```
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**READSAVE

RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD     SSSSSSSS    AAAAAA    VV      VV  EEEEEEEEEE
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD     SSSSSSSS    AAAAAA    VV      VV  EEEEEEEEEE
RR      RR  EE          AA      AA  DD      DD  SS          AA      AA  VV      VV  EE
RR      RR  EE          AA      AA  DD      DD  SS          AA      AA  VV      VV  EE
RR      RR  EE          AA      AA  DD      DD  SS          AA      AA  VV      VV  EE
RR      RR  EE          AA      AA  DD      DD  SS          AA      AA  VV      VV  EE
RRRRRRRR    EEEEEEEE    AA      AA  DD      DD   SSSSSS     AA      AA  VV      VV  EEEEEEEE
RRRRRRRR    EEEEEEEE    AA      AA  DD      DD   SSSSSS     AA      AA  VV      VV  EEEEEEEE
RR  RR      EE          AAAAAAAAAA  DD      DD         SS   AAAAAAAAAA  VV      VV  EE
RR  RR      EE          AAAAAAAAAA  DD      DD         SS   AAAAAAAAAA  VV      VV  EE
RR    RR    EE          AA      AA  DD      DD         SS   AA      AA   VV  VV    EE              ....
RR    RR    EE          AA      AA  DD      DD         SS   AA      AA   VV  VV    EE              ....
RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD    SSSSSSSS    AA      AA     VV      EEEEEEEEEE      ....
RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD    SSSSSSSS    AA      AA     VV      EEEEEEEEEE      ....

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 READSAVE (%TITLE 'Read Save Set'
0                 IDENT = 'V04-001'
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 !
1 !       This module contains routines to do the I/O involved in reading
1 !       save sets.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS user mode
1 !
1 !--
1 !
1 ! AUTHOR:  Andrew C. Goldstein, CREATION DATE:  2-Sep-1980  19:17
1 !
1 ! MODIFIED BY:
1 !
1 !       V04-001 LY0527          Larry Yetto             5-SEP-1984 11:12
1 !               Correct back space handling in INIT_SAVE_TAPE and FIN_IN_SAVE so that
1 !               they are consistant with the modifications previously done in
1 !               WRITESAVE.  We now backspace only 2 tape marks instead of three
1 !               at the end of a save set.
1 !
1 !       V03-014 LMP0272         L. Mark Pilant,         6-Jul-1984  8:48
1 !               Modify BACKUP to always use a full FIB.
```

```
!
!	V03-013	LMP0204		L. Mark Pilant,		7-Mar-1984  9:15
!		Correctly handle multi-reel save sets during a wildcard
!		restore or list.
!
!	V03-011	ACG0370		Andrew C. Goldstein,	8-Nov-1983  17:14
!		Fix block skip computation in tape repositioning
!
!	V03-010	ACG0332		Andrew C. Goldstein,	2-May-1983  13:39
!		Remove .B32 from BACKDEF require file
!
!	V03-009	ACG0328		Andrew C. Goldstein,	11-Apr-1983  16:06
!		Improve repositioning over bad tape
!
!	V03-008	ACG0313		Andrew C. Goldstein,	10-Feb-1983  21:40
!		Add another layer of routine around READY_NEXT_VOLUME
!		so that miscellaneous callers of it don't get the version
!		with the handler.
!
!	V03-007	MLJ0104		Martin L. Jack, 31-Jan-1983  4:18
!		Lower the severity of errors that occur during mounting
!		continuation volumes to informational so that they do not
!		affect the image return status.
!
!	V03-006	LMP0040		L. Mark Pilant,		20-Jul-1982  16:50
!		Correct a problem with the linkage to FMG$MATCH_NAME in the
!		routine MATCH_SSNAME.
!
!	V03-005	LMP0032		L. Mark Pilant,		21-Jun-1982  12:20
!		Add support for wildcard save set names on a LIST or
!		RESTORE operation.
!
!	V03-004	ACG0293		Andrew C. Goldstein,	7-Jun-1982  15:49
!		Fix handling of backspacing into BOT on verify pass
!
!	V03-003	ACG0280		Andrew C. Goldstein,	2-Apr-1982  16:48
!		Add dot to save set name with null type
!
!	V03-002	ACG0277		Andrew C. Goldstein,	30-Mar-1982  15:30
!		Check for spurious change in save set name
!
!	V03-001	ACG0276		Andrew C. Goldstein,	26-Mar-1982  18:23
!		Allow XOR groups to be one larger than indicated,
!		reject non-BACKUP blocks quietly
!
!	V02-012	ACG0235		Andrew C. Goldstein,	8-Dec-1981  21:35
!		Backspace tape over small save sets for verify
!
!	V02-011	MLJ0054		Martin L. Jack, 31-Oct-1981  15:03
!		Implement network save sets.  Move STAACP globals to common.
!
!	V02-010	ACG0217		Andrew C. Goldstein,	10-Sep-1981  18:01
!		Validate tape block size in reading save sets
!
!	V02-009	ACG0211		Andrew C. Goldstein,	29-Jul-1981  16:52
!		Implement sequential disk save sets
!
```

```
1 !          V02-008 ACG0209         Andrew C. Goldstein,    5-Jun-1981  15:09
1 !                  Stop on excessive input error rate
1 !
1 !          V02-007 MLJ0025         Martin L. Jack, 8-May-1981  13:48
1 !                  Reorganize qualifier database.  Move global variables into
1 !                  common.  Make routines non-global if possible.
1 !
1 !          V02-006 ACG0202         Andrew C. Goldstein,    23-Apr-1981  13:44
1 !                  Fix handling of /SAVE on magtapes
1 !
1 !          V02-004 MLJ0018         Martin L. Jack, 7-Apr-1981  21:09
1 !                  Correct signal parameters.
1 !
1 !          V02-003 MLJ0010         Martin L. Jack, 25-Mar-1981  15:38
1 !                  Reorganize global storage.  Remove limiting of buffer count as
1 !                  COMMAND now does this.  Change IQUA_FILE to IQUA_SAVE.  Ensure
1 !                  that QIO service failures are fatal.  Correct V02-001 to apply
1 !                  to continuation tapes.
1 !
1 !          V02-002 ACG0197         Andrew C. Goldstein,    5-Mar-1981  17:15
1 !                  Make not first reel error fatal if image restore
1 !
1 !          V02-001 ACG0193         Andrew C. Goldstein,    26-Feb-1981  17:04
1 !                  Allow null input save set name, add file type
1 !
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB';
1 REQUIRE 'SRC$:COMMON';
1 REQUIRE 'LIB$:BACKDEF';
1
1 EXTERNAL LITERAL
1          BACKUP$_OPENIN,
1          BACKUP$_READERR,
1          BACKUP$_CLOSEIN,
1          BACKUP$_SHORTBLOCK,
1          BACKUP$_BLOCKCRC,
1          BACKUP$_HDRCRC,
1          BACKUP$_BLOCKLOST,
1          BACKUP$_POSERROR,
1          BACKUP$_POSITERR,
1          BACKUP$_NOTBKBLOCK,
1          BACKUP$_INVBLKHDR,
1          BACKUP$_INVSTRUCT,
1          BACKUP$_INVBLKSIZE,
1          BACKUP$_INVFILEXT,
1          BACKUP$_LABELERR,
1          BACKUP$_NOTANSI,
1          BACKUP$_NOTSAVESET,
1          BACKUP$_NOT1STVOL,
1          BACKUP$_WRONGVOL,
1          BACKUP$_BADBLKSIZE,
1          BACKUP$_FATALERR,
1          BACKUP$_READERRS,
1          BACKUP$_SOFTRERRS,
1          BACKUP$_SSCHANGE,
```

```
1            BACKUP$_XORERRS,
1            BACKUP$_STARTVERIFY,
1            BACKUP$_RESUME,
1            BACKUP$_NEWSAVSET;
1
1 FORWARD ROUTINE
1            READY_NEXT_VOLUME,
1            NEXT_VOLUME;
1
1
1 G$DEFINE();              ! Define global common area
```

```
1 %SBTTL 'READ_BLOCK - read a save set block'
1 ROUTINE READ_BLOCK (READ_AHEAD, CRC_CHECK) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads a block from the input save set.
1 !
1 ! CALLING SEQUENCE:
1 !       READ_BLOCK (READ_AHEAD)
1 !
1 ! INPUT PARAMETERS:
1 !       READ_AHEAD: TRUE to enable asynchronous read ahead
1 !                   FALSE to read just one block
1 !       CRC_CHECK: TRUE to check block CRC's, etc.
1 !                  FALSE to accept any block
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       address of BCB of buffer read
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       INSQUE,
2       REMQUE,
2       TESTBITCS,
2       CRC;                                    ! compute CRC instruction
2
2 LOCAL
2       STATUS,                                 ! general status value
2       P               : REF BBLOCK,           ! pointer to chase BCB list
2       BCB             : REF BBLOCK,           ! buffer control block
2       BUFFER          : REF BBLOCK,           ! the I/O buffer itself
2       RAB             : REF BBLOCK,           ! input RAB
2       RECP            : REF BBLOCK,           ! pointer to end of records
2       BLOCK_CRC,                              ! CRC of data block
2       HDR_CRC,                                ! CRC of block header
2       CHECKSUM;                               ! output of CRC instruction
2
2 EXTERNAL ROUTINE
2       FILE_ERROR      : NOVALUE,              ! signal file-related error
2       GET_BUFFER,                             ! get an I/O buffer
```

```
2               FREE_BUFFER,                    ! free an I/O buffer
2               WAIT;                           ! wait for I/O completion
2
2   ! If reading from a file, allocate a buffer and read a block.
2   !
2
2   IF .QUAL[QUAL_SS_FILE]
2   THEN
3       BEGIN
3       IF NOT REMQUE (.INPUT_WAIT[0], BCB)
3       THEN
3           WAIT (.BCB)
3       ELSE
4           BEGIN
4           BCB = GET_BUFFER ();
4
4           RAB = RWSV_SAVE_FAB[FC_RAB];
4           RAB[RAB$W_USZ] = .BCB[BCB_SIZE];
4           RAB[RAB$L_UBF] = .BCB[BCB_BUFFER];
4           BCB[BCB_STATUS] = TRUE;
4           IF .RWSV_SAVE_FAB[FAB$V_BIO]
4           THEN
5               BEGIN
5               STATUS = $READ (RAB = .RAB);
5               RAB[RAB$L_BKT] = 0;
5               RWSV_SAVE_FAB[FAB$V_SQO] = TRUE;
5               END
4           ELSE
4               STATUS = $GET (RAB = .RAB);
4           BCB[BCB_IO_BCOUNT] = .RAB[RAB$W_RSZ];
4           IF NOT .STATUS
4           THEN
5               BEGIN
5               IF .STATUS EQL RMS$_EOF
5               THEN
5                   BCB[BCB_STATUS] = SS$_ENDOFFILE
5               ELSE
6                   BEGIN
6                   BCB[BCB_STATUS] = .RAB[RAB$L_STS];
6                   BCB[BCB_STATUS2] = .RAB[RAB$L_STV];
5                   END;
4               END;
3           END;
3       END
3
3   ! If we are reading from tape or sequential disk, grab I/O buffers
3   ! and issue read ahead QIO's until half the buffer pool is used. Then
3   ! wait for completion on the first buffer.
3   !
3
2   ELSE
3       BEGIN
3       P = .INPUT_WAIT[0];
4       DECR J FROM (
4           IF .READ_AHEAD
4           THEN (.COM_BUFF_COUNT+1) / 2
4           ELSE 1
```

```
                ) TO 1
            DO
                BEGIN
                IF .P NEQ INPUT_WAIT[0]
                THEN P = ..P
                ELSE
                    BEGIN
                    BCB = GET_BUFFER ();
                    CH$FILL (0, BBH$K_LENGTH, .BCB[BCB_BUFFER]);

                    IF .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
                    THEN
                        BEGIN
                        STATUS = $QIO  (CHAN = .RWSV_CHAN,
                                        FUNC = IO$_READLBLK,
                                        EFN  = BCB_S_READ,
                                        IOSB = BCB[BCB_IOSB],
                                        P1   = .BCB[BCB_BUFFER],
                                        P2   = .BCB[BCB_SIZE]
                                        );
                        END

                    ELSE
                        BEGIN
                        BCB[BCB_BLOCKNUM] = .RWSV_IN_VBN;
                        IF .RWSV_IN_VBN GEQU .RWSV_EOF
                        THEN
                            BEGIN
                            BCB[BCB_STATUS] = SS$_ENDOFFILE;
                            STATUS = 1;
                            END
                        ELSE
                            BEGIN
                            STATUS = $$QIO (CHAN = .RWSV_CHAN,
                                            FUNC = IO$_READVBLK,
                                            EFN  = BCB_S_READ,
                                            IOSB = BCB[BCB_IOSB],
                                            P1   = .BCB[BCB_BUFFER],
                                            P2   = .BCB[BCB_SIZE],
                                            P3   = .RWSV_IN_VBN
                                            );
                            RWSV_IN_VBN = .RWSV_IN_VBN + (.BCB[BCB_SIZE]+511) / 512;
                            END;
                        END;
                    IF NOT .STATUS
                    THEN
                        FILE_ERROR(
                            BACKUP$_READERR + STS$K_SEVERE,
                            .RWSV_SAVE_FAB,
                            .STATUS);

                    BCB[BCB_SUCC_ACT] = 0;
                    BCB[BCB_FAIL_ACT] = 0;
                    BCB[BCB_STATE] = BCB_S_READ;
                    INSQUE (.BCB, .INPUT_WAIT[1]);
                    END;
                END;
```

```
3
3           REMQUE (.INPUT_WAIT[0], BCB);
3           WAIT (.BCB);
3           BCB[BCB_STATUS2] = 0;
2           END;
2
2       ! Do basic validation checks on the buffer. It should be of the expected
2       ! length, and the CRC's, if present, should check.
2       !
2
2       IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFTAPE
2       THEN BCB[BCB_IO_STATUS] = TRUE;
2
2       BUFFER = .BCB[BCB_BUFFER];
2       IF NOT .BCB[BCB_IO_STATUS]
2       THEN
3           BEGIN
3           IF NOT .QUAL[QUAL_SS_FILE]
3           THEN BCB[BCB_IO_BCOUNT] = 0;
3           IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
3           THEN RETURN .BCB;
3           IF .RWSV_IN_ORGERR[0]
3           THEN
4               BEGIN
4               RWSV_IN_ORGERR[0] = .BCB[BCB_STATUS];
4               RWSV_IN_ORGERR[1] = 0;
3               END;
3           END
3
2       ELSE IF .CRC_CHECK
2       THEN
3           BEGIN
3           IF .BCB[BCB_IO_BCOUNT] LSSU .BCB[BCB_SIZE]
3           THEN
4               BEGIN
4               CH$FILL (0, .BCB[BCB_SIZE]-.BCB[BCB_IO_BCOUNT], .BUFFER+.BCB[BCB_IO_BCOUNT]);
4               BCB[BCB_STATUS] = BACKUP$_SHORTBLOCK;
3               END;
2           END;
2
2       IF .CRC_CHECK
2       THEN
3           BEGIN
3           BLOCK_CRC = .BUFFER[BBH$L_CRC];
3           HDR_CRC = .BUFFER[BBH$W_CHECKSUM];
3           BUFFER[BBH$L_CRC] = 0;
3           BUFFER[BBH$W_CHECKSUM] = 0;
3
3           CRC (RWSV_CRC16, %REF (0), %REF (BBH$K_LENGTH), .BUFFER, CHECKSUM);
3           IF .HDR_CRC NEQ .CHECKSUM
3           THEN BCB[BCB_STATUS] = BACKUP$_HDRCRC;
3
4           IF (
5               (.QUAL[QUAL_CRC] OR (.COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]))
4               AND NOT .BUFFER[BBH$V_NOCRC]
4               AND .BCB[BCB_STATUS]
4               AND
```

```
                    BEGIN
                    CRC (RWSV_AUTODIN, %REF (-1), BCB[BCB_SIZE], .BUFFER, CHECKSUM);
                    .BLOCK_CRC NEQ (NOT .CHECKSUM)
                    END
                )
        THEN BCB[BCB_STATUS] = BACKUP$_BLOCKCRC;

        BUFFER[BBH$L_CRC] = .BLOCK_CRC;
        BUFFER[BBH$W_CHECKSUM] = .ADR_CRC;

        IF NOT .BCB[BCB_STATUS]
        AND .RWSV_IN_ORGERR[0]
        THEN
            BEGIN
            RWSV_IN_ORGERR[0] = .BCB[BCB_STATUS];
            RWSV_IN_ORGERR[1] = 0;
            END;
        END;

  ! If this block read with an error, count it. If there are too many
  ! consecutive errors, complain. Medium offline is handled specially, since
  ! operator help is clearly necessary.
  !

    IF NOT .BCB[BCB_STATUS]
    THEN
        BEGIN
        RWSV_SEQ_ERRORS = .RWSV_SEQ_ERRORS + 1;
        IF .BCB[BCB_IO_STATUS] EQL SS$_MEDOFL
        OR .BCB[BCB_IO_STATUS] EQL SS$_VOLINV
        THEN
            BEGIN
            INSQUE (.BCB, RWSV_HOLD_LIST[0]);

  ! Other pending reads are also likely to fail with the same error. Clean
  ! them out.
  !

            UNTIL REMQUE (.INPUT_WAIT[0], BCB)
            DO
                BEGIN
                WAIT (.BCB);
                IF  .BCB[BCB_IO_STATUS] NEQ SS$_MEDOFL
                AND .BCB[BCB_IO_STATUS] NEQ SS$_VOLINV
                THEN
                    BEGIN
                    INSQUE (.BCB, INPUT_WAIT[0]);
                    BCB[BCB_STATE] = BCB_S_READ;
                    EXITLOOP;
                    END
                ELSE
                    FREE_BUFFER (.BCB);
                END;
            FILE_ERROR (BACKUP$_FATALERR, .RWSV_SAVE_FAB,
                        .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
            REMQUE (.RWSV_HOLD_LIST[0], BCB);
            END
```

```
4
4       ELSE IF (
4           IF .RWSV_SEQ_ERRORS GTRU 100
4           THEN TESTBITCS (COM_FLAGS[COM_CONTINUE])
4           ELSE FALSE)
3       THEN
4           BEGIN
4           INSQUE (.BCB, RWSV_HOLD_LIST[0]);
4           FILE_ERROR (BACKUP$_READERRS, .RWSV_SAVE_FAB,
4                   .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
4           REMQUE (.RWSV_HOLD_LIST[0], BCB);
3           END;
3       END
3
2   ELSE
2       RWSV_SEQ_ERRORS = 0;
2
2   .BCB
1   END;                                    ! End of routine READ_BLOCK

                                .TITLE  READSAVE Read Save Set
                                .IDENT  \V04-001\

                                .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
003B0 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

                                                            .EXTRN   BACKUP$_OPENIN, BACKUP$_READERR
                                                            .EXTRN   BACKUP$_CLOSEIN
                                                            .EXTRN   BACKUP$_SHORTBLOCK
                                                            .EXTRN   BACKUP$_BLOCKCRC
                                                            .EXTRN   BACKUP$_HDRCRC, BACKUP$_BLOCKLOST
                                                            .EXTRN   BACKUP$_POSERROR
                                                            .EXTRN   BACKUP$_POSITERR
                                                            .EXTRN   BACKUP$_NOTBKBLOCK
                                                            .EXTRN   BACKUP$_INVBLKHDR
                                                            .EXTRN   BACKUP$_INVSTRUCT
                                                            .EXTRN   BACKUP$_INVBLKSIZE
                                                            .EXTRN   BACKUP$_INVFILEXT
                                                            .EXTRN   BACKUP$_LABELERR
                                                            .EXTRN   BACKUP$_NOTANSI
                                                            .EXTRN   BACKUP$_NOTSAVESET
                                                            .EXTRN   BACKUP$_NOT1STVOL
                                                            .EXTRN   BACKUP$_WRONGVOL
                                                            .EXTRN   BACKUP$_BADBLKSIZE
                                                            .EXTRN   BACKUP$_FATALERR
                                                            .EXTRN   BACKUP$_READERRS
                                                            .EXTRN   BACKUP$_SOFTRERRS
                                                            .EXTRN   BACKUP$_SSCHANGE
                                                            .EXTRN   BACKUP$_XORERRS
                                                            .EXTRN   BACKUP$_STARTVERIFY
                                                            .EXTRN   BACKUP$_RESUME, BACKUP$_NEWSAVSET
                                                            .EXTRN   FILE_ERROR, GET_BUFFER
                                                            .EXTRN   FREE_BUFFER, WAIT
                                                            .EXTRN   SYS$READ, SYS$GET
                                                            .EXTRN   SYS$QIO, STA_QIO

                                                            .PSECT   CODE,NOWRT,2

                                        0FFC  00000 READ_BLOCK:
                                                            .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                 ; 1738
                     5B 00000000G   00  9E  00002           MOVAB    WAIT, R11                                            ;
                     5A 00000000'   EF  9E  00009           MOVAB    INPUT_WAIT, R10                                      ;
      7B      0153   CA             03  E1  00010           BBC      #3, QUAL+15, 6$                                      ; 1800
                     57         00  BA  0F  00016           REMQUE   @INPUT_WAIT, BCB                                     ; 1803
```

```
                                         07  1D 0001A          BVS     1$                                    ;
                                         57  DD 0001C          PUSHL   BCB                                   ; 1805
                          6B             01  FB 0001E          CALLS   #1, WAIT                              ;
                                         6B  11 00021          BRB     5$                                    ;
            00000000G     00             00  FB 00023 1$:      CALLS   #0, GET_BUFFER                        ; 1808
                          57             50  D0 0002A          MOVL    R0, BCB                               ;
                          50      00F0   CA  D0 0002D          MOVL    RWSV_SAVE_FAB, R0                     ; 1810
                          52        50   A0  9E 00032          MOVAB   80(R0), RAB                           ;
                   20     A2        08   A7  B0 00036          MOVW    8(BCB), 32(RAB)                       ; 1811
                   24     A2        0C   A7  D0 0003B          MOVL    12(BCB), 36(RAB)                      ; 1812
                   18     A7             01  D0 00040          MOVL    #1, 24(BCB)                           ; 1813
            1B     16     A0             05  E1 00044          BBC     #5, 22(R0), 2$                        ; 1814
                                         52  DD 00049          PUSHL   RAB                                   ; 1817
            00000000G     00             01  FB 0004B          CALLS   #1, SYS$READ                          ;
                          58             50  D0 00052          MOVL    R0, STATUS                            ;
                                    38   A2  D4 00055          CLRL    56(RAB)                               ; 1818
                          50      00F0   CA  D0 00058          MOVL    RWSV_SAVE_FAB, R0                     ; 1819
                   04     A0        40   8F  88 0005D          BISB2   #64, 4(R0)                            ;
                                         0C  11 00062          BRB     3$                                    ; 1814
                                         52  DD 00064 2$:      PUSHL   RAB                                   ; 1822
            00000000G     00             01  FB 00066          CALLS   #1, SYS$GET                           ;
                          58             50  D0 0006D          MOVL    R0, STATUS                            ;
                   1A     A7        22   A2  B0 00070 3$:      MOVW    34(RAB), 26(BCB)                      ; 1823
                          16             58  E8 00075          BLBS    STATUS, 5$                            ; 1824
            0001827A      8F             58  D1 00078          CMPL    STATUS, #98938                        ; 1827
                                         08  12 0007F          BNEQ    4$                                    ;
                   18     A7      0870   8F  3C 00081          MOVZWL  #2160, 24(BCB)                        ; 1829
                                         05  11 00087          BRB     5$                                    ;
                   18     A7        08   A2  7D 00089 4$:      MOVQ    8(RAB), 24(BCB)                       ; 1832
                                       00F2  31 0008E 5$:      BRW     19$                                   ; 1824
                          59             6A  D0 00091 6$:      MOVL    INPUT_WAIT, P                         ; 1846
                          0C        04   AC  E9 00094          BLBC    READ_AHEAD, 7$                        ; 1848
                          56      01C1   CA  9A 00098          MOVZBL  COM_BUFF_COUNT, R6                    ; 1849
                                         56  D6 0009D          INCL    R6                                    ;
                          56             02  C6 0009F          DIVL2   #2, R6                                ;
                                         03  11 000A2          BRB     8$                                    ;
                          56             01  D0 000A4 7$:      MOVL    #1, R6                                ; 1848
                                         56  D6 000A7 8$:      INCL    J                                     ; 1847
                                         0B  11 000A9          BRB     10$                                   ;
                          50             6A  9E 000AB 9$:      MOVAB   INPUT_WAIT, R0                        ; 1854
                          50             59  D1 000AE          CMPL    P, R0                                 ;
                                         06  13 000B1          BEQL    11$                                   ;
                          59             69  D0 000B3          MOVL    (P), P                                ; 1855
                                       00B6  31 000B6 10$:     BRW     16$                                   ;
            00000000G     00             00  FB 000B9 11$:     CALLS   #0, GET_BUFFER                        ; 1858
                          57             50  D0 000C0          MOVL    R0, BCB                               ;
 0100   8F          00    6E             00  2C 000C3          MOVC5   #0, (SP), #0, #256, @12(BCB)          ; 1859
                                    0C   B7     000CA                                                        ;
                          50      00F0   CA  D0 000CC          MOVL    RWSV_SAVE_FAB, R0                     ; 1861
            24     40     A0             05  E1 000D1          BBC     #5, 64(R0), 12$                       ;
                                         7E  7C 000D6          CLRQ    -(SP)                                 ; 1870
                                         7E  7C 000D8          CLRQ    -(SP)                                 ;
                          7E        08   A7  3C 000DA          MOVZWL  8(BCB), -(SP)                         ;
                                    0C   A7  DD 000DE          PUSHL   12(BCB)                               ;
                                         7E  7C 000E1          CLRQ    -(SP)                                 ;
                                    18   A7  9F 000E3          PUSHAB  24(BCB)                               ;
                                         21  DD 000E6          PUSHL   #33                                   ;
```

```
                         00F4  CA  DD 000E8          PUSHL   RWSV_CHAN                      ;
                               01  DD 000EC          PUSHL   #1                             ;
   00000000G   00              0C  FB 000EE          CALLS   #12, SYS$QIO                   ;
               58              50  D0 000F5          MOVL    R0, STATUS                     ;
                               54  11 000F8          BRB     14$                            ; 1861
               50        0120  CA  D0 000FA  12$:    MOVL    RWSV_IN_VBN, R0                ; 1875
          14   A7              50  D0 000FF          MOVL    R0, 20(BCB)                    ;
        012C   CA              50  D1 00103          CMPL    R0, RWSV_EOF                   ; 1876
                               0B  1F 00108          BLSSU   13$                            ;
          18   A7        0870  8F  3C 0010A          MOVZWL  #2160, 24(BCB)                 ; 1879
               58              01  D0 00110          MOVL    #1, STATUS                     ; 1880
                               39  11 00113          BRB     14$                            ; 1876
                               7E  7C 00115  13$:    CLRQ    -(SP)                          ; 1891
                               7E  D4 00117          CLRL    -(SP)                          ;
                               50  DD 00119          PUSHL   R0                             ;
               7E          08  A7  3C 0011B          MOVZWL  8(BCB), -(SP)                  ;
                           0C  A7  DD 0011F          PUSHL   12(BCB)                        ;
                               7E  7C 00122          CLRQ    -(SP)                          ;
                           18  A7  9F 00124          PUSHAB  24(BCB)                        ;
                               31  DD 00127          PUSHL   #49                            ;
                         00F4  CA  DD 00129          PUSHL   RWSV_CHAN                      ;
                               01  DD 0012D          PUSHL   #1                             ;
   00000000G   00              0C  FB 0012F          CALLS   #12, STA_QIO                   ;
