGS, 2$                  ; 2722
                    0F 00000000'  EF            03 E1 00034       BBC     #3, DIR_FLAGS, 2$                  ; 2724
                    0A       23   A6            03 E1 0003C       BBC     #3, 35(D), 2$
                       00000000'  EF            05 88 00041       BISB2   #5, DIR_STATUS                     ; 2727
                                              08B7 31 00048       BRW     96$                                ; 2728
                    4D 00000000'  EF            01 E0 0004B 2$:   BBS     #1, DIR_FLAGS, 4$                  ; 2745
                    45 00000000'  EF            05 E0 00053       BBS     #5, DIR_FLAGS, 4$
                    40       23   A6            00 E5 0005B       BBCC    #0, 35(D), 4$                      ; 2746
                                     00000000'  EF 96 00060       INCB    DIR_LEVELS                         ; 2752
                                  56       44   A6 9E 00066       MOVAB   68(R6), D                          ; 2753
                       00000000'  EF            56 D0 0006A       MOVL    D, DIR_SP
                                  50       1C   A6 9E 00071       MOVAB   28(D), R0                          ; 2754
                                  60       24   A8 D0 00075       MOVL    36(N), (R0)
                             04   A0       28   A8 B0 00079       MOVW    40(N), 4(R0)                       ; 2756
                             22   A6 00000000'  EF 90 0007E       MOVB    DIR_STRING, 34(D)                  ; 2757
                             14   A6            01 D0 00086       MOVL    #1, 20(D)                          ; 2758
                                     00000000'  EF 95 0008A       TSTB    DIR_STRING                         ; 2764
                                                06 13 00090       BEQL    3$
                                     00000000'  EF 96 00092       INCB    DIR_STRING
                       00000000'  EF       3B   A8 80 00098 3$:   ADDB2   59(N), DIR_STRING                  ; 2765
                                  5A       10   A6 9E 000A0 4$:   MOVAB   16(D), R10                         ; 2772
                                  6A            66 D1 000A4       CMPL    (D), (R10)
                                                03 1E 000A7       BGEQU   6$
                                              03B3 31 000A9 5$:   BRW     40$
                                  01       14   A6 D1 000AC 6$:   CMPL    20(D), #1                          ; 2773
                                                07 13 000B0       BEQL    7$
                             18   A6       14   A6 D1 000B2       CMPL    20(D), 24(D)
                                                F0 1A 000B7       BGTRU   5$
                             B8   AD       40   8F 9A 000B9 7$:   MOVZBL  #64, FIB_DESC                      ; 2791
                             BC   AD       C0   AD 9E 000BE       MOVAB   FIB, FIB_DESC+4                    ; 2792
0040    8F          00            6E            00 2C 000C3       MOVC5   #0, (SP), #0, #64, FIB             ; 2793
                                           C0   AD    000CA
                             C0   AD 00200000   8F D0 000CC       MOVL    #2097152, FIB                      ; 2794
                    08 00000000'  EF            02 E1 000D4       BBC     #2, QUAL+10, 8$                    ; 2795
                             C0   AD 00300000   8F D0 000DC       MOVL    #3145728, FIB
                                  57       1C   A6 9E 000E4 8$:   MOVAB   28(D), R7                          ; 2796
                             C4   AD            67 D0 000E8       MOVL    (R7), FIB+4
                             C8   AD       04   A7 B0 000EC       MOVW    4(R7), FIB+8                       ; 2798
                           FF60   CD 000A0042   8F D0 000F1       MOVL    #655426, ATR_DESC                  ; 2799
                           FF64   CD     FF6C   CD 9E 000FA       MOVAB   HEADER, ATR_DESC+4                 ; 2801
                                         FF68   CD D4 00101       CLRL    ATR_DESC+8                         ; 2802
                                  50 00000000'  EF D0 00105       MOVL    DIR_CHAN, R0                       ; 2808
                       0001FFFF   8F            50 D1 0010C       CMPL    R0, #131071
                                                23 1F 00113       BLSSU   9$
                                                7E D4 00115       CLRL    -(SP)
                                         FF60   CD 9F 00117       PUSHAB  ATR_DESC
                                                7E 7C 0011B       CLRQ    -(SP)
                                                7E D4 0011D       CLRL    -(SP)
                                           B8   AD 9F 0011F       PUSHAB  FIB_DESC
                                                7E 7C 00122       CLRQ    -(SP)
                                           B0  AD 9F 00124        PUSHAB   IOSB                        :
                                7E         72  8F 9A 00127        MOVZBL   #114, -(SP)                 :
                                               50 DD 0012B        PUSHL    R0                          :
                                               7E D4 0012D        CLRL     -(SP)                       :
                     00000000G  00             0C FB 0012F        CALLS    #12, STA_QIOW               :
                                               21 11 00136        BRB      10$                         :
                                               7E D4 00138  9$:   CLRL     -(SP)                       :
                                         FF60  CD 9F 0013A        PUSHAB   ATR_DESC                    :
                                               7E 7C 0013E        CLRQ     -(SP)                       :
                                               7E D4 00140        CLRL     -(SP)                       :
                                           B8  AD 9F 00142        PUSHAB   FIB_DESC                    :
                                               7E 7C 00145        CLRQ     -(SP)                       :
                                           B0  AD 9F 00147        PUSHAB   IOSB                        :
                                7E         72  8F 9A 0014A        MOVZBL   #114, -(SP)                 :
                                               50 DD 0014E        PUSHL    R0                          :
                                               7E D4 00150        CLRL     -(SP)                       :
                     00000000G  9F             0C FB 00152        CALLS    #12, @#SYS$QIOW             :
                                59             50 D0 00159 10$:   MOVL     R0, STATUS                  :
                                04             59 E9 0015C        BLBC     STATUS, 11$                 : 2809
                                59         B0  AD 3C 0015F        MOVZWL   IOSB, STATUS                :
                                24             59 B1 00163 11$:   CMPW     STATUS, #36                 : 2817
                                               08 12 00166        BNEQ     12$                         :
                  03 00000000'  EF             01 E1 00168        BBC      #1, COM_FLAGS, 13$          : 2818
                                             00E3 31 00170 12$:   BRW      16$                         :
                  F8        23  A6             02 E0 00173 13$:   BBS      #2, 35(D), 12$              : 2819
                                           0C  A6 D5 00178        TSTL     12(D)                       : 2820
                                               F3 12 0017B        BNEQ     12$                         :
                            08  A6       0200  8F 3C 0017D        MOVZWL   #512, 8(D)                  : 2830
                                7E       0200  8F 3C 00183        MOVZWL   #512, -(SP)                 : 2831
                     00000000G  00             01 FB 00188        CALLS    #1, GET_VM                  :
                            0C  A6             50 D0 0018F        MOVL     R0, 12(D)                   :
                          00A4  CE         40  8F 9A 00193        MOVZBL   #64, FIB_DESC               : 2836
                          FF5C  CD         64  AE 9E 00199        MOVAB    FIB, FIB_DESC+4             : 2837
0040    8F        00            6E             00 2C 0019F        MOVC5    #0, (SP), #0, #64, FIB      : 2838
                                           64  AE    001A6                                             :
                            6E  AE             67 D0 001A8        MOVL     (R7), FIB+10                : 2839
                            72  AE         04  A7 B0 001AC        MOVW     4(R7), FIB+14               : 2841
                            5C  AE         56  8F 9A 001B1        MOVZBL   #86, FNA_DESC               : 2846
                            60  AE         04  AE 9E 001B6        MOVAB    FNA, FNA_DESC+4             : 2847
                                7E         38  A6 3C 001BB        MOVZWL   56(D), -(SP)                : 2852
                                52         24  A6 9E 001BF        MOVAB    36(D), R2                   :
                                               52 DD 001C3        PUSHL    R2                          :
                                           64  AE 9F 001C5        PUSHAB   FNA_DESC                    :
                                           68  AE 9F 001C8        PUSHAB   FNA_DESC                    :
                                         FE11  CF 9F 001CB        PUSHAB   P.AAI                       :
                     00000000G  00             05 FB 001CF        CALLS    #5, SYS$FAO                 :
                                               7E 7C 001D6        CLRQ     -(SP)                       : 2862
                                               7E 7C 001D8        CLRQ     -(SP)                       :
                                           6C  AE 9F 001DA        PUSHAB   FNA_DESC                    :
                                         FF58  CD 9F 001DD        PUSHAB   FIB_DESC                    :
                                               7E 7C 001E1        CLRQ     -(SP)                       :
                                           B0  AD 9F 001E3        PUSHAB   IOSB                        :
                                               32 DD 001E6        PUSHL    #50                         :
                                    00000000'  EF DD 001E8        PUSHL    DIR_CHAN                    :
                                               7E D4 001EE        CLRL     -(SP)                       :
                     00000000G  00             0C FB 001F0        CALLS    #12, SYS$QIOW               :
                                59             50 D0 001F7        MOVL     R0, STATUS                  :
```

