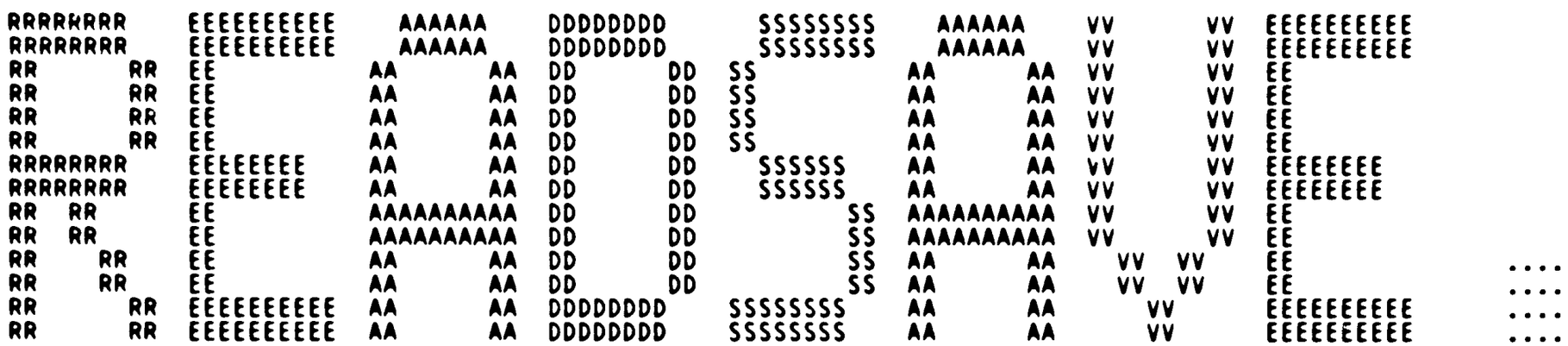
```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         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  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
```


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
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 AA AA DD DD SS AA AA VV VV EE
RR RR EEEEEEEEEE AA AA DDDDDDDD SSSSSSSS AA AA VV EEEEEEEEEE
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1        0001  0 MODULE READSAVE (%TITLE 'Read Save Set'
;     2        0002  0                  IDENT = 'V04-001'
;     3        0003  0                  ) =
;     4        0004  1 BEGIN
;     5        0005  1
;     6        0006  1 !
;     7        0007  1 !**************************************************************************
;     8        0008  1 !*                                                                        *
;     9        0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    10        0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    11        0011  1 !*  ALL RIGHTS RESERVED.                                                   *
;    12        0012  1 !*                                                                        *
;    13        0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14        0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15        0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16        0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17        0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18        0018  1 !*  TRANSFERRED.                                                           *
;    19        0019  1 !*                                                                        *
;    20        0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21        0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22        0022  1 !*  CORPORATION.                                                           *
;    23        0023  1 !*                                                                        *
;    24        0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25        0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    26        0026  1 !*                                                                        *
;    27        0027  1 !*                                                                        *
;    28        0028  1 !**************************************************************************
;    29        0029  1 !
;    30        0030  1
;    31        0031  1 !++
;    32        0032  1 ! FACILITY:
;    33        0033  1 !       Backup/Restore
;    34        0034  1 !
;    35        0035  1 ! ABSTRACT:
;    36        0036  1 !
;    37        0037  1 !       This module contains routines to do the I/O involved in reading
;    38        0038  1 !       save sets.
;    39        0039  1 !
;    40        0040  1 ! ENVIRONMENT:
;    41        0041  1 !
;    42        0042  1 !       VAX/VMS user mode
;    43        0043  1 !
;    44        0044  1 !--
;    45        0045  1 !
;    46        0046  1 ! AUTHOR:  Andrew C. Goldstein, CREATION DATE:  2-Sep-1980  19:17
;    47        0047  1 !
;    48        0048  1 ! MODIFIED BY:
;    49        0049  1 !
;    50        0050  1 !       V04-001 LY0527          Larry Yetto             5-SEP-1984 11:12
;    51        0051  1 !               Correct back space handling in INIT_SAVE_TAPE and FIN_IN_SAVE so that
;    52        0052  1 !               they are consistant with the modifications previously done in
;    53        0053  1 !               WRITESAVE.  We now backspace only 2 tape marks instead of three
;    54        0054  1 !               at the end of a save set.
;    55        0055  1 !
;    56        0056  1 !       V03-014 LMP0272         L. Mark Pilant,         6-Jul-1984  8:48
;    57        0057  1 !               Modify BACKUP to always use a full FIB.
```

```
;    58          0058  1 ;
;    59          0059  1 ;       V03-013 LMP0204         L. Mark Pilant,         7-Mar-1984  9:15
;    60          0060  1 ;               Correctly handle multi-reel save sets during a wildcard
;    61          0061  1 ;               restore or list.
;    62          0062  1 ;
;    63          0063  1 ;       V03-011 ACG0370         Andrew C. Goldstein,    8-Nov-1983  17:14
;    64          0064  1 ;               Fix block skip computation in tape repositioning
;    65          0065  1 ;
;    66          0066  1 ;       V03-010 ACG0332         Andrew C. Goldstein,    2-May-1983  13:39
;    67          0067  1 ;               Remove .B32 from BACKDEF require file
;    68          0068  1 ;
;    69          0069  1 ;       V03-009 ACG0328         Andrew C. Goldstein,    11-Apr-1983  16:06
;    70          0070  1 ;               Improve repositioning over bad tape
;    71          0071  1 ;
;    72          0072  1 ;       V03-008 ACG0313         Andrew C. Goldstein,    10-Feb-1983  21:40
;    73          0073  1 ;               Add another layer of routine around READY_NEXT_VOLUME
;    74          0074  1 ;               so that miscellaneous callers of it don't get the version
;    75          0075  1 ;               with the handler.
;    76          0076  1 ;
;    77          0077  1 ;       V03-007 MLJ0104         Martin L. Jack, 31-Jan-1983  4:18
;    78          0078  1 ;               Lower the severity of errors that occur during mounting
;    79          0079  1 ;               continuation volumes to informational so that they do not
;    80          0080  1 ;               affect the image return status.
;    81          0081  1 ;
;    82          0082  1 ;       V03-006 LMP0040         L. Mark Pilant,         20-Jul-1982  16:50
;    83          0083  1 ;               Correct a problem with the linkage to FMG$MATCH_NAME in the
;    84          0084  1 ;               routine MATCH_SSNAME.
;    85          0085  1 ;
;    86          0086  1 ;       V03-005 LMP0032         L. Mark Pilant,         21-Jun-1982  12:20
;    87          0087  1 ;               Add support for wildcard save set names on a LIST or
;    88          0088  1 ;               RESTORE operation.
;    89          0089  1 ;
;    90          0090  1 ;       V03-004 ACG0293         Andrew C. Goldstein,    7-Jun-1982  15:49
;    91          0091  1 ;               Fix handling of backspacing into BOT on verify pass
;    92          0092  1 ;
;    93          0093  1 ;       V03-003 ACG0280         Andrew C. Goldstein,    2-Apr-1982  16:48
;    94          0094  1 ;               Add dot to save set name with null type
;    95          0095  1 ;
;    96          0096  1 ;       V03-002 ACG0277         Andrew C. Goldstein,    30-Mar-1982  15:30
;    97          0097  1 ;               Check for spurious change in save set name
;    98          0098  1 ;
;    99          0099  1 ;       V03-001 ACG0276         Andrew C. Goldstein,    26-Mar-1982  18:23
;   100          0100  1 ;               Allow XOR groups to be one larger than indicated,
;   101          0101  1 ;               reject non-BACKUP blocks quietly
;   102          0102  1 ;
;   103          0103  1 ;       V02-012 ACG0235         Andrew C. Goldstein,    8-Dec-1981  21:35
;   104          0104  1 ;               Backspace tape over small save sets for verify
;   105          0105  1 ;
;   106          0106  1 ;       V02-011 MLJ0054         Martin L. Jack, 31-Oct-1981  15:03
;   107          0107  1 ;               Implement network save sets.  Move STAACP globals to common.
;   108          0108  1 ;
;   109          0109  1 ;       V02-010 ACG0217         Andrew C. Goldstein,    10-Sep-1981  18:01
;   110          0110  1 ;               Validate tape block size in reading save sets
;   111          0111  1 ;
;   112          0112  1 ;       V02-009 ACG0211         Andrew C. Goldstein,    29-Jul-1981  16:52
;   113          0113  1 ;               Implement sequential disk save sets
;   114          0114  1 ;
```