               58              50  D0 00136          MOVL    R0, STATUS                     ;
               50          08  A7  3C 00139          MOVZWL  8(BCB), R0                     ; 1892
               50        01FF  C0  9E 0013D          MOVAB   511(R0), R0                    ;
               50    00000200  8F  C6 00142          DIVL2   #512, R0                       ;
        0120   CA              50  C0 00149          ADDL2   R0, RWSV_IN_VBN                ;
               13              58  E8 0014E  14$:    BLBS    STATUS, 15$                    ; 1895
                               58  DD 00151          PUSHL   STATUS                         ; 1900
                         00F0  CA  DD 00153          PUSHL   RWSV_SAVE_FAB                  ; 1899
                    00000000G  8F  DD 00157          PUSHL   #BACKUP$_READERR+4             ; 1898
   00000000G   00              03  FB 0015D          CALLS   #3, FILE_ERROR                 ;
                           20  A7  7C 00164  15$:    CLRQ    32(BCB)                        ; 1902
          0A   A7              01  90 00167          MOVB    #1, 10(BCB)                    ; 1904
          04   BA              67  0E 0016B          INSQUE  (BCB), @INPUT_WAIT+4           ; 1905
               02              56  F5 0016F  16$:    SOBGTR  J, 17$                         ; 1847
                               03  11 00172          BRB     18$                            ;
                             FF34  31 00174  17$:    BRW     9$                             ;
               57          00  BA  0F 00177  18$:    REMQUE  @INPUT_WAIT, BCB               ; 1909
                               57  DD 0017B          PUSHL   BCB                            ; 1910
               6B              01  FB 0017D          CALLS   #1, WAIT                       ;
                           1C  A7  D4 00180          CLRL    28(BCB)                        ; 1911
               58          18  A7  9E 00183  19$:    MOVAB   24(BCB), R8                    ; 1918
        0878   8F              68  B1 00187          CMPW    (R8), #2168                    ;
                               03  12 0018C          BNEQ    20$                            ;
               68              01  B0 0018E          MOVW    #1, (R8)                       ; 1919
               56          0C  A7  D0 00191  20$:    MOVL    12(BCB), BUFFER                ; 1921
               23              68  E8 00195          BLBS    (R8), 23$                      ; 1922
   03   0153   CA              03  E0 00198          BBS     #3, QUAL+15, 21$               ; 1925
                           1A  A7  B4 0019E          CLRW    26(BCB)                        ; 1926
        0870   8F              68  B1 001A1  21$:    CMPW    (R8), #2160                    ; 1927
                               03  12 001A6          BNEQ    22$                            ;
                             0137  31 001A8          BRW     37$                            ;
               2E        0118  CA  E9 001AB  22$:    BLBC    RWSV_IN_ORGERR, 24$            ; 1929
        0118   CA              68  D0 001B0          MOVL    (R8), RWSV_IN_ORGERR           ; 1932
                         011C  CA  D4 001B5          CLRL    RWSV_IN_ORGERR+4               ; 1933
```

```
                                  23  11 001B9        BRB     24$                                      ; 1922
                   1F        08   AC  E9 001BB 23$:   BLBC    CRC_CHECK, 24$                           ; 1937
                   51        1A   A7  3C 001BF        MOVZWL  26(BCB), R1                              ; 1940
                   51        08   A7  B1 001C3        CMPW    8(BCB), R1                               ;
                             15   1B     001C7        BLEQU   24$                                      ;
                   50        08   A7  3C 001C9        MOVZWL  8(BCB), R0                               ; 1943
                             50   51  C2 001CD        SUBL2   R1, R0                                   ;
    50        00   6E             00  2C 001D0        MOVC5   #0, (SP), #0, R0, (R1)[BUFFER]           ;
                               6146      001D5                                                         ;
                   68 00000000G   8F  D0 001D7        MOVL    #BACKUP$_SHORTBLOCK, (R8)                ; 1944
                   71        08   AC  E9 001DE 24$:   BLBC    CRC_CHECK, 28$                           ; 1948
                   59        24   A6  D0 001E2        MOVL    36(BUFFER), BLOCK_CRC                    ; 1951
                   54      00FE   C6  9E 001E6        MOVAB   254(BUFFER), R4                          ; 1952
                   55        64   3C     001EB        MOVZWL  (R4), HDR_CRC                            ;
                             24   A6  D4 001EE        CLRL    36(BUFFER)                               ; 1953
                             64   B4     001F1        CLRW    (R4)                                     ; 1954
    66   0100   8F 00        50   AA  0B 001F3        CRC     RWSV_CRC16, #0, #256, (BUFFER)           ; 1956
                   50        55   D1     001FB        CMPL    HDR_CRC, CHECKSUM                        ; 1957
                   07        13          001FE        BEQL    25$                                      ;
                   68 00000000G   8F  D0 00200        MOVL    #BACKUP$_HDRCRC, (R8)                    ; 1958
            0C   014D   CA        01  E0 00207 25$:   BBS     #1, QUAL+9, 26$                          ; 1961
            28   01BE   CA        03  E1 0020D        BBC     #3, COM_FLAGS, 27$                       ;
                           0153   CA  95 00213        TSTB    QUAL+15                                  ;
                             22   18     00217        BGEQ    27$                                      ;
                   1E        2C   A6  E8 00219 26$:   BLBS    44(BUFFER), 27$                          ; 1962
                   1B        68   E9     0021D        BLBC    (R8), 27$                                ; 1963
    66        08   A7 FFFFFFFF 8F  0090  CA  0B 00220 CRC     RWSV_AUTODIN, #-1, 8(BCB), (BUFFER)      ; 1966
                   50        50   D2     0022C        MCOML   CHECKSUM, R0                             ; 1967
                   50        59   D1     0022F        CMPL    BLOCK_CRC, R0                            ;
                   07        13          00232        BEQL    27$                                      ;
                   68 00000000G   8F  D0 00234        MOVL    #BACKUP$_BLOCKCRC, (R8)                  ; 1970
            24     A6        59   D0     0023B 27$:   MOVL    BLOCK_CRC, 36(BUFFER)                    ; 1972
                   64        55   B0     0023F        MOVW    HDR_CRC, (R4)                            ; 1973
                   0E        68   E8     00242        BLBS    (R8), 28$                                ; 1975
                   09      0118   CA  E9 00245        BLBC    RWSV_IN_ORGERR, 28$                      ; 1976
          0118     CA        68   D0     0024A        MOVL    (R8), RWSV_IN_ORGERR                     ; 1979
                           011C   CA  D4 0024F        CLRL    RWSV_IN_ORGERR+4                         ; 1980
                   03        68   E9     00253 28$:   BLBC    (R8), 29$                                ; 1989
                               0085   31 00256        BRW     36$                                      ;
                           013E   CA  B6 00259 29$:   INCW    RWSV_SEQ_ERRORS                          ; 1992
          01A4     8F        68   B1     0025D        CMPW    (R8), #420                               ; 1993
                   07        13          00262        BEQL    30$                                      ;
          0254     8F        68   B1     00264        CMPW    (R8), #596                                 1994
                   44        12          00269        BNEQ    34$                                      ;
            48     AA        67   0E     0026B 30$:   INSQUE  (BCB), RWSV_HOLD_LIST                    ; 1997
                   57        00   BA  0F 0026F 31$:   REMQUE  @INPUT_WAIT, BCB                         ; 2003
                   29        1D          00273        BVS     33$                                      ;
                   57        DD          00275        PUSHL   BCB                                        2006
                   6B        01   FB     00277        CALLS   #1, WAIT
          01A4     8F        18   A7  B1 0027A        CMPW    24(BCB), #420                              2007
                   11        13          00280        BEQL    32$
          0254     8F        18   A7  B1 00282        CMPW    24(BCB), #596                              2008
                   09        13          00288        BEQL    32$
                   6A        67   0E     0028A        INSQUE  (BCB), INPUT_WAIT                        ; 2011
            0A     A7        01   90     0028D        MOVB    #1, 10(BCB)                              ; 2012
                   0B        11          00291        BRB     33$                                      ; 2010
                   57        DD          00293 32$:   PUSHL   BCB                                      ; 2016
```

```
          00000000G  00           01  FB 00295        CALLS   #1, FREE_BUFFER                 ;
                                  D1  11 0029C        BRB     31$                             ; 2003
                     7E     0118  CA  7D 0029E 33$:   MOVQ    RWSV_IN_ORGERR, -(SP)           ; 2019
                            00F0  CA  DD 002A3        PUSHL   RWSV_SAVE_FAB                   ; 2018
                       00000000G  8F  DD 002A7        PUSHL   #BACKUP$_FATALERR               ;
                                  22  11 002AD        BRB     35$                             ;
               0064  8F     013E  CA  B1 002AF 34$:   CMPW    RWSV_SEQ_ERRORS, #100           ; 2024
                                  2A  1B 002B6        BLEQU   37$                             ;
         24    01BE  CA           05  E2 002B8        BBSS    #5, COM_FLAGS, 37$              ; 2025
                 48  AA           67  0E 002BE        INSQUE  (BCB), RWSV_HOLD_LIST           ; 2029
                     7E     0118  CA  7D 002C2        MOVQ    RWSV_IN_ORGERR, -(SP)           ; 2031
                            00F0  CA  DD 002C7        PUSHL   RWSV_SAVE_FAB                   ; 2030
                       00000000G  8F  DD 002CB        PUSHL   #BACKUP$_READERRS               ;
          00000000G  00           04  FB 002D1 35$:   CALLS   #4, FILE_ERROR                  ;
                     57       48  BA  0F 002D8        REMQUE  @RWSV_HOLD_LIST, BCB            ; 2032
                                  04  11 002DC        BRB     37$                             ; 1989
                            013E  CA  B4 002DE 36$:   CLRW    RWSV_SEQ_ERRORS                 ; 2037
                     50           57  D0 002E2 37$:   MOVL    BCB, R0                         ; 2040
                                      04 002E5        RET                                     ;

; Routine Size:  742 bytes,    Routine Base:  CODE + 0000
```

```
1 %SBTTL 'READ_SEQ_BLOCK - read a sequenced save set block'
1 ROUTINE READ_SEQ_BLOCK (READ_AHEAD) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads a save set block and applies sequence number
1 !       checking and rewrite error recovery.
1 !
1 ! CALLING SEQUENCE:
1 !       READ_SEQ_BLOCK (READ_AHEAD)
1 !
1 ! INPUT PARAMETERS:
1 !       READ_AHEAD: TRUE to enable asynchronous read ahead
1 !                   FALSE to read only one block
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       BCB address of block read
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       REMQUE;
2
2 LOCAL
2       BCB             : REF BBLOCK,   ! BCB of block read
2       BUFFER          : REF BBLOCK,   ! address of data block read
2       BLOCK_COUNT,                    ! count of blocks read forward
2       BK_BLOCK_COUNT,                 ! count of backup blocks seen
2       TM_COUNT;                       ! count of tape marks crossed
2
2 EXTERNAL ROUTINE
2       SKIP_RECORD,                    ! skip tape records
2       SKIP_TM,                        ! skip tape marks
2       WAIT,                           ! wait for I/O completion
2       FREE_BUFFER;                    ! free an I/O buffer
2
2 ! The outer loop handles re-reading a data block after a forward
2 ! retry has been tried and found unsuccessful.
2 !
2
2 DECR TRY FROM 2 TO 1
```

```
2 DO
3     BEGIN
3
3 ! Loop, reading blocks from the input medium. If an error occurrs,
3 ! leave the loop for forward retry. Keep reading blocks if a sequence
3 ! number inversion occurrs. Note that a block header CRC error is totally
3 ! fatal. Since the header cannot be trusted, the block must be flushed.
3 ! Blocks not written by BACKUP are also flushed at this level.
3 !
3
3     WHILE TRUE
3     DO
4         BEGIN
4         BCB = READ_BLOCK (.READ_AHEAD, TRUE);
4         BUFFER = .BCB[BCB_BUFFER];
4         IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
4         THEN RETURN .BCB;
4         IF .BCB[BCB_STATUS] NEQ BACKUP$_HDRCRC
4         AND .BUFFER[BBH$W_SUBSYS] EQL BACKUP$K_BACKUP
4         AND .BUFFER[BBH$L_NUMBER] GEQU .RWSV_IN_SEQ
4         THEN EXITLOOP;
4         FREE_BUFFER (.BCB);
3         END;
3
4     IF (.BCB[BCB_IO_STATUS] AND
4         (.BUFFER[BBH$L_NUMBER] EQL .RWSV_IN_SEQ OR .RWSV_IN_SEQ EQL 0))
3     OR .TRY
3     THEN
3         EXITLOOP
3     ELSE
4         BEGIN
4         RWSV_IN_ERRORS = .RWSV_IN_ERRORS + 1;
4         IF .QUAL[QUAL_SS_FILE]
4         OR NOT .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
4         THEN EXITLOOP;
3         END;
3
3 ! If an error has occurred, search forward, limited by the buffer depth
3 ! of the writer, for a rewrite of the block.
3 !
3
3     FREE_BUFFER (.BCB);
3     BLOCK_COUNT = 0;
3     BK_BLOCK_COUNT = 0;
3     TM_COUNT = 0;
3     IF .RWSV_LOOKAHEAD EQL 0 THEN RWSV_LOOKAHEAD = 10;
3     DO
4         BEGIN
4         BCB = READ_BLOCK (FALSE, TRUE);
4         BUFFER = .BCB[BCB_BUFFER];
4         IF .BCB[BCB_IO_STATUS]
4         AND .BUFFER[BBH$W_SUBSYS] EQL BACKUP$K_BACKUP
4         AND .BUFFER[BBH$L_NUMBER] EQL .RWSV_IN_SEQ
4         THEN RETURN .BCB;
4         FREE_BUFFER (.BCB);
4         IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
4         THEN
```

```
5                BEGIN
5                TM_COUNT = .TM_COUNT + 1;
5                EXITLOOP;
4                END;
4            IF .TM_COUNT EQL 0
4            THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
4            IF .BUFFER[BBH$W_SUBSYS] EQL BACKUP$K_BACKUP
4            THEN BK_BLOCK_COUNT = .BK_BLOCK_COUNT + 1;
4            END
3        UNTIL .BK_BLOCK_COUNT GEQU .RWSV_LOOKAHEAD
3        AND .BCB[BCB_STATUS] NEQ BACKUP$_HDRCRC
3        AND .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ GEQU .RWSV_LOOKAHEAD;
3
3    ! Failed to find a rewrite of the block. Flush out the input wait list,
3    ! then backspace the tape to the original block, less one for good measure,
3    ! and retry the read.
3    !
3
3        UNTIL REMQUE (.INPUT_WAIT[0], BCB)
3        DO
4            BEGIN
4            WAIT (.BCB);
4            IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
4            THEN TM_COUNT = .TM_COUNT + 1;
4            IF .TM_COUNT EQL 0
4            THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
4            FREE_BUFFER (.BCB);
3            END;
3
3        IF .TM_COUNT NEQ 0
3        THEN SKIP_TM (-.TM_COUNT);
3        IF SKIP_RECORD (-.BLOCK_COUNT - 2) EQL SS$_ENDOFFILE
3        THEN SKIP_TM (1);
2        END;                                        ! end of outer loop
2
2  .BCB
1  END;                                              ! End of routine READ_SEQ_BLOCK


                                                      .EXTRN  SKIP_RECORD, SKIP_TM

                                OFFC 00000 READ_SEQ_BLOCK:
                                                      .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 2042
             5B 00000000G  00  9E 00002               MOVAB   SKIP_TM, R11                              ;
             5A 00000000G  8F  D0 00009               MOVL    #BACKUP$_HDRCRC, R10                      ;
             59 00000000G  00  9E 00010               MOVAB   FREE_BUFFER, R9                           ;
             58 00000000'  EF  9E 00017               MOVAB   RWSV_IN_SEQ, R8                           ;
             56                02  D0 0001E           MOVL    #2, TRY                                   ; 2097
                               01  DD 00021  1$:      PUSHL   #1                                        ; 2111
                           04  AC  DD 00023           PUSHL   READ_AHEAD                                ;
        FCEF  CF               02  FB 00026           CALLS   #2, READ_BLOCK                            ;
              54               50  D0 0002B           MOVL    R0, BCB                                   ;
              52           0C  A4  D0 0002E           MOVL    12(BCB), BUFFER                           ; 2112
        0870  8F           18  A4  B1 00032           CMPW    24(BCB), #2160                            ; 2113
                               26  13 00038           BEQL    5$                                        ;
              5A           18  A4  D1 0003A           CMPL    24(BCB), R10                              ; 2115
```

```
                                  0C  13 0003E        BEQL    2$                                     ;
                 01        04     A2  B1 00040        CMPW    4(BUFFER), #1                          ; 2116
                                  06  12 00044        BNEQ    2$                                     ;
                 68        08     A2  D1 00046        CMPL    8(BUFFER), RWSV_IN_SEQ                 ; 2117
                                  07  1E 0004A        BGEQU   3$                                     ;
                                  54  DD 0004C 2$:    PUSHL   BCB                                    ; 2119
                 69               01  FB 0004E        CALLS   #1, FREE_BUFFER                        ;
                                  CE  11 00051        BRB     1$                                     ; 2108
                 06        18     A4  E9 00053 3$:    BLBC    24(BCB), 4$                            ; 2122
                                  07  13 00057        BEQL    5$                                     ; 2123
                                  68  D5 00059        TSTL    RWSV_IN_SEQ                            ;
                                  03  13 0005B        BEQL    5$                                     ;
                 03               56  E9 0005D 4$:    BLBC    TRY, 6$                                ; 2124
                                00D3  31 00060 5$:    BRW     19$                                    ;
                           18     A8  B6 00063 6$:    INCW    RWSV_IN_ERRORS                         ; 2129
     F5     57   A8               03  E0 00066        BBS     #3, QUAL+15, 5$                        ; 2130
                 50        F4     A8  D0 0006B        MOVL    RWSV_SAVE_FAB, R0                      ; 2131
     EC     40   A0               05  E1 0006F        BBC     #5, 64(R0), 5$                         ;
                                  54  DD 00074        PUSHL   BCB                                    ; 2139
                 69               01  FB 00076        CALLS   #1, FREE_BUFFER                        ;
                                  55  D4 00079        CLRL    BLOCK_COUNT                            ; 2140
                                  57  D4 0007B        CLRL    BK_BLOCK_COUNT                         ; 2141
                                  53  D4 0007D        CLRL    TM_COUNT                               ; 2142
                           12     A8  95 0007F        TSTB    RWSV_LOOKAHEAD                         ; 2143
                                  04  12 00082        BNEQ    7$                                     ;
            12   A8               0A  90 00084        MOVB    #10, RWSV_LOOKAHEAD                    ;
                                  01  DD 00088 7$:    PUSHL   #1                                     ; 2146
                                  7E  D4 0008A        CLRL    -(SP)                                  ;
          FC89   CF               02  FB 0008C        CALLS   #2, READ_BLOCK                         ;
                 54               50  D0 00091        MOVL    R0, BCB                                ;
                 52        0C     A4  D0 00094        MOVL    12(BCB), BUFFER                        ; 2147
                 0C        18     A4  E9 00098        BLBC    24(BCB), 8$                            ; 2148
                 01        04     A2  B1 0009C        CMPW    4(BUFFER), #1                          ; 2149
                                  06  12 000A0        BNEQ    8$                                     ;
                 68        08     A2  D1 000A2        CMPL    8(BUFFER), RWSV_IN_SEQ                 ; 2150
                                  B8  13 000A6        BEQL    5$                                     ;
                                  54  DD 000A8 8$:    PUSHL   BCB                                    ; 2152
                 69               01  FB 000AA        CALLS   #1, FREE_BUFFER                        ;
          0870   8F        18     A4  B1 000AD        CMPW    24(BCB), #2160                         ; 2153
                                  04  12 000B3        BNEQ    9$                                     ;
                                  53  D6 000B5        INCL    TM_COUNT                               ; 2156
                                  29  11 000B7        BRB     12$                                    ; 2155
                                  53  D5 000B9 9$:    TSTL    TM_COUNT                               ; 2159
                                  02  12 000BB        BNEQ    10$                                    ;
                                  55  D6 000BD        INCL    BLOCK_COUNT                            ; 2160
                 01        04     A2  B1 000BF 10$:   CMPW    4(BUFFER), #1                          ; 2161
                                  02  12 000C3        BNEQ    11$                                    ;
                                  57  D6 000C5        INCL    BK_BLOCK_COUNT                         ; 2162
57     12   A8   08               00  ED 000C7 11$:   CMPZV   #0, #8, RWSV_LOOKAHEAD, BK_BLOCK_COUNT ; 2164
                                  B9  1A 000CD        BGTRU   7$                                     ;
                 5A        18     A4  D1 000CF        CMPL    24(BCB), R10                           ; 2165
                                  B3  13 000D3        BEQL    7$                                     ;
            50   08   A2          68  C3 000D5        SUBL3   RWSV_IN_SEQ, 8(BUFFER), R0             ; 2166
50     12   A8   08               00  ED 000DA        CMPZV   #0, #8, RWSV_LOOKAHEAD, R0             ;
                                  A6  1A 000E0        BGTRU   7$                                     ;
                 54      FF04     D8  0F 000E2 12$:   REMQUE  @INPUT_WAIT, BCB                       ; 2173
                                  20  1D 000E7        BVS     15$                                    ;
```

```
                              54 DD 000E9          PUSHL   BCB                        ; 2176
          00000000G 00        01 FB 000EB          CALLS   #1, WAIT                   ;
               0870 8F    18  A4 B1 000F2          CMPW    24(BCB), #2160             ; 2177
                              02 12 000F8          BNEQ    13$                        ;
                              53 D6 000FA          INCL    TM_COUNT                   ; 2178
                              53 D5 000FC  13$:    TSTL    TM_COUNT                   ; 2179
                              02 12 000FE          BNEQ    14$                        ;
                              55 D6 00100          INCL    BLOCK_COUNT                ; 2180
                              54 DD 00102  14$:    PUSHL   BCB                        ; 2181
                    69        01 FB 00104          CALLS   #1, FREE_BUFFER            ;
                              D9 11 00107          BRB     12$                        ; 2173
                              53 D5 00109  15$:    TSTL    TM_COUNT                   ; 2184
                              06 13 0010B          BEQL    16$                        ;
                    7E        53 CE 0010D          MNEGL   TM_COUNT, -(SP)            ; 2185
                    6B        01 FB 00110          CALLS   #1, SKIP_TM                ;
                          02  A5 9F 00113  16$:    PUSHAB  2(BLOCK_COUNT)             ; 2186
                    6E        6E CE 00116          MNEGL   (SP), (SP)                 ;
          00000000G 00        01 FB 00119          CALLS   #1, SKIP_RECORD            ;
           00000870 8F        50 D1 00120          CMPL    R0, #2160                  ;
                              05 12 00127          BNEQ    17$                        ;
                              01 DD 00129          PUSHL   #1                         ; 2187
                    6B        01 FB 0012B          CALLS   #1, SKIP_TM                ;
                    02        56 F5 0012E  17$:    SOBGTR  TRY, 18$                   ; 2097
                              03 11 00131          BRB     19$                        ;
                            FEEB 31 00133  18$:    BRW     1$                         ;
                    50        54 D0 00136  19$:    MOVL    BCB, R0                    ; 2191
                                 04 00139          RET                                ;

; Routine Size:  314 bytes,    Routine Base:  CODE + 02E6
```

```
1 %SBTTL 'REPOSITION - reposition save set'
1 ROUTINE REPOSITION (BLOCK_NEEDED, ADDRESS, CUR_BCB) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !        This routine repositions the input save set to the specified
1 !        sequence number and reads that block.
1 !
1 ! CALLING SEQUENCE:
1 !        REPOSITION (BLOCK_NEEDED, ADDRESS, CUR_BCB)
1 !
1 ! INPUT PARAMETERS:
1 !        BLOCK_NEEDED: sequence number of desired block
1 !        ADDRESS: address of RFA or VBN, if file or seq disk, respectively
1 !        CUR_BCB: BCB of last block read
1 !
1 ! IMPLICIT INPUTS:
1 !        NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !        NONE
1 !
1 ! ROUTINE VALUE:
1 !        BCB of desired block
1 !
1 ! SIDE EFFECTS:
1 !        NONE
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2        REMQUE;
2
2 MAP
2        CUR_BCB          : REF BBLOCK;    ! input BCB arg
2
2 LOCAL
2        STATUS,                           ! general status value
2        BOT,                              ! flag indicating tape backspaced into BOT
2        RAB              : REF BBLOCK,    ! RAB for reading input file
2        PREV_SEQ,                         ! sequence number of last block read
2        BLOCK_COUNT,                      ! count of records to backspace
2        TM_COUNT,                         ! count of EOF marks to backspace
2        BCB              : REF BBLOCK,    ! current BCB
2        BUFFER           : REF BBLOCK;    ! current I/O buffer
2
2 EXTERNAL ROUTINE
2        WAIT,                             ! wait for I/O completion
2        FILE_ERROR,                       ! signal file related error
2        FREE_BUFFER,                      ! free an I/O buffer
```

```
2           SKIP_RECORD,                    ! skip tape records
2           SKIP_TM;                        ! skip tape marks
2
2   ! First wait out pending reads, tracking the tape position.
2   !
2
2   BUFFER = .CUR_BCB[BCB_BUFFER];
2   PREV_SEQ = -1;
2   BLOCK_COUNT = .BUFFER[BBH$L_NUMBER] - .BLOCK_NEEDED + 1;
2   TM_COUNT = 0;
2   IF .CUR_BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
2   THEN TM_COUNT = .TM_COUNT + 1;
2   IF .CUR_BCB[BCB_STATE] NEQ BCB_S_IDLE
2   THEN FREE_BUFFER (.CUR_BCB);
2
2   UNTIL REMQUE (.INPUT_WAIT[0], BCB)
2   DO
3       BEGIN
3       WAIT (.BCB);
3       IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
3       THEN TM_COUNT = .TM_COUNT + 1;
3       IF .TM_COUNT EQL 0
3       THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
3       FREE_BUFFER (.BCB);
3       END;
2
2   ! Now find the first block of the group by backspacing, reading, and
2   ! checking the sequence number. This may take several tries, since
2   ! we don't know whether there are block rewrites to space over.
2   !