```
                                04                59 E9 001FA        BLBC    STATUS, 14$                           ; 2863
                                59          B0    AD 3C 001FD        MOVZWL  IOSB, STATUS                          ;
                          0910  8F                59 B1 00201  14$:  CMPW    STATUS, #2320                         ; 2870
                                                  08 12 00206        BNEQ    15$                                   ;
                                            18    A6 D4 00208        CLRL    24(D)                                 ; 2873
                                59                01 D0 0020B        MOVL    #1, STATUS                            ; 2874
                                                  46 11 0020E        BRB     16$                                   ; 2870
                     00000000'  EF                02 90 00210  15$:  MOVB    #2, DIR_STRUCLEV                      ; 2885
                            14  A6                02 D0 00217        MOVL    #2, 20(D)                             ; 2886
                            18  A6                01 D0 0021B        MOVL    #1, 24(D)                             ; 2887
                                50                62 3C 0021F        MOVZWL  (R2), R0                              ; 2892
                                                  50 D6 00222        INCL    R0                                    ;
                                50                01 8A 00224        BICB2   #1, L                                 ;
                                53          0C    A6 D0 00227        MOVL    12(D), P                              ; 2893
              63                50                0C A1 0022B        ADDW3   #12, L, (P)                           ; 2894
                            02  A3        0090    CE B0 0022F        MOVW    FIB+44, 2(P)                          ; 2895
                                            04    A3 94 00235        CLRB    4(P)                                  ; 2896
                            05  A3                62 90 00238        MOVB    (R2), 5(P)                            ; 2897
      50      00            04  B2                62 2C 0023C        MOVC5   (R2), @4(R2), #0, L, 6(P)             ; 2902
                                            06    A3    00242
                                63          38    A6 B0 00244        MOVW    56(D), (P)                            ; 2903
                            02  A3          68    AE D0 00248        MOVL    FIB+4, 2(P)                           ; 2904
                            06  A3          6C    AE B0 0024D        MOVW    FIB+8, 6(P)                           ; 2906
                                6A          08    A3 9E 00252        MOVAB   8(R3), (R10)                          ; 2909
                                35                59 E8 00256  16$:  BLBS    STATUS, 19$                           ; 2914
                                                  59 DD 00259        PUSHL   STATUS                                ; 2925
                                     00000000'    EF 95 0025B        TSTB    DIR_STRING                            ; 2924
                                                  07 12 00261        BNEQ    17$                                   ;
                                50        EE15    CF 9E 00263        MOVAB   MFD, R0                               ;
                                                  07 11 00268        BRB     18$                                   ;
                                50   00000000'    EF 9E 0026A  17$:  MOVAB   DIR_STRING, R0                        ;
                                                  50 DD 00271  18$:  PUSHL   R0                                    ;
                                     00000000'    EF DD 00273        PUSHL   DIR_DEV_DESC                          ; 2923
                                                  02 DD 00279        PUSHL   #2                                    ; 2920
                                     00000000G    8F DD 0027B        PUSHL   #BACKUP$_OPENDIR                      ;
                     00000000G  00                05 FB 00281        CALLS   #5, LIB$SIGNAL                        ;
                                            18    A6 D4 00288        CLRL    24(D)                                 ; 2931
                                                01D1 31 0028B        BRW     40$                                   ; 2914
                                            0C    A6 D5 0028E  19$:  TSTL    12(D)                                 ; 2940
                                                  03 13 00291        BEQL    20$                                   ;
                                                00A3 31 00293        BRW     27$                                   ;
                                50          18    A6 9E 00296  20$:  MOVAB   24(D), R0                             ; 2950
                                                  60 D4 0029A        CLRL    (R0)                                  ;
                     00000000'  EF        FF73    CD 90 0029C        MOVB    HEADER+7, DIR_STRUCLEV                ; 2951
                                02   00000000'    EF 91 002A5        CMPB    DIR_STRUCLEV, #2                      ; 2952
                                                  22 12 002AC        BNEQ    22$                                   ;
              4E            A1  AD                05 E1 002AE        BBC     #5, HEADER+53, 24$                    ; 2958
              60            88  AD                10 9C 002B3        ROTL    #16, RECATTR+8, (R0)                  ; 2961
                                            8C    AD B5 002B8        TSTW    RECATTR+12                            ; 2962
                                                  02 12 002BB        BNEQ    21$                                   ;
                                                  60 D7 002BD        DECL    (R0)                                  ; 2963
                            3A  A6          AC    AD B0 002BF  21$:  MOVW    HEADER+64, 58(D)                      ; 2964
                            3C  A6          A8    AD D0 002C4        MOVL    HEADER+60, 60(D)                      ; 2965
                            40  A6          9E    AD B0 002C9        MOVW    HEADER+50, 64(D)                      ; 2966
                                                  31 11 002CE        BRB     24$                                   ; 2952
                                01        FF7A    CD 91 002D0  22$:  CMPB    RECATTR, #1                           ; 2975
                                                  2A 12 002D5        BNEQ    24$                                   ;
```