```
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
```

```
;  184          1737  1 %SBTTL 'READ_BLOCK - read a save set block'
;  185          1738  1 ROUTINE READ_BLOCK (READ_AHEAD, CRC_CHECK) =
;  186          1739  1
;  187          1740  1 !++
;  188          1741  1 !
;  189          1742  1 ! FUNCTIONAL DESCRIPTION:
;  190          1743  1 !
;  191          1744  1 !       This routine reads a block from the input save set.
;  192          1745  1 !
;  193          1746  1 ! CALLING SEQUENCE:
;  194          1747  1 !       READ_BLOCK (READ_AHEAD)
;  195          1748  1 !
;  196          1749  1 ! INPUT PARAMETERS:
;  197          1750  1 !       READ_AHEAD: TRUE to enable asynchronous read ahead
;  198          1751  1 !                   FALSE to read just one block
;  199          1752  1 !       CRC_CHECK: TRUE to check block CRC's, etc.
;  200          1753  1 !                  FALSE to accept any block
;  201          1754  1 !
;  202          1755  1 ! IMPLICIT INPUTS:
;  203          1756  1 !       NONE
;  204          1757  1 !
;  205          1758  1 ! OUTPUT PARAMETERS:
;  206          1759  1 !       NONE
;  207          1760  1 !
;  208          1761  1 ! IMPLICIT OUTPUTS:
;  209          1762  1 !       NONE
;  210          1763  1 !
;  211          1764  1 ! ROUTINE VALUE:
;  212          1765  1 !       address of BCB of buffer read
;  213          1766  1 !
;  214          1767  1 ! SIDE EFFECTS:
;  215          1768  1 !       NONE
;  216          1769  1 !
;  217          1770  1 !--
;  218          1771  1
;  219          1772  2 BEGIN
;  220          1773  2
;  221          1774  2 BUILTIN
;  222          1775  2         INSQUE,
;  223          1776  2         REMQUE,
;  224          1777  2         TESTBITCS,
;  225          1778  2         CRC;                            ! compute CRC instruction
;  226          1779  2
;  227          1780  2 LOCAL
;  228          1781  2         STATUS,                         ! general status value    -
;  229          1782  2         P               : REF BBLOCK,   ! pointer to chase BCB list
;  230          1783  2         BCB             : REF BBLOCK,   ! buffer control block
;  231          1784  2         BUFFER          : REF BBLOCK,   ! the I/O buffer itself
;  232          1785  2         RAB             : REF BBLOCK,   ! input RAB
;  233          1786  2         RECP            : REF BBLOCK,   ! pointer to end of records
;  234          1787  2         BLOCK_CRC,                      ! CRC of data block
;  235          1788  2         HDR_CRC,                        ! CRC of block header
;  236          1789  2         CHECKSUM;                       ! output of CRC instruction
;  237          1790  2
;  238          1791  2 EXTERNAL ROUTINE
;  239          1792  2         FILE_ERROR      : NOVALUE,      ! signal file-related error
;  240          1793  2         GET_BUFFER,                     ! get an I/O buffer
```

```
;  241          1794  2          FREE_BUFFER,                    ! free an I/O buffer
;  242          1795  2          WAIT;                           ! wait for I/O completion
;  243          1796  2
;  244          1797  2 ! If reading from a file, allocate a buffer and read a block.
;  245          1798  2 !
;  246          1799  2
;  247          1800  2 IF .QUAL[QUAL_SS_FILE]
;  248          1801  2 THEN
;  249          1802  3     BEGIN
;  250          1803  3     IF NOT REMQUE (.INPUT_WAIT[0], BCB)
;  251          1804  3     THEN
;  252          1805  3         WAIT (.BCB)
;  253          1806  3     ELSE
;  254          1807  4         BEGIN
;  255          1808  4         BCB = GET_BUFFER ();
;  256          1809  4
;  257          1810  4         RAB = RWSV_SAVE_FAB[FC_RAB];
;  258          1811  4         RAB[RAB$W_USZ] = .BCB[BCB_SIZE];
;  259          1812  4         RAB[RAB$L_UBF] = .BCB[BCB_BUFFER];
;  260          1813  4         BCB[BCB_STATUS] = TRUE;
;  261          1814  4         IF .RWSV_SAVE_FAB[FAB$V_BIO]
;  262          1815  4         THEN
;  263          1816  5             BEGIN
;  264          1817  5             STATUS = $READ (RAB = .RAB);
;  265          1818  5             RAB[RAB$L_BKT] = 0;
;  266          1819  5             RWSV_SAVE_FAB[FAB$V_SQO] = TRUE;
;  267          1820  5             END
;  268          1821  4         ELSE
;  269          1822  4             STATUS = $GET (RAB = .RAB);
;  270          1823  4         BCB[BCB_IO_BCOUNT] = .RAB[RAB$W_RSZ];
;  271          1824  4         IF NOT .STATUS
;  272          1825  4         THEN
;  273          1826  5             BEGIN
;  274          1827  5             IF .STATUS EQL RMS$_EOF
;  275          1828  5             THEN
;  276          1829  5                 BCB[BCB_STATUS] = SS$_ENDOFFILE
;  277          1830  5             ELSE
;  278          1831  6                 BEGIN
;  279          1832  6                 BCB[BCB_STATUS] = .RAB[RAB$L_STS];
;  280          1833  6                 BCB[BCB_STATUS2] = .RAB[RAB$L_STV];
;  281          1834  5                 END;
;  282          1835  4             END;
;  283          1836  3         END;
;  284          1837  3     END
;  285          1838  3
;  286          1839  3 ! If we are reading from tape or sequential disk, grab I/O buffers
;  287          1840  3 ! and issue read ahead QIO's until half the buffer pool is used. Then
;  288          1841  3 ! wait for completion on the first buffer.
;  289          1842  3 !
;  290          1843  3
;  291          1844  2 ELSE
;  292          1845  3     BEGIN
;  293          1846  3     P = .INPUT_WAIT[0];
;  294          1847  4     DECR J FROM (
;  295          1848  4         IF .READ_AHEAD
;  296          1849  4         THEN (.COM_BUFF_COUNT+1) / 2
;  297          1850  4         ELSE 1
```

```
;  298          1851  3          ) TO 1
;  299          1852  3      DO
;  300          1853  4          BEGIN
;  301          1854  4          IF .P NEQ INPUT_WAIT[0]
;  302          1855  4          THEN P = ..P
;  303          1856  4          ELSE
;  304          1857  5              BEGIN
;  305          1858  5              BCB = GET_BUFFER ();
;  306          1859  5              CH$FILL (0, BBH$K_LENGTH, .BCB[BCB_BUFFER]);
;  307          1860  5
;  308          1861  5              IF .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
;  309          1862  5              THEN
;  310          1863  6                  BEGIN
;  311        P 1864  6                  STATUS = $QIO  (CHAN = .RWSV_CHAN,
;  312        P 1865  6                                  FUNC = IO$_READLBLK,
;  313        P 1866  6                                  EFN  = BCB_S_READ,
;  314        P 1867  6                                  IOSB = BCB[BCB_IOSB],
;  315        P 1868  6                                  P1   = .BCB[BCB_BUFFER],
;  316        P 1869  6                                  P2   = .BCB[BCB_SIZE]
;  317          1870  6                                  );
;  318          1871  6                  END
;  319          1872  6
;  320          1873  5              ELSE
;  321          1874  6                  BEGIN
;  322          1875  6                  BCB[BCB_BLOCKNUM] = .RWSV_IN_VBN;
;  323          1876  6                  IF .RWSV_IN_VBN GEQU .RWSV_EOF
;  324          1877  6                  THEN
;  325          1878  7                      BEGIN
;  326          1879  7                      BCB[BCB_STATUS] = SS$_ENDOFFILE;
;  327          1880  7                      STATUS = 1;
;  328          1881  7                      END
;  329          1882  6                  ELSE
;  330          1883  7                      BEGIN
;  331        P 1884  7                      STATUS = S$QIO (CHAN = .RWSV_CHAN,
;  332        P 1885  7                                      FUNC = IO$_READVBLK,
;  333        P 1886  7                                      EFN  = BCB_S_READ,
;  334        P 1887  7                                      IOSB = BCB[BCB_IOSB],
;  335        P 1888  7                                      P1   = .BCB[BCB_BUFFER],
;  336        P 1889  7                                      P2   = .BCB[BCB_SIZE],
;  337        P 1890  7                                      P3   = .RWSV_IN_VBN
;  338          1891  7                                      );
;  339          1892  7                      RWSV_IN_VBN = .RWSV_IN_VBN + (.BCB[BCB_SIZE]+511) / 512;
;  340          1893  6                      END;
;  341          1894  5                  END;
;  342          1895  5              IF NOT .STATUS
;  343          1896  5              THEN
;  344          1897  5                  FILE_ERROR(
;  345          1898  5                      BACKUP$_READERR + STS$K_SEVERE,
;  346          1899  5                      .RWSV_SAVE_FAB,
;  347          1900  5                      .STATUS);
;  348          1901  5
;  349          1902  5              BCB[BCB_SUCC_ACT] = 0;
;  350          1903  5              BCB[BCB_FAIL_ACT] = 0;
;  351          1904  5              BCB[BCB_STATE] = BCB_S_READ;
;  352          1905  5              INSQUE (.BCB, .INPUT_WAIT[1]);
;  353          1906  4              END;
;  354          1907  3      END;
```