2
2   RWSV_IN_SEQ = 0;
2   WHILE TRUE
2   DO
3       BEGIN
3       IF NOT .QUAL[QUAL_SS_FILE]
3       THEN
4           BEGIN
4           IF .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
4           THEN
5               BEGIN
5               BOT = FALSE;
5               IF .TM_COUNT NEQ 0
5               THEN SKIP_TM (-.TM_COUNT);
5               IF .BLOCK_COUNT GTR 0
5               THEN IF SKIP_RECORD (-.BLOCK_COUNT) EQL SS$_ENDOFFILE
5               THEN
6                   BEGIN
6                   SKIP_TM (1);
6                   BOT = TRUE;
5                   END;
5               END
4           ELSE
4               RWSV_IN_VBN = ..ADDRESS;
4           END
3       ELSE
4           BEGIN
```

```
4             RAB = RWSV_SAVE_FAB[FC_RAB];
4             IF .BBLOCK[RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_NET]
4             THEN
5                 BEGIN
5                 STATUS = $DISCONNECT (RAB = .RAB);
5                 IF NOT .STATUS
5                 THEN
6                     BEGIN
6                     FILE_ERROR (BACKUP$_POSITERR, .RWSV_SAVE_FAB,
6                             .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
5                     END;
5                 RWSV_SAVE_FAB[FAB$V_SQO] = FALSE;
5                 STATUS = $CONNECT (RAB = .RAB);
5                 IF NOT .STATUS
5                 THEN
6                     BEGIN
6                     FILE_ERROR (BACKUP$_POSITERR, .RWSV_SAVE_FAB,
6                             .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
5                     END;
5                 RAB[RAB$L_BKT] = ..ADDRESS;
5                 END
4             ELSE
5                 BEGIN
5                 CH$MOVE (RAB$S_RFA, .ADDRESS, RAB[RAB$W_RFA]);
5                 RAB[RAB$B_RAC] = RAB$C_RFA;
5                 STATUS = $FIND (RAB = .RAB);
5                 IF NOT .STATUS
5                 THEN
6                     BEGIN
6                     FILE_ERROR (BACKUP$_POSERROR, .RWSV_SAVE_FAB,
6                             .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
6                     RETURN (READ_SEQ_BLOCK (FALSE));
5                     END;
5                 RAB[RAB$B_RAC] = RAB$C_SEQ;
4                 END;
3             END;
3
3         BCB = READ_SEQ_BLOCK (FALSE);
3         BUFFER = .BCB[BCB_BUFFER];
3         TM_COUNT = 0;
3         IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
3         THEN
4             BEGIN
4             TM_COUNT = 1;
4             BLOCK_COUNT = .BLOCK_COUNT + 1;
4             END
3         ELSE IF .BUFFER[BBH$L_NUMBER] GEQU .PREV_SEQ
3         THEN BLOCK_COUNT = .BLOCK_COUNT + .BUFFER[BBH$L_NUMBER] - .PREV_SEQ + 1
3         ELSE BLOCK_COUNT = .BUFFER[BBH$L_NUMBER] - .BLOCK_NEEDED + 1;
3         PREV_SEQ = .BUFFER[BBH$L_NUMBER];
3
3         IF .BUFFER[BBH$L_NUMBER] LEQU .BLOCK_NEEDED
3         OR .QUAL[QUAL_SS_FILE]
3         OR NOT .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
3         OR .BOT
3         THEN EXITLOOP;
3         FREE_BUFFER (.BCB);
```

```
;  812      2363 2      END;
;  813      2364 2
;  814      2365 2  ! We may have spaced to before the block needed. If so, read forward
;  815      2366 2  ! until we get there. If it is really bad, we could skip the one we want
;  816      2367 2  ! and end up farther ahead. C'est la vie.
;  817      2368 2  !
;  818      2369 2
;  819      2370 2  RWSV_IN_SEQ = .BLOCK_NEEDED;
;  820      2371 2  IF .BUFFER[BBH$L_NUMBER] LSSU .BLOCK_NEEDED
;  821      2372 2  THEN
;  822      2373 3      BEGIN
;  823      2374 3      FREE_BUFFER (.BCB);
;  824      2375 3      BCB = READ_SEQ_BLOCK (TRUE);
;  825      2376 2      END;
;  826      2377 2
;  827      2378 2  .BCB
;  828      2379 1  END;                                  ! End of routine REPOSITION


                                                           .EXTRN  SYS$DISCONNECT, SYS$CONNECT
                                                           .EXTRN  SYS$FIND

                                  0FFC 00000 REPOSITION:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       ; 2193
                5E             04   C2 00002               SUBL2   #4, SP                                      ;
                50       0C    AC   D0 00005               MOVL    CUR_BCB, R0                                 ; 2255
                56       0C    A0   D0 00009               MOVL    12(R0), BUFFER                              ;
                7E             01   CE 0000D               MNEGL   #1, PREV_SEQ                                ; 2256
   58     08    A6       04    AC   C3 00010               SUBL3   BLOCK_NEEDED, 8(BUFFER), R8                 ; 2257
                               58   D6 00016               INCL    BLOCK_COUNT                                 ;
                               57   D4 00018               CLRL    TM_COUNT                                    ; 2258
        0870    8F       18    A0   B1 0001A               CMPW    24(R0), #2160                               ; 2259
                               02   12 00020               BNEQ    1$                                          ;
                               57   D6 00022               INCL    TM_COUNT                                    ; 2260
                         0A    A0   95 00024 1$:           TSTB    10(R0)                                      ; 2261
                               09   13 00027               BEQL    3$                                          ;
                               50   DD 00029               PUSHL   R0                                          ; 2262
   00000000G    00             01   FB 0002B 2$:           CALLS   #1, FREE_BUFFER                             ;
                5A 00000000'   FF   0F 00032 3$:           REMQUE  @INPUT_WAIT, BCB                            ; 2264
                               1D   1D 00039               BVS     6$                                          ;
                               5A   DD 0003B               PUSHL   BCB                                         ; 2267
   00000000G    00             01   FB 0003D               CALLS   #1, WAIT                                    ;
        0870    8F       18    AA   B1 00044               CMPW    24(BCB), #2160                              ; 2268
                               02   12 0004A               BNEQ    4$                                          ;
                               57   D6 0004C               INCL    TM_COUNT                                    ; 2269
                               57   D5 0004E 4$:           TSTL    TM_COUNT                                    ; 2270
                               02   12 00050               BNEQ    5$                                          ;
                               58   D6 00052               INCL    BLOCK_COUNT                                 ; 2271
                               5A   DD 00054 5$:           PUSHL   BCB                                         ; 2272
                               D3   11 00056               BRB     2$                                          ;
                   00000000'   EF   D4 00058 6$:           CLRL    RWSV_IN_SEQ                                 ; 2280
                50 00000000'   EF   D0 0005E 7$:           MOVL    RWSV_SAVE_FAB, R0                           ; 2287
   44 00000000'    EF          03   E0 00065               BBS     #3, QUAL+T5, 11$                            ; 2284
   35     40       A0          05   E1 0006D               BBC     #5, 64(R0), 9$                              ; 2287
                               5B   D4 00072               CLRL    BOT                                         ; 2290
                               57   D5 00074               TSTL    TM_COUNT                                    ; 2291
```

```
                                      0A   13 00076       BEQL    8$                                ;
                         7E           57   CE 00078       MNEGL   TM_COUNT, -(SP)                   ; 2292
            00000000G    00           01   FB 0007B       CALLS   #1, SKIP_TM                       ;
                                      58   D5 00082 8$:   TSTL    BLOCK_COUNT                       ; 2293
                                      29   15 00084       BLEQ    10$                               ;
                         7E           58   CE 00086       MNEGL   BLOCK_COUNT, -(SP)                ; 2294
            00000000G    00           01   FB 00089       CALLS   #1, SKIP_RECORD                   ;
            00000870     8F           50   D1 00090       CMPL    R0, #2160                         ;
                                      16   12 00097       BNEQ    10$                               ;
                                      01   DD 00099       PUSHL   #1                                ; 2297
            00000000G    00           01   FB 0009B       CALLS   #1, SKIP_TM                       ;
                         5B           01   D0 000A2       MOVL    #1, BOT                           ; 2298
                                      7C   11 000A5       BRB     14$                               ; 2287
            00000000'    EF      08   BC   D0 000A7 9$:   MOVL    @ADDRESS, RWSV_IN_VBN             ; 2302
                                      72   11 000AF 10$:  BRB     14$                               ; 2284
         59 00000000'    EF 00000050  8F   C1 000B1 11$:  ADDL3   #80, RWSV_SAVE_FAB, RAB           ; 2306
         63       41     A0           05   E1 000BD       BBC     #5, 65(R0), 15$                   ; 2307
                                      59   DD 000C2       PUSHL   RAB                               ; 2310
            00000000G    00           01   FB 000C4       CALLS   #1, SYS$DISCONNECT                ;
                  04     AE           50   D0 000CB       MOVL    R0, STATUS                        ;
                         17      04   AE   E8 000CF       BLBS    STATUS, 12$                       ; 2311
                         7E      08   A9   7D 000D3       MOVQ    8(RAB), -(SP)                     ; 2315
                        00000000'     EF   DD 000D7       PUSHL   RWSV_SAVE_FAB                     ; 2314
                        00000000G     8F   DD 000DD       PUSHL   #BACKUP$_POSITERR                 ;
            00000000G    00           04   FB 000E3       CALLS   #4, FILE_ERROR                    ;
                         50 00000000' EF   D0 000EA 12$:  MOVL    RWSV_SAVE_FAB, R0                 ; 2317
                  04     A0      40   8F   8A 000F1       BICB2   #64, 4(R0)                        ;
                                      59   DD 000F6       PUSHL   RAB                               ; 2318
            00000000G    00           01   FB 000F8       CALLS   #1, SYS$CONNECT                   ;
                  04     AE           50   D0 000FF       MOVL    R0, STATUS                        ;
                         17      04   AE   E8 00103       BLBS    STATUS, 13$                       ; 2319
                         7E      08   A9   7D 00107       MOVQ    8(RAB), -(SP)                     ; 2323
                        00000000'     EF   DD 0010B       PUSHL   RWSV_SAVE_FAB                     ; 2322
                        00000000G     8F   DD 00111       PUSHL   #BACKUP$_POSITERR                 ;
            00000000G    00           04   FB 00117       CALLS   #4, FILE_ERROR                    ;
                  38     A9      08   BC   D0 0011E 13$:  MOVL    @ADDRESS, 56(RAB)                 ; 2325
                                      3D   11 00123 14$:  BRB     17$                               ; 2307
     10   A9      08     BC           06   28 00125 15$:  MOVC3   #6, @ADDRESS, 16(RAB)             ; 2329
                  1E     A9           02   90 0012B       MOVB    #2, 30(RAB)                       ; 2330
                                      59   DD 0012F       PUSHL   RAB                               ; 2331
            00000000G    00           01   FB 00131       CALLS   #1, SYS$FIND                      ;
                  04     AE           50   D0 00138       MOVL    R0, STATUS                        ;
                         1F      04   AE   E8 0013C       BLBS    STATUS, 16$                       ; 2332
                         7E      08   A9   7D 00140       MOVQ    8(RAB), -(SP)                     ; 2336
                        00000000'     EF   DD 00144       PUSHL   RWSV_SAVE_FAB                     ; 2335
                        00000000G     8F   DD 0014A       PUSHL   #BACKUP$_POSERROR                 ;
            00000000G    00           04   FB 00150       CALLS   #4, FILE_ERROR                    ;
                                      7E   D4 00157       CLRL    -(SP)                             ; 2337
                FD68     CF           01   FB 00159       CALLS   #1, READ_SEQ_BLOCK                ;
                                           04 0015E       RET                                       ;
                                 1E   A9   94 0015F 16$:  CLRB    30(RAB)                           ; 2339
                                      7E   D4 00162 17$:  CLRL    -(SP)                             ; 2343
                FD5D     CF           01   FB 00164       CALLS   #1, READ_SEQ_BLOCK                ;
                         5A           50   D0 00169       MOVL    R0, BCB                           ;
                         56      0C   AA   D0 0016C       MOVL    12(BCB), BUFFER                   ; 2344
                                      57   D4 00170       CLRL    TM_COUNT                          ; 2345
                0870     8F      18   AA   B1 00172       CMPW    24(BCB), #2160                    ; 2346
```

```
                                  07   12 00178          BNEQ    18$                                  ;
                57                01   D0 0017A          MOVL    #1, TM_COUNT                          ; 2349
                                  58   D6 0017D          INCL    BLOCK_COUNT                           ; 2350
                                  1A   11 0017F          BRB     21$                                   ; 2346
                6E         08     A6   D1 00181  18$:    CMPL    8(BUFFER), PREV_SEQ                   ; 2352
                                  0A   1F 00185          BLSSU   19$                                   ;
      50        58         08     A6   C1 00187          ADDL3   8(BUFFER), BLOCK_COUNT, R0            ; 2353
                50                6E   C2 0018C          SUBL2   PREV_SEQ, R0                          ;
                                  06   11 0018F          BRB     20$                                   ;
      50   08   A6         04     AC   C3 00191  19$:    SUBL3   BLOCK_NEEDED, 8(BUFFER), R0           ; 2354
                58         01     A0   9E 00197  20$:    MOVAB   1(R0), BLOCK_COUNT                    ;
                6E         08     A6   D0 0019B  21$:    MOVL    8(BUFFER), PREV_SEQ                   ; 2355
           04   AC         08     A6   D1 0019F          CMPL    8(BUFFER), BLOCK_NEEDED               ; 2357
                                  23   1B 001A4          BLEQU   22$                                   ;
      1B 00000000'  EF            03   E0 001A6          BBS     #3, QUAL+15, 22$                      ; 2358
                50 00000000'      EF   D0 001AE          MOVL    RWSV_SAVE_FAB, R0                     ; 2359
      0F   40   A0                05   E1 001B5          BBC     #5, 64(R0), 22$                       ;
                0C                5B   E8 001BA          BLBS    BOT, 22$                              ; 2360
                                  5A   DD 001BD          PUSHL   BCB                                   ; 2362
         00000000G  00            01   FB 001BF          CALLS   #1, FREE_BUFFER                       ;
                                FE95   31 001C6          BRW     7$                                    ; 2281
         00000000'  EF         04 AC   D0 001C9  22$:    MOVL    BLOCK_NEEDED, RWSV_IN_SEQ             ; 2370
           04   AC         08     A6   D1 001D1          CMPL    8(BUFFER), BLOCK_NEEDED               ; 2371
                                  13   1E 001D6          BGEQU   23$                                   ;
                                  5A   DD 001D8          PUSHL   BCB                                   ; 2374
         00000000G  00            01   FB 001DA          CALLS   #1, FREE_BUFFER                       ;
                                  01   DD 001E1          PUSHL   #1                                    ; 2375
              FCDE  CF            01   FB 001E3          CALLS   #1, READ_SEQ_BLOCK                    ;
                5A                50   D0 001E8          MOVL    R0, BCB                               ;
                50                5A   D0 001EB  23$:    MOVL    BCB, R0                               ; 2379
                                       04 001EE          RET                                           ;

; Routine Size:  495 bytes,    Routine Base:  CODE + 0420
```

```
%SBTTL 'XORCIZE - apply XOR recovery'
ROUTINE XORCIZE (BLOCK_NEEDED, CUR_BCB) =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine applies XOR recovery to recover the indicated
!       lost block. This is done by backspacing to the start of the
!       current group and XORing all blocks (except the one in question)
!       up through the next XOR block.
!
! CALLING SEQUENCE:
!       XORCIZE (BLOCK_NEEDED, CUR_BCB)
!
! INPUT PARAMETERS:
!       BLOCK_NEEDED: sequence number of block to be recovered
!       CUR_BCB: BCB of block just read
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       BCB of recovered block
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN

BUILTIN
        INSQUE;

MAP
        CUR_BCB         : REF BBLOCK;   ! input BCB arg

LOCAL
        RAB             : REF BBLOCK,   ! RAB for reading input file
        SAVE_ADDR       : BBLOCK [RAB$S_RFA], ! saved file address
        SAVE_ERRORS,                    ! save soft error count
        P1,                             ! XOR buffer pointer
        P2,                             ! XOR buffer pointer
        BCB             : REF BBLOCK,   ! BCB of current buffer
        XOR_BCB         : REF BBLOCK,   ! BCB of XOR buffer
        BUFFER          : REF BBLOCK;   ! current I/O buffer

EXTERNAL ROUTINE
        FREE_BUFFER;                    ! free an I/O buffer

```

```
2 ! Save the current position if the input is a file or seq disk.
2 !
2
2 SAVE_ERRORS = .RWSV_IN_ERRORS;
2 IF .RWSV_IN_GROUP_SIZE EQL 0 THEN RETURN .CUR_BCB;
2 SAVE_ADDR = .CUR_BCB[BCB_BLOCKNUM];
2 IF .QUAL[QUAL_SS_FILE]
2 THEN
3     BEGIN
3     RAB = RWSV_SAVE_FAB[FC_RAB];
3     CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], SAVE_ADDR);
2     END;
2
2 ! Special case the situation in which the block lost is the first
2 ! of an XOR group, since no special repositioning is necessary.
2 !
2
2 BUFFER = .CUR_BCB[BCB_BUFFER];
2 IF .BLOCK_NEEDED EQL .RWSV_IN_XOR_SEQ + 1
2 THEN
3     BEGIN
3     RWSV_IN_SEQ = .BLOCK_NEEDED + 1;
3     IF .BUFFER[BBH$L_NUMBER] EQL .BLOCK_NEEDED
3     THEN
4         BEGIN
4         FREE_BUFFER (.CUR_BCB);
4         BCB = READ_SEQ_BLOCK (TRUE);
4         END
3     ELSE IF .BUFFER[BBH$L_NUMBER] EQL .RWSV_IN_SEQ
3     THEN BCB = .CUR_BCB
3     ELSE RETURN .CUR_BCB;
3     END
3
2 ELSE
2     BCB = REPOSITION (.RWSV_IN_XOR_SEQ + 1, RWSV_IN_XOR_RFA, .CUR_BCB);
2 BUFFER = .BCB[BCB_BUFFER];
2
2 ! Accumulate the blocks in an XOR buffer, skipping the one we are
2 ! after. Note that we allow for a group size one larger than specified,
2 ! to accomodate some boundary conditions on sequential disk.
2 ! Temporarily decrement the buffer count to account for the XOR buffer
2 ! we are sitting on, to prevent the read-ahead from eating too many buffers.
2 !
2
2 COM_BUFF_COUNT = .COM_BUFF_COUNT - 1;
2 XOR_BCB = 0;
3 IF (
3 DECR J FROM MINU (.RWSV_IN_GROUP_SIZE+1, 100) TO 1
3 DO
4     BEGIN
4     IF .XOR_BCB NEQ 0
4     THEN
5         BEGIN
5         BCB = READ_SEQ_BLOCK (TRUE);
5         BUFFER = .BCB[BCB_BUFFER];
4         END;
4
```

```
4       IF NOT .BCB[BCB_STATUS]
4       OR .BUFFER[BBH$L_NUMBER] GTRU .RWSV_IN_SEQ
4       THEN EXITLOOP -1;
4
4       IF .XOR_BCB EQL 0
4       THEN
4           XOR_BCB = .BCB
4       ELSE
5           BEGIN
5           P1 = .BUFFER + BBH$K_COMMON;
5           P2 = .XOR_BCB[BCB_BUFFER] + BBH$K_COMMON;
5           DECR J FROM (.BCB[BCB_SIZE]-BBH$K_COMMON)/16 TO 1
5           DO
6               BEGIN
6               .P2 = ..P2 XOR ..P1;
6               P1 = .P1 + 4;
6               P2 = .P2 + 4;
6               .P2 = ..P2 XOR ..P1;
6               P1 = .P1 + 4;
6               P2 = .P2 + 4;
6               .P2 = ..P2 XOR ..P1;
6               P1 = .P1 + 4;
6               P2 = .P2 + 4;
6               .P2 = ..P2 XOR ..P1;
6               P1 = .P1 + 4;
6               P2 = .P2 + 4;
5               END;
4           END;
4
4       RWSV_IN_SEQ = .BUFFER[BBH$L_NUMBER] + 1;
4       IF .RWSV_IN_SEQ EQL .BLOCK_NEEDED
4       THEN RWSV_IN_SEQ = .RWSV_IN_SEQ + 1;
4       IF .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_XORBLOCK
4       THEN EXITLOOP 0;
4       IF .BCB NEQ .XOR_BCB THEN FREE_BUFFER (.BCB);
4       END
3   )
3
3   ! If XOR recovery was not successful, reposition to the desired block
3   ! and return it.
3   !
3
2   THEN
3       BEGIN
3       IF .XOR_BCB NEQ 0
3       THEN FREE_BUFFER (.XOR_BCB);
3       BCB =  REPOSITION (.BLOCK_NEEDED, SAVE_ADDR, .BCB);
3       RWSV_IN_ERRORS = .SAVE_ERRORS;
3       COM_BUFF_COUNT = .COM_BUFF_COUNT + 1;
3       .BCB
3       END
3
3   ! If XOR recovery was successful, reposition to the block following
3   ! the one recovered, and link it into the FRONT of the input wait
3   ! list, so it will be read next. Return the recovered block.
3   !
3
```

```
; 1001          2551  2 ELSE
; 1002          2552  3     BEGIN
; 1003          2553  3     IF .BUFFER[BBH$L_NUMBER] NEQ .BLOCK_NEEDED + 1
; 1004          2554  3     THEN BCB = REPOSITION (.BLOCK_NEEDED + 1, SAVE_ADDR, .BCB);
; 1005          2555  3     RWSV_IN_SEQ = .BLOCK_NEEDED;
; 1006          2556  3     IF .BCB NEQ .XOR_BCB
; 1007          2557  3     THEN
; 1008          2558  4         BEGIN
; 1009          2559  4         BCB[BCB_STATE] = BCB_S_READ;
; 1010          2560  4         INSQUE (.BCB, INPUT_WAIT[0]);
; 1011          2561  3         END;
; 1012          2562  3     RWSV_IN_ERRORS = .SAVE_ERRORS - 1;
; 1013          2563  3     RWSV_IN_XORUSE = .RWSV_IN_XORUSE + 1;
; 1014          2564  3     COM_BUFF_COUNT = .COM_BUFF_COUNT + 1;
; 1015          2565  3     BUFFER = .XOR_BCB[BCB_BUFFER];
; 1016          2566  3     BUFFER[BBH$W_APPLIC] = BACKUP$K_DATABLOCK;
; 1017          2567  3     BUFFER[BBH$L_NUMBER] = .BLOCK_NEEDED;
; 1018          2568  3     .XOR_BCB
; 1019          2569  3     END
; 1020          2570  3
; 1021          2571  1 END;                                      ! End of routine XORCIZE



                                        0FFC 00000 XORCIZE:.WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2381
                  5B 00000000G    00   9E 00002         MOVAB   FREE_BUFFER, R11                              ;
                  5A 00000000'    EF   9E 00009         MOVAB   RWSV_IN_SEQ, R10                              ;
                  5E              08   C2 00010         SUBL2   #8, SP                                        ;
                  59        18    AA   3C 00013         MOVZWL  RWSV_IN_ERRORS, SAVE_ERRORS                   ; 2440
                            14    AA   D5 00017         TSTL    RWSV_IN_GROUP_SIZE                            ; 2441
                                  05   12 0001A         BNEQ    1$                                            ;
                  50        08    AC   D0 0001C         MOVL    CUR_BCB, R0                                   ;
                                       04 00020         RET                                                   ;
                  56        08    AC   D0 00021 1$:     MOVL    CUR_BCB, R6                                   ; 2442
                  6E        14    A6   D0 00025         MOVL    20(R6), SAVE_ADDR                             ;
0E        57      AA              03   E1 00029         BBC     #3, QUAL+15, 2$                               ; 2443
50        F4      AA 00000050     8F   C1 0002E         ADDL3   #80, RWSV_SAVE_FAB, RAB                       ; 2446
6E        10      A0              06   28 00037         MOVC3   #6, 16(RAB), SAVE_ADDR                        ; 2447
                  52        0C    A6   D0 0003C 2$:     MOVL    12(R6), BUFFER                                ; 2454
                  58        04    AC   D0 00040         MOVL    BLOCK_NEEDED, R8                              ; 2455
50        08      AA              01   C1 00044         ADDL3   #1, RWSV_IN_XOR_SEQ, R0                       ;
                  50              58   D1 00049         CMPL    R8, R0                                        ;
                                  27   12 0004C         BNEQ    5$                                            ;
                  6A        01    A8   9E 0004E         MOVAB   1(R8), RWSV_IN_SEQ                            ; 2458
                  58        08    A2   D1 00052         CMPL    8(BUFFER), R8                                 ; 2459
                                  0E   12 00056         BNEQ    3$                                            ;
                                  56   DD 00058         PUSHL   R6                                            ; 2462
                  6B              01   FB 0005A         CALLS   #1, FREE_BUFFER                               ;
                                  01   DD 0005D         PUSHL   #1                                            ; 2463
          FC73    CF              01   FB 0005F         CALLS   #1, READ_SEQ_BLOCK                            ;
                                  1B   11 00064         BRB     6$                                            ;
                  6A        08    A2   D1 00066 3$:     CMPL    8(BUFFER), RWSV_IN_SEQ                        ; 2465
                                  05   12 0006A         BNEQ    4$                                            ;
                  55              56   D0 0006C         MOVL    R6, BCB                                       ; 2466
                                  13   11 0006F         BRB     7$                                            ;
```

```
                     50          56  D0 00071  4$:     MOVL    R6, R0                                  ; 2467
                                     04 00074          RET
                                 56  DD 00075  5$:     PUSHL   R6                                      ; 2471
                     0C          AA  9F 00077          PUSHAB  RWSV_IN_XOR_RFA                         ;
                                 50  DD 0007A          PUSHL   R0                                      ;
            FD90     CF          03  FB 0007C          CALLS   #3, REPOSITION                          ;
                     55          50  D0 00081  6$:     MOVL    R0, BCB                                 ;
                     52   0C     A5  D0 00084  7$:     MOVL    12(BCB), BUFFER                         ; 2472
                        00C5     CA  97 00088          DECB    COM_BUFF_COUNT                          ; 2481
                                 56  D4 0008C          CLRL    XOR_BCB                                 ; 2482
       50      14    AA          01  C1 0008E          ADDL3   #1, RWSV_IN_GROUP_SIZE, R0              ; 2484
                     57          50  D0 00093          MOVL    R0, R7                                  ;
        00000064     8F          57  D1 00096          CMPL    R7, #100                                ;
                                 04  1B 0009D          BLEQU   8$                                      ;
                     57   64     8F  9A 0009F          MOVZBL  #100, R7                                ;
                                 57  D6 000A3  8$:     INCL    J                                       ;
                                 67  11 000A5          BRB     16$                                     ;
                                 56  D5 000A7  9$:     TSTL    XOR_BCB                                 ; 2487
                                 0E  13 000A9          BEQL    10$                                     ;
                                 01  DD 000AB          PUSHL   #1                                      ; 2490
            FC25     CF          01  FB 000AD          CALLS   #1, READ_SEQ_BLOCK                      ;
                     55          50  D0 000B2          MOVL    R0, BCB                                 ;
                     52   0C     A5  D0 000B5          MOVL    12(BCB), BUFFER                         ; 2491
                     54   18     A5  E9 000B9  10$:    BLBC    24(BCB), 17$                            ; 2494
                     6A   08     A2  D1 000BD          CMPL    8(BUFFER), RWSV_IN_SEQ                  ; 2495
                                 4E  1A 000C1          BGTRU   17$                                     ;
                                 56  D5 000C3          TSTL    XOR_BCB                                 ; 2498
                                 05  12 000C5          BNEQ    11$                                     ;
                     56          55  D0 000C7          MOVL    BCB, XOR_BCB                            ; 2500
                                 26  11 000CA          BRB     14$                                     ;
                     53   20     A2  9E 000CC  11$:    MOVAB   32(R2), P1                              ; 2503
       54      0C    A6          20  C1 000D0          ADDL3   #32, 12(XOR_BCB), P2                    ; 2504
                     50   08     A5  3C 000D5          MOVZWL  8(BCB), R0                              ; 2505
                     50          20  C2 000D9          SUBL2   #32, R0                                 ;
                     50          10  C6 000DC          DIVL2   #16, R0                                 ;
                                 50  D6 000DF          INCL    J                                       ;
                                 0C  11 000E1          BRB     13$                                     ;
                     84          83  CC 000E3  12$:    XORL2   (P1)+, (P2)+                            ; 2508
                     84          83  CC 000E6          XORL2   (P1)+, (P2)+                            ; 2511
                     84          83  CC 000E9          XORL2   (P1)+, (P2)+                            ; 2514
                     84          83  CC 000EC          XORL2   (P1)+, (P2)+                            ; 2517
                     F1          50  F5 000EF  13$:    SOBGTR  J, 12$                                  ; 2505
       6A      08    A2          01  C1 000F2  14$:    ADDL3   #1, 8(BUFFER), RWSV_IN_SEQ              ; 2523
                     58          6A  D1 000F7          CMPL    RWSV_IN_SEQ, R8                         ; 2524
                                 02  12 000FA          BNEQ    15$                                     ;
                                 6A  D6 000FC          INCL    RWSV_IN_SEQ                             ; 2525
                     02   06     A2  B1 000FE  15$:    CMPW    6(BUFFER), #2                           ; 2526
                                 2E  13 00102          BEQL    19$                                     ;
                     56          55  D1 00104          CMPL    BCB, XOR_BCB                            ; 2528
                                 05  13 00107          BEQL    16$                                     ;
                                 55  DD 00109          PUSHL   BCB                                     ;
                     6B          01  FB 0010B          CALLS   #1, FREE_BUFFER                         ;
                     96          57  F5 0010E  16$:    SOBGTR  J, 9$                                   ; 2484
                                 56  D5 00111  17$:    TSTL    XOR_BCB                                 ; 2538
                                 05  13 00113          BEQL    18$                                     ;
                                 56  DD 00115          PUSHL   XOR_BCB                                 ; 2539
                     6B          01  FB 00117          CALLS   #1, FREE_BUFFER                         ;
```

```
                                 55  DD 0011A 18$:   PUSHL   BCB                                        ; 2540
                         04      AE  9F 0011C        PUSHAB  SAVE_ADDR                                  ;
                                 58  DD 0011F        PUSHL   R8                                         ;
              FCEB   CF          03  FB 00121        CALLS   #3, REPOSITION                             ;
                     55          50  D0 00126        MOVL    R0, BCB                                    ;
                18   AA          59  B0 00129        MOVW    SAVE_ERRORS, RWSV_IN_ERRORS                ; 2541
                     50          55  D0 0012D        MOVL    BCB, R0                                    ; 2543
                                 42  11 00130        BRB     22$                                        ;
                     50  01      A8  9E 00132 19$:   MOVAB   1(R8), R0                                  ; 2553
                     50  08      A2  D1 00136        CMPL    8(BUFFER), R0                              ;
                                 0F  13 0013A        BEQL    20$                                        ;
                                 55  DD 0013C        PUSHL   BCB                                        ; 2554
                         04      AE  9F 0013E        PUSHAB  SAVE_ADDR                                  ;
                                 50  DD 00141        PUSHL   R0                                         ;
              FCC9   CF          03  FB 00143        CALLS   #3, REPOSITION                             ;
                     55          50  D0 00148        MOVL    R0, BCB                                    ;
                     6A          58  D0 0014B 20$:   MOVL    R8, RWSV_IN_SEQ                            ; 2555
                     56          55  D1 0014E        CMPL    BCB, XOR_BCB                               ; 2556
                                 09  13 00151        BEQL    21$                                        ;
                0A   A5          01  90 00153        MOVB    #1, 10(BCB)                                ; 2559
              FF04   CA          65  0E 00157        INSQUE  (BCB), INPUT_WAIT                          ; 2560
     18   AA         59          01  A3 0015C 21$:   SUBW3   #1, SAVE_ERRORS, RWSV_IN_ERRORS            ; 2562
                         1A      AA  B6 00161        INCW    RWSV_IN_XORUSE                             ; 2563
                     52  0C      A6  D0 00164        MOVL    12(XOR_BCB), BUFFER                        ; 2565
                06   A2          01  B0 00168        MOVW    #1, 6(BUFFER)                              ; 2566
                08   A2  04      AC  D0 0016C        MOVL    BLOCK_NEEDED, 8(BUFFER)                    ; 2567
                     50          56  D0 00171        MOVL    XOR_BCB, R0                                ; 2568
                       00C5      CA  96 00174 22$:   INCB    COM_BUFF_COUNT                             ; 2542
                                     04 00178        RET                                                ; 2571

; Routine Size:  377 bytes,    Routine Base:  CODE + 060F
```

```
1 %SBTTL 'MATCH_SSNAME - match save set name string'
1 ROUTINE MATCH_SSNAME (LABEL_BUFFER) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine determines whether the specified tape header label
1 !       matches the save set file name in COM_SSNAME.