```
                  10       FF7C   CD  B1 002D7       CMPW     RECATTR+2, #16                    ; 2976
                                  23  12 002DC       BNEQ     24$                               ;
     60       82  AD              10  9C 002DE       ROTL     #16, RECATTR+8, (R0)              ; 2979
                             86   AD  B5 002E3       TSTW     RECATTR+12                        ; 2980
                                  02  12 002E6       BNEQ     23$                               ;
                                  60  D7 002E8       DECL     (R0)                              ; 2981
              3A  A6       FF76   CD  B0 002EA 23$:  MOVW     HEADER+10, 58(D)                  ; 2982
              3C  A6       FF74   CD  9A 002F0       MOVZBL   HEADER+8, 60(D)                   ; 2983
              3E  A6       FF75   CD  9B 002F6       MOVZBW   HEADER+9, 62(D)                   ; 2984
              40  A6         98   AD  B0 002FC       MOVW     HEADER+44, 64(D)                  ; 2985
                                  60  D5 00301 24$:  TSTL     (R0)                              ; 2993
                                  34  13 00303       BEQL     27$                               ;
08   A6           60              09  78 00305       ASHL     #9, (R0), 8(D)                    ; 3002
        00002000  8F         08   A6  D1 0030A       CMPL     8(D), #8192                       ; 3004
                                  17  1B 00312       BLEQU    26$                               ;
                  01  00000000'   EF  91 00314       CMPB     DIR_STRUCLEV, #1                  ; 3005
                                  08  12 0031B       BNEQ     25$                               ;
                      00000000'   EF  D5 0031D       TSTL     DIR_SEL_LATEST                    ;
                                  06  15 00323       BLEQ     26$                               ;
              08  A6       2000   8F  3C 00325 25$:  MOVZWL   #8192, 8(D)                       ; 3007
                             08   A6  DD 0032B 26$:  PUSHL    8(D)                              ; 3012
        00000000G 00              01  FB 0032E       CALLS    #1, GET_VM                        ;
              0C  A6              50  D0 00335       MOVL     R0, 12(D)                         ;
              18  A6         14   A6  D1 00339 27$:  CMPL     20(D), 24(D)                      ; 3020
                                  2B  1A 0033E       BGTRU    30$                               ;
     50       18  A6         14   A6  C3 00340       SUBL3    20(D), 24(D), R0                  ; 3032
     50           50              09  78 00346       ASHL     #9, R0, R0                        ;
                  51       0200   C0  9E 0034A       MOVAB    512(R0), R1                       ;
                  50         08   A6  D0 0034F       MOVL     8(D), R0                          ;
                  51              50  D1 00353       CMPL     R0, R1                            ;
                                  03  1B 00356       BLEQU    28$                               ;
                  50              51  D0 00358       MOVL     R1, R0                            ;
                  6A         0C B640  9E 0035B 28$:  MOVAB    @12(D)[PROC_LENGTH], (R10)        ; 3033
                                  66  7C 00360       CLRQ     (D)                               ; 3034
                  53         0C   A6  D0 00362       MOVL     12(D), READ_ADDRESS               ; 3042
                  6A              53  D1 00366 29$:  CMPL     READ_ADDRESS, (R10)               ; 3043
                                  03  1F 00369       BLSSU    31$                               ;
                                00B3  31 0036B 30$:  BRW      38$                               ;
     50           6A              53  C3 0036E 31$:  SUBL3    READ_ADDRESS, (R10), R0           ; 3053
        0000FE00  8F              50  D1 00372       CMPL     R0, #65024                        ; 3052
                                  05  1B 00379       BLEQU    32$                               ;
                  50       FE00   8F  3C 0037B       MOVZWL   #65024, R0                        ;
                  52              50  D0 00380 32$:  MOVL     R0, READ_LENGTH                   ; 3051
                  50  00000000'   EF  D0 00383       MOVL     DIR_CHAN, R0                      ; 3064
        0001FFFF  8F              50  D1 0038A       CMPL     R0, #131071                       ;
                                  1F  1F 00391       BLSSU    33$                               ;
                                  7E  7C 00393       CLRQ     -(SP)                             ;
                                  7E  D4 00395       CLRL     -(SP)                             ;
                             14   A6  DD 00397       PUSHL    20(D)                             ;
                                  52  DD 0039A       PUSHL    READ_LENGTH                       ;
                                  53  DD 0039C       PUSHL    READ_ADDRESS                      ;
                                  7E  7C 0039E       CLRQ     -(SP)                             ;
                             B0   AD  9F 003A0       PUSHAB   IOSB                              ;
                                  31  DD 003A3       PUSHL    #49                               ;
                                  50  DD 003A5       PUSHL    R0                                ;
                                  7E  D4 003A7       CLRL     -(SP)                             ;
        00000000G 00              0C  FB 003A9       CALLS    #12, STA_QIOW                     ;
```