```
;  355          1908  3
;  356          1909  3      REMQUE (.INPUT_WAIT[0], BCB);
;  357          1910  3      WAIT (.BCB);
;  358          1911  3      BCB[BCB_STATUS2] = 0;
;  359          1912  2      END;
;  360          1913  2
;  361          1914  2  ! Do basic validation checks on the buffer. It should be of the expected
;  362          1915  2  ! length, and the CRC's, if present, should check.
;  363          1916  2  !
;  364          1917  2
;  365          1918  2  IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFTAPE
;  366          1919  2  THEN BCB[BCB_IO_STATUS] = TRUE;
;  367          1920  2
;  368          1921  2  BUFFER = .BCB[BCB_BUFFER];
;  369          1922  2  IF NOT .BCB[BCB_IO_STATUS]
;  370          1923  2  THEN
;  371          1924  3      BEGIN
;  372          1925  3      IF NOT .QUAL[QUAL_SS_FILE]
;  373          1926  3      THEN BCB[BCB_IO_BCOUNT] = 0;
;  374          1927  3      IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
;  375          1928  3      THEN RETURN .BCB;
;  376          1929  3      IF .RWSV_IN_ORGERR[0]
;  377          1930  3      THEN
;  378          1931  4          BEGIN
;  379          1932  4          RWSV_IN_ORGERR[0] = .BCB[BCB_STATUS];
;  380          1933  4          RWSV_IN_ORGERR[1] = 0;
;  381          1934  3          END;
;  382          1935  3      END
;  383          1936  3
;  384          1937  2  ELSE IF .CRC_CHECK
;  385          1938  2  THEN
;  386          1939  3      BEGIN
;  387          1940  3      IF .BCB[BCB_IO_BCOUNT] LSSU .BCB[BCB_SIZE]
;  388          1941  3      THEN
;  389          1942  4          BEGIN
;  390          1943  4          CH$FILL (0, .BCB[BCB_SIZE]-.BCB[BCB_IO_BCOUNT], .BUFFER+.BCB[BCB_IO_BCOUNT]);
;  391          1944  4          BCB[BCB_STATUS] = BACKUP$_SHORTBLOCK;
;  392          1945  3          END;
;  393          1946  2      END;
;  394          1947  2
;  395          1948  2  IF .CRC_CHECK
;  396          1949  2  THEN
;  397          1950  3      BEGIN
;  398          1951  3      BLOCK_CRC = .BUFFER[BBH$L_CRC];
;  399          1952  3      HDR_CRC = .BUFFER[BBH$W_CHECKSUM];
;  400          1953  3      BUFFER[BBH$L_CRC] = 0;
;  401          1954  3      BUFFER[BBH$W_CHECKSUM] = 0;
;  402          1955  3
;  403          1956  3      CRC (RWSV_CRC16, %REF (0), %REF (BBH$K_LENGTH), .BUFFER, CHECKSUM);
;  404          1957  3      IF .HDR_CRC NEQ .CHECKSUM
;  405          1958  3      THEN BCB[BCB_STATUS] = BACKUP$_HDRCRC;
;  406          1959  3
;  407          1960  4      IF (
;  408          1961  5          (.QUAL[QUAL_CRC] OR (.COM_FLAGS[COM_VERIFYING] AND .QUAL[QUAL_OSAV]))
;  409          1962  4          AND NOT .BUFFER[BBH$V_NOCRC]
;  410          1963  4          AND .BCB[BCB_STATUS]
;  411          1964  4          AND
```

```
;  412          1965  5              BEGIN
;  413          1966  5              CRC (RWSV_AUTODIN, %REF (-1), BCB[BCB_SIZE], .BUFFER, CHECKSUM);
;  414          1967  6              .BLOCK_CRC NEQ (NOT .CHECKSUM)
;  415          1968  5              END
;  416          1969  4          )
;  417          1970  3      THEN BCB[BCB_STATUS] = BACKUP$_BLOCKCRC;
;  418          1971  3
;  419          1972  3      BUFFER[BBH$L_CRC] = .BLOCK_CRC;
;  420          1973  3      BUFFER[BBH$W_CHECKSUM] = .ADR_CRC;
;  421          1974  3
;  422          1975  3      IF NOT .BCB[BCB_STATUS]
;  423          1976  3      AND .RWSV_IN_ORGERR[0]
;  424          1977  3      THEN
;  425          1978  4          BEGIN
;  426          1979  4          RWSV_IN_ORGERR[0] = .BCB[BCB_STATUS];
;  427          1980  4          RWSV_IN_ORGERR[1] = 0;
;  428          1981  3          END;
;  429          1982  2      END;
;  430          1983  2
;  431          1984  2  ! If this block read with an error, count it. If there are too many
;  432          1985  2  ! consecutive errors, complain. Medium offline is handled specially, since
;  433          1986  2  ! operator help is clearly necessary.
;  434          1987  2  !
;  435          1988  2
;  436          1989  2  IF NOT .BCB[BCB_STATUS]
;  437          1990  2  THEN
;  438          1991  3      BEGIN
;  439          1992  3      RWSV_SEQ_ERRORS = .RWSV_SEQ_ERRORS + 1;
;  440          1993  3      IF .BCB[BCB_IO_STATUS] EQL SS$_MEDOFL
;  441          1994  3      OR .BCB[BCB_IO_STATUS] EQL SS$_VOLINV
;  442          1995  3      THEN
;  443          1996  4          BEGIN
;  444          1997  4          INSQUE (.BCB, RWSV_HOLD_LIST[0]);
;  445          1998  4
;  446          1999  4  ! Other pending reads are also likely to fail with the same error. Clean
;  447          2000  4  ! them out.
;  448          2001  4  !
;  449          2002  4
;  450          2003  4          UNTIL REMQUE (.INPUT_WAIT[0], BCB)
;  451          2004  4          DO
;  452          2005  5              BEGIN
;  453          2006  5              WAIT (.BCB);
;  454          2007  5              IF  .BCB[BCB_IO_STATUS] NEQ SS$_MEDOFL
;  455          2008  5              AND .BCB[BCB_IO_STATUS] NEQ SS$_VOLINV
;  456          2009  5              THEN
;  457          2010  6                  BEGIN
;  458          2011  6                  INSQUE (.BCB, INPUT_WAIT[0]);
;  459          2012  6                  BCB[BCB_STATE] = BCB_S_READ;
;  460          2013  6                  EXITLOOP;
;  461          2014  6                  END
;  462          2015  5              ELSE
;  463          2016  5                  FREE_BUFFER (.BCB);
;  464          2017  4              END;
;  465          2018  4          FILE_ERROR (BACKUP$_FATALERR, .RWSV_SAVE_FAB,
;  466          2019  4                      .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
;  467          2020  4          REMQUE (.RWSV_HOLD_LIST[0], BCB);
;  468          2021  4          END
```