1 !
1 ! CALLING SEQUENCE:
1 !       MATCH_SSNAME (LABEL_BUFFER)
1 !
1 ! INPUT PARAMETERS:
1 !       LABEL_BUFFER: buffer containing tape header label
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       TRUE if label matches
1 !       FALSE if not
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2
2 LABEL_BUFFER    : REF BBLOCK;           ! buffer containing tape label
2
2 LINKAGE
2       L_MATCH_NAME    = JSB (REGISTER = 4, REGISTER = 5,
2                             REGISTER = 2, REGISTER = 3) :
2                             NOPRESERVE (2,3,4,5);
2
2 LOCAL
2       LABEL_LENGTH;                   ! Length of the HDR1 file identifier
2
2 EXTERNAL ROUTINE
2       FMG$MATCH_NAME  : L_MATCH_NAME; ! Compare names with wildcards
2
2 ! The specified file name matches (1) if it is null or (2) if it matches
2 ! exactly or (3) if the specified file name has a trailing dot and the tape
2 ! file name has a null type (with no dot).
2 !
2
2 IF
```

```
; 1080    2629 2       .COM_SSNAME[DSC$W_LENGTH] EQL 0
; 1081    2630 2  OR
; 1082    2631 3       (.COM_SSNAME[DSC$W_LENGTH] EQL 1
; 1083    2632 3         AND .VECTOR[.COM_SSNAME[DSC$A_POINTER], 0; ,BYTE] EQL '.')
; 1084    2633 2  THEN RETURN TRUE;
; 1085    2634 2
; 1086    2635 2  LABEL_LENGTH = HD1$S_FILEID;
; 1087    2636 2  WHILE .VECTOR[LABEL_BUFFER[HD1$T_FILEID], .LABEL_LENGTH-1; ,BYTE] EQL %C' '
; 1088    2637 2  AND .LABEL_LENGTH GTR 0
; 1089    2638 2  DO LABEL_LENGTH = .LABEL_LENGTH - 1;
; 1090    2639 2
; 1091    2640 2  IF
; 1092    2641 2       FMG$MATCH_NAME(.COM_SSNAME[DSC$W_LENGTH],
; 1093    2642 2                      .COM_SSNAME[DSC$A_POINTER],
; 1094    2643 2                      .LABEL_LENGTH,
; 1095    2644 2                      LABEL_BUFFER[HD1$T_FILEID])
; 1096    2645 3  OR (
; 1097    2646 3       .VECTOR[.COM_SSNAME[DSC$A_POINTER], .COM_SSNAME[DSC$W_LENGTH]-1; ,BYTE] EQL '.'
; 1098    2647 3       AND
; 1099    2648 3       FMG$MATCH_NAME(.COM_SSNAME[DSC$W_LENGTH]-1,
; 1100    2649 3                      .COM_SSNAME[DSC$A_POINTER],
; 1101    2650 3                      .LABEL_LENGTH,
; 1102    2651 3                      LABEL_BUFFER[HD1$T_FILEID])
; 1103    2652 3       )
; 1104    2653 2  THEN TRUE
; 1105    2654 2  ELSE FALSE
; 1106    2655 2
; 1107    2656 1  END;                                      ! End of routine MATCH_SSNAME


                                                .EXTRN  FMG$MATCH_NAME

                                 0FFC 00000 MATCH_SSNAME:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11           ; 2573
           59 00000000G   00   9E 00002         MOVAB   FMG$MATCH_NAME, R9                              :
           58 00000000'   EF   9E 00009         MOVAB   COM_SSNAME+4, R8                                :
           54          FC A8   3C 00010         MOVZWL  COM_SSNAME, R4                                  ; 2629
                          54   13 00014         BEQL    4$                                              :
           01             54   B1 00016         CMPW    R4, #1                                          ; 2631
                          06   12 00019         BNEQ    1$                                              :
           2E          00 B8   91 0001B         CMPB    @COM_SSNAME+4, #46                              ; 2632
                          49   13 0001F         BEQL    4$                                              :
           57             11   D0 00021 1$:     MOVL    #17, LABEL_LENGTH                               ; 2635
           50          04 AC   D0 00024         MOVL    LABEL_BUFFER, R0                                ; 2636
           56          04 A0   9E 00028         MOVAB   4(R0), R6                                       :
           20        FF A746   91 0002C 2$:     CMPB    -1(LABEL_LENGTH)[R6], #32                       :
                          08   12 00031         BNEQ    3$                                              :
                          57   D5 00033         TSTL    LABEL_LENGTH                                    ; 2637
                          04   15 00035         BLEQ    3$                                              :
                          57   D7 00037         DECL    LABEL_LENGTH                                    ; 2638
                          F1   11 00039         BRB     2$                                              :
           53             56   D0 0003B 3$:     MOVL    R6, R3                                          ; 2644
           52             57   D0 0003E         MOVL    LABEL_LENGTH, R2                                :
           55             68   D0 00041         MOVL    COM_SSNAME+4, R5                                :
                          69   16 00044         JSB     FMG$MATCH_NAME                                  :
           21             50   E8 00046         BLBS    R0, 4$                                          :
```

```
                          50      FC   A8  3C 00049        MOVZWL  COM_SSNAME, R0                 ; 2646
                          50           68  C0 0004D        ADDL2   COM_SSNAME+4, R0               ;
                          2E      FF   A0  91 00050        CMPB    -1(R0), #46                    ;
                                       18  12 00054        BNEQ    5$                             ;
                          54      FC   A8  3C 00056        MOVZWL  COM_SSNAME, R4                 ; 2648
                                       54  D7 0005A        DECL    R4                             ;
                          53           56  D0 0005C        MOVL    R6, R3                         ; 2651
                          52           57  D0 0005F        MOVL    LABEL_LENGTH, R2               ;
                          55           68  D0 00062        MOVL    COM_SSNAME+4, R5               ;
                                       69  16 00065        JSB     FMG$MATCH_NAME                 ;
                          04           50  E9 00067        BLBC    R0, 5$                         ;
                          50           01  D0 0006A 4$:    MOVL    #1, R0                         ; 2640
                                           04 0006D        RET                                    ;
                                       50  D4 0006E 5$:    CLRL    R0                             ;
                                           04 00070        RET                                    ; 2656

; Routine Size:  113 bytes,    Routine Base:  CODE + 0788
```

```
1 %SBTTL 'INIT_SAVE_DISK - initialize sequential disk save set'
1 ROUTINE INIT_SAVE_DISK (CONTINUE, FILE_ID) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine opens the save set file on an offline save
1 !       set disk.
1 !
1 ! CALLING SEQUENCE:
1 !       INIT_SAVE_DISK (CONTINUE)
1 !
1 ! INPUT PARAMETERS:
1 !       CONTINUE: TRUE to open next file segment
1 !                 FALSE to indicate normal open
1 !       FILE_ID: address of file ID if CONTINUE is true
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
2 BUILTIN
2       ROT;
2
2 MAP
2       FILE_ID         : REF BBLOCK;   ! continuation file ID arg
2
2 LOCAL
2       FAB             : REF BBLOCK,   ! pointer to save set FAB
2       NAM             : REF BBLOCK,   ! pointer to NAM block
2       STATUS,                         ! general status value
2       IO_STATUS       : VECTOR [4, WORD], ! I/O status block
2       FIB             : BBLOCK [FIB$C_LENGTH], ! FIB to access file
2       FIB_DESC        : VECTOR [2],   ! descriptor for above
2       RECATTR         : BBLOCK [FAT$C_LENGTH], ! record attributes buffer
2       SEG_NUMBER      : WORD,         ! file segment number
2       ATT_CONTROL     : BBLOCK [20];  ! attribute control list
2
2 BIND
2       ATT_CONTROL0    = ATT_CONTROL+00 : BBLOCK,
2       ATT_CONTROL1    = ATT_CONTROL+08 : BBLOCK,
```

```
2           ATTR_END          = ATT_CONTROL+16;
2
2   EXTERNAL ROUTINE
2           READY_DISK,                         ! ready disk for input
2           INIT_DIR_SCAN,                      ! initialize directory scan
2           FIND_NEXT,                          ! find a file
2           FREE_DIR_DATA,                      ! clean up file scan context
2           FILE_ERROR;                         ! signal file related error
2
2
2   ! Set up the input disk.
2   !
2
2   FAB = .RWSV_SAVE_FAB;
2   FAB[FAB$L_STS] = 1;
2   FAB[FAB$L_STV] = STA_IN_CHAN;
2   NAM = .FAB[FAB$L_NAM];
2   IF NOT .CONTINUE
2   THEN
3       BEGIN
3       READY_DISK (0);
3
3   ! Use the file scan logic to find the file.
3   !
3
3       INIT_DIR_SCAN (
3               STA_IN_CHAN,
3               .NAM,
3               BBLOCK [.QUAL[QUAL_INPU_LIST], QUAL_DEV_DESC],
3               BBLOCK [.QUAL[QUAL_INPU_LIST], QUAL_EXP_DESC],
3               FALSE,
3               .INPUT_MTL[MTL_RVN_BASE],
3               0);
3       STATUS = FIND_NEXT ();
3       IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB, SS$_NOSUCHFILE);
3       FREE_DIR_DATA ();
3
3   ! Set up the FIB and issue the QIO to open the file.
3   !
3
3       CH$FILL (0, FIB$C_LENGTH, FIB);
3       FIB[FIB$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
3       FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
3       FIB[FIB$W_FID_RVN] = .NAM[NAM$W_FID_RVN];
3       END
3
3   ! A file continuation is simply opened by file ID.
3   !
3
2   ELSE
3       BEGIN
3       IF .INPUT_MTL[MTL_SEQ_DISK]
3       THEN READY_DISK (0);
3       CH$FILL (0, FIB$C_LENGTH, FIB);
3       FIB[FIB$W_FID_NUM] = .FILE_ID[FID$W_NUM];
3       FIB[FIB$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
3       FIB[FIB$W_FID_RVN] = .FILE_ID[FID$W_RVN];
```

```
  3         FIB[FIB$L_EXVBN] = .RWSV_IN_VBN;
  2         END;
  2
  2 FIB_DESC[0] = FIB$C_LENGTH;
  2 FIB_DESC[1] = FIB;
  2
  2 ATT_CONTROL0[ATR$W_SIZE] = ATR$S_RECATTR;
  2 ATT_CONTROL0[ATR$W_TYPE] = ATR$C_RECATTR;
  2 ATT_CONTROL0[ATR$L_ADDR] = RECATTR;
  2 ATT_CONTROL1[ATR$W_SIZE] = ATR$S_SEGNUM;
  2 ATT_CONTROL1[ATR$W_TYPE] = ATR$C_SEGNUM;
  2 ATT_CONTROL1[ATR$L_ADDR] = SEG_NUMBER;
  2 ATTR_END = 0;
  2
P 2 STATUS = $QIOW (CHAN = STA_IN_CHAN,
P 2                 IOSB = IO_STATUS,
P 2                 FUNC = IO$_ACCESS OR IO$M_ACCESS,
P 2                 P1   = FIB_DESC,
P 2                 P5   = ATT_CONTROL
  2                 );
  2 IF .STATUS THEN STATUS = .IO_STATUS[0];
  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB, .STATUS);
  2
  2 ! Validate attributes of the save set file.
  2 !
  2
  2 IF .RECATTR[FAT$B_RTYPE] NEQ FAT$C_FIXED
  2 OR .RECATTR[FAT$B_RATTRIB] NEQ 0
  2 OR .(RECATTR[FAT$W_RSIZE])<0,9> NEQ 0
  2 OR .RECATTR[FAT$W_RSIZE] LSSU 2048
  2 THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
  2
  2 IF NOT .CONTINUE
  2 THEN
  3     BEGIN
  3     IF .SEG_NUMBER NEQ 0
  3     THEN
  3         IF .QUAL[QUAL_IMAG]
  3         THEN FILE_ERROR (BACKUP$_NOT1STVOL+STS$K_SEVERE, .FAB)
  3         ELSE FILE_ERROR (BACKUP$_NOT1STVOL, .FAB);
  3     QUAL[QUAL_BLOC_VALUE] = .RECATTR[FAT$W_RSIZE];
  3     RWSV_EOF = ROT (.RECATTR[FAT$L_EFBLK], 16);
  3     IF .RECATTR[FAT$W_FFBYTE] EQL 0
  3     THEN RWSV_EOF = .RWSV_EOF - 1;
  3     RWSV_VOL_NUMBER = .FIB[FIB$B_FID_RVN];
  3     RWSV_SEG_NUMBER = .SEG_NUMBER;
  3     END
  3
  2 ELSE
  3     BEGIN
  3     IF .SEG_NUMBER NEQ .RWSV_SEG_NUMBER
  3     THEN FILE_ERROR (BACKUP$_WRONGVOL, .FAB);
  3     IF .RECATTR[FAT$W_RSIZE] NEQ .QUAL[QUAL_BLOC_VALUE]
  3     THEN FILE_ERROR (BACKUP$_BADBLKSIZE, .FAB);
  3     END;
  2
  1 END;                                    ! End of routine INIT_SAVE_DISK
```

```
; INFO#250                      L1:2806
; Referenced LOCAL symbol SEG_NUMBER is probably not initialized


                                                                        .EXTRN  READY_DISK, INIT_DIR_SCAN
                                                                        .EXTRN  FIND_NEXT, FREE_DIR_DATA
                                                                        .EXTRN  STA_QIOW

                                                   0FFC 00000 INIT_SAVE_DISK:
                                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 2658
                              5B 00000000G   00   9E 00002              MOVAB   READY_DISK, R11                                 ;
                              5A 00000000G   00   9E 00009              MOVAB   FILE_ERROR, R10                                 ;
                              59 00000000'   EF   9E 00010              MOVAB   INPUT_MTL, R9                                   ;
                              5E      FF78   CE   9E 00017              MOVAB   -136(SP), SP                                    ;
                              57      F93C   C9   D0 0001C              MOVL    RWSV_SAVE_FAB, FAB                              ; 2727
                        08    A7             01   D0 00021              MOVL    #1, 8(FAB)                                      ; 2728
                        0C    A7 0001FFFF    8F   D0 00025              MOVL    #131071, 12(FAB)                                ; 2729
                              56        28   A7   D0 0002D              MOVL    40(FAB), NAM                                    ; 2730
                              64        04   AC   E8 00031              BLBS    CONTINUE, 2$                                    ; 2731
                                             7E   D4 00035              CLRL    -(SP)                                           ; 2734
                              6B             01   FB 00037              CALLS   #1, READY_DISK                                  ;
                                             7E   D4 0003A              CLRL    -(SP)                                           ; 2743
                              50             69   D0 0003C              MOVL    INPUT_MTL, R0                                   ; 2745
                              7E        30   A0   9A 0003F              MOVZBL  48(R0), -(SP)                                   ;
                                             7E   D4 00043              CLRL    -(SP)                                           ; 2743
             7E      F990     C9             08   C1 00045              ADDL3   #8, QUAL, -(SP)                                 ;
             7E      F990     C9             10   C1 0004B              ADDL3   #16, QUAL, -(SP)                                ; 2742
                                             56   DD 00051              PUSHL   NAM                                             ; 2743
                                 0001FFFF    8F   DD 00053              PUSHL   #131071                                         ;
                   00000000G  00             07   FB 00059              CALLS   #7, INIT_DIR_SCAN                               ;
                   00000000G  00             00   FB 00060              CALLS   #0, FIND_NEXT                                   ; 2747
                              58             50   D0 00067              MOVL    R0, STATUS                                      ;
                              10             58   E8 0006A              BLBS    STATUS, 1$                                      ; 2748
                              7E      0910   8F   3C 0006D              MOVZWL  #2320, -(SP)                                    ;
                                             57   DD 00072              PUSHL   FAB                                             ;
                                 00000000G   8F   DD 00074              PUSHL   #BACKUP$_OPENIN+4                               ;
                              6A             03   FB 0007A              CALLS   #3, FILE_ERROR                                  ;
                   00000000G  00             00   FB 0007D 1$:          CALLS   #0, FREE_DIR_DATA                               ; 2749
  0040    8F           00     6E             00   2C 00084              MOVC5   #0, (SP), #0, #64, FIB                          ; 2754
                                        40   AE      0008B                                                              ;
                        44    AE        24   A6   D0 0008D              MOVL    36(NAM), FIB+4                                  ; 2755
                        48    AE        28   A6   B0 00092              MOVW    40(NAM), FIB+8                                  ; 2757
                                             28   11 00097              BRB     4$                                              ; 2731
                              50             69   D0 00099 2$:          MOVL    INPUT_MTL, R0                                   ; 2765
                              05        31   A0   E9 0009C              BLBC    49(R0), 3$                                      ;
                                             7E   D4 000A0              CLRL    -(SP)                                           ; 2766
                              6B             01   FB 000A2              CALLS   #1, READY_DISK                                  ;
  0040    8F           00     6E             00   2C 000A5 3$:          MOVC5   #0, (SP), #0, #64, FIB                          ; 2767
                                        40   AE      000AC                                                              ;
                              50        08   AC   D0 000AE              MOVL    FILE_ID, R0                                     ; 2768
                        44    AE             60   D0 000B2              MOVL    (R0), FIB+4                                     ;
                        48    AE        04   A0   B0 000B6              MOVW    4(R0), FIB+8                                    ; 2770
                        5C    AE      F96C   C9   D0 000BB              MOVL    RWSV_IN_VBN, FIB+28                             ; 2771
                        38    AE        40   8F   9A 000C1 4$:          MOVZBL  #64, FIB_DESC                                   ; 2774
                        3C    AE        40   AE   9E 000C6              MOVAB   FIB, FIB_DESC+4                                 ; 2775
                        04    AE 00040020    8F   D0 000CB              MOVL    #262176, ATT_CONTROL0                           ; 2777
```

```
                 08   AE        18   AE   9E 000D3        MOVAB   RECATTR, ATT_CONTROL0+4              : 2779
                 0C   AE  00280002   8F   D0 000D8        MOVL    #2621442, ATT_CONTROL1               : 2780
                 10   AE             6E   9E 000E0        MOVAB   SEG_NUMBER, ATT_CONTROL1+4           : 2782
                                14   AE   D4 000E4        CLRL    ATTR_END                             : 2783
                                     7E   D4 000E7        CLRL    -(SP)                                : 2790
                                08   AE   9F 000E9        PUSHAB  ATT_CONTROL                          :
                                     7E   7C 000EC        CLRQ    -(SP)                                :
                                     7E   D4 000EE        CLRL    -(SP)                                :
                                4C   AE   9F 000F0        PUSHAB  FIB_DESC                             :
                                     7E   7C 000F3        CLRQ    -(SP)                                :
                                F8   AD   9F 000F5        PUSHAB  IO_STATUS                            :
                      7E        72   8F   9A 000F8        MOVZBL  #114, -(SP)                          :
                          0001FFFF   8F   DD 000FC        PUSHL   #131071                              :
                                     7E   D4 00102        CLRL    -(SP)                                :
          00000000G   00             0C   FB 00104        CALLS   #12, STA_QIOW                        :
                      58             50   D0 0010B        MOVL    R0, STATUS                           :
                      07             58   E9 0010E        BLBC    STATUS, 5$                           : 2791
                      58        F8   AD   3C 00111        MOVZWL  IO_STATUS, STATUS                    :
                      0C             58   E8 00115        BLBS    STATUS, 6$                           : 2792
                      7E             57   7D 00118 5$:    MOVQ    FAB, -(SP)                           :
                          00000000G  8F   DD 0011B        PUSHL   #BACKUP$_OPENIN+4                    :
                      6A             03   FB 00121        CALLS   #3, FILE_ERROR                       :
                      01        18   AE   91 00124 6$:    CMPB    RECATTR, #1                          : 2797
                                     15   12 00128        BNEQ    7$                                   :
                                19   AE   95 0012A        TSTB    RECATTR+1                            : 2798
                                     10   12 0012D        BNEQ    7$                                   :
               01FF   8F        1A   AE   B3 0012F        BITW    RECATTR+2, #511                      : 2799
                                     08   12 00135        BNEQ    7$                                   :
               0800   8F        1A   AE   B1 00137        CMPW    RECATTR+2, #2048                     : 2800
                                     0B   1E 0013D        BGEQU   8$                                   :
                                     57   DD 0013F 7$:    PUSHL   FAB                                  : 2801
                          00000000G  8F   DD 00141        PUSHL   #BACKUP$_NOTSAVESET                  :
                      6A             02   FB 00147        CALLS   #2, FILE_ERROR                       :
                      52             6E   3C 0014A 8$:    MOVZWL  SEG_NUMBER, R2                       : 2806
                      3F        04   AC   E8 0014D        BLBS    CONTINUE, 13$                        : 2811
                      1B             13 00151             BEQL    11$                                  : 2806
            0A      F99A   C9             03   E1 00153   BBC     #3, QUAL+10, 9$                      : 2808
                                     57   DD 00159        PUSHL   FAB                                  : 2809
                          00000000G  8F   DD 0015B        PUSHL   #BACKUP$_NOT1STVOL+4                 :
                                     08   11 00161        BRB     10$                                  :
                                     57   DD 00163 9$:    PUSHL   FAB                                  : 2810
                          00000000G  8F   DD 00165        PUSHL   #BACKUP$_NOT1STVOL                   :
                      6A             02   FB 0016B 10$:   CALLS   #2, FILE_ERROR                       :
               F9D8   C9        1A   AE   B0 0016E 11$:   MOVW    RECATTR+2, QUAL+72                   : 2811
   F978   C9     20   AE             10   9C 00174        ROTL    #16, RECATTR+8, RWSV_EOF             : 2812
                                24   AE   B5 0017B        TSTW    RECATTR+12                           : 2813
                                     04   12 0017E        BNEQ    12$                                  :
                              F978   C9   D7 00180        DECL    RWSV_EOF                             : 2814
               F930   C9        48   AE   9B 00184 12$:   MOVZBW  FIB+8, RWSV_VOL_NUMBER               : 2815
               F932   C9             52   B0 0018A        MOVW    R2, RWSV_SEG_NUMBER                  : 2816
                                          04 0018F        RET                                          : 2803
                      52      F932   C9   B1 00190 13$:   CMPW    RWSV_SEG_NUMBER, R2                  : 2821
                                     0B   13 00195        BEQL    14$                                  :
                                     57   DD 00197        PUSHL   FAB                                  : 2822
                          00000000G  8F   DD 00199        PUSHL   #BACKUP$_WRONGVOL                    :
                      6A             02   FB 0019F        CALLS   #2, FILE_ERROR                       :
               F9D8   C9        1A   AE   B1 001A2 14$:   CMPW    RECATTR+2, QUAL+72                   : 2823
```

```
                                0B  13 001A8         BEQL    15$                                  ;
                                57  DD 001AA         PUSHL   FAB                                  ; 2824
                   00000000G    8F  DD 001AC         PUSHL   #BACKUP$_BADBLKSIZE                  ;
              6A                02  FB 001B2         CALLS   #2, FILE_ERROR                       ;
                                    04 001B5 15$:    RET                                          ; 2827

; Routine Size:  438 bytes,    Routine Base:  CODE + 07F9
```

```
1 %SBTTL 'INIT_SAVE_TAPE - initialize save set tape'
1 ROUTINE INIT_SAVE_TAPE (VERIFY) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine sets up the input tape for reading.