```
                        1D  11 003B0          BRB      34$
                        7E  7C 003B2 33$:     CLRQ     -(SP)
                        7E  D4 003B4          CLRL     -(SP)
                    14  A6  DD 003B6          PUSHL    20(D)
                        52  DD 003B9          PUSHL    READ_LENGTH
                        53  DD 003BB          PUSHL    READ_ADDRESS
                        7E  7C 003BD          CLRQ     -(SP)
                    B0  AD  9F 003BF          PUSHAB   IOSB
                        31  DD 003C2          PUSHL    #49
                        50  DD 003C4          PUSHL    R0
                        7E  D4 003C6          CLRL     -(SP)
00000000G 9F            0C  FB 003C8          CALLS    #12, @#SYS$QIOW
          59            50  D0 003CF 34$:     MOVL     R0, STATUS
          53            52  C0 003D2          ADDL2    READ_LENGTH, READ_ADDRESS                   3065
          52  00000200  8F  C6 003D5          DIVL2    #512, R2                                    3066
       14 A6            52  C0 003DC          ADDL2    R2, 20(D)
          0A            59  E9 003E0          BLBC     STATUS, 35$                                 3067
          59        B0  AD  3C 003E3          MOVZWL   IOSB, STATUS
          03            59  E9 003E7          BLBC     STATUS, 35$                                 3068
                      FF79  31 003EA          BRW      29$
                        59  DD 003ED 35$:     PUSHL    STATUS                                      3076
             00000000'  EF  95 003EF          TSTB     DIR_STRING                                  3075
                        07  12 003F5          BNEQ     36$
          50      EC81  CF  9E 003F7          MOVAB    MFD, R0
                        07  11 003FC          BRB      37$
          50 00000000'  EF  9E 003FE 36$:     MOVAB    DIR_STRING, R0
                        50  DD 00405 37$:     PUSHL    R0
             00000000'  EF  DD 00407          PUSHL    DIR_DEV_DESC                                3074
                        02  DD 0040D          PUSHL    #2                                          3071
             00000000G  8F  DD 0040F          PUSHL    #BACKUP$_READDIR
00000000G 00            05  FB 00415          CALLS    #5, LIB$SIGNAL
                    18  A6  D4 0041C          CLRL     24(D)                                       3082
                        6A  D4 0041F          CLRL     (R10)                                       3083
          50 00000000'  EF  D0 00421 38$:     MOVL     DIR_CHAN, R0                                3094
0001FFFF  8F            50  D1 00428          CMPL     R0, #131071
                        18  1F 0042F          BLSSU    39$
                        7E  7C 00431          CLRQ     -(SP)
                        7E  7C 00433          CLRQ     -(SP)
                        7E  7C 00435          CLRQ     -(SP)
                        7E  7C 00437          CLRQ     -(SP)
          7E            34  7D 00439          MOVQ     #52, -(SP)
                        50  DD 0043C          PUSHL    R0
                        7E  D4 0043E          CLRL     -(SP)
00000000G 00            0C  FB 00440          CALLS    #12, STA_QIOW
                        16  11 00447          BRB      40$
                        7E  7C 00449 39$:     CLRQ     -(SP)
                        7E  7C 0044B          CLRQ     -(SP)
                        7E  7C 0044D          CLRQ     -(SP)
                        7E  7C 0044F          CLRQ     -(SP)
          7E            34  7D 00451          MOVQ     #52, -(SP)
                        50  DD 00454          PUSHL    R0
                        7E  D4 00456          CLRL     -(SP)
00000000G 9F            0C  FB 00458          CALLS    #12, @#SYS$QIOW
          6A            66  D1 0045F 40$:     CMPL     (D), (R10)                                  3101
                        6A  1F 00462          BLSSU    45$
                    0C  A6  D5 00464          TSTL     12(D)                                       3107
                        0B  13 00467          BEQL     41$
```

```
                                7E          08    A6  7D 00469            MOVQ    8(D), -(SP)                      ; 3109
                     00000000G  00                02  FB 0046D            CALLS   #2, FREE_VM                      ;
                                            28    A6  D5 00474 41$:       TSTL    40(D)                            ; 3110
                                                  0E  13 00477            BEQL    42$                              ;
                                            28    A6  DD 00479            PUSHL   40(D)                            ; 3112
                                7E          50    8F  9A 0047C            MOVZBL  #80, -(SP)                       ;
                     00000000G  00                02  FB 00480            CALLS   #2, FREE_VM                      ;
                                            30    A6  D5 00487 42$:       TSTL    48(D)                            ; 3113
                                                  0E  13 0048A            BEQL    43$                              ;
                                            30    A6  DD 0048C            PUSHL   48(D)                            ; 3115
                                7E          50    8F  9A 0048F            MOVZBL  #80, -(SP)                       ;
                     00000000G  00                02  FB 00493            CALLS   #2, FREE_VM                      ;
                     00000000'  EF          22    A6  90 0049A 43$:       MOVB    34(D), DIR_STRING                ; 3120
   0044    8F          00       6E                00  2C 004A2            MOVC5   #0, (SP), #0, #68, (D)           ; 3121
                                                  66     004A9                                                     ;
                                56                3C  C2 004AA            SUBL2   #60, D                           ; 3122
                     00000000'  EF                76  7E 004AD            MOVAQ   -(D), DIR_SP                     ;
                                     00000000'    EF  97 004B4            DECB    DIR_LEVELS                       ; 3123
                                                  12  12 004BA            BNEQ    45$                              ; 3128
                                            03    A8  94 004BC 44$:       CLRB    3(N)                             ; 3131
                                            26    A8  D4 004BF            CLRL    38(N)                            ; 3132
                                            24    A8  B4 004C2            CLRW    36(N)                            ;
                                            2C    A8  D4 004C5            CLRL    44(N)                            ; 3133
                                            2A    A8  B4 004C8            CLRW    42(N)                            ;
                                                03B2  31 004CB            BRW     87$                              ; 3134
                                57                66  D0 004CE 45$:       MOVL    (D), R                           ; 3141
                                59          04    A6  D0 004D1            MOVL    4(D), V                          ; 3142
                  03 00000000'  EF                01  E5 004D5            BBCC    #1, DIR_FLAGS, 46$               ; 3143
                                                012B  31 004DD            BRW     63$                              ;
                             01 00000000'     EF  91 004E0 46$:       CMPB    DIR_STRUCLEV, #1                 ; 3146
                                                  5B  12 004E7            BNEQ    54$                              ;
                                                  57  D5 004E9 47$:       TSTL    R                                ; 3154
                                                  06  12 004EB            BNEQ    48$                              ;
                                57          0C    A6  D0 004ED            MOVL    12(D), R                         ; 3156
                                                  05  11 004F1            BRB     49$                              ;
                                57                10  C0 004F3 48$:       ADDL2   #16, R                           ; 3159
                                                  59  D6 004F6            INCL    V                                ; 3160
                           10   A6                57  D1 004F8 49$:       CMPL    R, 16(D)                         ; 3166
                                                  6F  1E 004FC            BGEQU   57$                              ;
                                                  67  B5 004FE            TSTW    (R)                              ; 3171
                                                  E7  13 00500            BEQL    47$                              ;
                                            42    A6  B4 00502            CLRW    66(D)                            ; 3179
                                     00000000'    EF  D5 00505            TSTL    DIR_SEL_LATEST                   ; 3180
                                                  34  14 0050B            BGTR    53$                              ;
                51         0C   A6          08    A6  C1 0050D            ADDL3   8(D), 12(D), R1                  ; 3184
                                51                10  C2 00513            SUBL2   #16, R1                          ;
                                50          0C    A6  D0 00516            MOVL    12(D), P                         ; 3191
                                                  20  11 0051A            BRB     52$                              ;
                           06   A7          06    A0  D1 0051C 50$:       CMPL    6(P), 6(R)                       ;
                                                  16  12 00521            BNEQ    51$                              ;
                           0A   A7          0A    A0  D1 00523            CMPL    10(P), 10(R)                     ; 3192
                                                  0F  12 00528            BNEQ    51$                              ;
                           0E   A7          0E    A0  B1 0052A            CMPW    14(P), 14(R)                     ; 3193
                                                  08  1B 0052F            BLEQU   51$                              ;
                           42   A6        8000    8F  B0 00531            MOVW    #-32768, 66(D)                   ; 3196
                                                  08  11 00537            BRB     53$                              ; 3195
                                50                10  C0 00539 51$:       ADDL2   #16, P                           ; 3182
```