```
;  469          2022  4
;  470          2023  4     ELSE IF (
;  471          2024  4         IF .RWSV_SEQ_ERRORS GTRU 100
;  472          2025  4         THEN TESTBITCS (COM_FLAGS[COM_CONTINUE])
;  473          2026  4         ELSE FALSE)
;  474          2027  3     THEN
;  475          2028  4         BEGIN
;  476          2029  4         INSQUE (.BCB, RWSV_HOLD_LIST[0]);
;  477          2030  4         FILE_ERROR (BACKUP$_READERRS, .RWSV_SAVE_FAB,
;  478          2031  4                     .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
;  479          2032  4         REMQUE (.RWSV_HOLD_LIST[0], BCB);
;  480          2033  3         END;
;  481          2034  3     END
;  482          2035  3
;  483          2036  2 ELSE
;  484          2037  2     RWSV_SEQ_ERRORS = 0;
;  485          2038  2
;  486          2039  2 .BCB
;  487          2040  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               ;
78       0153    CA            03  E1 00010        BBC     #3, @@AL+15, 6$               ; 1800
                 57      00    BA  0F 00016        REMQUE  @INPUT_WAIT, R0               ; 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
```

```
;  489          2041  1 %SBTTL 'READ_SEQ_BLOCK - read a sequenced save set block'
;  490          2042  1 ROUTINE READ_SEQ_BLOCK (READ_AHEAD) =
;  491          2043  1
;  492          2044  1 !++
;  493          2045  1 !
;  494          2046  1 ! FUNCTIONAL DESCRIPTION:
;  495          2047  1 !
;  496          2048  1 !       This routine reads a save set block and applies sequence number
;  497          2049  1 !       checking and rewrite error recovery.
;  498          2050  1 !
;  499          2051  1 ! CALLING SEQUENCE:
;  500          2052  1 !       READ_SEQ_BLOCK (READ_AHEAD)
;  501          2053  1 !
;  502          2054  1 ! INPUT PARAMETERS:
;  503          2055  1 !       READ_AHEAD: TRUE to enable asynchronous read ahead
;  504          2056  1 !                   FALSE to read only one block
;  505          2057  1 !
;  506          2058  1 ! IMPLICIT INPUTS:
;  507          2059  1 !       NONE
;  508          2060  1 !
;  509          2061  1 ! OUTPUT PARAMETERS:
;  510          2062  1 !       NONE
;  511          2063  1 !
;  512          2064  1 ! IMPLICIT OUTPUTS:
;  513          2065  1 !       NONE
;  514          2066  1 !
;  515          2067  1 ! ROUTINE VALUE:
;  516          2068  1 !       BCB address of block read
;  517          2069  1 !
;  518          2070  1 ! SIDE EFFECTS:
;  519          2071  1 !       NONE
;  520          2072  1 !
;  521          2073  1 !--
;  522          2074  1
;  523          2075  2 BEGIN
;  524          2076  2
;  525          2077  2 BUILTIN
;  526          2078  2         REMQUE;
;  527          2079  2
;  528          2080  2 LOCAL
;  529          2081  2         BCB             : REF BBLOCK,   ! BCB of block read
;  530          2082  2         BUFFER          : REF BBLOCK,   ! address of data block read
;  531          2083  2         BLOCK_COUNT,                    ! count of blocks read forward
;  532          2084  2         BK_BLOCK_COUNT,                 ! count of backup blocks seen
;  533          2085  2         TM_COUNT;                       ! count of tape marks crossed
;  534          2086  2
;  535          2087  2 EXTERNAL ROUTINE
;  536          2088  2         SKIP_RECORD,                    ! skip tape records
;  537          2089  2         SKIP_TM,                        ! skip tape marks
;  538          2090  2         WAIT,                           ! wait for I/O completion
;  539          2091  2         FREE_BUFFER;                    ! free an I/O buffer
;  540          2092  2
;  541          2093  2 ! The outer loop handles re-reading a data block after a forward
;  542          2094  2 ! retry has been tried and found unsuccessful.
;  543          2095  2 !
;  544          2096  2
;  545          2097  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
```

```
;  603          2155  5              BEGIN
;  604          2156  5              TM_COUNT = .TM_COUNT + 1;
;  605          2157  5              EXITLOOP;
;  606          2158  4              END;
;  607          2159  4          IF .TM_COUNT EQL 0
;  608          2160  4          THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
;  609          2161  4          IF .BUFFER[BBH$W_SUBSYS] EQL BACKUP$K_BACKUP
;  610          2162  4          THEN BK_BLOCK_COUNT = .BK_BLOCK_COUNT + 1;
;  611          2163  4          END
;  612          2164  3      UNTIL .BK_BLOCK_COUNT GEQU .RWSV_LOOKAHEAD
;  613          2165  3      AND .BCB[BCB_STATUS] NEQ BACKUP$_HDRCRC
;  614          2166  3      AND .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ GEQU .RWSV_LOOKAHEAD;
;  615          2167  3
;  616          2168  3  ! Failed to find a rewrite of the block. Flush out the input wait list,
;  617          2169  3  ! then backspace the tape to the original block, less one for good measure,
;  618          2170  3  ! and retry the read.
;  619          2171  3  !
;  620          2172  3
;  621          2173  3      UNTIL REMQUE (.INPUT_WAIT[0], BCB)
;  622          2174  3      DO
;  623          2175  4          BEGIN
;  624          2176  4          WAIT (.BCB);
;  625          2177  4          IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
;  626          2178  4          THEN TM_COUNT = .TM_COUNT + 1;
;  627          2179  4          IF .TM_COUNT EQL 0
;  628          2180  4          THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
;  629          2181  4          FREE_BUFFER (.BCB);
;  630          2182  3          END;
;  631          2183  3
;  632          2184  3      IF .TM_COUNT NEQ 0
;  633          2185  3      THEN SKIP_TM (-.TM_COUNT);
;  634          2186  3      IF SKIP_RECORD (-.BLOCK_COUNT - 2) EQL SS$_ENDOFFILE
;  635          2187  3      THEN SKIP_TM (1);
;  636          2188  2      END;                                ! end of outer loop
;  637          2189  2
;  638          2190  2 .BCB
;  639          2191  1 END;                                    ! End of routine READ_SEQ_BLOCK


                                                        .EXTRN  SKIP_RECORD, SKIP_TM

                                             0FFC 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
                                0003  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
                   0000000G  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$:    PUSHAB  2(BLOCK_COUNT)               ; 2186
                             6E         6E  CE 00116         MNEGL   (SP), (SP)                   ;
                   0000000G  00         01  FB 00119         CALLS   #1, SKIP_RECORD              ;
                   00000870  8F         50  D1 00120         CMPL    R0, #2160                    ;
                                        05  12 00127         BNEQ    17$                          ;
                                        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
```

```
;  641          2192  1 %SBTTL 'REPOSITION - reposition save set'
;  642          2193  1 ROUTINE REPOSITION (BLOCK_NEEDED, ADDRESS, CUR_BCB) =
;  643          2194  1
;  644          2195  1 !++
;  645          2196  1 !
;  646          2197  1 ! FUNCTIONAL DESCRIPTION:
;  647          2198  1 !
;  648          2199  1 !       This routine repositions the input save set to the specified
;  649          2200  1 !       sequence number and reads that block.
;  650          2201  1 !
;  651          2202  1 ! CALLING SEQUENCE:
;  652          2203  1 !       REPOSITION (BLOCK_NEEDED, ADDRESS, CUR_BCB)
;  653          2204  1 !
;  654          2205  1 ! INPUT PARAMETERS:
;  655          2206  1 !       BLOCK_NEEDED: sequence number of desired block
;  656          2207  1 !       ADDRESS: address of RFA or VBN, if file or seq disk, respectively
;  657          2208  1 !       CUR_BCB: BCB of last block read
;  658          2209  1 !
;  659          2210  1 ! IMPLICIT INPUTS:
;  660          2211  1 !       NONE
;  661          2212  1 !
;  662          2213  1 ! OUTPUT PARAMETERS:
;  663          2214  1 !       NONE
;  664          2215  1 !
;  665          2216  1 ! IMPLICIT OUTPUTS:
;  666          2217  1 !       NONE
;  667          2218  1 !
;  668          2219  1 ! ROUTINE VALUE:
;  669          2220  1 !       BCB of desired block
;  670          2221  1 !
;  671          2222  1 ! SIDE EFFECTS:
;  672          2223  1 !       NONE
;  673          2224  1 !
;  674          2225  1 !--
;  675          2226  1
;  676          2227  2 BEGIN
;  677          2228  2
;  678          2229  2 BUILTIN
;  679          2230  2         REMQUE;
;  680          2231  2
;  681          2232  2 MAP
;  682          2233  2         CUR_BCB         : REF BBLOCK;   ! input BCB arg
;  683          2234  2
;  684          2235  2 LOCAL
;  685          2236  2         STATUS,                         ! general status value
;  686          2237  2         BOT,                            ! flag indicating tape backspaced into BOT
;  687          2238  2         RAB             : REF BBLOCK,   ! RAB for reading input file
;  688          2239  2         PREV_SEQ,                       ! sequence number of last block read
;  689          2240  2         BLOCK_COUNT,                    ! count of records to backspace
;  690          2241  2         TM_COUNT,                       ! count of EOF marks to backspace
;  691          2242  2         BCB             : REF BBLOCK,   ! current BCB
;  692          2243  2         BUFFER          : REF BBLOCK;   ! current I/O buffer
;  693          2244  2
;  694          2245  2 EXTERNAL ROUTINE
;  695          2246  2         WAIT,                           ! wait for I/O completion
;  696          2247  2         FILE_ERROR,                     ! signal file related error
;  697          2248  2         FREE_BUFFER,                    ! free an I/O buffer
```