1 !
1 ! CALLING SEQUENCE:
1 !       INIT_SAVE_TAPE (VERIFY)
1 !
1 ! INPUT PARAMETERS:
1 !       VERIFY: TRUE to indicate setup for verify pass
1 !               FALSE to indicate normal initialization
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
2 LOCAL
2       FAB             : REF BBLOCK,   ! pointer to input FAB
2       STATUS,                         ! the usual status value
2       FILE_SECTION,                   ! file section number
2       TAPE_CHAR       : BBLOCK [4],   ! magtape device characteristics
2       LABEL_BUFFER    : BBLOCK [90],  ! buffer for tape labels
2       IO_STATUS       : VECTOR [4,WORD]; ! I/O status block
2
2 EXTERNAL ROUTINE
2       FILE_ERROR,                     ! signal file related error
2       READY_TAPE,                     ! prepare tape for I/O
2       SENSE_CHAR,                     ! sense magtape characteristics
2       REWIND,                         ! rewind tape
2       SKIP_TM,                        ! skip tape marks
2       READ_LABEL,                     ! read and check tape label
2       LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL);
2                                       ! convert decimal to binary
2
2 ! Set up the input tape. Rewind it if called for; if we are setting up a
2 ! small save set written /NOREWIND for verify, backspace over it rather
2 ! than rewinding to save time.
2 !
```

```
2
2 FAB = .RWSV_SAVE_FAB;
2 TAPE_CHAR = READY_TAPE (FALSE);
2 IF .QUAL[QUAL_REWI] AND NOT .INPUT_FLAGS[INPUT_REWOUND]
2 THEN
3     BEGIN
3     REWIND ();
3     TAPE_CHAR[MT$V_BOT] = TRUE;
3     INPUT_FLAGS[INPUT_REWOUND] = TRUE;
3     END
3
2 ELSE IF .VERIFY
2 THEN
3     BEGIN
3     IF .RWSV_OUT_BLOCK_COUNT LSSU 1000
3     THEN
4         BEGIN
4         SKIP_TM (-3);
4         TAPE_CHAR = SENSE_CHAR ();
4         END
4
3     ELSE
4         BEGIN
4         REWIND ();
4         TAPE_CHAR[MT$V_BOT] = TRUE;
3         END;
2     END;
2
2 ! If the tape is at BOT, read and verify the volume header label.
2 !
2
2 IF .TAPE_CHAR[MT$V_BOT]
2 THEN
3     BEGIN
3     STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
3
3     WHILE TRUE
3     DO
4         BEGIN
4         STATUS = READ_LABEL (LABEL_BUFFER);
4         IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
4         IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'HDR1' THEN EXITLOOP;
3         END;
2     END;
2
2 ! Search for a HDR1 record that matches the desired save set name.
2 !
2
2 IF NOT .TAPE_CHAR[MT$V_BOT]
2 OR NOT MATCH_SSNAME (LABEL_BUFFER)
2 THEN
3     BEGIN
3     WHILE TRUE
3     DO
4         BEGIN
4         STATUS = SKIP_TM (1);
```

```
                IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);

                STATUS = READ_LABEL (LABEL_BUFFER, 'HDR1');
                IF .STATUS
                AND MATCH_SSNAME (LABEL_BUFFER)
                THEN EXITLOOP;

                IF .STATUS EQL SS$_ENDOFVOLUME
                THEN
                    BEGIN
                    SKIP_TM (-2);
                    IF .INPUT_FLAGS[INPUT_WILDSAVE]
                    THEN COM_FLAGS[COM_EOV] = TRUE
                    ELSE
                        BEGIN
                        COM_FLAGS[COM_EOV] = FALSE;
                        FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE,
                                    .FAB, SS$_NOSUCHFILE);
                        END;
                    RETURN;
                    END;
                END;
            END;

        INPUT_FLAGS[INPUT_SSFOUND] = TRUE;                  ! Note that save set was found

        ! Notify of a new save set if requested.
        !
        IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .QUAL[QUAL_LOG]
        THEN SIGNAL (BACKUP$_NEWSAVSET, 2,
                    HD1$S_FILEID, LABEL_BUFFER[HD1$T_FILEID]);

        ! Check other HDR1 fields.
        !

        CH$MOVE (HD1$S_FILESETID, LABEL_BUFFER[HD1$T_FILESETID], RWSV_FILESET_ID);
        IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], RWSV_FILE_NUMBER)
        THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
        IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESECNO], FILE_SECTION)
        THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
        IF .FILE_SECTION NEQ 1
        THEN
            BEGIN
            IF .QUAL[QUAL_IMAG]
            THEN FILE_ERROR ((BACKUP$_NOT1STVOL AND NOT $FIELDMASK (STS$V_SEVERITY)) + STS$K_SEVERE, .FAB)
            ELSE FILE_ERROR (BACKUP$_NOT1STVOL, .FAB);
            END;
        RWSV_VOL_NUMBER = .FILE_SECTION;

        ! Read and check HDR2.
        !

        STATUS = READ_LABEL (LABEL_BUFFER, 'HDR2');
        IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
        IF .LABEL_BUFFER[HD2$B_RECFORMAT] NEQ 'F'
        OR CH$NEQ (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN],
                   HD2$S_BLOCKLEN, LABEL_BUFFER[HD2$T_BLOCKLEN])
```

```
; 1452      2999    2 THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
; 1453      3000    2 IF NOT LIB$CVT_DTB (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN], QUAL[QUAL_BLOC_VALUE])
; 1454      3001    2 THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
; 1455      3002    2 IF .QUAL[QUAL_BLOC_VALUE] LSSU 2048
; 1456      3003    2 OR .QUAL[QUAL_BLOC_VALUE] GEQU 65536
; 1457      3004    2 THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
; 1458      3005    2
; 1459      3006    2 ! Skip to start of the data records.
; 1460      3007    2 !
; 1461      3008    2
; 1462      3009    2 SKIP_TM (1);
; 1463      3010    2
; 1464      3011    1 END;                                  ! End of routine INIT_SAVE_TAPE


                                                                .EXTRN  READY_TAPE, SENSE_CHAR
                                                                .EXTRN  REWIND, READ_LABEL
                                                                .EXTRN  LIB$CVT_DTB

                                            0FFC 00000 INIT_SAVE_TAPE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 2829
                        5B 00000000G    00   9E 00002           MOVAB   SKIP_TM, R11                                    ;
                        5A 00000000G    8F   D0 00009           MOVL    #BACKUP$_LABELERR, R10                          ;
                        59 00000000G    00   9E 00010           MOVAB   FILE_ERROR, R9                                  ;
                        58 00000000'    EF   9E 00017           MOVAB   INPUT_FLAGS, R8                                 ;
                        5E        98    AE   9E 0001E           MOVAB   -104(SP), SP                                    ;
                        56      FEE4    C8   D0 00022           MOVL    RWSV_SAVE_FAB, FAB                              ; 2886
                                        7E   D4 00027           CLRL    -(SP)                                           ; 2887
              00000000G 00              01   FB 00029           CALLS   #1, READY_TAPE                                  ;
                        52              50   D0 00030           MOVL    R0, TAPE_CHAR                                   ;
                        16      FF45    C8   E9 00033           BLBC    QUAL+13, 1$                                     ; 2888
                                        68   95 00038           TSTB    INPUT_FLAGS                                     ;
                                        12   19 0003A           BLSS    1$                                              ;
              00000000G 00              00   FB 0003C           CALLS   #0, REWIND                                      ; 2891
   52           01      10              01   F0 00043           INSV    #1, #16, #1, TAPE_CHAR                          ; 2892
                        68        80    8F   88 00048           BISB2   #128, INPUT_FLAGS                               ; 2893
                                        2D   11 0004C           BRB     3$                                              ; 2888
                        29        04    AC   E9 0004E 1$:       BLBC    VERIFY, 3$                                      ; 2896
              000003E8  8F      FF2C    C8   D1 00052           CMPL    RWSV_OUT_BLOCK_COUNT, #1000                     ; 2899
                                        12   1E 0005B           BGEQU   2$                                              ;
                        7E              03   CE 0005D           MNEGL   #3, -(SP)                                       ; 2902
                        6B              01   FB 00060           CALLS   #1, SKIP_TM                                     ;
              00000000G 00              00   FB 00063           CALLS   #0, SENSE_CHAR                                  ; 2903
                        52              50   D0 0006A           MOVL    R0, TAPE_CHAR                                   ;
                                        0C   11 0006D           BRB     3$                                              ; 2899
              00000000G 00              00   FB 0006F 2$:       CALLS   #0, REWIND                                      ; 2908
   52           01      10              01   F0 00076           INSV    #1, #16, #1, TAPE_CHAR                          ; 2909
                4F      52              10   E1 0007B 3$:       BBC     #16, TAPE_CHAR, 6$                              ; 2916
                          314C4F56      8F   DD 0007F           PUSHL   #827084630                                      ; 2919
                                  10    AE   9F 00085           PUSHAB  LABEL_BUFFER                                    ;
              00000000G 00              02   FB 00088           CALLS   #2, READ_LABEL                                  ;
                        57              50   D0 0008F           MOVL    R0, STATUS                                      ;
                        08              57   E8 00092           BLBS    STATUS, 4$                                      ; 2920
                        7E              56   7D 00095           MOVQ    FAB, -(SP)                                      ;
                                        5A   DD 00098           PUSHL   R10                                             ;
                        69              03   FB 0009A           CALLS   #3, FILE_ERROR                                  ;
```

```
                                     0C  AE  9F 0009D 4$:    PUSHAB  LABEL_BUFFER                                    ; 2925
                   00000000G  00         01  FB 000A0        CALLS   #1, READ_LABEL                                  ;
                              57         50  D0 000A7        MOVL    R0, STATUS                                      ;
                              08         57  E8 000AA        BLBS    STATUS, 5$                                      ; 2926
                              7E         56  7D 000AD        MOVQ    FAB, -(SP)                                      ;
                                         5A  DD 000B0        PUSHL   R10                                             ;
                              69         03  FB 000B2        CALLS   #3, FILE_ERROR                                  ;
                   31524448   8F     0C  AE  D1 000B5 5$:    CMPL    LABEL_BUFFER, #827475016                        ; 2927
                                         DE  12 000BD        BNEQ    4$                                              ;
             0B               52         10  E1 000BF        BBC     #16, TAPE_CHAR, 6$                              ; 2934
                                     0C  AE  9F 000C3        PUSHAB  LABEL_BUFFER                                    ; 2935
                       FD0E   CF         01  FB 000C6        CALLS   #1, MATCH_SSNAME                                ;
                              61         50  E8 000CB        BLBS    R0, 10$                                         ;
                                         01  DD 000CE 6$:    PUSHL   #1                                              ; 2941
                              6B         01  FB 000D0        CALLS   #1, SKIP_TM                                     ;
                              57         50  D0 000D3        MOVL    R0, STATUS                                      ;
                              08         57  E8 000D6        BLBS    STATUS, 7$                                      ; 2942
                              7E         56  7D 000D9        MOVQ    FAB, -(SP)                                      ;
                                         5A  DD 000DC        PUSHL   R10                                             ;
                              69         03  FB 000DE        CALLS   #3, FILE_ERROR                                  ;
                                31524448 8F  DD 000E1 7$:    PUSHL   #827475016                                      ; 2944
                                     10  AE  9F 000E7        PUSHAB  LABEL_BUFFER                                    ;
                   00000000G  00         02  FB 000EA        CALLS   #2, READ_LABEL                                  ;
                              57         50  D0 000F1        MOVL    R0, STATUS                                      ;
                              0B         57  E9 000F4        BLBC    STATUS, 8$                                      ; 2945
                                     0C  AE  9F 000F7        PUSHAB  LABEL_BUFFER                                    ; 2946
                       FCDA   CF         01  FB 000FA        CALLS   #1, MATCH_SSNAME                                ;
                              2D         50  E8 000FF        BLBS    R0, 10$                                         ;
                   000009A0   8F         57  D1 00102 8$:    CMPL    STATUS, #2464                                   ; 2949
                                         C3  12 00109        BNEQ    6$                                              ;
                              7E         02  CE 0010B        MNEGL   #2, -(SP)                                       ; 2952
                              6B         01  FB 0010E        CALLS   #1, SKIP_TM                                     ;
             05               68         06  E1 00111        BBC     #6, INPUT_FLAGS, 9$                             ; 2953
                         B2   A8         01  88 00115        BISB2   #1, COM_FLAGS                                   ; 2954
                                         04  00119           RET                                                     ;
                         B2   A8         01  8A 0011A 9$:    BICB2   #1, COM_FLAGS                                   ; 2957
                              7E    0910 8F  3C 0011E        MOVZWL  #2320, -(SP)                                    ; 2958
                                         56  DD 00123        PUSHL   FAB                                             ; 2959
                               00000000G 8F  DD 00125        PUSHL   #BACKUP$_OPENIN+4                               ; 2958
                              69         03  FB 0012B        CALLS   #3, FILE_ERROR                                  ;
                                         04  0012E           RET                                                     ; 2951
                         01   A8         01  88 0012F 10$:   BISB2   #1, INPUT_FLAGS+1                               ; 2966
             1A               68         06  E1 00133        BBC     #6, INPUT_FLAGS, 11$                            ; 2970
             14        FF43   C8         01  E1 00137        BBC     #1, QUAL+11, 11$                                ;
                                     10  AE  9F 0013D        PUSHAB  LABEL_BUFFER+4                                  ; 2972
                                         11  DD 00140        PUSHL   #17                                             ;
                                         02  DD 00142        PUSHL   #2                                              ;
                               00000000G 8F  DD 00144        PUSHL   #BACKUP$_NEWSAVSET                              ;
                   00000000G  00         04  FB 0014A        CALLS   #4, LIB$SIGNAL                                  ;
       FEC4     C8        21  AE         06  28 00151 11$:   MOVC3   #6, LABEL_BUFFER+21, RWSV_FILESET_ID            ; 2977
                                   FEDC  C8  9F 00158        PUSHAB  RWSV_FILE_NUMBER                                ; 2978
                                     2F  AE  9F 0015C        PUSHAB  LABEL_BUFFER+31                                 ;
                                         04  DD 0015F        PUSHL   #4                                              ;
                   00000000G  00         03  FB 00161        CALLS   #3, LIB$CVT_DTB                                 ;
                              0D         50  E8 00168        BLBS    R0, 12$                                         ;
                               00000000G 8F  DD 0016B        PUSHL   #BACKUP$_NOTANSI                                ; 2979
                                         56  DD 00171        PUSHL   FAB                                             ;
```

```
                                     5A  DD 00173         PUSHL   R10                                       ;
                       69            03  FB 00175         CALLS   #3, FILE_ERROR                            ;
                                     5E  DD 00178 12$:    PUSHL   SP                                        ; 2980
                              2B     AE  9F 0017A         PUSHAB  LABEL_BUFFER+27                           ;
                                     04  DD 0017D         PUSHL   #4                                        ;
              00000000G       00     03  FB 0017F         CALLS   #3, LIB$CVT_DTB                           ;
                              0D     50  E8 00186         BLBS    R0, 13$                                   ;
                       00000000G     8F  DD 00189         PUSHL   #BACKUP$_NOTANSI                          ; 2981
                                     56  DD 0018F         PUSHL   FAB                                       ;
                                     5A  DD 00191         PUSHL   R10                                       ;
                       69            03  FB 00193         CALLS   #3, FILE_ERROR                            ;
                       01            6E  D1 00196 13$:    CMPL    FILE_SECTION, #1                          ; 2982
                                     1B  13 00199         BEQL    16$                                       ;
          0A      FF42 C8            03  E1 0019B         BBC     #3, QUAL+10, 14$                          ; 2985
                                     56  DD 001A1         PUSHL   FAB                                       ; 2986
                       00000000*     8F  DD 001A3         PUSHL   #<<BACKUP$_NOT1STVOL&-8>+4>               ;
                                     08  11 001A9         BRB     15$                                       ;
                                     56  DD 001AB 14$:    PUSHL   FAB                                       ; 2987
                       00000000G     8F  DD 001AD         PUSHL   #BACKUP$_NOT1STVOL                        ;
                       69            02  FB 001B3 15$:    CALLS   #2, FILE_ERROR                            ;
                  FED8 C8            6E  B0 001B6 16$:    MOVW    FILE_SECTION, RWSV_VOL_NUMBER             ; 2989
                       32524448      8F  DD 001BB         PUSHL   #844252232                                ; 2994
                              10     AE  9F 001C1         PUSHAB  LABEL_BUFFER                              ;
              00000000G       00     02  FB 001C4         CALLS   #2, READ_LABEL                            ;
                              57     50  D0 001CB         MOVL    R0, STATUS                                ;
                              08     57  E8 001CE         BLBS    STATUS, 17$                               ; 2995
                              7E     56  7D 001D1         MOVQ    FAB, -(SP)                                ;
                                     5A  DD 001D4         PUSHL   R10                                       ;
                       69            03  FB 001D6         CALLS   #3, FILE_ERROR                            ;
                    46 8F      10    AE  91 001D9 17$:    CMPB    LABEL_BUFFER+4, #70                       ; 2996
                                     08  12 001DE         BNEQ    18$                                       ;
       11 AE        16 AE            05  29 001E0         CMPC3   #5, LABEL_BUFFER+10, LABEL_BUFFER+5       ; 2998
                                     0B  13 001E6         BEQL    19$                                       ;
                                     56  DD 001E8 18$:    PUSHL   FAB                                       ; 2999
                       00000000G     8F  DD 001EA         PUSHL   #BACKUP$_NOTSAVESET                       ;
                       69            02  FB 001F0         CALLS   #2, FILE_ERROR                            ;
                              80     A8  9F 001F3 19$:    PUSHAB  QUAL+72                                   ; 3000
                              1A     AE  9F 001F6         PUSHAB  LABEL_BUFFER+10                           ;
                                     05  DD 001F9         PUSHL   #5                                        ;
              00000000G       00     03  FB 001FB         CALLS   #3, LIB$CVT_DTB                           ;
                              0D     50  E8 00202         BLBS    R0, 20$                                   ;
                       00000000G     8F  DD 00205         PUSHL   #BACKUP$_NOTANSI                          ; 3001
                                     56  DD 0020B         PUSHL   FAB                                       ;
                                     5A  DD 0020D         PUSHL   R10                                       ;
                       69            03  FB 0020F         CALLS   #3, FILE_ERROR                            ;
                  0800 8F      80    A8  B1 00212 20$:    CMPW    QUAL+72, #2048                            ; 3003
                                     0B  1E 00218         BGEQU   21$                                       ;
                                     56  DD 0021A         PUSHL   FAB                                       ; 3004
                       00000000G     8F  DD 0021C         PUSHL   #BACKUP$_NOTSAVESET                       ;
                       69            02  FB 00222         CALLS   #2, FILE_ERROR                            ;
                                     01  DD 00225 21$:    PUSHL   #1                                        ; 3009
                       6B            01  FB 00227         CALLS   #1, SKIP_TM                               ;
                                         04 0022A         RET                                               ; 3011

; Routine Size:  555 bytes,    Routine Base:  CODE + 09AF
```

```
%SBTTL 'INIT_IN_SAVE - initialize save set for reading'
GLOBAL ROUTINE INIT_IN_SAVE (VERIFY) : NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine initializes the input save set.
!
! CALLING SEQUENCE:
!       INIT_IN_SAVE (VERIFY)
!
! INPUT PARAMETERS:
!       VERIFY: TRUE to indicate setup for verify pass
!               FALSE to indicate normal initialization
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN

LOCAL
        FAB             : REF BBLOCK,   ! pointer to input FAB
        RAB             : REF BBLOCK,   ! pointer to input RAB
        STATUS,                         ! the usual status value
        DEVICE_DESC     : VECTOR[2];    ! Input device desc if mounted /FOREIGN

EXTERNAL ROUTINE
        FILE_ERROR,                     ! signal file related error
        INIT_BUFFERS,                   ! initialize the buffer pool
        EXTRACT_FILENAME;               ! extract file name, type, and version

RWSV_IN_ERRORS = 0;
RWSV_IN_XORUSE = 0;
RWSV_IN_SEQ = 1;
RWSV_IN_SEQ_0 = 1;
RWSV_IN_XOR_RFA = 1;
RWSV_IN_VBN = 1;
RWSV_IN_VBN_0 = 1;

IF NOT .VERIFY
THEN
    BEGIN
```

```
    RWSV_VOL_NUMBER = 1;
    RWSV_SEG_NUMBER = 0;

! Do a UFO open on the input FAB and save away the channel.
!

    RWSV_SAVE_QUAL = .QUAL[QUAL_INPU_LIST];
    RWSV_SAVE_FAB = FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
    FAB[FAB$V_NAM] = TRUE;
    IF .QUAL[QUAL_SS_FILE]
    THEN
        BEGIN
        IF .BBLOCK[FAB[FAB$L_DEV], DEV$V_NET]
        THEN
            FAB[FAB$V_BIO] = FAB[FAB$V_SQO] = TRUE;

        FAB[FAB$V_GET] = TRUE;
        END
    ELSE
        FAB[FAB$V_UFO] = TRUE;

    IF .QUAL[QUAL_SS_FILE]
    OR .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
    THEN
        BEGIN

! If the device is not mounted foreign do a $OPEN; otherwise do a $ASSIGN
!

        IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
        THEN STATUS = $OPEN (FAB = .FAB)
        ELSE
            BEGIN
            DEVICE_DESC[0] = .BBLOCK[.FAB[FAB$L_NAM], NAM$B_DEV];
            DEVICE_DESC[1] = .BBLOCK[.FAB[FAB$L_NAM], NAM$L_DEV];
            STATUS = FAB[FAB$L_STS] = $ASSIGN (DEVNAM = DEVICE_DESC,
                                               CHAN = FAB[FAB$L_STV]);
            END;
        IF NOT .STATUS
        THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
                          .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
        END
    ELSE
        BEGIN
        INIT_SAVE_DISK (0);
        END;

    IF .QUAL[QUAL_SS_FILE]
    THEN
        BEGIN
        RAB = FAB[FC_RAB];
        STATUS = $CONNECT (RAB = .RAB);
        IF NOT .STATUS
        THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
                         .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
        IF .FAB[FAB$B_RFM] NEQ FAB$C_FIX
        OR .FAB[FAB$B_ORG] NEQ FAB$C_SEQ
```

```
            OR .BBLOCK[FAB[FAB$L_DEV], DEV$V_NET] AND .(FAB[FAB$W_MRS])<0,9> NEQ 0
            THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
            QUAL[QUAL_BLOC_VALUE] = .FAB[FAB$W_MRS];
            CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
            END
        ELSE
            RWSV_CHAN = .FAB[FAB$L_STV];

        EXTRACT_FILENAME (.FAB, COM_SSNAME);
        END

    ! Otherwise do setup for input verification.
    !

    ELSE
        BEGIN
        FAB = .RWSV_SAVE_FAB;
        RAB = FAB[FC_RAB];
        IF .QUAL[QUAL_SS_FILE]
        THEN
            BEGIN
            STATUS = $REWIND (RAB = .RAB);
            IF NOT .STATUS
            THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
                             .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
            CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
            END;
        END;
    ALT_SSNAME[0] = 0;

    ! Set up the input tape.
    !

    IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
    AND NOT .QUAL[QUAL_SS_FILE]
    THEN
        BEGIN
        INIT_SAVE_TAPE (.VERIFY);
        IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .COM_FLAGS[COM_EOV] THEN RETURN;
        END;

    ! Set up the buffer pool.
    !