```
                   51             50  D1 0053C 52$:      CMPL    P, R1                                  ;
                                  DB  1B 0053F           BLEQU   50$                                    ;
                                00C0  31 00541 53$:      BRW     62$                                    ;
                                  57  D5 00544 54$:      TSTL    R                                      ; 3219
                                  06  12 00546           BNEQ    55$                                    ;
                   57         0C  A6  D0 00548           MOVL    12(D), R                               ; 3221
                                  1B  11 0054C           BRB     56$                                    ;
                                  59  D5 0054E 55$:      TSTL    V                                      ; 3223
                                  17  13 00550           BEQL    56$                                    ;
                   50             67  3C 00552           MOVZWL  (R), R0                                ; 3226
                   50      02 A740    9E 00555           MOVAB   2(R)[R0], NEXT_RECORD                  ;
                   59             08  C0 0055A           ADDL2   #8, V                                  ; 3227
                   51         F8  A0  9E 0055D           MOVAB   -8(R0), R1                             ; 3228
                   51             59  D1 00561           CMPL    V, R1                                  ;
                                  DB  1B 00564           BLEQU   53$                                    ;
                   57             50  D0 00566           MOVL    NEXT_RECORD, R                         ; 3229
             10    A6             57  D1 00569 56$:      CMPL    R, 16(D)                               ; 3235
                                  D2  1E 0056D 57$:      BGEQU   53$                                    ;
                   51             67  3C 0056F           MOVZWL  (R), R1                                ; 3238
           FFFF    8F             51  B1 00572           CMPW    R1, #65535                             ;
                                  17  12 00577           BNEQ    58$                                    ;
     52            57         0C  A6  C3 00579           SUBL3   12(D), R, R2                           ; 3244
                   52       01FF  8F  AA 0057E           BICW2   #511, R2                               ;
                   52         0C  A6  C0 00583           ADDL2   12(D), R2                              ;
                   57       0200  C2  9E 00587           MOVAB   512(R2), R                             ;
                                  59  D4 0058C           CLRL    V                                      ; 3245
                                  B4  11 0058E           BRB     54$                                    ; 3238
                   50      02 A741    9E 00590 58$:      MOVAB   2(R)[R1], NEXT_RECORD                  ; 3253
     52            57         0C  A6  C3 00595           SUBL3   12(D), R, R2                           ; 3257
                   52       01FF  8F  AA 0059A           BICW2   #511, R2                               ;
                   52         0C  A6  C0 0059F           ADDL2   12(D), R2                              ;
                   52       0200  C2  9E 005A3           MOVAB   512(R2), R2                            ;
                   52             50  D1 005A8           CMPL    NEXT_RECORD, R2                        ;
                                  23  1E 005AB           BGEQU   59$                                    ;
                   20             51  E8 005AD           BLBS    R1, 59$                                ; 3259
                   0E             51  B1 005B0           CMPW    R1, #14                                ; 3260
                                  1B  1F 005B3           BLSSU   59$                                    ;
                   50         05  A7  9A 005B5           MOVZBL  5(R), R0                               ; 3266
                   50      07 A047    9E 005B9           MOVAB   7(R0)[R], R0                           ;
     59            50             01  CB 005BE           BICL3   #1, R0, V                              ;
                   07         04  A7  93 005C2           BITB    4(R), #7                               ; 3267
                                  08  12 005C6           BNEQ    59$                                    ;
                   52             08  C2 005C8           SUBL2   #8, R2                                 ; 3268
                   52             59  D1 005CB           CMPL    V, R2                                  ;
                                  34  1F 005CE           BLSSU   62$                                    ;
                        00000000' EF  95 005D0 59$:      TSTB    DIR_STRING                             ; 3281
                                  07  12 005D6           BNEQ    60$                                    ;
                   50       EAA0  CF  9E 005D8           MOVAB   MFD, R0                                ;
                                  07  11 005DD           BRB     61$                                    ;
                   50 00000000'   EF  9E 005DF 60$:      MOVAB   DIR_STRING, R0                         ;
                                  50  DD 005E6 61$:      PUSHL   R0                                     ;
                        00000000' EF  DD 005E8           PUSHL   DIR_DEV_DESC                           ; 3280
                                  02  DD 005EE           PUSHL   #2                                     ; 3277
                        00000000G 8F  DD 005F0           PUSHL   #BACKUP$_BADDIR                        ;
         00000000G 00             04  FB 005F6           CALLS   #4, LIB$SIGNAL                         ;
                   57         10  A6  D0 005FD           MOVL    16(D), R                               ; 3286
                              18  A6  D4 00601           CLRL    24(D)                                  ; 3287
```