```
;  698          2249  2          SKIP_RECORD,                    ! skip tape records
;  699          2250  2          SKIP_TM;                        ! skip tape marks
;  700          2251  2
;  701          2252  2 ! First wait out pending reads, tracking the tape position.
;  702          2253  2 !
;  703          2254  2
;  704          2255  2 BUFFER = .CUR_BCB[BCB_BUFFER];
;  705          2256  2 PREV_SEQ = -1;
;  706          2257  2 BLOCK_COUNT = .BUFFER[BBH$L_NUMBER] - .BLOCK_NEEDED + 1;
;  707          2258  2 TM_COUNT = 0;
;  708          2259  2 IF .CUR_BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
;  709          2260  2 THEN TM_COUNT = .TM_COUNT + 1;
;  710          2261  2 IF .CUR_BCB[BCB_STATE] NEQ BCB_S_IDLE
;  711          2262  2 THEN FREE_BUFFER (.CUR_BCB);
;  712          2263  2
;  713          2264  2 UNTIL REMQUE (.INPUT_WAIT[0], BCB)
;  714          2265  2 DO
;  715          2266  3     BEGIN
;  716          2267  3     WAIT (.BCB);
;  717          2268  3     IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
;  718          2269  3     THEN TM_COUNT = .TM_COUNT + 1;
;  719          2270  3     IF .TM_COUNT EQL 0
;  720          2271  3     THEN BLOCK_COUNT = .BLOCK_COUNT + 1;
;  721          2272  3     FREE_BUFFER (.BCB);
;  722          2273  2     END;
;  723          2274  2
;  724          2275  2 ! Now find the first block of the group by backspacing, reading, and
;  725          2276  2 ! checking the sequence number. This may take several tries, since
;  726          2277  2 ! we don't know whether there are block rewrites to space over.
;  727          2278  2 !
;  728          2279  2
;  729          2280  2 RWSV_IN_SEQ = 0;
;  730          2281  2 WHILE TRUE
;  731          2282  2 DO
;  732          2283  3     BEGIN
;  733          2284  3     IF NOT .QUAL[QUAL_SS_FILE]
;  734          2285  3     THEN
;  735          2286  4         BEGIN
;  736          2287  4         IF .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
;  737          2288  4         THEN
;  738          2289  5             BEGIN
;  739          2290  5             BOT = FALSE;
;  740          2291  5             IF .TM_COUNT NEQ 0
;  741          2292  5             THEN SKIP_TM (-.TM_COUNT);
;  742          2293  5             IF .BLOCK_COUNT GTR 0
;  743          2294  5             THEN IF SKIP_RECORD (-.BLOCK_COUNT) EQL SS$_ENDOFFILE
;  744          2295  5             THEN
;  745          2296  6                 BEGIN
;  746          2297  6                 SKIP_TM (1);
;  747          2298  6                 BOT = TRUE;
;  748          2299  5                 END;
;  749          2300  5             END
;  750          2301  4         ELSE
;  751          2302  4             RWSV_IN_VBN = ..ADDRESS;
;  752          2303  4         END
;  753          2304  3     ELSE
;  754          2305  4         BEGIN
```

```
;  755          2306  4          RAB = RWSV_SAVE_FAB[FC_RAB];
;  756          2307  4          IF .BBLOCK[RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_NET]
;  757          2308  4          THEN
;  758          2309  5              BEGIN
;  759          2310  5              STATUS = $DISCONNECT (RAB = .RAB);
;  760          2311  5              IF NOT .STATUS
;  761          2312  5              THEN
;  762          2313  6                  BEGIN
;  763          2314  6                  FILE_ERROR (BACKUP$_POSITERR, .RWSV_SAVE_FAB,
;  764          2315  6                          .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
;  765          2316  5                  END;
;  766          2317  5              RWSV_SAVE_FAB[FAB$V_SQO] = FALSE;
;  767          2318  5              STATUS = $CONNECT (RAB = .RAB);
;  768          2319  5              IF NOT .STATUS
;  769          2320  5              THEN
;  770          2321  6                  BEGIN
;  771          2322  6                  FILE_ERROR (BACKUP$_POSITERR, .RWSV_SAVE_FAB,
;  772          2323  6                          .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
;  773          2324  5                  END;
;  774          2325  5              RAB[RAB$L_BKT] = ..ADDRESS;
;  775          2326  5              END
;  776          2327  4          ELSE
;  777          2328  5              BEGIN
;  778          2329  5              CH$MOVE (RAB$S_RFA, .ADDRESS, RAB[RAB$W_RFA]);
;  779          2330  5              RAB[RAB$B_RAC] = RAB$C_RFA;
;  780          2331  5              STATUS = $FIND (RAB = .RAB);
;  781          2332  5              IF NOT .STATUS
;  782          2333  5              THEN
;  783          2334  6                  BEGIN
;  784          2335  6                  FILE_ERROR (BACKUP$_POSERROR, .RWSV_SAVE_FAB,
;  785          2336  6                          .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
;  786          2337  6                  RETURN (READ_SEQ_BLOCK (FALSE));
;  787          2338  5                  END;
;  788          2339  5              RAB[RAB$B_RAC] = RAB$C_SEQ;
;  789          2340  4              END;
;  790          2341  3          END;
;  791          2342  3
;  792          2343  3      BCB = READ_SEQ_BLOCK (FALSE);
;  793          2344  3      BUFFER = .BCB[BCB_BUFFER];
;  794          2345  3      TM_COUNT = 0;
;  795          2346  3      IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
;  796          2347  3      THEN
;  797          2348  4          BEGIN
;  798          2349  4          TM_COUNT = 1;
;  799          2350  4          BLOCK_COUNT = .BLOCK_COUNT + 1;
;  800          2351  4          END
;  801          2352  3      ELSE IF .BUFFER[BBH$L_NUMBER] GEQU .PREV_SEQ
;  802          2353  3      THEN BLOCK_COUNT = .BLOCK_COUNT + .BUFFER[BBH$L_NUMBER] - .PREV_SEQ + 1
;  803          2354  3      ELSE BLOCK_COUNT = .BUFFER[BBH$L_NUMBER] - .BLOCK_NEEDED + 1;
;  804          2355  3      PREV_SEQ = .BUFFER[BBH$L_NUMBER];
;  805          2356  3
;  806          2357  3      IF .BUFFER[BBH$L_NUMBER] LEQU .BLOCK_NEEDED
;  807          2358  3      OR .QUAL[QUAL_SS_FILE]
;  808          2359  3      OR NOT .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
;  809          2360  3      OR .BOT
;  810          2361  3      THEN EXITLOOP;
;  811          2362  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
```