    IF NOT .VERIFY
    THEN
        BEGIN
        INIT_BUFFERS (.QUAL[QUAL_BUFF_VALUE], .QUAL[QUAL_BLOC_VALUE]);
        RWSV_IN_GROUP_SIZE = 1^31-1;
        END;

    END;                                    ! End of routine INIT_IN_SAVE


                                                    .EXTRN  INIT_BUFFERS, EXTRACT_FILENAME
                                                    .EXTRN  SYS$OPEN, SYS$ASSIGN
```

```
                                                    .EXTRN  SYS$REWIND

                                 03FC 00000         .ENTRY  INIT_IN_SAVE, Save R2,R3,R4,R5,R6,R7,R8,R9      ; 3013
              59 00000000G  8F    D0 00002          MOVL    #BACKUP$_OPENIN+4, R9                          ;
              58 00000000G  00    9E 00009          MOVAB   FILE_ERROR, R8                                 ;
              57 00000000'  EF    9E 00010          MOVAB   RWSV_IN_XOR_RFA, R7                            ;
              5E            08    C2 00017          SUBL2   #8, SP                                         ;
                        0C  A7    D4 0001A          CLRL    RWSV_IN_ERRORS                                 ; 3058
           F4  A7           01    D0 0001D          MOVL    #1, RWSV_IN_SEQ                                ; 3060
           F8  A7           01    D0 00021          MOVL    #1, RWSV_IN_SEQ_0                              ; 3061
               67           01    D0 00025          MOVL    #1, RWSV_IN_XOR_RFA                            ; 3062
           18  A7           01    D0 00028          MOVL    #1, RWSV_IN_VBN                                ; 3063
           1C  A7           01    D0 0002C          MOVL    #1, RWSV_IN_VBN_0                              ; 3064
51      4B  A7              01    03 EF 00030       EXTZV   #3, #1, QUAL[+15, R1                           ; 3078
               03       04  AC    E9 00036          BLBC    VERIFY, 1$                                     ; 3066
                        00EC      31 0003A          BRW     15$                                            ;
           DC  A7           01    D0 0003D 1$:      MOVL    #1, RWSV_VOL_NUMBER                            ; 3069
           E4  A7       3C  A7    D0 00041          MOVL    QUAL, RWSV_SAVE_QUAL                           ; 3075
               50       E4  A7    D0 00046          MOVL    RWSV_SAVE_QUAL, R0                             ; 3076
               56       04  A0    D0 0004A          MOVL    4(R0), FAB                                     ;
           E8  A7           56    D0 0004E          MOVL    FAB, RWSV_SAVE_FAB                             ;
           07  A6           01    88 00052          BISB2   #1, 7(FAB)                                     ; 3077
               14           51    E9 00056          BLBC    R1, 3$                                         ; 3078
      09   41  A6           05    E1 00059          BBC     #5, 65(FAB), 2$                                ; 3081
           04  A6       40  8F    88 0005E          BISB2   #64, 4(FAB)                                    ; 3083
           16  A6           20    88 00063          BISB2   #32, 22(FAB)                                   ;
           16  A6           02    88 00067 2$:      BISB2   #2, 22(FAB)                                    ; 3085
                            04    11 0006B          BRB     4$                                             ; 3078
           06  A6           02    88 0006D 3$:      BISB2   #2, 6(FAB)                                     ; 3088
               05           51    E8 00071 4$:      BLBS    R1, 5$                                         ; 3090
      42   40  A6           05    E1 00074          BBC     #5, 64(FAB), 8$                                ; 3091
               0B       43  A6    E8 00079 5$:      BLBS    67(FAB), 6$                                    ; 3098
                            56    DD 0007D          PUSHL   FAB                                            ; 3099
      00000000G  00         01    FB 0007F          CALLS   #1, SYS$OPEN                                   ;
                            20    11 00086          BRB     7$                                             ;
               50       28  A6    D0 00088 6$:      MOVL    40(FAB), R0                                    ; 3102
               6E       39  A0    9A 0008C          MOVZBL  57(R0), DEVICE_DESC                            ;
           04  AE       44  A0    D0 00090          MOVL    68(R0), DEVICE_DESC+4                          ; 3103
                            7E    7C 00095          CLRQ    -(SP)                                          ; 3105
                        0C  A6    9F 00097          PUSHAB  12(FAB)                                        ;
                        0C  AE    9F 0009A          PUSHAB  DEVICE_DESC                                    ;
      00000000G  00         04    FB 0009D          CALLS   #4, SYS$ASSIGN                                 ;
           08  A6           50    D0 000A4          MOVL    R0, 8(FAB)                                     ;
               53           50    D0 000A8 7$:      MOVL    R0, STATUS                                     ; 3104
               14           53    E8 000AB          BLBS    STATUS, 9$                                     ; 3107
               7E       08  A6    7D 000AE          MOVQ    8(FAB), -(SP)                                  ; 3109
                            56    DD 000B2          PUSHL   FAB                                            ; 3108
                            59    DD 000B4          PUSHL   R9                                             ;
               68           04    FB 000B6          CALLS   #4, FILE_ERROR                                 ;
                            07    11 000B9          BRB     9$                                             ; 3090
                            7E    D4 000BB 8$:      CLRL    -(SP)                                          ; 3113
         FB5D  CF           01    FB 000BD          CALLS   #1, INIT_SAVE_DISK                             ;
      4E   4B  A7           03    E1 000C2 9$:      BBC     #3, QUAL+15, 13$                               ; 3116
               52       50  A6    9E 000C7          MOVAB   80(R6), RAB                                    ; 3119
                            52    DD 000CB          PUSHL   RAB                                            ; 3120
      00000000G  00         01    FB 000CD          CALLS   #1, SYS$CONNECT                                ;
               53           50    D0 000D4          MOVL    R0, STATUS                                     ;
```

```
                         0B                 53  E8 000D7          BLBS    STATUS, 10$                                   ; 3121
                         7E          08     A2  7D 000DA          MOVQ    8(RAB), -(SP)                                 ; 3123
                                            56  DD 000DE          PUSHL   FAB                                           ; 3122
                                            59  DD 000E0          PUSHL   R9                                            ;
                         68                 04  FB 000E2          CALLS   #4, FILE_ERROR                                ;
                         01          1F     A6  91 000E5 10$:     CMPB    31(FAB), #1                                   ; 3124
                                            12  12 000E9          BNEQ    11$                                           ;
                                     1D     A6  95 000EB          TSTB    29(FAB)                                       ; 3125
                                            0D  12 000EE          BNEQ    11$                                           ;
        13          41   A6                 05  E1 000F0          BBC     #5, 65(FAB), 12$                              ; 3126
                  01FF   8F          36     A6  B3 000F5          BITW    54(FAB), #511                                 ;
                                            0B  13 000FB          BEQL    12$                                           ;
                                            56  DD 000FD 11$:     PUSHL   FAB                                           ; 3127
                              00000000G     8F  DD 000FF          PUSHL   #BACKUP$_NOTSAVESET                           ;
                         68                 02  FB 00105          CALLS   #2, FILE_ERROR                                ;
                  0084   C7          36     A6  B0 00108 12$:     MOVW    54(FAB), QUAL+72                              ; 3128
        67          10   A2                 06  28 0010E          MOVC3   #6, 16(RAB), RWSV_IN_XOR_RFA                  ; 3129
                                            05  11 00113          BRB     14$                                           ; 3116
                    EC   A7          0C     A6  D0 00115 13$:     MOVL    12(FAB), RWSV_CHAN                            ; 3132
                                   00AC     C7  9F 0011A 14$:     PUSHAB  COM_SSNAME                                    ; 3134
                                            56  DD 0011E          PUSHL   FAB                                           ;
             00000000G   00                 02  FB 00120          CALLS   #2, EXTRACT_FILENAME                          ;
                                            2A  11 00127          BRB     17$                                           ; 3066
                         56          E8     A7  D0 00129 15$:     MOVL    RWSV_SAVE_FAB, FAB                            ; 3142
                         52          50     A6  9E 0012D          MOVAB   80(R6), RAB                                   ; 3143
                         1F                 51  E9 00131          BLBC    R1, 17$                                       ; 3144
                                            52  DD 00134          PUSHL   RAB                                           ; 3147
             00000000G   00                 01  FB 00136          CALLS   #1, SYS$REWIND                                ;
                         53                 50  D0 0013D          MOVL    R0, STATUS                                    ;
                         0B                 53  E8 00140          BLBS    STATUS, 16$                                   ; 3148
                         7E          08     A2  7D 00143          MOVQ    8(RAB), -(SP)                                 ; 3150
                                            56  DD 00147          PUSHL   FAB                                           ; 3149
                                            59  DD 00149          PUSHL   R9                                            ;
                         68                 04  FB 0014B          CALLS   #4, FILE_ERROR                                ;
        67          10   A2                 06  28 0014E 16$:     MOVC3   #6, 16(RAB), RWSV_IN_XOR_RFA                  ; 3151
                                   00DC     C7  94 00153 17$:     CLRB    ALT_SSNAME                                    ; 3154
        18          40   A6                 05  E1 00157          BBC     #5, 64(FAB), 18$                              ; 3159
        13          4B   A7                 03  E0 0015C          BBS     #3, QUAL+15, 18$                              ; 3160
                                     04     AC  DD 00161          PUSHL   VERIFY                                        ; 3163
                  FC6C   CF                 01  FB 00164          CALLS   #1, INIT_SAVE_TAPE                            ;
        05        0104   C7                 06  E1 00169          BBC     #6, INPUT_FLAGS, 18$                          ; 3164
                         1D        00B6     C7  E8 0016F          BLBS    COM_FLAGS, 19$                                ;
                         19          04     AC  E8 00174 18$:     BLBS    VERIFY, 19$                                   ; 3170
                         7E        0084     C7  3C 00178          MOVZWL  QUAL+72, -(SP)                                ; 3173
                         7E        0088     C7  9A 0017D          MOVZBL  QUAL+76, -(SP)                                ;
             00000000G   00                 02  FB 00182          CALLS   #2, INIT_BUFFERS                              ;
                    08   A7  7FFFFFFF       8F  D0 00189          MOVL    #2147483647, RWSV_IN_GROUP_SIZE               ; 3174
                                                04 00191 19$:     RET                                                   ; 3177

; Routine Size:  402 bytes,    Routine Base:  CODE + 0BDA
```

```
1 %SBTTL 'READ_BUFFER - read a save set buffer'
1 GLOBAL ROUTINE READ_BUFFER =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads a corrected buffer from the input save
1 !       set, applying XOR recovery if necessary.
1 !
1 ! CALLING SEQUENCE:
1 !       READ_BUFFER ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       BCB address of buffer read
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       K,                                      ! count of blocks missed
2       RAB             : REF BBLOCK,           ! RAB to read input file
2       BCB             : REF BBLOCK,           ! BCB of block read
2       BUFFER          : REF BBLOCK;           ! data buffer read
2
2 EXTERNAL ROUTINE
2       FILE_ERROR,                             ! signal file related error
2       FREE_BUFFER;                            ! release buffer to free list
2
2 ! Loop, reading buffers until we get an acceptable data block. Others
2 ! are treated appropriately.
2 !
2
2 IF .RWSV_XORSIZE NEQ 0 THEN RWSV_IN_GROUP_SIZE = .RWSV_XORSIZE;
2 WHILE TRUE
2 DO
3     BEGIN
3     RWSV_IN_ORGERR[0] = TRUE;
3     RWSV_IN_ORGERR[1] = 0;
3     BCB = READ_SEQ_BLOCK (TRUE);
3     IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
```

```
3           THEN
4               BEGIN
4               BCB = NEXT_VOLUME (.BCB);
4               IF .BCB EQL 0
4               THEN RETURN 0;
3               END;
3
3           IF .(RWSV_IN_XOR_RFA)<0,32> EQL 0
3           AND .(RWSV_IN_XOR_RFA+4)<0,16> EQL 0
3           THEN
4               BEGIN
4               RAB = RWSV_SAVE_FAB[FC_RAB];
4               CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
3               END;
3           BUFFER = .BCB[BCB_BUFFER];
3
3       ! Check if a block has been skipped. If exactly one has, then apply XOR
3       ! recovery to get it back. If recovery fails, report the error unless the
3       ! missed block was an XOR block. We do not count or report block skip
3       ! errors if we're trying to read the first save set block, and there was
3       ! no I/O error; this is the condition of being handed a continuation
3       ! volume to start with.
3       !
3
3           IF .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ EQL 1
4           AND (.RWSV_IN_SEQ NEQ .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE + 1
4               OR .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_XORBLOCK)
4           AND (.RWSV_IN_SEQ NEQ 1 OR NOT .RWSV_IN_ORGERR[0])
3           THEN
4               BEGIN
4               BCB = XORCIZE (.RWSV_IN_SEQ, .BCB);
4               BUFFER = .BCB[BCB_BUFFER];
3               END;
3
3           K = .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ;
3           IF .K NEQ 0
3           THEN
4               BEGIN
5               IF (.RWSV_IN_SEQ NEQ 1 OR NOT .RWSV_IN_ORGERR[0])
4               THEN
5                   BEGIN
6                   IF (.RWSV_IN_SEQ NEQ .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE
6                       OR .K NEQ 1)
5                   THEN
6                       BEGIN
6                       FILE_ERROR (BACKUP$_BLOCKLOST, .RWSV_SAVE_FAB,
6                                   .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
6                       RWSV_IN_ERRORS = .RWSV_IN_ERRORS - 1;
5                       END;
4                   END
4               ELSE
4                   RWSV_IN_ERRORS = .RWSV_IN_ERRORS - 1;
4               IF .RWSV_IN_GROUP_SIZE NEQ 0
4               THEN
5                   BEGIN
5                   IF .BUFFER[BBH$L_NUMBER] GEQU .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE
5                   THEN RWSV_IN_XOR_SEQ = .RWSV_IN_XOR_SEQ
```

```
6                                   + (.BUFFER[BBH$L_NUMBER] - .RWSV_IN_XOR_SEQ)
5                                   / .RWSV_IN_GROUP_SIZE * .RWSV_IN_GROUP_SIZE;
4                   END;
3           END;
3
3   ! Do basic validation checks to make sure we can understand this block.
3   ! Note that this is done even if the block reads with an error, since
3   ! the header data is protected by the header CRC.
3   !
3
3       IF .BUFFER[BBH$W_SIZE] LSSU BBH$K_LENGTH
3       THEN SIGNAL (BACKUP$_INVBLKHDR)
3
3   ! Process the data block. If there has been a read error, attempt to
3   ! recover it. If the error persists, report it.
3   !
3
3       ELSE
4           BEGIN
4           IF .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_DATABLOCK
4           THEN
5               BEGIN
5               IF .BUFFER[BBH$W_STRUCLEV] NEQ BBH$K_LEVEL1
5               THEN SIGNAL (BACKUP$_INVSTRUCT)
5               ELSE
6                   BEGIN
6                   IF NOT .BCB[BCB_STATUS]
6                   THEN
7                       BEGIN
7                       BCB = XORCIZE (.RWSV_IN_SEQ, .BCB);
7                       BUFFER = .BCB[BCB_BUFFER];
6                       END;
6
6                   IF NOT .BCB[BCB_STATUS]
6                   THEN
7                       BEGIN
7                       FILE_ERROR (BACKUP$_READERR+STS$K_ERROR,
7                                   .RWSV_SAVE_FAB, .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
7                       RWSV_IN_ERRORS = .RWSV_IN_ERRORS = 1;
6                       END;
6
6   ! Check that the save set name in the block header matches the labels.
6   ! A mismatch indicates running off the end of an incomplete tape.
6   ! If we are instructed to continue, use the new save set name.
6   !
6
6                   IF CH$NEQ (.ALT_SSNAME[0]+1, ALT_SSNAME,
6                           .(BUFFER[BBH$T_SSNAME])<0,8>+1, BUFFER[BBH$T_SSNAME], 255)
6                   THEN
7                       BEGIN
7                       IF .ALT_SSNAME[0] NEQ 0
7                       THEN FILE_ERROR (BACKUP$_SSCHANGE, .RWSV_SAVE_FAB);
7                       CH$COPY (.(BUFFER[BBH$T_SSNAME])<0,8>+1, BUFFER[BBH$T_SSNAME],
7                               0, BBH$S_SSNAME, ALT_SSNAME);
6                       END;
6
6                   IF .BUFFER[BBH$L_BLOCKSIZE] NEQ .BCB[BCB_SIZE]
```

```
; 1804      3349  6                 THEN SIGNAL (BACKUP$_INVBLKSIZE);
; 1805      3350  6                 RWSV_IN_SEQ = .BUFFER[BBH$L_NUMBER] + 1;
; 1806      3351  6                 EXITLOOP;
; 1807      3352  5                 END;
; 1808      3353  4             END;
; 1809      3354  4
; 1810      3355  4 ! If the buffer contains an XOR block, note its sequence number.
; 1811      3356  4 !
; 1812      3357  4
; 1813      3358  4         IF .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_XORBLOCK
; 1814      3359  4         THEN
; 1815      3360  5             BEGIN
; 1816      3361  5             RWSV_IN_XOR_SEQ = .BUFFER[BBH$L_NUMBER];
; 1817      3362  5             RWSV_IN_XOR_RFA = .BCB[BCB_BLOCKNUM];
; 1818      3363  5             IF .QUAL[QUAL_SS_FILE]
; 1819      3364  5             THEN
; 1820      3365  6                 BEGIN
; 1821      3366  6                 RAB = RWSV_SAVE_FAB[FC_RAB];
; 1822      3367  6                 CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
; 1823      3368  5                 END;
; 1824      3369  4             END;
; 1825      3370  4
; 1826      3371  3         END;                              ! end of major buffer valid condition
; 1827      3372  3
; 1828      3373  3     RWSV_IN_SEQ = .BUFFER[BBH$L_NUMBER] + 1;
; 1829      3374  3     FREE_BUFFER (.BCB);
; 1830      3375  2     END;
; 1831      3376  2
; 1832      3377  2 .BCB
; 1833      3378  1 END;                                      ! End of routine READ_BUFFER



                                    0FFC 00000              .ENTRY  READ_BUFFER, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 3179
                                                                    R10,R11
                  5B 00000000G   00  9E 00002               MOVAB   FILE_ERROR, R11                               ;
                  5A 00000000'   EF  9E 00009               MOVAB   RWSV_IN_XOR_SEQ, R10                          ;
                  50       0B    AA  9A 00010               MOVZBL  RWSV_XORSIZE, R0                              ; 3227
                                 04  13 00014               BEQL    1$                                            ;
         0C       AA             50  D0 00016               MOVL    R0, RWSV_IN_GROUP_SIZE                        ;
         14       AA             01  7D 0001A  1$:          MOVQ    #1, RWSV_IN_ORGERR                            ; 3231
                                 01  DD 0001E               PUSHL   #1                                            ; 3233
       F555       CF             01  FB 00020               CALLS   #1, READ_SEQ_BLOCK                            ;
                  57             50  D0 00025               MOVL    R0, BCB                                       ;
       0870       8F       18    A7  B1 00028               CMPW    24(BCB), #2160                                ; 3234
                                 0F  12 0002E               BNEQ    2$                                            ;
                                 57  DD 00030               PUSHL   BCB                                           ; 3237
      0000V       CF             01  FB 00032               CALLS   #1, NEXT_VOLUME                               ;
                  57             50  D0 00037               MOVL    R0, BCB                                       ;
                                 03  12 0003A               BNEQ    2$                                            ; 3238
                                 50  D4 0003C               CLRL    R0                                            ; 3239
                                 04     0003E               RET                                                   ;
                           04    AA  D5 0003F  2$:          TSTL    RWSV_IN_XOR_RFA                               ; 3242
                                 14  12 00042               BNEQ    3$                                            ;
                           08    AA  B5 00044               TSTW    RWSV_IN_XOR_RFA+4                             ; 3243
```

```
                                    0F   12 00047          BNEQ    3$                                          ;
          58      EC   AA 00000050  8F   C1 00049          ADDL3   #80, RWSV_SAVE_FAB, RAB                     ; 3246
    04    AA      10   A8           06   28 00052          MOVC3   #6, 16(RAB), RWSV_IN_XOR_RFA                ; 3247
                  56         0C     A7   D0 00058 3$:      MOVL    12(BCB), BUFFER                             ; 3249
                  50         F8     AA   D0 0005C          MOVL    RWSV_IN_SEQ, R0                             ; 3259
                  51         01     A0   9E 00060          MOVAB   1(R0), R1                                   ;
                  51         08     A6   D1 00064          CMPL    8(BUFFER), R1                               ;
                  2B                     12 00068          BNEQ    6$                                          ;
          51               6A  0C   AA   C1 0006A          ADDL3   RWSV_IN_GROUP_SIZE, RWSV_IN_XOR_SEQ, R1     ; 3260
                                    51   D6 0006F          INCL    R1                                          ;
                  51                50   D1 00071          CMPL    R0, R1                                      ;
                                    06   12 00074          BNEQ    4$                                          ;
                  02         06     A6   B1 00076          CMPW    6(BUFFER), #2                               ; 3261
                                    19   12 0007A          BNEQ    6$                                          ;
                  01                50   D1 0007C 4$:      CMPL    R0, #1                                      ; 3262
                                    04   12 0007F          BNEQ    5$                                          ;
                  10         14     AA   E8 00081          BLBS    RWSV_IN_ORGERR, 6$                          ;
                           0081     8F   BB 00085 5$:      PUSHR   #^M<R0,R7>                                  ; 3265
             F815     CF            02   FB 00089          CALLS   #2, XORCIZE                                 ;
                  57                50   D0 0008E          MOVL    R0, BCB                                     ;
                  56         0C     A7   D0 00091          MOVL    12(BCB), BUFFER                             ; 3266
                  50         F8     AA   D0 00095 6$:      MOVL    RWSV_IN_SEQ, R0                             ; 3269
          59      08   A6           50   C3 00099          SUBL3   R0, 8(BUFFER), K                            ;
                                    49   13 0009E          BEQL    11$                                         ; 3270
                  01                50   D1 000A0          CMPL    R0, #1                                      ; 3273
                                    04   12 000A3          BNEQ    7$                                          ;
                  1F         14     AA   E8 000A5          BLBS    RWSV_IN_ORGERR, 9$                          ;
          51      6A         0C     AA   C1 000A9 7$:      ADDL3   RWSV_IN_GROUP_SIZE, RWSV_IN_XOR_SEQ, R1     ; 3276
                  51                50   D1 000AE          CMPL    R0, R1                                      ;
                                    05   12 000B1          BNEQ    8$                                          ;
                  01                59   D1 000B3          CMPL    K, #1                                       ; 3277
                                    13   13 000B6          BEQL    10$                                         ;
                  7E         14     AA   7D 000B8 8$:      MOVQ    RWSV_IN_ORGERR, -(SP)                       ; 3281
                             EC     AA   DD 000BC          PUSHL   RWSV_SAVE_FAB                               ; 3280
                      00000000G     8F   DD 000BF          PUSHL   #BACKUP$_BLOCKLOST                          ;
                             6B     04   FB 000C5          CALLS   #4, FILE_ERROR                              ;
                             10     AA   B7 000C8 9$:      DECW    RWSV_IN_ERRORS                              ; 3286
                  51         0C     AA   D0 000CB 10$:     MOVL    RWSV_IN_GROUP_SIZE, R1                      ; 3287
                                    18   13 000CF          BEQL    11$                                         ;
          50           6A           51   C1 000D1          ADDL3   R1, RWSV_IN_XOR_SEQ, R0                     ; 3290
                       50    08     A6   D1 000D5          CMPL    8(BUFFER), R0                               ;
                                    0E   1F 000D9          BLSSU   11$                                         ;
          50      08   A6           6A   C3 000DB          SUBL3   RWSV_IN_XOR_SEQ, 8(BUFFER), R0              ; 3292
                       50           51   C6 000E0          DIVL2   R1, R0                                      ; 3293
                       50           51   C4 000E3          MULL2   R1, R0                                      ;
                       6A           50   C0 000E6          ADDL2   R0, RWSV_IN_XOR_SEQ                         ;
             0100      8F           66   B1 000E9 11$:     CMPW    (BUFFER), #256                              ; 3302
                                    10   1E 000EE          BGEQU   12$                                         ;
                      00000000G     8F   DD 000F0          PUSHL   #BACKUP$_INVBLKHDR                          ; 3303
        00000000G      00           01   FB 000F6          CALLS   #1, LIB$SIGNAL                              ;
                                  00BE   31 000FD          BRW     20$                                         ;
                       01    06     A6   B1 00100 12$:     CMPW    6(BUFFER), #1                               ; 3311
                                    15   12 00104          BNEQ    13$                                         ;
             0101      8F    20     A6   B1 00106          CMPW    32(BUFFER), #257                            ; 3314
                                    0F   13 0010C          BEQL    14$                                         ;
                      00000000G     8F   DD 0010E          PUSHL   #BACKUP$_INVSTRUCT                          ; 3315
        00000000G      00           01   FB 00114          CALLS   #1, LIB$SIGNAL                              ;
```

```
                                            7E  11 0011B 13$:   BRB     19$                                          ;
                              28        18  A7  E8 0011D 14$:   BLBS    24(BCB), 15$                                 ; 3318
                                            57  DD 00121        PUSHL   BCB                                          ; 3321
                                        F8  AA  DD 00123        PUSHL   RWSV_IN_SEQ                                  ;
                     F778     CF            02  FB 00126        CALLS   #2, XORCIZE                                  ;
                              57            50  D0 0012B        MOVL    R0, BCB                                      ;
                              56        0C  A7  D0 0012E        MOVL    12(BCB), BUFFER                              ; 3322
                              13        18  A7  E8 00132        BLBS    24(BCB), 15$                                 ; 3325
                              7E        14  AA  7D 00136        MOVQ    RWSV_IN_ORGERR, -(SP)                        ; 3329
                                        EC  AA  DD 0013A        PUSHL   RWSV_SAVE_FAB                                ;
                                 00000000G  8F  DD 0013D        PUSHL   #BACKUP$_READERR+2                           ; 3328
                              6B            04  FB 00143        CALLS   #4, FILE_ERROR                               ;
                                        10  AA  B7 00146        DECW    RWSV_IN_ERRORS                               ; 3330
                              50      00E0  CA  9A 00149 15$:   MOVZBL  ALT_SSNAME, R0                               ; 3338
                                            50  D6 0014E        INCL    R0                                           ;
                              54        30  A6  9A 00150        MOVZBL  48(BUFFER), R4                               ; 3339
                                            54  D6 00154        INCL    R4                                           ;
         54       FF    8F   00E0     CA    50  2D 00156        CMPC5   R0, ALT_SSNAME, #255, R4, 48(BUFFER)         ;
                                        30  A6     0015E                                                             ;
                                            1B  13 00160        BEQL    17$                                          ;
                                      00E0  CA  95 00162        TSTB    ALT_SSNAME                                   ; 3342
                                            0C  13 00166        BEQL    16$                                          ;
                                        EC  AA  DD 00168        PUSHL   RWSV_SAVE_FAB                                ; 3343
                                 00000000G  8F  DD 0016B        PUSHL   #BACKUP$_SSCHANGE                            ;
                              6B            02  FB 00171        CALLS   #2, FILE_ERROR                               ;
         20             00     30     A6    54  2C 00174 16$:   MOVC5   R4, 48(BUFFER), #0, #32, ALT_SSNAME          ; 3344
                                      00E0  CA     0017A                                                             ;
    28   A6       08    A7     10           00  ED 0017D 17$:   CMPZV   #0, #16, 8(BCB), 40(BUFFER)                  ; 3348
                                            0D  13 00184        BEQL    18$                                          ;
                                 00000000G  8F  DD 00186        PUSHL   #BACKUP$_INVBLKSIZE                          ; 3349
                  00000000G   00            01  FB 0018C        CALLS   #1, LIB$SIGNAL                               ;
                  F8    AA     08     A6    01  C1 00193 18$:   ADDL3   #1, 8(BUFFER), RWSV_IN_SEQ                   ; 3350
                                            35  11 00199        BRB     21$                                          ; 3317
                              02        06  A6  B1 0019B 19$:   CMPW    6(BUFFER), #2                                ; 3358
                                            1D  12 0019F        BNEQ    20$                                          ;
                              6A        08  A6  D0 001A1        MOVL    8(BUFFER), RWSV_IN_XOR_SEQ                   ; 3361
                       04     AA        14  A7  D0 001A5        MOVL    20(BCB), RWSV_IN_XOR_RFA                     ; 3362
                        0F    4F        AA  03  E1 001AA        BBC     #3, QUAL+15, 20$                             ; 3363
                        58    EC    AA 00000050  8F  C1 001AF   ADDL3   #80, RWSV_SAVE_FAB, RAB                      ; 3366
                  04    AA     10     A8    06  28 001B8        MOVC3   #6, 16(RAB), RWSV_IN_XOR_RFA                 ; 3367
                  F8    AA     08     A6    01  C1 001BE 20$:   ADDL3   #1, 8(BUFFER), RWSV_IN_SEQ                   ; 3373
                                            57  DD 001C4        PUSHL   BCB                                          ; 3374
                  00000000G   00            01  FB 001C6        CALLS   #1, FREE_BUFFER                              ;
                                          FE4A  31 001CD        BRW     1$                                           ; 3228
                              50            57  D0 001D0 21$:   MOVL    BCB, R0                                      ; 3378
                                                04 001D3        RET                                                  ;

; Routine Size:  468 bytes,    Routine Base:  CODE + 0D6C
```

```
1 %SBTTL 'FIN_IN_SAVE - finish reading save set'
1 GLOBAL ROUTINE FIN_IN_SAVE (CONTINUE) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine completes the reading of the input save set.
1 !
1 ! CALLING SEQUENCE:
1 !       FIN_IN_SAVE (CONTINUE)
1 !