```
                          66           57  D0 00604 62$:       MOVL    R, (D)                                   : 3297
                    04    A6           59  D0 00607            MOVL    V, 4(D)                                  : 3298
                    10    A6           57  D1 0060B 63$:       CMPL    R, 16(D)                                 : 3304
                                       03  1F 0060F            BLSSU   64$                                      :
                                     FA37  31 00611            BRW     2$                                       :
                    5B          24     A8  9E 00614 64$:       MOVAB   36(N), R11                               : 3340
                    02  00000000'      EF  91 00618            CMPB    DIR_STRUCLEV, #2                         : 3317
                                       48  12 0061F            BNEQ    67$                                      :
                                42     A6  B7 00621            DECW    66(D)                                    : 3320
                    55          05     A7  9A 00624            MOVZBL  5(R), R5                                 : 3322
                    54          2C     A6  9E 00628            MOVAB   44(D), R4                                :
                    55                 64  B1 0062C            CMPW    (R4), R5                                 :
                                       08  12 0062F            BNEQ    65$                                      :
   04     B4        06    A7           55  29 00631            CMPC3   R5, 6(R), @4(R4)                         : 3325
                                       0C  13 00637            BEQL    66$                                      :
                                42     A6  B4 00639 65$:       CLRW    66(D)                                    : 3330
                          64           55  B0 0063C            MOVW    R5, (R4)                                 : 3331
   04     B4        06    A7           55  28 0063F            MOVC3   R5, 6(R), @4(R4)                         : 3335
                          5A    05     A7  9E 00645 66$:       MOVAB   5(R), NAME                               : 3337
                          6E           69  32 00649            CVTWL   (V), VERSION                             : 3338
              00000000'   EF    02     A7  B0 0064C            MOVW    2(R), DIR_VERLIMIT                       : 3339
                          6B    02     A9  B0 00654            MOVW    2(V), (R11)                              : 3340
                    26    A8    04     A9  D0 00658            MOVL    4(V), 38(N)                              : 3341
                                28     A8  95 0065D            TSTB    40(N)                                    : 3343
                                       39  12 00660            BNEQ    69$                                      :
                    28    A8    20     A6  90 00662            MOVB    32(D), 40(N)                             : 3344
                                       32  11 00667            BRB     69$                                      : 3317
                                E5     AD  9F 00669 67$:       PUSHAB  FILE_NAME+1                              : 3348
                                       57  DD 0066C            PUSHL   R                                        :
              00000000G   00           02  FB 0066E            CALLS   #2, MAKE_STRING                          :
   E5     AD              50           3B  3A 00675            LOCC    #59, T, FILE_NAME+1                      : 3349
                                       02  12 0067A            BNEQ    68$                                      :
                                       51  D4 0067C            CLRL    R1                                       :
                          50    E5     AD  9E 0067E 68$:       MOVAB   FILE_NAME+1, R0                          :
   E4     AD              51           50  83 00682            SUBB3   R0, R1, FILE_NAME                        :
                          5A    E4     AD  9E 00687            MOVAB   FILE_NAME, NAME                          : 3350
                          6E    0E     A7  32 0068B            CVTWL   14(R), VERSION                           : 3351
                          6B           67  B0 0068F            MOVW    (R), (R11)                               : 3352
                    26    A8    02     A7  B0 00692            MOVW    2(R), 38(N)                              : 3353
                    28    A8           01  B0 00697            MOVW    #1, 40(N)                                : 3354
                    F8    AD    02     A8  9A 0069B 69$:       MOVZBL  2(N), RSA_DESC                           : 3360
                    FC    AD    04     A8  D0 006A0            MOVL    4(N), RSA_DESC+4                         : 3361
                                       6E  DD 006A5            PUSHL   VERSION                                  : 3369
                                       5A  DD 006A7            PUSHL   NAME                                     :
                          00000000'    EF  95 006A9            TSTB    DIR_STRING                               :
                                       07  12 006AF            BNEQ    70$                                      :
                          50      E9C7 CF  9E 006B1            MOVAB   MFD, R0                                  :
                                       07  11 006B6            BRB     71$                                      :
                          50 00000000' EF  9E 006B8 70$:       MOVAB   DIR_STRING, R0                           :
                                       50  DD 006BF 71$:       PUSHL   R0                                       :
                          00000000'    EF  DD 006C1            PUSHL   DIR_DEV_DESC                             :
                                F8     AD  9F 006C7            PUSHAB  RSA_DESC                                 :
                                F8     AD  9F 006CA            PUSHAB  RSA_DESC                                 :
                              F927     CF  9F 006CD            PUSHAB  P.AAK                                    :
              00000000G   00           07  FB 006D1            CALLS   #7, SYS$FAO                              :
                    03    A8    F8     AD  90 006D8            MOVB    RSA_DESC, 3(N)                           : 3370
                                       58  DD 006DD            PUSHL   N                                        : 3371
```

```
                 00000000G  00           01  FB 006DF          CALLS   #1, INIT_NAMEBLOCK                          ;
                            50       1C  A6  9E 006E6          MOVAB   28(D), R0                                   ; 3376
                       2A   A8           60  D0 006EA          MOVL    (R0), 42(N)                                 ;
                       2E   A8       04  A0  B0 006EE          MOVW    4(R0), 46(N)                                ; 3378
                                00000000' EF  94 006F3         CLRB    DIR_STATUS                                  ; 3383
                         08  00000000' EF  91 006F9           CMPB    DIR_LEVELS, #8                              ; 3385
                                         1F  1A 00700          BGTRU   72$                                         ;
                            04       3C  A8  91 00702          CMPB    60(N), #4                                   ; 3386
                                         19  12 00706          BNEQ    72$                                         ;
                 5249442E   8F       50  B8  D1 00708          CMPL    @80(N), #1380533294                         ; 3387
                                         0F  12 00710          BNEQ    72$                                         ;
                            01           6E  D1 00712          CMPL    VERSION, #1                                 ; 3389
                                         0A  12 00715          BNEQ    72$                                         ;
                            04           6B  B1 00717          CMPW    (R11), #4                                   ; 3390
                                         08  12 0071A          BNEQ    73$                                         ;
                                     29  A8  95 0071C          TSTB    41(N)                                       ; 3391
                                         03  12 0071F          BNEQ    73$                                         ;
                                       00F0  31 00721  72$:    BRW     81$                                         ;
                 00000000'  EF           01  88 00724  73$:    BISB2   #1, DIR_STATUS                              ; 3401
                       DC   AD  00000000' EF  9A 0072B         MOVZBL  DIR_STRING, DIR_DESC                        ; 3407
                       E0   AD  00000000' EF  9E 00733         MOVAB   DIR_STRING+1, DIR_DESC+4                    ; 3408
                                     DC  AD  D5 0073B          TSTL    DIR_DESC                                    ; 3409
                                         0F  13 0073E          BEQL    74$                                         ;
                                     DC  AD  D6 00740          INCL    DIR_DESC                                    ; 3412
                            50  00000000' EF  9E 00743         MOVAB   DIR_STRING, R0                              ; 3413
                    DC BD40              2E  90 0074A          MOVB    #46, @DIR_DESC[R0]                          ;
                            51       3B  A8  9A 0074F  74$:    MOVZBL  59(N), R1                                   ; 3416
                            50  00000000' EF  9E 00753         MOVAB   DIR_STRING+1, R0                            ; 3418
           DC BD40     4C   B8           51  28 0075A          MOVC3   R1, @76(N), @DIR_DESC[R0]                   ;
                            50       3B  A8  9A 00761          MOVZBL  59(N), R0                                   ; 3419
                       DC   AD           50  C0 00765          ADDL2   R0, DIR_DESC                                ;
                            56       44  A6  9E 00769          MOVAB   68(R6), D                                   ; 3424
                            7E       50  8F  9A 0076D          MOVZBL  #80, -(SP)                                  ; 3429
                 00000000G  00           01  FB 00771          CALLS   #1, GET_VM                                  ;
                       28   A6           50  D0 00778          MOVL    R0, 40(D)                                   ;
                            7E       50  8F  9A 0077C          MOVZBL  #80, -(SP)                                  ; 3430
                 00000000G  00           01  FB 00780          CALLS   #1, GET_VM                                  ;
                       30   A6           50  D0 00787          MOVL    R0, 48(D)                                   ;
                            05  00000000' EF  E9 0078B         BLBC    DIR_FLAGS, 75$                              ; 3437
                            50           07  D0 00792          MOVL    #7, STATUS                                  ; 3439
                                         1C  11 00795          BRB     76$                                         ;
                                00000000' EF  9F 00797  75$:   PUSHAB  DIR_SEL_NTV                                 ; 3445
                                     38  A6  9F 0079D          PUSHAB  56(D)                                       ;
                                     24  A6  9F 007A0          PUSHAB  36(R6)                                      ; 3444
                                00000000' EF  9F 007A3         PUSHAB  DIR_SEL_DIR                                 ; 3441
                                     DC  AD  9F 007A9          PUSHAB  DIR_DESC                                    ;
                 00000000G  00           05  FB 007AC          CALLS   #5, MATCH_DIRECTORY                         ; 3445
               32           50           01  E1 007B3  76$:    BBC     #1, STATUS, 79$                             ; 3449
                            51  00000000' EF  D0 007B7         MOVL    DIR_SP, R1                                  ; 3452
                 00000000'  EF           04  88 007BE          BISB2   #4, DIR_STATUS                              ;
                       23   A1           01  88 007C5          BISB2   #1, 35(R1)                                  ;
                            0B           50  E9 007C9          BLBC    STATUS, 77$                                 ; 3453
                 00000000'  EF           08  88 007CC          BISB2   #8, DIR_STATUS                              ;
                       23   A6           02  88 007D3          BISB2   #2, 35(D)                                   ;
               04           50           02  E1 007D7  77$:    BBC     #2, STATUS, 78$                             ; 3454
                       23   A6           04  88 007DB          BISB2   #4, 35(D)                                   ;
               2A           50           03  E1 007DF  78$:    BBC     #3, STATUS, 80$                             ; 3455
```