```
;  830          2380  1 %SBTTL 'XORCIZE - apply XOR recovery'
;  831          2381  1 ROUTINE XORCIZE (BLOCK_NEEDED, CUR_BCB) =
;  832          2382  1
;  833          2383  1 !++
;  834          2384  1 !
;  835          2385  1 ! FUNCTIONAL DESCRIPTION:
;  836          2386  1 !
;  837          2387  1 !       This routine applies XOR recovery to recover the indicated
;  838          2388  1 !       lost block. This is done by backspacing to the start of the
;  839          2389  1 !       current group and XORing all blocks (except the one in question)
;  840          2390  1 !       up through the next XOR block.
;  841          2391  1 !
;  842          2392  1 ! CALLING SEQUENCE:
;  843          2393  1 !       XORCIZE (BLOCK_NEEDED, CUR_BCB)
;  844          2394  1 !
;  845          2395  1 ! INPUT PARAMETERS:
;  846          2396  1 !       BLOCK_NEEDED: sequence number of block to be recovered
;  847          2397  1 !       CUR_BCB: BCB of block just read
;  848          2398  1 !
;  849          2399  1 ! IMPLICIT INPUTS:
;  850          2400  1 !       NONE
;  851          2401  1 !
;  852          2402  1 ! OUTPUT PARAMETERS:
;  853          2403  1 !       NONE
;  854          2404  1 !
;  855          2405  1 ! IMPLICIT OUTPUTS:
;  856          2406  1 !       NONE
;  857          2407  1 !
;  858          2408  1 ! ROUTINE VALUE:
;  859          2409  1 !       BCB of recovered block
;  860          2410  1 !
;  861          2411  1 ! SIDE EFFECTS:
;  862          2412  1 !       NONE
;  863          2413  1 !
;  864          2414  1 !--
;  865          2415  1
;  866          2416  2 BEGIN
;  867          2417  2
;  868          2418  2 BUILTIN
;  869          2419  2         INSQUE;
;  870          2420  2
;  871          2421  2 MAP
;  872          2422  2         CUR_BCB         : REF BBLOCK;   ! input BCB arg
;  873          2423  2
;  874          2424  2 LOCAL
;  875          2425  2         RAB             : REF BBLOCK,   ! RAB for reading input file
;  876          2426  2         SAVE_ADDR       : BBLOCK [RAB$S_RFA], ! saved file address
;  877          2427  2         SAVE_ERRORS,                    ! save soft error count
;  878          2428  2         P1,                             ! XOR buffer pointer
;  879          2429  2         P2,                             ! XOR buffer pointer
;  880          2430  2         BCB             : REF BBLOCK,   ! BCB of current buffer
;  881          2431  2         XOR_BCB         : REF BBLOCK,   ! BCB of XOR buffer
;  882          2432  2         BUFFER          : REF BBLOCK;   ! current I/O buffer
;  883          2433  2
;  884          2434  2 EXTERNAL ROUTINE
;  885          2435  2         FREE_BUFFER;                    ! free an I/O buffer
;  886          2436  2
```

```
;  887          2437  2 ! Save the current position if the input is a file or seq disk.
;  888          2438  2 !
;  889          2439  2
;  890          2440  2 SAVE_ERRORS = .RWSV_IN_ERRORS;
;  891          2441  2 IF .RWSV_IN_GROUP_SIZE EQL 0 THEN RETURN .CUR_BCB;
;  892          2442  2 SAVE_ADDR = .CUR_BCB[BCB_BLOCKNUM];
;  893          2443  2 IF .QUAL[QUAL_SS_FILE]
;  894          2444  2 THEN
;  895          2445  3     BEGIN
;  896          2446  3     RAB = RWSV_SAVE_FAB[FC_RAB];
;  897          2447  3     CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], SAVE_ADDR);
;  898          2448  2     END;
;  899          2449  2
;  900          2450  2 ! Special case the situation in which the block lost is the first
;  901          2451  2 ! of an XOR group, since no special repositioning is necessary.
;  902          2452  2 !
;  903          2453  2
;  904          2454  2 BUFFER = .CUR_BCB[BCB_BUFFER];
;  905          2455  2 IF .BLOCK_NEEDED EQL .RWSV_IN_XOR_SEQ + 1
;  906          2456  2 THEN
;  907          2457  3     BEGIN
;  908          2458  3     RWSV_IN_SEQ = .BLOCK_NEEDED + 1;
;  909          2459  3     IF .BUFFER[BBH$L_NUMBER] EQL .BLOCK_NEEDED
;  910          2460  3     THEN
;  911          2461  4         BEGIN
;  912          2462  4         FREE_BUFFER (.CUR_BCB);
;  913          2463  4         BCB = READ_SEQ_BLOCK (TRUE);
;  914          2464  4         END
;  915          2465  3     ELSE IF .BUFFER[BBH$L_NUMBER] EQL .RWSV_IN_SEQ
;  916          2466  3     THEN BCB = .CUR_BCB
;  917          2467  3     ELSE RETURN .CUR_BCB;
;  918          2468  3     END
;  919          2469  3
;  920          2470  2 ELSE
;  921          2471  2     BCB = REPOSITION (.RWSV_IN_XOR_SEQ + 1, RWSV_IN_XOR_RFA, .CUR_BCB);
;  922          2472  2 BUFFER = .BCB[BCB_BUFFER];
;  923          2473  2
;  924          2474  2 ! Accumulate the blocks in an XOR buffer, skipping the one we are
;  925          2475  2 ! after. Note that we allow for a group size one larger than specified,
;  926          2476  2 ! to accomodate some boundary conditions on sequential disk.
;  927          2477  2 ! Temporarily decrement the buffer count to account for the XOR buffer
;  928          2478  2 ! we are sitting on, to prevent the read-ahead from eating too many buffers.
;  929          2479  2 !
;  930          2480  2
;  931          2481  2 COM_BUFF_COUNT = .COM_BUFF_COUNT - 1;
;  932          2482  2 XOR_BCB = 0;
;  933          2483  3 IF 1
;  934          2484  3 DECR J FROM MINU (.RWSV_IN_GROUP_SIZE+1, 100) TO 1
;  935          2485  3 DO
;  936          2486  4     BEGIN
;  937          2487  4     IF .XOR_BCB NEQ 0
;  938          2488  4     THEN
;  939          2489  5         BEGIN
;  940          2490  5         BCB = READ_SEQ_BLOCK (TRUE);
;  941          2491  5         BUFFER = .BCB[BCB_BUFFER];
;  942          2492  4         END;
;  943          2493  4
```

```
;  944          2494  4      IF NOT .BCB[BCB_STATUS]
;  945          2495  4      OR .BUFFER[BBH$L_NUMBER] GTRU .RWSV_IN_SEQ
;  946          2496  4      THEN EXITLOOP -1;
;  947          2497  4
;  948          2498  4      IF .XOR_BCB EQL 0
;  949          2499  4      THEN
;  950          2500  4          XOR_BCB = .BCB
;  951          2501  4      ELSE
;  952          2502  5          BEGIN
;  953          2503  5          P1 = .BUFFER + BBH$K_COMMON;
;  954          2504  5          P2 = .XOR_BCB[BCB_BUFFER] + BBH$K_COMMON;
;  955          2505  5          DECR J FROM (.BCB[BCB_SIZE]-BBH$K_COMMON)/16 TO 1
;  956          2506  5          DO
;  957          2507  6              BEGIN
;  958          2508  6              .P2 = ..P2 XOR ..P1;
;  959          2509  6              P1 = .P1 + 4;
;  960          2510  6              P2 = .P2 + 4;
;  961          2511  6              .P2 = ..P2 XOR ..P1;
;  962          2512  6              P1 = .P1 + 4;
;  963          2513  6              P2 = .P2 + 4;
;  964          2514  6              .P2 = ..P2 XOR ..P1;
;  965          2515  6              P1 = .P1 + 4;
;  966          2516  6              P2 = .P2 + 4;
;  967          2517  6              .P2 = ..P2 XOR ..P1;
;  968          2518  6              P1 = .P1 + 4;
;  969          2519  6              P2 = .P2 + 4;
;  970          2520  5              END;
;  971          2521  4          END;
;  972          2522  4
;  973          2523  4      RWSV_IN_SEQ = .BUFFER[BBH$L_NUMBER] + 1;
;  974          2524  4      IF .RWSV_IN_SEQ EQL .BLOCK_NEEDED
;  975          2525  4      THEN RWSV_IN_SEQ = .RWSV_IN_SEQ + 1;
;  976          2526  4      IF .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_XORBLOCK
;  977          2527  4      THEN EXITLOOP 0;
;  978          2528  4      IF .BCB NEQ .XOR_BCB THEN FREE_BUFFER (.BCB);
;  979          2529  4      END
;  980          2530  3 )
;  981          2531  3
;  982          2532  3 ! If XOR recovery was not successful, reposition to the desired block
;  983          2533  3 ! and return it.
;  984          2534  3 !
;  985          2535  3
;  986          2536  2 THEN
;  987          2537  3      BEGIN
;  988          2538  3      IF .XOR_BCB NEQ 0
;  989          2539  3      THEN FREE_BUFFER (.XOR_BCB);
;  990          2540  3      BCB =  REPOSITION (.BLOCK_NEEDED, SAVE_ADDR, .BCB);
;  991          2541  3      RWSV_IN_ERRORS = .SAVE_ERRORS;
;  992          2542  3      COM_BUFF_COUNT = .COM_BUFF_COUNT + 1;
;  993          2543  3      .BCB
;  994          2544  3      END
;  995          2545  3
;  996          2546  3 ! If XOR recovery was successful, reposition to the block following
;  997          2547  3 ! the one recovered, and link it into the FRONT of the input wait
;  998          2548  3 ! list, so it will be read next. Return the recovered block.
;  999          2549  3 !
; 1000          2550  3
```