1 ! INPUT PARAMETERS:
1 !       CONTINUE: TRUE if there is a continuation tape
1 !                 FALSE if this is the last reel
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
2 BUILTIN
2       REMQUE;
2
2 LOCAL
2       STATUS,                         ! general status value
2       TM_COUNT,                       ! count of tape marks crossed
2       BCB             : REF BBLOCK,   ! pointer to current BCB
2       FAB             : REF BBLOCK;   ! input FAB
2
2 EXTERNAL ROUTINE
2       WAIT,                           ! wait for I/O completion
2       FREE_BUFFER,                    ! free an I/O buffer
2       SKIP_TM,                        ! skip tape marks
2       UNLOAD,                         ! rewind and unload tape
2       STA_DISMOUNT,                   ! dismount save set disk
2       RESTORE_VERIFY_REEL,            ! verify input save set volume
2       FILE_ERROR;                     ! signal file related error
2
2 ! If the input is a file, close it.
2 !
2
2 FAB = .RWSV_SAVE_FAB;
```

```
2 IF .QUAL[QUAL_SS_FILE]
2 THEN
3     BEGIN
3     IF NOT .QUAL[QUAL_VERI]
3     OR .COM_FLAGS[COM_VERIFYING]
3     THEN
4         BEGIN
4         STATUS = $CLOSE (FAB = .FAB);
4         IF NOT .STATUS
4         THEN FILE_ERROR (BACKUP$_CLOSEIN+STS$K_ERROR, .FAB,
4                          .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
3         END;
3     END
3
3 ! For a tape, wait out the pending read aheads, and backspace the tape
3 ! over the trailer label set so that appending won't get lost.
3 !
3
2 ELSE
3     BEGIN
3     IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
3     THEN
4         BEGIN
4         TM_COUNT = 0;
4         UNTIL REMQUE (.INPUT_WAIT[0], BCB)
4         DO
5             BEGIN
5             WAIT (.BCB);
5             FREE_BUFFER (.BCB);
5             IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
5             THEN TM_COUNT = .TM_COUNT + 1;
4             END;
4
4         SKIP_TM (-.TM_COUNT);
4
4         IF NOT .QUAL[QUAL_VERI]
4         OR .COM_FLAGS[COM_VERIFYING]
4         THEN
5             BEGIN
5             IF .CONTINUE
5             THEN UNLOAD ();
5
5             $DASSGN (CHAN = .RWSV_CHAN);
5             RWSV_CHAN = 0;
4             END;
4         END
4
4 ! For a save set disk, wait out any pending reads. Then deaccess the file
4 ! and dismount the volume.
4 !
4
3     ELSE
4         BEGIN
4         UNTIL REMQUE (.INPUT_WAIT[0], BCB)
4         DO
5             BEGIN
5             WAIT (.BCB);
```

```
                            FREE_BUFFER (.BCB);
                            END;

                        IF NOT .QUAL[QUAL_VERI]
                        OR .COM_FLAGS[COM_VERIFYING]
                        THEN
                            BEGIN
                            $QIOW (CHAN = .RWSV_CHAN,
                                    FUNC = IO$_DEACCESS
                                    );
                            IF .CURRENT_MTL[MTL_SEQ_DISK]
                            THEN
                                BEGIN
                                STA_DISMOUNT (.RWSV_VOL_NUMBER);
                                $QIOW (CHAN = .CURRENT_VCB[VCB_CHAN],
                                        FUNC = IO$_UNLOAD
                                        );
                                END;
                            END;
                        END;
                    END;

    ! Inform the user of any input errors that we might have graciously
    ! recovered.
    !

    IF .RWSV_IN_ERRORS NEQ 0
    THEN FILE_ERROR (BACKUP$_SOFTRERRS, .FAB, .RWSV_IN_ERRORS);
    IF .RWSV_IN_XORUSE NEQ 0
    THEN FILE_ERROR (BACKUP$_XORERRS, .FAB, .RWSV_IN_XORUSE);


    ! Run a verify pass if requested (and if this is not already a verify pass).
    !

    IF NOT .COM_FLAGS[COM_VERIFYING]
    THEN
        BEGIN
        IF .QUAL[QUAL_VERI]
        THEN
            BEGIN
            SIGNAL (BACKUP$_STARTVERIFY);
            COM_FLAGS[COM_VERIFYING] = TRUE;
            IF .RWSV_SEG_NUMBER EQL 0
            THEN INIT_IN_SAVE (TRUE)
            ELSE
                IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
                THEN READY_NEXT_VOLUME ();
            RWSV_IN_SEQ = .RWSV_IN_SEQ_0;
            RWSV_IN_VBN = .RWSV_IN_VBN_0;
            RWSV_IN_ERRORS = 0;
            RWSV_IN_XORUSE = 0;
            RESTORE_VERIFY_REEL ();
            FIN_IN_SAVE (.CONTINUE);
            COM_FLAGS[COM_VERIFYING] = FALSE;
            END;
        IF .CONTINUE THEN SIGNAL (BACKUP$_RESUME, 1, .RWSV_VOL_NUMBER+1);
```

```
; 2006          3550  2        END;
; 2007          3551  2
; 2008          3552  1 END;                                   ! End of routine FIN_IN_SAVE


                                                                 .EXTRN  UNLOAD, STA_DISMOUNT
                                                                 .EXTRN  RESTORE_VERIFY_REEL
                                                                 .EXTRN  SYS$CLOSE, SYS$DASSGN
                                                                 .EXTRN  SYS$QIOW

                                          03FC 00000             .ENTRY  FIN_IN_SAVE, Save R2,R3,R4,R5,R6,R7,R8,R9     ; 3380
                    59 00000000G     00  9E 00002                MOVAB   LIB$SIGNAL, R9                                 ;
                    58 00000000G     00  9E 00009                MOVAB   FREE_BUFFER, R8                                ;
                    57 00000000G     00  9E 00010                MOVAB   WAIT, R7                                       ;
                    56 00000000G     00  9E 00017                MOVAB   FILE_ERROR, R6                                 ;
                    55 00000000'     EF  9E 0001E                MOVAB   COM_FLAGS, R5                                  ;
                    54        FF32   C5  D0 00025                MOVL    RWSV_SAVE_FAB, FAB                             ; 3435
      26       95   A5               03  E1 0002A                BBC     #3, QUAL+15, 2$                                ; 3436
                          93         A5  95 0002F                TSTB    QUAL+13                                        ; 3439
                                     04  18 00032                BGEQ    1$                                             ;
      6E            65               03  E1 00034                BBC     #3, COM_FLAGS, 7$                              ; 3440
                                     54  DD 00038 1$:            PUSHL   FAB                                            ; 3443
         00000000G  00               01  FB 0003A                CALLS   #1, SYS$CLOSE                                  ;
                    62               50  E8 00041                BLBS    STATUS, 7$                                     ; 3444
                    7E        08     A4  7D 00044                MOVQ    8(FAB), -(SP)                                  ; 3446
                                     54  DD 00048                PUSHL   FAB                                            ; 3445
                       00000000G     8F  DD 0004A                PUSHL   #BACKUP$_CLOSEIN+2                             ;
                    66               04  FB 00050                CALLS   #4, FILE_ERROR                                 ;
                                     51  11 00053                BRB     7$                                             ; 3436
      4E       40   A4               05  E1 00055 2$:            BBC     #5, 64(FAB), 8$                                ; 3456
                                     52  D4 0005A                CLRL    TM_COUNT                                       ; 3459
                    53        FE42   D5  0F 0005C 3$:            REMQUE  @INPUT_WAIT, BCB                               ; 3460
                                     16  1D 00061                BVS     4$                                             ;
                                     53  DD 00063                PUSHL   BCB                                            ; 3463
                    67               01  FB 00065                CALLS   #1, WAIT                                       ;
                                     53  DD 00068                PUSHL   BCB                                            ; 3464
                    68               01  FB 0006A                CALLS   #1, FREE_BUFFER                                ;
            0870    8F        18     A3  B1 0006D                CMPW    24(BCB), #2160                                 ; 3465
                                     E7  12 00073                BNEQ    3$                                             ;
                                     52  D6 00075                INCL    TM_COUNT                                       ; 3466
                                     E3  11 00077                BRB     3$                                             ; 3460
                    7E               52  CE 00079 4$:            MNEGL   TM_COUNT, -(SP)                                ; 3469
         00000000G  00               01  FB 0007C                CALLS   #1, SKIP_TM                                    ;
                          93         A5  95 00083                TSTB    QUAL+13                                        ; 3471
                                     04  18 00086                BGEQ    5$                                             ;
      34            65               03  E1 00088                BBC     #3, COM_FLAGS, 10$                             ; 3472
                    07        04     AC  E9 0008C 5$:            BLBC    CONTINUE, 6$                                   ; 3475
         00000000G  00               00  FB 00090                CALLS   #0, UNLOAD                                     ; 3476
                              FF36   C5  DD 00097 6$:            PUSHL   RWSV_CHAN                                      ; 3478
         00000000G  00               01  FB 0009B                CALLS   #1, SYS$DASSGN                                 ;
                              FF36   C5  D4 000A2                CLRL    RWSV_CHAN                                      ; 3479
                                     66  11 000A6 7$:            BRB     12$                                            ; 3456
                    53        FE42   D5  0F 000A8 8$:            REMQUE  @INPUT_WAIT, BCB                               ; 3489
                                     0C  1D 000AD                BVS     9$                                             ;
                                     53  DD 000AF                PUSHL   BCB                                            ; 3492
                    67               01  FB 000B1                CALLS   #1, WAIT                                       ;
```

```
                                   53  DD 000B4          PUSHL   BCB                                   ; 3493
                      68           01  FB 000B6          CALLS   #1, FREE_BUFFER                       ;
                                   ED  11 000B9          BRB     8$                                    ; 3489
                             93    A5  95 000BB  9$:     TSTB    QUAL+13                               ; 3496
                                   04  18 000BE          BGEQ    11$                                   ;
          4A          65           03  E1 000C0  10$:    BBC     #3, COM_FLAGS, 12$                    ; 3497
                                   7E  7C 000C4  11$:    CLRQ    -(SP)                                 ; 3502
                                   7E  7C 000C6          CLRQ    -(SP)                                 ;
                                   7E  7C 000C8          CLRQ    -(SP)                                 ;
                                   7E  7C 000CA          CLRQ    -(SP)                                 ;
                      7E           34  7D 000CC          MOVQ    #52, -(SP)                            ;
                           FF36    C5  DD 000CF          PUSHL   RWSV_CHAN                             ;
                                   7E  D4 000D3          CLRL    -(SP)                                 ;
            00000000G 00           0C  FB 000D5          CALLS   #12, STA_QIOW                         ;
                      50   05FE    C5  D0 000DC          MOVL    CURRENT_MTL, R0                       ; 3503
                      29     31    A0  E9 000E1          BLBC    49(R0), 12$                           ;
                      7E   FF26    C5  3C 000E5          MOVZWL  RWSV_VOL_NUMBER, -(SP)                ; 3506
            00000000G 00           01  FB 000EA          CALLS   #1, STA_DISMOUNT                      ;
                                   7E  7C 000F1          CLRQ    -(SP)                                 ; 3509
                                   7E  7C 000F3          CLRQ    -(SP)                                 ;
                                   7E  7C 000F5          CLRQ    -(SP)                                 ;
                                   7E  7C 000F7          CLRQ    -(SP)                                 ;
                      7E           01  7D 000F9          MOVQ    #1, -(SP)                             ;
                      50   0602    C5  D0 000FC          MOVL    CURRENT_VCB, R0                       ;
                      7E     08    A0  3C 00101          MOVZWL  8(R0), -(SP)                          ;
                                   7E  D4 00105          CLRL    -(SP)                                 ;
            00000000G 00           0C  FB 00107          CALLS   #12, SYS$QIOW                         ;
                      50   FF56    C5  3C 0010E  12$:    MOVZWL  RWSV_IN_ERRORS, R0                    ; 3519
                                   0D  13 00113          BEQL    13$                                   ;
                                   50  DD 00115          PUSHL   R0                                    ; 3520
                                   54  DD 00117          PUSHL   FAB                                   ;
                      00000000G    8F  DD 00119          PUSHL   #BACKUP$_SOFTRERRS                    ;
                      66           03  FB 0011F          CALLS   #3, FILE_ERROR                        ;
                      50   FF58    C5  3C 00122  13$:    MOVZWL  RWSV_IN_XORUSE, R0                    ; 3521
                                   0D  13 00127          BEQL    14$                                   ;
                                   50  DD 00129          PUSHL   R0                                    ; 3522
                                   54  DD 0012B          PUSHL   FAB                                   ;
                      00000000G    8F  DD 0012D          PUSHL   #BACKUP$_XORERRS                      ;
                      66           03  FB 00133          CALLS   #3, FILE_ERROR                        ;
          64          65           03  E0 00136  14$:    BBS     #3, COM_FLAGS, 18$                    ; 3528
                             93    A5  95 0013A          TSTB    QUAL+13                               ; 3531
                                   49  18 0013D          BGEQ    17$                                   ;
                      00000000G    8F  DD 0013F          PUSHL   #BACKUP$_STARTVERIFY                  ; 3534
                      69           01  FB 00145          CALLS   #1, LIB$SIGNAL                        ;
                      65           08  88 00148          BISB2   #8, COM_FLAGS                         ; 3535
                           FF28    C5  B5 0014B          TSTW    RWSV_SEG_NUMBER                       ; 3536
                                   09  12 0014F          BNEQ    15$                                   ;
                                   01  DD 00151          PUSHL   #1                                    ; 3537
               FB42   CF           01  FB 00153          CALLS   #1, INIT_IN_SAVE                      ;
                                   0A  11 00158          BRB     16$                                   ;
          05     40   A4           05  E1 0015A  15$:    BBC     #5, 64(FAB), 16$                      ; 3539
              0000V   CF           00  FB 0015F          CALLS   #0, READY_NEXT_VOLUME                 ; 3540
               FF3E   C5   FF42    C5  D0 00164  16$:    MOVL    RWSV_IN_SEQ_0, RWSV_IN_SEQ            ; 3541
               FF62   C5   FF66    C5  D0 0016B          MOVL    RWSV_IN_VBN_0, RWSV_IN_VBN            ; 3542
                           FF56    C5  D4 00172          CLRL    RWSV_IN_ERRORS                        ; 3543
            00000000G 00           00  FB 00176          CALLS   #0, RESTORE_VERIFY_REEL               ; 3545
                             04    AC  DD 0017D          PUSHL   CONTINUE                              ; 3546
```

```
                    FE7B   CF           01   FB 00180          CALLS    #1, FIN_IN_SAVE                      ;
                           65           08   8A 00185          BICB2    #8, COM_FLAGS                        ; 3547
                           12      04   AC   E9 00188 17$:     BLBC     CONTINUE, 18$                        ; 3549
                           7E    FF26   C5   3C 0018C          MOVZWL   RWSV_VOL_NUMBER, -(SP)               ;
                                        6E   D6 00191          INCL     (SP)                                 ;
                                        01   DD 00193          PUSHL    #1                                   ;
                             00000000G  8F   DD 00195          PUSHL    #BACKUP$_RESUME                      ;
                           69           03   FB 0019B          CALLS    #3, LIB$SIGNAL                       ;
                                             04 0019E 18$:     RET                                           ; 3552

; Routine Size:  415 bytes,     Routine Base:  CODE + 0F40
```

```
1 %SBTTL 'NEXTVOL_HANDLER - handler for next volume processing'
1 ROUTINE NEXTVOL_HANDLER (SIGNAL, MECHANISM) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is the condition handler for the TRY_NEXT_VOLUME
1 !       routine. It resignals the error with informational severity and then
1 !       returns failure, causing a retry.
1 !
1 ! CALLING SEQUENCE:
1 !       NEXTVOL_HANDLER (SIGNAL, MECHANISM)
1 !
1 ! INPUT PARAMETERS:
1 !       SIGNAL: signal argument list
1 !       MECHANISM: mechanism arg list
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       TRUE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       CALLG;
2
2 MAP
2       SIGNAL          : REF BBLOCK,   ! signal arg
2       MECHANISM       : REF BBLOCK;   ! mechanism arg
2
2 EXTERNAL ROUTINE
2       STA_DISMOUNT,                   ! dismount volume
2       LIB$SIGNAL                      : ADDRESSING_MODE (GENERAL);
2
2 ! Fix up the severity of the error being signalled. Then resignal it
2 ! and unwind.
2 !
2
2 IF .SIGNAL[CHF$L_SIG_NAME] NEQ SS$_UNWIND
2 AND NOT .SIGNAL[CHF$[ _SIG_NAME]
2 THEN
3     BEGIN
3     IF NOT .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
```

```
; 2067          3610  3        THEN
; 2068          3611  4            BEGIN
; 2069        P 3612  4            S$QIOW (CHAN = STA_IN_CHAN,
; 2070        P 3613  4                    FUNC = IO$_DEACCESS
; 2071          3614  4                    );
; 2072          3615  4            STA_DISMOUNT (.RWSV_VOL_NUMBER);
; 2073          3616  3            END;
; 2074          3617  3        BBLOCK [SIGNAL[CHF$L_SIG_NAME], STS$V_SEVERITY] = STS$K_INFO;
; 2075          3618  3        SIGNAL[CHF$L_SIG_ARGS] = .SIGNAL[CHF$L_SIG_ARGS] - 2;
; 2076          3619  3        CALLG (.SIGNAL, LIB$SIGNAL);
; 2077          3620  3        MECHANISM[CHF$L_MCH_SAVR0] = FALSE;
; 2078          3621  3        $UNWIND ();
; 2079          3622  2        END;
; 2080          3623  2
; 2081          3624  2 SS$_RESIGNAL
; 2082          3625  1 END;                                     ! End of routine NEXTVOL_HANDLER


                                                                .EXTRN  LIB$SIGNAL, SYS$UNWIND

                                             0004 00000 NEXTVOL_HANDLER:
                                                                .WORD   Save R2                          ; 3554
                        52         04   AC  D0 00002            MOVL    SIGNAL, R2                       ; 3605
             00000920   8F         04   A2  D1 00006            CMPL    4(R2), #2336                     ;
                                        58  13 0000E            BEQL    2$                               ;
                        54         04   A2  E8 00010            BLBS    4(R2), 2$                        ; 3606
                        50  00000000'   EF  D0 00014            MOVL    RWSV_SAVE_FAB, R0                ; 3609
                  28         40   A0    05  E0 0001B            BBS     #5, 64(R0), 1$                   ;
                                        7E  7C 00020            CLRQ    -(SP)                            ; 3614
                                        7E  7C 00022            CLRQ    -(SP)                            ;
                                        7E  7C 00024            CLRQ    -(SP)                            ;
                                        7E  7C 00026            CLRQ    -(SP)                            ;
                        7E              34  7D 00028            MOVQ    #52, -(SP)                       ;
                           0001FFFF     8F  DD 0002B            PUSHL   #131071                          ;
                                        7E  D4 00031            CLRL    -(SP)                            ;
             00000000G  00              0C  FB 00033            CALLS   #12, STA_QIOW                    ;
                        7E  00000000'   EF  3C 0003A            MOVZWL  RWSV_VOL_NUMBER, -(SP)           ; 3615
             00000000G  00              01  FB 00041            CALLS   #1, STA_DISMOUNT                 ;
      04   A2          03   00          03  F0 00048 1$:        INSV    #3, #0, #3, 4(R2)                ; 3617
                        62              02  C2 0004E            SUBL2   #2, (R2)                         ; 3618
             00000000G  00              62  FA 00051            CALLG   (R2), LIB$SIGNAL                 ; 3619
                        50         08   AC  D0 00058            MOVL    MECHANISM, R0                    ; 3620
                                   0C   A0  D4 0005C            CLRL    12(R0)                           ;
                                        7E  7C 0005F            CLRQ    -(SP)                            ; 3621
             00000000G  00              02  FB 00061            CALLS   #2, SYS$UNWIND                   ;
                        50       0918   8F  3C 00068 2$:        MOVZWL  #2328, R0                        ; 3625
                                            04 0006D            RET                                      ;

; Routine Size:  110 bytes,    Routine Base:  CODE + 10DF
```

```
1 %SBTTL 'READY_NEXT_VOLUME - set up next volume'
1 GLOBAL ROUTINE READY_NEXT_VOLUME (FILE_ID) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads and verifies the labels on the candidate
1 !       continuation volume.
1 !
1 ! CALLING SEQUENCE:
1 !       READY_NEXT_VOLUME ()
1 !
1 ! INPUT PARAMETERS:
1 !       FILE_ID: address of continuation file ID if disk
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       TRUE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FILE_ID         : REF BBLOCK;   ! file ID arg
2
2 LOCAL
2       FAB             : REF BBLOCK,   ! pointer to input FAB
2       STATUS,                         ! the usual status value
2       BUFFER_SIZE,                    ! size of I/O buffers
2       NUMBER,                         ! output for decimal convert
2       LABEL_BUFFER    : BBLOCK [90];  ! buffer for tape labels
2
2 EXTERNAL ROUTINE
2       FILE_ERROR,                     ! signal file related error
2       READY_TAPE,                     ! prepare tape for I/O
2       REWIND,                         ! rewind tape
2       SKIP_TM,                        ! skip tape marks
2       READ_LABEL,                     ! read and check tape label
2       LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL);
2                                       ! convert decimal to binary
2
2 FAB = .RWSV_SAVE_FAB;
2 RWSV_IN_ERRORS = 0;
2 RWSV_IN_XORUSE = 0;
```

```

! Set up the input tape. Rewind it if called for.
!

IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
THEN
    BEGIN
    READY_TAPE (FALSE);
    REWIND ();

! Read and verify the volume header label. Then scan for HDR1.
!

    STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
    IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);

    WHILE TRUE
    DO
        BEGIN
        STATUS = READ_LABEL (LABEL_BUFFER);
        IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
        IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'HDR1' THEN EXITLOOP;
        END;

! Check HDR1 for the correct save set name, file set ID, and file
! & section numbers.
!

    IF NOT MATCH_SSNAME (LABEL_BUFFER)

    OR CH$NEQ (HD1$S_FILESETID, LABEL_BUFFER[HD1$T_FILESETID],
               HD1$S_FILESETID, RWSV_FILESET_ID)

    OR
        BEGIN
        IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], NUMBER)
        THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
        .NUMBER NEQ .RWSV_FILE_NUMBER
        END

    OR
        BEGIN
        IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESECNO], NUMBER)
        THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
        .NUMBER NEQ .RWSV_VOL_NUMBER
        END

    THEN FILE_ERROR (BACKUP$_WRONGVOL, .FAB);

! Read and check HDR2.
!

    STATUS = READ_LABEL (LABEL_BUFFER, 'HDR2');
    IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
    IF .LABEL_BUFFER[HD2$B_RECFORMAT] NEQ 'F'
    OR CH$NEQ (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN],
               HD2$S_BLOCKLEN, LABEL_BUFFER[HD2$T_BLOCKLEN])
```

```
; 2198          3740  3          THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
; 2199          3741  3
; 2200          3742  3          IF NOT LIB$CVT_DTB (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN], BUFFER_SIZE)
; 2201          3743  3          THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
; 2202          3744  3          IF .BUFFER_SIZE NEQ .BBLOCK [.FREE_LIST[0], BCB_SIZE]
; 2203          3745  3          THEN FILE_ERROR (BACKUP$_BADBLKSIZE, .FAB);
; 2204          3746  3
; 2205          3747  3      ! Skip to start of the data records.
; 2206          3748  3      !
; 2207          3749  3
; 2208          3750  3          SKIP_TM (1);
; 2209          3751  3          END
; 2210          3752  3
; 2211          3753  3      ! Sequential disk is all handled by INIT_SAVE_DISK.
; 2212          3754  3      !