```
                      23   A6         08  88 007E3         BISB2   #8, 35(D)                         ;
                                      24  11 007E7         BRB     80$                               ; 3449
                                 28   A6  DD 007E9 79$:    PUSHL   40(D)                             ; 3459
                           7E    50   8F  9A 007EC         MOVZBL  #80, -(SP)                        ;
                 00000000G 00         02  FB 007F0         CALLS   #2, FREE_VM                       ;
                                 30   A6  DD 007F7         PUSHL   48(D)                             ; 3460
                           7E    50   8F  9A 007FA         MOVZBL  #80, -(SP)                        ;
                 00000000G 00         02  FB 007FE         CALLS   #2, FREE_VM                       ;
  0044   8F          00    6E         00  2C 00805         MOVC5   #0, (SP), #0, #68, (D)            ; 3461
                                      66     0080C                                                   ;
                        56 00000000'  EF  D0 0080D 80$:    MOVL    DIR_SP, D                         ; 3467
                        15 00000000'  EF  E9 00814 81$:    BLBC    DIR_FLAGS, 84$                    ; 3474
                03 00000000'   EF         03  E0 0081B     BBS     #3, DIR_FLAGS, 83$                ; 3475
                                    00B8  31 00823 82$:    BRW     92$                               ;
                           04         6B  B1 00826 83$:    CMPW    (R11), #4                         ; 3476
                                      F8  12 00829         BNEQ    82$                               ;
                                 29   A8  95 0082B         TSTB    41(N)                             ; 3477
                                      F3  12 0082E         BNEQ    82$                               ;
                        01 00000000'  EF  91 00830 84$:    CMPB    DIR_STRUCLEV, #1                  ; 3490
                                      13  12 00837         BNEQ    85$                               ;
                        50 00000000'  EF  9A 00839         MOVZBL  DIR_LEVELS, R0                    ; 3492
                 00000000'EF40        59  D1 00840         CMPL    V, DIR_SCANLIMIT-4[R0]            ;
                                      40  1B 00848         BLEQU   89$                               ;
                                      25  11 0084A         BRB     86$                               ;
                           50    24   A6  9E 0084C 85$:    MOVAB   36(D), R0                         ; 3494
                                      60  B5 00850         TSTW    (R0)                              ;
                                      36  13 00852         BEQL    89$                               ;
                           7E    38   A6  3C 00854         MOVZWL  56(D), -(SP)                      ; 3503
                                 04   A0  DD 00858         PUSHL   4(R0)                             ; 3502
                           7E         60  3C 0085B         MOVZWL  (R0), -(SP)                       ; 3501
                                 0C   AE  DD 0085E         PUSHL   VERSION                           ; 3500
                                 01   AA  9F 00861         PUSHAB  1(NAME)                           ; 3499
                           7E         6A  9A 00864         MOVZBL  (NAME), -(SP)                     ;
                 00000000G 00         06  FB 00867         CALLS   #6, TERMINATE_SCAN                ;
                           19         50  E9 0086E         BLBC    R0, 89$                           ;
              0A 00000000'  EF        02  E1 00871 86$:    BBC     #2, DIR_FLAGS, 88$                ; 3507
                 00000000'  EF        02  88 00879         BISB2   #2, DIR_FLAGS                     ; 3510
                                      50  D4 00880 87$:    CLRL    R0                                ; 3511
                                      04     00882         RET                                       ;
                           66    10   A6  D0 00883 88$:    MOVL    16(D), (D)                        ; 3517
                                 18   A6  D4 00887         CLRL    24(D)                             ; 3518
                      10   A6         66  D1 0088A 89$:    CMPL    (D), 16(D)                        ; 3525
                                      5E  1E 0088E         BGEQU   93$                               ;
              59      23   A6         01  E1 00890         BBC     #1, 35(D), 93$                    ; 3526
                        50 00000000'  EF  D0 00895         MOVL    DIR_SEL_LATEST, R0                ; 3527
                                      08  14 0089C         BGTR    90$                               ;
       50        42   A6         10   00  EC 0089E         CMPV    #0, #16, 66(D), R0                ;
                                      48  12 008A4         BNEQ    93$                               ;
              0A 00000000'  EF        03  E1 008A6 90$:    BBC     #3, DIR_FLAGS, 91$                ; 3528
                           04         6B  B1 008AE         CMPW    (R11), #4                         ; 3529
                                      05  12 008B1         BNEQ    91$                               ;
                                 29   A8  95 008B3         TSTB    41(N)                             ; 3530
                                      36  13 008B6         BEQL    93$                               ;
                           50    03   A8  9A 008B8 91$:    MOVZBL  3(N), R0                          ; 3539
                           50    04   A8  C0 008BC         ADDL2   4(N), R0                          ;
              DC   AD      50    4C   A8  C3 008C0         SUBL3   76(N), R0, NTV_DESC               ;
                      E0   AD    4C   A8  D0 008C6         MOVL    76(N), NTV_DESC+4                 ; 3540
```