```
;  1001         2551  2 ELSE
;  1002         2552  2     BEGIN
;  1003         2553  2     IF .BUFFER[BBH$L_NUMBER] NEQ .BLOCK_NEEDED + 1
;  1004         2554  2     THEN BCB = REPOSITION (.BLOCK_NEEDED + 1, SAVE_ADDR, .BCB);
;  1005         2555  2     RWSV_IN_SEQ = .BLOCK_NEEDED;
;  1006         2556  2     IF .BCB NEQ .XOR_BCB
;  1007         2557  2     THEN
;  1008         2558  3         BEGIN
;  1009         2559  3         BCB[BCB_STATE] = BCB_S_READ;
;  1010         2560  3         INSQUE (.BCB, INPUT_WAIT[0]);
;  1011         2561  3         END;
;  1012         2562  2     RWSV_IN_ERRORS = .SAVE_ERRORS - 1;
;  1013         2563  2     RWSV_IN_XORUSE = .RWSV_IN_XORUSE + 1;
;  1014         2564  2     COM_BUFF_COUNT = .COM_BUFF_COUNT + 1;
;  1015         2565  2     BUFFER = .XOR_BCB[BCB_BUFFER];
;  1016         2566  2     BUFFER[BBH$W_APPLIC] = BACKUP$K_DATABLOCK;
;  1017         2567  2     BUFFER[BBH$L_NUMBER] = .BLOCK_NEEDED;
;  1018         2568  2     .XOR_BCB
;  1019         2569  2     END
;  1020         2570  1
;  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$:    PUSHL   BCB                                  ; 2540
                      04      AE  9F 0011C         PUSHAB  SAVE_ADDR                            ;
                              58  DD 0011F         PUSHL   R8                                   ;
          FCEB    CF          03  FB 00121         CALLS   #3, REPOSITION                       ;
                  55          50  D0 00126         MOVL    R0, BCB                              ;
            18    AA          59  B0 00129         MOVW    SAVE_ERRORS, RWSV_IN_ERRORS          ; 2541
                  50          55  D0 0012D         MOVL    BCB, R0                              ; 2543
                              42  11 00130         BRB     22$                                  ;
                  5C  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
```

```
;  1023          2572  1 %SBTTL 'MATCH_SSNAME - match save set name string'
;  1024          2573  1 ROUTINE MATCH_SSNAME (LABEL_BUFFER) =
;  1025          2574  1
;  1026          2575  1 !++
;  1027          2576  1 !
;  1028          2577  1 ! FUNCTIONAL DESCRIPTION:
;  1029          2578  1 !
;  1030          2579  1 !       This routine determines whether the specified tape header label
;  1031          2580  1 !       matches the save set file name in COM_SSNAME.
;  1032          2581  1 !
;  1033          2582  1 ! CALLING SEQUENCE:
;  1034          2583  1 !       MATCH_SSNAME (LABEL_BUFFER)
;  1035          2584  1 !
;  1036          2585  1 ! INPUT PARAMETERS:
;  1037          2586  1 !       LABEL_BUFFER: buffer containing tape header label
;  1038          2587  1 !
;  1039          2588  1 ! IMPLICIT INPUTS:
;  1040          2589  1 !       NONE
;  1041          2590  1 !
;  1042          2591  1 ! OUTPUT PARAMETERS:
;  1043          2592  1 !       NONE
;  1044          2593  1 !
;  1045          2594  1 ! IMPLICIT OUTPUTS:
;  1046          2595  1 !       NONE
;  1047          2596  1 !
;  1048          2597  1 ! ROUTINE VALUE:
;  1049          2598  1 !       TRUE if label matches
;  1050          2599  1 !       FALSE if not
;  1051          2600  1 !
;  1052          2601  1 ! SIDE EFFECTS:
;  1053          2602  1 !       NONE
;  1054          2603  1 !
;  1055          2604  1 !--
;  1056          2605  1
;  1057          2606  2 BEGIN
;  1058          2607  2
;  1059          2608  2 MAP
;  1060          2609  2
;  1061          2610  2 LABEL_BUFFER    : REF BBLOCK;           ! buffer containing tape label
;  1062          2611  2
;  1063          2612  2 LINKAGE
;  1064          2613  2         L_MATCH_NAME    = JSB (REGISTER = 4, REGISTER = 5,
;  1065          2614  2                                REGISTER = 2, REGISTER = 3) :
;  1066          2615  2                                NOPRESERVE (2,3,4,5);
;  1067          2616  2
;  1068          2617  2 LOCAL
;  1069          2618  2         LABEL_LENGTH;                   ! Length of the HDR1 file identifier
;  1070          2619  2
;  1071          2620  2 EXTERNAL ROUTINE
;  1072          2621  2         FMG$MATCH_NAME  : L_MATCH_NAME; ! Compare names with wildcards
;  1073          2622  2
;  1074          2623  2 ! The specified file name matches (1) if it is null or (2) if it matches
;  1075          2624  2 ! exactly or (3) if the specified file name has a trailing dot and the tape
;  1076          2625  2 ! file name has a null type (with no dot).
;  1077          2626  2 !
;  1078          2627  2
;  1079          2628  2 IF
```

```
2           .COM_SSNAME[DSC$W_LENGTH] EQL 0
2       OR
3           (.COM_SSNAME[DSC$W_LENGTH] EQL 1
3            AND .VECTOR[.COM_SSNAME[DSC$A_POINTER], 0; ,BYTE] EQL '.')
2       THEN RETURN TRUE;
2
2       LABEL_LENGTH = HD1$S_FILEID;
2       WHILE .VECTOR[LABEL_BUFFER[HD1$T_FILEID], .LABEL_LENGTH-1; ,BYTE] EQL %C' '
2       AND .LABEL_LENGTH GTR 0
2       DO LABEL_LENGTH = .LABEL_LENGTH - 1;
2
2       IF
2           FMG$MATCH_NAME(.COM_SSNAME[DSC$W_LENGTH],
2                          .COM_SSNAME[DSC$A_POINTER],
2                          .LABEL_LENGTH,
2                          LABEL_BUFFER[HD1$T_FILEID])
3       OR (
3           .VECTOR[.COM_SSNAME[DSC$A_POINTER], .COM_SSNAME[DSC$W_LENGTH]-1; ,BYTE] EQL '.'
3           AND
3           FMG$MATCH_NAME(.COM_SSNAME[DSC$W_LENGTH]-1,
3                          .COM_SSNAME[DSC$A_POINTER],
3                          .LABEL_LENGTH,
3                          LABEL_BUFFER[HD1$T_FILEID])
3           )
2       THEN TRUE
2       ELSE FALSE
2
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 00C50          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,
```