; 2213          3755  3
; 2214          3756  2      ELSE
; 2215          3757  2          INIT_SAVE_DISK (1, .FILE_ID);
; 2216          3758  2
; 2217          3759  2      TRUE
; 2218          3760  1      END;                                    ! End of routine READY_NEXT_VOLUME


                                          0FFC 00000          .ENTRY  READY_NEXT_VOLUME, Save R2,R3,R4,R5,R6,R7,- ; 3627
                                                                      R8,R9,R10,R11                                ;
                  5B 00000000G   8F  D0 00002          MOVL    #BACKUP$_NOTANSI, R11                        ;
                  5A 00000000G   00  9E 00009          MOVAB   LIB$CVT_DTB, R10                             ;
                  59 00000000G   00  9E 00010          MOVAB   READ_LABEL, R9                               ;
                  58 00000000'   EF  9E 00017          MOVAB   RWSV_SAVE_FAB, R8                            ;
                  57 00000000G   8F  D0 0001E          MOVL    #BACKUP$_LABELERR, R7                        ;
                  56 00000000G   00  9E 00025          MOVAB   FILE_ERROR, R6                               ;
                  5E          9C   AE  9E 0002C          MOVAB   -100(SP), SP                                 ;
                  54               68  D0 00030          MOVL    RWSV_SAVE_FAB, FAB                           ; 3680
                              24   A8  D4 00033          CLRL    RWSV_IN_ERRORS                               ; 3681
       03          40         A4   05  E0 00036          BBS     #5, 64(FAB), 1$                              ; 3687
                                 010C  31 0003B          BRW     13$                                          ;
                                   7E  D4 0003E  1$:     CLRL    -(SP)                                        ; 3690
            00000000G  00          01  FB 00040          CALLS   #1, READY_TAPE                               ;
            00000000G  00          00  FB 00047          CALLS   #0, REWIND                                   ; 3691
                        314C4F56   8F  DD 0004E          PUSHL   #827084630                                   ; 3696
                              0C   AE  9F 00054          PUSHAB  LABEL_BUFFER                                 ;
                  69               02  FB 00057          CALLS   #2, READ_LABEL                               ;
                  55               50  D0 0005A          MOVL    R0, STATUS                                   ;
                  07               55  E8 0005D          BLBS    STATUS, 2$                                   ; 3697
                                   30  BB 00060          PUSHR   #^M<R4,R5>                                   ;
                                   57  DD 00062          PUSHL   R7                                           ;
                  66               03  FB 00064          CALLS   #3, FILE_ERROR                               ;
                              08   AE  9F 00067  2$:     PUSHAB  LABEL_BUFFER                                 ; 3702
                  69               01  FB 0006A          CALLS   #1, READ_LABEL                               ;
                  55               50  D0 0006D          MOVL    R0, STATUS                                   ;
                  07               55  E8 00070          BLBS    STATUS, 3$                                   ; 3703
                                   30  BB 00073          PUSHR   #^M<R4,R5>                                   ;
                                   57  DD 00075          PUSHL   R7                                           ;
                  66               03  FB 00077          CALLS   #3, FILE_ERROR                               ;
```

```
                  31524448   8F        08  AE  D1 0007A 3$:   CMPL    LABEL_BUFFER, #827475016                   ; 3704
                                           E3  12 00082       BNEQ    2$                                          ;
                                       08  AE  9F 00084       PUSHAB  LABEL_BUFFER                                ; 3711
                         F5AF   CF         01  FB 00087       CALLS   #1, MATCH_SSNAME                            ;
                                42         50  E9 0008C       BLBC    R0, 6$                                      ;
            E0     A8      1D   AE         06  29 0008F       CMPC3   #6, LABEL_BUFFER+21, RWSV_FILESET_ID        ; 3713
                                           3A  12 00095       BNEQ    6$                                          ;
                                           5E  DD 00097       PUSHL   SP                                          ; 3718
                                       2B  AE  9F 00099       PUSHAB  LABEL_BUFFER+31                             ;
                                           04  DD 0009C       PUSHL   #4                                          ;
                                6A         03  FB 0009E       CALLS   #3, LIB$CVT_DTB                             ;
                                09         50  E8 000A1       BLBS    R0, 4$                                      ;
                                     0810  8F  BB 000A4       PUSHR   #^M<R4,R11>                                 ; 3719
                                           57  DD 000A8       PUSHL   R7                                          ;
                                66         03  FB 000AA       CALLS   #3, FILE_ERROR                              ;
                           F8   A8         6E  D1 000AD 4$:   CMPL    NUMBER, RWSV_FILE_NUMBER                    ; 3720
                                           1E  12 000B1       BNEQ    6$                                          ;
                                           5E  DD 000B3       PUSHL   SP                                          ; 3725
                                       27  AE  9F 000B5       PUSHAB  LABEL_BUFFER+27                             ;
                                           04  DD 000B8       PUSHL   #4                                          ;
                                6A         03  FB 000BA       CALLS   #3, LIB$CVT_DTB                             ;
                                09         50  E8 000BD       BLBS    R0, 5$                                      ;
                                     0810  8F  BB 000C0       PUSHR   #^M<R4,R11>                                 ; 3726
                                           57  DD 000C4       PUSHL   R7                                          ;
                                66         03  FB 000C6       CALLS   #3, FILE_ERROR                              ;
       6E          F4     A8    10         00  ED 000C9 5$:   CMPZV   #0, #16, RWSV_VOL_NUMBER, NUMBER            ; 3727
                                           0B  13 000CF       BEQL    7$                                          ;
                                           54  DD 000D1 6$:   PUSHL   FAB                                         ; 3730
                                00000000G  8F  DD 000D3       PUSHL   #BACKUP$_WRONGVOL                           ;
                                66         02  FB 000D9       CALLS   #2, FILE_ERROR                              ;
                                32524448   8F  DD 000DC 7$:   PUSHL   #844252232                                  ; 3735
                                       0C  AE  9F 000E2       PUSHAB  LABEL_BUFFER                                ;
                                69         02  FB 000E5       CALLS   #2, READ_LABEL                              ;
                                55         50  D0 000E8       MOVL    R0, STATUS                                  ;
                                07         55  E8 000EB       BLBS    STATUS, 8$                                  ; 3736
                                           30  BB 000EE       PUSHR   #^M<R4,R5>                                  ;
                                           57  DD 000F0       PUSHL   R7                                          ;
                                66         03  FB 000F2       CALLS   #3, FILE_ERROR                              ;
                           46   8F     0C  AE  91 000F5 8$:   CMPB    LABEL_BUFFER+4, #70                         ; 3737
                                           08  12 000FA       BNEQ    9$                                          ;
            0D     AE      12   AE         05  29 000FC       CMPC3   #5, LABEL_BUFFER+10, LABEL_BUFFER+5         ; 3739
                                           0B  13 00102       BEQL    10$                                         ;
                                           54  DD 00104 9$:   PUSHL   FAB                                         ; 3740
                                00000000G  8F  DD 00106       PUSHL   #BACKUP$_NOTSAVESET                         ;
                                66         02  FB 0010C       CALLS   #2, FILE_ERROR                              ;
                                       04  AE  9F 0010F 10$:  PUSHAB  BUFFER_SIZE                                 ; 3742
                                       16  AE  9F 00112       PUSHAB  LABEL_BUFFER+10                             ;
                                           05  DD 00115       PUSHL   #5                                          ;
                                6A         03  FB 00117       CALLS   #3, LIB$CVT_DTB                             ;
                                09         50  E8 0011A       BLBS    R0, 11$                                     ;
                                     0810  8F  BB 0011D       PUSHR   #^M<R4,R11>                                 ; 3743
                                           57  DD 00121       PUSHL   R7                                          ;
                                66         03  FB 00123       CALLS   #3, FILE_ERROR                              ;
                                50   FF08  C8  D0 00126 11$:  MOVL    FREE_LIST, R0                               ; 3744
  04   AE          08     A0    10         00  ED 0012B       CMPZV   #0, #16, 8(R0), BUFFER_SIZE                 ;
                                           0B  13 00132       BEQL    12$                                         ;
                                           54  DD 00134       PUSHL   FAB                                         ; 3745
```

```
                      00000000G  8F   DD 00136          PUSHL   #BACKUP$_BADBLKSIZE           ;
                   66            02   FB 0013C          CALLS   #2, FILE_ERROR                ;
                                 01   DD 0013F  12$:    PUSHL   #1                            ; 3750
          00000000G  00          01   FB 00141          CALLS   #1, SKIP_TM                   ;
                                 0A   11 00148          BRB     14$                           ; 3687
                            04   AC   DD 0014A  13$:    PUSHL   FILE_ID                       ; 3757
                                 01   DD 0014D          PUSHL   #1                            ;
              F558  CF           02   FB 0014F          CALLS   #2, INIT_SAVE_DISK            ;
                    50           01   D0 00154  14$:    MOVL    #1, R0                        ; 3760
                                      04 00157          RET                                   ;

; Routine Size:  344 bytes,    Routine Base:  CODE + 114D
```

```
1 %SBTTL 'TRY_NEXT_VOLUME - set up next volume under handler'
1 GLOBAL ROUTINE TRY_NEXT_VOLUME (FILE_ID) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads and verifies the labels on the candidate
1 !       continuation volume.
1 !
1 ! CALLING SEQUENCE:
1 !       TRY_NEXT_VOLUME ()
1 !
1 ! INPUT PARAMETERS:
1 !       FILE_ID: address of continuation file ID if disk
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       TRUE if volume is OK
1 !       FALSE if not; retry requested
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FILE_ID         : REF BBLOCK;   ! file ID arg
2
2 ! This routine simply wraps a handler around the READY_NEXT_VOLUME
2 ! routine so that errors cause a non-fatal message and error return,
2 ! rather than an exit.
2 !
2
2 ENABLE NEXTVOL_HANDLER;
2 READY_NEXT_VOLUME (.FILE_ID)
2
1 END;                                    ! End of routine TRY_NEXT_VOLUME
```

```
                        0000 00000        .ENTRY  TRY_NEXT_VOLUME, Save nothing      ; 3762
         6D      000A  CF   DE 00002        MOVAL   1$, (FP)                           ; 3795
                   04  AC   DD 00007        PUSHL   FILE_ID                            ; 3806
   FE99  CF            01   FB 0000A        CALLS   #1, READY_NEXT_VOLUME              ;
```

```
                                    04 0000F        RET                                        ; 3808
                                  0000 00010 1$:    .WORD   Save nothing                       ; 3795
                                7E  D4 00012        CLRL    -(SP)                              ;
                                5E  DD 00014        PUSHL   SP                                 ;
                  7E        04  AC  7D 00016        MOVQ    4(AP), -(SP)                       ;
            FE1B  CF            03  FB 0001A        CALLS   #3, NEXTVOL_HANDLER                ;
                                    04 0001F        RET                                        ;

; Routine Size:  32 bytes,    Routine Base:  CODE + 12A5
```

```
1 %SBTTL 'NEXT_VOLUME - switch to continuation volume'
1 ROUTINE NEXT_VOLUME (CUR_BCB) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is invoked when end of medium is encountered on
1 !       the input save set. It determines if there is a continuation
1 !       medium, and if so, readies it for use and returns the first block.
1 !
1 ! CALLING SEQUENCE:
1 !       NEXT_VOLUME (CUR_BCB)
1 !
1 ! INPUT PARAMETERS:
1 !       CUR_BCB: BCB of last read (containing end of file status)
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       BCB of first block on new volume, or 0 if no more volumes
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       CUR_BCB         : REF BBLOCK;   ! BCB input arg
2
2 LOCAL
2       STATUS,                         ! the usual status value
2       IO_STATUS       : VECTOR [4, WORD], ! I/O status block
2       FAB             : REF BBLOCK,   ! pointer to input FAB
2       BCB             : REF BBLOCK,   ! BCB of buffer in use
2       BUFFER          : REF BBLOCK,   ! current I/O buffer
2       NUMBER,                         ! output for decimal convert
2       ATT_CONTROL     : BBLOCK [12],  ! attribute control list
2       FILE_ID         : BBLOCK [FID$C_LENGTH], ! extension file ID
2       DEVICE_DESC     : VECTOR[2];    ! Input device desc if mounted /FOREIGN
2
2 EXTERNAL ROUTINE
2       FILE_ERROR,                     ! signal file related error
2       UNLOAD,                         ! rewind and unload tape
2       FREE_BUFFER;                    ! free an I/O buffer
2
2
2 ! Discard the BCB that reported the EOF. Return 0 (no continuation) if the
```

```
  2 ! save set is being handled through the file system.
  2 !
  2
  2 FAB = .RWSV_SAVE_FAB;
  2 RWSV_IN_VBN = .CUR_BCB[BCB_BLOCKNUM];
  2 FREE_BUFFER (.CUR_BCB);
  2
  2 IF .QUAL[QUAL_SS_FILE]
  2 THEN RETURN 0;
  2
  2 ! For tape, read the trailer labels and see if a continuation exists.
  2 !
  2
  2 IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
  2 THEN
  3     BEGIN
  3     BCB = READ_BLOCK (FALSE, FALSE);
  3     FREE_BUFFER (.BCB);
  3
  3     IF NOT .BCB[BCB_IO_STATUS]
  3     THEN
  4         BEGIN
  4         IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFVOLUME
  4         THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI)
  4         ELSE FILE_ERROR (BACKUP$_LABELERR, .FAB, .BCB[BCB_IO_STATUS]);
  4         END
  4
  3     ELSE IF .BCB[BCB_IO_BCOUNT] NEQ 80
  3     THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
  3
  3     BUFFER = .BCB[BCB_BUFFER];
  3     IF .BUFFER[HD1$L_HD1LID] EQL 'EOF1' OR .COM_FLAGS[COM_VERIFYING]
  3     THEN RETURN 0;
  3     IF .BUFFER[HD1$L_HD1LID] NEQ 'EOV1'
  3     THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
  3     END
  3
  3 ! For sequential disk, read the extension file ID and sequence number.
  3 !
  3
  2 ELSE
  3     BEGIN
  3     IF .COM_FLAGS[COM_VERIFYING] THEN RETURN 0;
  3     IF NOT .INPUT_MTL[MTL_SEQ_DISK] THEN RETURN 0;
  3     ATT_CONTROL[ATR$W_SIZE] = ATR$S_EXTFID;
  3     ATT_CONTROL[ATR$W_TYPE] = ATR$C_EXTFID;
  3     ATT_CONTROL[ATR$L_ADDR] = FILE_ID;
  3     ATT_CONTROL+8 = 0;
P 3     STATUS = $QIOW (CHAN = .RWSV_CHAN,
P 3                     FUNC = IO$_ACCESS,
P 3                     IOSB = IO_STATUS,
P 3                     P5   = ATT_CONTROL
  3                     );
  3     IF .STATUS THEN STATUS = .IO_STATUS[0];
  3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENIN, .FAB, .STATUS);
  3     IF .FILE_ID[FID$W_NUM] EQL 0
  3     AND .FILE_ID[FID$B_RVN] EQL 0
```

```
  3      THEN RETURN 0;
  3      IF .INPUT_MTL[MTL_SEQ_DISK]
  3      AND .FILE_ID[FID$B_RVN] NEQ .RWSV_VOL_NUMBER + 1
  3      THEN FILE_ERROR (BACKUP$_INVFILEXT, .FAB);
  2      END;
  2
  2  ! Finish processing of the current volume. Then set up the new input
  2  ! volume. Loop until the operator gets it right.
  2  !
  2
  2  FIN_IN_SAVE (TRUE);
  2  RWSV_SEG_NUMBER = .RWSV_SEG_NUMBER + 1;
  2
  2  ! Do a UFO open on the input FAB and save away the channel.
  2  !
  2
  2  RWSV_SAVE_QUAL = .RWSV_SAVE_QUAL[QUAL_NEXT];
  2  IF .RWSV_SAVE_QUAL EQL 0 THEN RWSV_SAVE_QUAL = .QUAL[QUAL_INPU_LIST];
  2  RWSV_SAVE_FAB = FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
  2
  2  IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
  2  THEN
  3      BEGIN
  3      RWSV_VOL_NUMBER = .RWSV_VOL_NUMBER + 1;
  3      FAB[FAB$V_NAM] = TRUE;
  3      FAB[FAB$V_UFO] = TRUE;
  3
  3  ! If the device is not mounted foreign do a $OPEN; otherwise do a $ASSIGN
  3  !
  3
  3      IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
  4      THEN STATUS = $OPEN (FAB = .FAB)
  3      ELSE
  4          BEGIN
  4          DEVICE_DESC[0] = .BBLOCK[.FAB[FAB$L_NAM], NAM$B_DEV];
  4          DEVICE_DESC[1] = .BBLOCK[.FAB[FAB$L_NAM], NAM$L_DEV];
P 4          STATUS = FAB[FAB$L_STS] = $ASSIGN (DEVNAM = DEVICE_DESC,
  4                                             CHAN = FAB[FAB$L_STV]);
  3          END;
  3      IF NOT .STATUS
  3      THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
  3                       .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
  3      RWSV_CHAN = .FAB[FAB$L_STV];
  3
  3      UNTIL TRY_NEXT_VOLUME ()
  3      DO UNLOAD ();
  3      END
  3
  2  ELSE
  3      BEGIN
  3      RWSV_VOL_NUMBER = .FILE_ID[FID$B_RVN];
  3      UNTIL TRY_NEXT_VOLUME (FILE_ID)
  3      DO
  4          BEGIN
  4          IF NOT .INPUT_MTL[MTL_SEQ_DISK]
  4          THEN FILE_ERROR (BACKUP$_READERR+STS$K_SEVERE, .FAB);
P 4          $QIOW (CHAN = .CURRENT_VCB[VCB_CHAN],
```

```
P 4                  FUNC = IO$_UNLOAD
  4                  );
  3            END;
  2         END;
  2
  2  ! Read the first block from the input volume.
  2  !
  2
  2  RWSV_IN_SEQ_0 = .RWSV_IN_SEQ;
  2  RWSV_IN_VBN_0 = .RWSV_IN_VBN;
  2  READ_SEQ_BLOCK (TRUE)
  2
  1  END;                                                ! End of routine NEXT_VOLUME


                                      01FC 00000 NEXT_VOLUME:
                                                      .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 3810
                 58 00000000G  8F  D0 00002           MOVL    #BACKUP$_LABELERR, R8                    ;
                 57 00000000G  8F  D0 00009           MOVL    #BACKUP$_NOTANSI, R7                     ;
                 56 00000000G  00  9E 00010           MOVAB   FREE_BUFFER, R6                          ;
                 55 00000000G  00  9E 00017           MOVAB   FILE_ERROR, R5                           ;
                 54 00000000'  EF  9E 0001E           MOVAB   RWSV_SAVE_QUAL, R4                       ;
                 5E            24  C2 00025           SUBL2   #36, SP                                  ;
                 53        04  A4  D0 00028           MOVL    RWSV_SAVE_FAB, FAB                       ; 3869
                 50        04  AC  D0 0002C           MOVL    CUR_BCB, R0                              ; 3870
          34  A4           14  A0  D0 00030           MOVL    20(R0), RWSV_IN_VBN                      ;
                               50  DD 00035           PUSHL   R0                                       ;
                 66            01  FB 00037           CALLS   #1, FREE_BUFFER                          ; 3871
      03      67  A4           03  E1 0003A           BBC     #3, QUAL+15, 1$                          ; 3873
                             00BE  31 0003F           BRW     10$                                      ;
      5A      40  A3           05  E1 00042 1$:       BBC     #5, 64(FAB), 7$                          ; 3879
                               7E  7C 00047           CLRQ    -(SP)                                    ; 3882
            ECED  CF           02  FB 00049           CALLS   #2, READ_BLOCK                           ;
                  52           50  D0 0004E           MOVL    R0, BCB                                  ;
                               52  DD 00051           PUSHL   BCB                                      ; 3883
                  66           01  FB 00053           CALLS   #1, FREE_BUFFER                          ;
                  0E       18  A2  E8 00056           BLBS    24(BCB), 2$                              ; 3885
            09A0  8F       18  A2  B1 0005A           CMPW    24(BCB), #2464                           ; 3888
                               0E  13 00060           BEQL    3$                                       ;
                  7E       18  A2  3C 00062           MOVZWL  24(BCB), -(SP)                           ; 3890
                               0A  11 00066           BRB     4$                                       ;
            0050  8F       1A  A2  B1 00068 2$:       CMPW    26(BCB), #80                             ; 3893
                               09  13 0006E           BEQL    5$                                       ;
                               57  DD 00070 3$:       PUSHL   R7                                       ; 3894
                               53  DD 00072 4$:       PUSHL   FAB                                      ;
                               58  DD 00074           PUSHL   R8                                       ;
                  65           03  FB 00076           CALLS   #3, FILE_ERROR                           ;
                  50       0C  A2  D0 00079 5$:       MOVL    12(BCB), BUFFER                          ; 3896
        31464F45  8F           60  D1 0007D           CMPL    (BUFFER), #826691397                     ; 3897
                               7A  13 00084           BEQL    10$                                      ;
      74    00D2  C4           03  E0 00086           BBS     #3, COM_FLAGS, 10$                       ;
        31564F45  8F           60  D1 0008C           CMPL    (BUFFER), #827739973                     ; 3899
                               09  13 00093           BEQL    6$                                       ;
                       0088    8F  BB 00095           PUSHR   #^M<R3,R7>                               ; 3900
```

```
                                    58  DD 00099         PUSHL   R8                                  ;
                    65              03  FB 0009B         CALLS   #3, FILE_ERROR                      ;
                                  0084  31 0009E  6$:    BRW     12$                                 ; 3879
          59        00D2  C4        03  E0 000A1  7$:    BBS     #3, COM_FLAGS, 10$                  ; 3908
                    50    06C8      C4  D0 000A7         MOVL    INPUT_MTL, R0                       ; 3909
                    50      31      A0  E9 000AC         BLBC    49(R0), 10$                         ;
              10    AE  00270006    8F  D0 000B0         MOVL    #2555910, ATT_CONTROL               ; 3910
              14    AE      08      AE  9E 000B8         MOVAB   FILE_ID, ATT_CONTROL+4              ; 3912
                            18      AE  D4 000BD         CLRL    ATT_CONTROL+8                       ; 3913
                                    7E  D4 000C0         CLRL    -(SP)                               ; 3918
                            14      AE  9F 000C2         PUSHAB  ATT_CONTROL                         ;
                                    7E  7C 000C5         CLRQ    -(SP)                               ;
                                    7E  7C 000C7         CLRQ    -(SP)                               ;
                                    7E  7C 000C9         CLRQ    -(SP)                               ;
                            3C      AE  9F 000CB         PUSHAB  IO_STATUS                           ;
                                    32  DD 000CE         PUSHL   #50                                 ;
                            08      A4  DD 000D0         PUSHL   RWSV_CHAN                           ;
                                    7E  D4 000D3         CLRL    -(SP)                               ;
         00000000G  00              0C  FB 000D5         CALLS   #12, STA_QIOW                       ;
                    52              50  D0 000DC         MOVL    R0, STATUS                          ;
                    07              52  E9 000DF         BLBC    STATUS, 8$                          ; 3919
                    52      1C      AE  3C 000E2         MOVZWL  IO_STATUS, STATUS                   ;
                    0D              52  E8 000E6         BLBS    STATUS, 9$                          ; 3920
                                    52  DD 000E9  8$:    PUSHL   STATUS                              ;
                                    53  DD 000EB         PUSHL   FAB                                 ;
                        00000000G   8F  DD 000ED         PUSHL   #BACKUP$_OPENIN                     ;
                    65              03  FB 000F3         CALLS   #3, FILE_ERROR                      ;
                            08      AE  B5 000F6  9$:    TSTW    FILE_ID                             ; 3921
                                    08  12 000F9         BNEQ    11$                                 ;
                            0C      AE  B5 000FB         TSTW    FILE_ID+4                           ; 3922
                                    03  12 000FE         BNEQ    11$                                 ;
                                    50  D4 00100  10$:   CLRL    R0                                  ; 3923
                                        04 00102         RET                                         ;
                    50    06C8      C4  D0 00103  11$:   MOVL    INPUT_MTL, R0                       ; 3924
                    19      31      A0  E9 00108         BLBC    49(R0), 12$                         ;
                    50      F8      A4  3C 0010C         MOVZWL  RWSV_VOL_NUMBER, R0                 ; 3925
                                    50  D6 00110         INCL    R0                                  ;
   50      0C   AE  08              00  ED 00112         CMPZV   #0, #8, FILE_ID+4, R0               ;
                                    0B  13 00118         BEQL    12$                                 ;
                                    53  DD 0011A         PUSHL   FAB                                 ; 3926
                        00000000G   8F  DD 0011C         PUSHL   #BACKUP$_INVFILEXT                  ;
                    65              02  FB 00122         CALLS   #2, FILE_ERROR                      ;
                                    01  DD 00125  12$:   PUSHL   #1                                  ; 3933
              FB4F  CF              01  FB 00127         CALLS   #1, FIN_IN_SAVE                     ;
                            FA      A4  B6 0012C         INCW    RWSV_SEG_NUMBER                     ; 3934
                    74              94  D0 0012F         MOVL    @RWSV_SAVE_QUAL, RWSV_SAVE_QUAL     ; 3939
                                    04  12 00132         BNEQ    13$                                 ; 3940
                    64      58      A4  D0 00134         MOVL    QUAL, RWSV_SAVE_QUAL                ;
                    50              64  D0 00138  13$:   MOVL    RWSV_SAVE_QUAL, R0                  ; 3941
                    53      04      A0  D0 0013B         MOVL    4(R0), FAB                          ;
              04    A4              53  D0 0013F         MOVL    FAB, RWSV_SAVE_FAB                  ;
          63  40    A3              05  E1 00143         BBC     #5, 64(FAB), 18$                    ; 3943
                            F8      A4  B6 00148         INCW    RWSV_VOL_NUMBER                     ; 3946
              06    A3    0102      8F  A8 0014B         BISW2   #258, 6(FAB)                        ; 3947
                    0B      43      A3  E8 00151         BLBS    67(FAB), 14$                        ; 3953
                                    53  DD 00155         PUSHL   FAB                                 ; 3954
         00000000G  00              01  FB 00157         CALLS   #1, SYS$OPEN                        ;
```

```
                                20   11 0015E          BRB     15$                                    ;
             50         28      A3   D0 00160 14$:     MOVL    40(FAB), R0                            ; 3957
             6E         39      A0   9A 00164          MOVZBL  57(R0), DEVICE_DESC                    ;
        04   AE         44      A0   D0 00168          MOVL    68(R0), DEVICE_DESC+4                  ; 3958
                                7E   7C 0016D          CLRQ    -(SP)                                  ; 3960
                        0C      A3   9F 0016F          PUSHAB  12(FAB)                                ;
                        0C      AE   9F 00172          PUSHAB  DEVICE_DESC                            ;
 00000000G   00                 04   FB 00175          CALLS   #4, SYS$ASSIGN                         ;
        08   A3                 50   D0 0017C          MOVL    R0, 8(FAB)                             ;
             52                 50   D0 00180 15$:     MOVL    R0, STATUS                             ; 3959
             0F                 52   E8 00183          BLBS    STATUS, 16$                            ; 3962
             7E         08      A3   7D 00186          MOVQ    8(FAB), -(SP)                          ; 3964
                                53   DD 0018A          PUSHL   FAB                                    ; 3963
                  00000000G     8F   DD 0018C          PUSHL   #BACKUP$_OPENIN+4                      ;
             65                 04   FB 00192          CALLS   #4, FILE_ERROR                         ;
        08   A4         0C      A3   D0 00195 16$:     MOVL    12(FAB), RWSV_CHAN                     ; 3965
      FE41   CF                 00   FB 0019A 17$:     CALLS   #0, TRY_NEXT_VOLUME                    ; 3967
             4C                 50   E8 0019F          BLBS    R0, 21$                                ;
 00000000G   00                 00   FB 001A2          CALLS   #0, UNLOAD                             ; 3968
                                EF   11 001A9          BRB     17$                                    ;
        F8   A4         0C      AE   9B 001AB 18$:     MOVZBW  FILE_ID+4, RWSV_VOL_NUMBER             ; 3973
                        08      AE   9F 001B0 19$:     PUSHAB  FILE_ID                                ; 3974
      FE28   CF                 01   FB 001B3          CALLS   #1, TRY_NEXT_VOLUME                    ;
             33                 50   E8 001B8          BLBS    R0, 21$                                ;
             50       06C8      C4   D0 001BB          MOVL    INPUT_MTL, R0                          ; 3977
             0B         31      A0   E8 001C0          BLBS    49(R0), 20$                            ;
                                53   DD 001C4          PUSHL   FAB                                    ; 3978
                  00000000G     8F   DD 001C6          PUSHL   #BACKUP$_READERR+4                     ;
             65                 02   FB 001CC          CALLS   #2, FILE_ERROR                         ;
                                7E   7C 001CF 20$:     CLRQ    -(SP)                                  ; 3981
                                7E   7C 001D1          CLRQ    -(SP)                                  ;
                                7E   7C 001D3          CLRQ    -(SP)                                  ;
                                7E   7C 001D5          CLRQ    -(SP)                                  ;
             7E                 01   7D 001D7          MOVQ    #1, -(SP)                              ;
             50       06D4      C4   D0 001DA          MOVL    CURRENT_VCB, R0                        ;
             7E         08      A0   3C 001DF          MOVZWL  8(R0), -(SP)                           ;
                                7E   D4 001E3          CLRL    -(SP)                                  ;
 00000000G   00                 0C   FB 001E5          CALLS   #12, SYS$QIOW                          ;
                                C2   11 001EC          BRB     19$                                    ; 3974
        14   A4         10      A4   D0 001EE 21$:     MOVL    RWSV_IN_SEQ, RWSV_IN_SEQ_0             ; 3988
        38   A4         34      A4   D0 001F3          MOVL    RWSV_IN_VBN, RWSV_IN_VBN_0             ; 3989
                                01   DD 001F8          PUSHL   #1                                     ; 3990
      EE22   CF                 01   FB 001FA          CALLS   #1, READ_SEQ_BLOCK                     ;
                                     04 001FF          RET                                            ; 3992

; Routine Size:  512 bytes,    Routine Base:  CODE + 12C5


; 2453          3993   1
; 2454          3994   1 END
; 2455          3995   0 ELUDOM



                                                       .EXTRN  LIB$SIGNAL
```

```
;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  COMMON                          2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                            5317  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       114         0      1000         00:01.9


; Information:    1
; Warnings:       0
; Errors:         0


;                               COMMAND QUALIFIERS

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

; Size:           5317 code + 2124 data bytes
; Run Time:          01:34.5
; Elapsed Time:      04:59.9
; Lines/CPU Min:       2536
; Lexemes/CPU-Min: 24401
; Memory Used:  460 pages
; Compilation Complete
```

0012 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RESTART
LIS

MAIN
LIS

READSAVE
LIS

LISTJOUR
LIS

OTHERMSG
LIS

MATCH
LIS