```
                            00000000'  EF   9F 008CB          PUSHAB  DIR_SEL_NTV                                  ; 3541
                                   DC  AD   9F 008D1          PUSHAB  NTV_DESC                                     ;
             00000000G  00              02   FB 008D4          CALLS   #2, MATCH_FILENAME                           ;
                        10              50   E9 008DB          BLBC    R0, 93$                                      ;
                        1D  00000000'   EF   E9 008DE 92$:     BLBC    DIR_STATUS, 96$                              ; 3544
             00000000'  EF              02   88 008E5          BISB2   #2, DIR_STATUS                               ; 3545
                                        14   11 008EC          BRB     96$                                          ; 3543
          03 00000000'  EF              03   E0 008EE 93$:     BBS     #3, DIR_FLAGS, 95$                           ; 3554
                                      F752   31 008F6 94$:     BRW     2$                                           ;
                        F9          23  A6   E9 008F9 95$:     BLBC    35(D), 94$                                   ;
        F4          23  A6              03   E1 008FD          BBC     #3, 35(D), 94$                               ;
                        50              01   D0 00902 96$:     MOVL    #1, R0                                       ; 3562
                                             04 00905          RET                                                  ;

; Routine Size:  2310 bytes,    Routine Base:  CODE + 0F84
```

```
1 %SBTTL 'FREE_DIR_DATA - free directory scan context'
1 GLOBAL ROUTINE FREE_DIR_DATA: NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine deletes the directory scan context.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       Directory scan context.
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       Directory scan context.
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 ! Free any dynamic storage that is currently allocated.
2 !
2 INCRA D FROM DIR_STACK TO DIR_STACK+D_K_NLEVELS*D_S_ENTRY-D_S_ENTRY BY D_S_ENTRY DO
3     BEGIN
3     MAP
3         D:      REF BBLOCK;     ! Pointer to directory stack entry
3
3     IF .D[D_BUF_ADDR] NEQ 0
3     THEN
3         FREE_VM(.D[D_BUF_LEN], .D[D_BUF_ADDR]);
3     IF .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER] NEQ 0
3     THEN
3         FREE_VM(DIR$S_NAME, .BBLOCK[D[D_TERM_DESC], DSC$A_POINTER]);
3     IF .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER] NEQ 0
3     THEN
3         FREE_VM(DIR$S_NAME, .BBLOCK[D[D_NAME_DESC], DSC$A_POINTER]);
2     END;
2
2
2 ! Reinitialize the impure storage.
2 !
2 CH$FILL(0, DIR_END-DIR_BEG, DIR_BEG);
1 END;
```

```
                                         00FC 00000        .ENTRY  FREE_DIR_DATA, Save R2,R3,R4,R5,R6,R7          ; 3564
                   57 00000000'   EF   9E 00002            MOVAB   DIR_STACK, R7                                  ;
                   56 00000000G   00   9E 00009            MOVAB   FREE_VM, R6                                    ;
                   52             67   9E 00010            MOVAB   DIR_STACK, R2                                  ; 3595
                   53       0220  C7   9E 00013            MOVAB   DIR_STACK+544, R3                              ;
                                  2E   11 00018            BRB     5$                                             ;
                              0C  A2   D5 0001A 1$:        TSTL    12(D)                                          ; 3600
                                  07   13 0001D            BEQL    2$                                             ;
                   7E         08  A2   7D 0001F            MOVQ    8(D), -(SP)                                    ; 3602
                   66             02   FB 00023            CALLS   #2, FREE_VM                                    ;
                              28  A2   D5 00026 2$:        TSTL    40(D)                                          ; 3603
                                  0A   13 00029            BEQL    3$                                             ;
                              28  A2   DD 0002B            PUSHL   40(D)                                          ; 3605
                   7E         50  8F   9A 0002E            MOVZBL  #80, -(SP)                                     ;
                   66             02   FB 00032            CALLS   #2, FREE_VM                                    ;
                              30  A2   D5 00035 3$:        TSTL    48(D)                                          ; 3606
                                  0A   13 00038            BEQL    4$                                             ;
                              30  A2   DD 0003A            PUSHL   48(D)                                          ; 3608
                   7E         50  8F   9A 0003D            MOVZBL  #80, -(SP)                                     ;
                   66             02   FB 00041            CALLS   #2, FREE_VM                                    ;
                   52         44  A2   9E 00044 4$:        MOVAB   68(R2), D                                      ; 3595
                   53             52   D1 00048 5$:        CMPL    D, R3                                          ;
                                  CD   1B 0004B            BLEQU   1$                                             ;
  03CC   8F          00    6E     00   2C 0004D            MOVC5   #0, (SP), #0, #972, DIR_BEG                    ; 3614
                            FEA0  C7      00054                                                                   ;
                                       04 00057            RET                                                    ; 3615

; Routine Size:  88 bytes,    Routine Base:  CODE + 188A
; 2523            3616  1 END
; 2524            3617  0 ELUDOM



                                                                                           .EXTRN  LIB$SIGNAL

;                                        PSECT SUMMARY
;
;       Name                        Bytes                          Attributes
;
;   COMMON                              2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;   CODE                                6370  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                  Library Statistics
;
;                                                    -------- Symbols --------          Pages        Processing
;       File                                         Total    Loaded   Percent          Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1                    18619       196         1          1000           00:02.3



;                                        COMMAND QUALIFIERS

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

; Size:          6262 code + 2232 data bytes
; Run Time:         01:53.6
; Elapsed Time:     06:24.4
; Lines/CPU Min:      1911
; Lexemes/CPU-Min: 26165
; Memory Used:  875 pages
; Compilation Complete
```

0011 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIST
LIS

JOURNAL
LIS

FASTSCAN
LIS

[illegible]