```
;  1166          2714  2          ATTR_END        = ATT_CONTROL+16;
;  1167          2715  2
;  1168          2716  2 EXTERNAL ROUTINE
;  1169          2717  2          READY_DISK,                     ! ready disk for input
;  1170          2718  2          INIT_DIR_SCAN,                  ! initialize directory scan
;  1171          2719  2          FIND_NEXT,                      ! find a file
;  1172          2720  2          FREE_DIR_DATA,                  ! clean up file scan context
;  1173          2721  2          FILE_ERROR;                     ! signal file related error
;  1174          2722  2
;  1175          2723  2
;  1176          2724  2 ! Set up the input disk.
;  1177          2725  2 !
;  1178          2726  2
;  1179          2727  2 FAB = .RWSV_SAVE_FAB;
;  1180          2728  2 FAB[FAB$L_STS] = 1;
;  1181          2729  2 FAB[FAB$L_STV] = STA_IN_CHAN;
;  1182          2730  2 NAM = .FAB[FAB$L_NAM];
;  1183          2731  2 IF NOT .CONTINUE
;  1184          2732  2 THEN
;  1185          2733  3     BEGIN
;  1186          2734  3     READY_DISK (0);
;  1187          2735  3
;  1188          2736  3 ! Use the file scan logic to find the file.
;  1189          2737  3 !
;  1190          2738  3
;  1191          2739  3     INIT_DIR_SCAN (
;  1192          2740  3             STA_IN_CHAN,
;  1193          2741  3             .NAM,
;  1194          2742  3             BBLOCK [.QUAL[QUAL_INPU_LIST], QUAL_DEV_DESC],
;  1195          2743  3             BBLOCK [.QUAL[QUAL_INPU_LIST], QUAL_EXP_DESC],
;  1196          2744  3             FALSE,
;  1197          2745  3             .INPUT_MTL[MTL_RVN_BASE],
;  1198          2746  3             0);
;  1199          2747  3     STATUS = FIND_NEXT ();
;  1200          2748  3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB, SS$_NOSUCHFILE);
;  1201          2749  3     FREE_DIR_DATA ();
;  1202          2750  3
;  1203          2751  3 ! Set up the FIB and issue the QIO to open the file.
;  1204          2752  3 !
;  1205          2753  3
;  1206          2754  3     CH$FILL (0, FIB$C_LENGTH, FIB);
;  1207          2755  3     FIB[FIB$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
;  1208          2756  3     FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
;  1209          2757  3     FIB[FIB$W_FID_RVN] = .NAM[NAM$W_FID_RVN];
;  1210          2758  3     END
;  1211          2759  3
;  1212          2760  3 ! A file continuation is simply opened by file ID.
;  1213          2761  3 !
;  1214          2762  3
;  1215          2763  2 ELSE
;  1216          2764  3     BEGIN
;  1217          2765  3     IF .INPUT_MTL[MTL_SEQ_DISK]
;  1218          2766  3     THEN READY_DISK (0);
;  1219          2767  3     CH$FILL (0, FIB$C_LENGTH, FIB);
;  1220          2768  3     FIB[FIB$W_FID_NUM] = .FILE_ID[FID$W_NUM];
;  1221          2769  3     FIB[FIB$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
;  1222          2770  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

                                                     OFFC 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
```

```
;  1281          2828  1 %SBTTL 'INIT_SAVE_TAPE - initialize save set tape'
;  1282          2829  1 ROUTINE INIT_SAVE_TAPE (VERIFY) : NOVALUE =
;  1283          2830  1
;  1284          2831  1 !++
;  1285          2832  1 !
;  1286          2833  1 ! FUNCTIONAL DESCRIPTION:
;  1287          2834  1 !
;  1288          2835  1 !       This routine sets up the input tape for reading.
;  1289          2836  1 !
;  1290          2837  1 ! CALLING SEQUENCE:
;  1291          2838  1 !       INIT_SAVE_TAPE (VERIFY)
;  1292          2839  1 !
;  1293          2840  1 ! INPUT PARAMETERS:
;  1294          2841  1 !       VERIFY: TRUE to indicate setup for verify pass
;  1295          2842  1 !               FALSE to indicate normal initialization
;  1296          2843  1 !
;  1297          2844  1 ! IMPLICIT INPUTS:
;  1298          2845  1 !       NONE
;  1299          2846  1 !
;  1300          2847  1 ! OUTPUT PARAMETERS:
;  1301          2848  1 !       NONE
;  1302          2849  1 !
;  1303          2850  1 ! IMPLICIT OUTPUTS:
;  1304          2851  1 !       NONE
;  1305          2852  1 !
;  1306          2853  1 ! ROUTINE VALUE:
;  1307          2854  1 !       NONE
;  1308          2855  1 !
;  1309          2856  1 ! SIDE EFFECTS:
;  1310          2857  1 !       NONE
;  1311          2858  1 !
;  1312          2859  1 !--
;  1313          2860  1
;  1314          2861  2 BEGIN
;  1315          2862  2
;  1316          2863  2 LOCAL
;  1317          2864  2         FAB             : REF BBLOCK,   ! pointer to input FAB
;  1318          2865  2         STATUS,                         ! the usual status value
;  1319          2866  2         FILE_SECTION,                   ! file section number
;  1320          2867  2         TAPE_CHAR       : BBLOCK [4],   ! magtape device characteristics
;  1321          2868  2         LABEL_BUFFER    : BBLOCK [90],  ! buffer for tape labels
;  1322          2869  2         IO_STATUS       : VECTOR [4,WORD]; ! I/O status block
;  1323          2870  2
;  1324          2871  2 EXTERNAL ROUTINE
;  1325          2872  2         FILE_ERROR,                     ! signal file related error
;  1326          2873  2         READY_TAPE,                     ! prepare tape for I/O
;  1327          2874  2         SENSE_CHAR,                     ! sense magtape characteristics
;  1328          2875  2         REWIND,                         ! rewind tape
;  1329          2876  2         SKIP_TM,                        ! skip tape marks
;  1330          2877  2         READ_LABEL,                     ! read and check tape label
;  1331          2878  2         LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL);
;  1332          2879  2                                         ! convert decimal to binary
;  1333          2880  2
;  1334          2881  2 ! Set up the input tape. Rewind it if called for; if we are setting up a
;  1335          2882  2 ! small save set written /NOREWIND for verify, backspace over it rather
;  1336          2883  2 ! than rewinding to save time.
;  1337          2884  2 !
```

```
;  1338          2885  2
;  1339          2886  2 FAB = .RWSV_SAVE_FAB;
;  1340          2887  2 TAPE_CHAR = READY_TAPE (FALSE);
;  1341          2888  2 IF .QUAL[QUAL_REWI] AND NOT .INPUT_FLAGS[INPUT_REWOUND]
;  1342          2889  2 THEN
;  1343          2890  3     BEGIN
;  1344          2891  3     REWIND ();
;  1345          2892  3     TAPE_CHAR[MT$V_BOT] = TRUE;
;  1346          2893  3     INPUT_FLAGS[INPUT_REWOUND] = TRUE;
;  1347          2894  3     END
;  1348          2895  3
;  1349          2896  2 ELSE IF .VERIFY
;  1350          2897  2 THEN
;  1351          2898  3     BEGIN
;  1352          2899  3     IF .RWSV_OUT_BLOCK_COUNT LSSU 1000
;  1353          2900  3     THEN
;  1354          2901  4         BEGIN
;  1355          2902  4         SKIP_TM (-3);
;  1356          2903  4         TAPE_CHAR = SENSE_CHAR ();
;  1357          2904  4         END
;  1358          2905  4
;  1359          2906  3     ELSE
;  1360          2907  4         BEGIN
;  1361          2908  4         REWIND ();
;  1362          2909  4         TAPE_CHAR[MT$V_BOT] = TRUE;
;  1363          2910  3         END;
;  1364          2911  2     END;
;  1365          2912  2
;  1366          2913  2 ! If the tape is at BOT, read and verify the volume header label.
;  1367          2914  2 !
;  1368          2915  2
;  1369          2916  2 IF .TAPE_CHAR[MT$V_BOT]
;  1370          2917  2 THEN
;  1371          2918  3     BEGIN
;  1372          2919  3     STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
;  1373          2920  3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1374          2921  3
;  1375          2922  3     WHILE TRUE
;  1376          2923  3     DO
;  1377          2924  4         BEGIN
;  1378          2925  4         STATUS = READ_LABEL (LABEL_BUFFER);
;  1379          2926  4         IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1380          2927  4         IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'HDR1' THEN EXITLOOP;
;  1381          2928  3         END;
;  1382          2929  2     END;
;  1383          2930  2
;  1384          2931  2 ! Search for a HDR1 record that matches the desired save set name.
;  1385          2932  2 !
;  1386          2933  2
;  1387          2934  2 IF NOT .TAPE_CHAR[MT$V_BOT]
;  1388          2935  2 OR NOT MATCH_SSNAME (LABEL_BUFFER)
;  1389          2936  2 THEN
;  1390          2937  3     BEGIN
;  1391          2938  3     WHILE TRUE
;  1392          2939  3     DO
;  1393          2940  4         BEGIN
;  1394          2941  4         STATUS = SKIP_TM (1);
```

```
;  1395          2942  4          IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1396          2943  4
;  1397          2944  4          STATUS = READ_LABEL (LABEL_BUFFER, 'HDR1');
;  1398          2945  4          IF .STATUS
;  1399          2946  4          AND MATCH_SSNAME (LABEL_BUFFER)
;  1400          2947  4          THEN EXITLOOP;
;  1401          2948  4
;  1402          2949  4          IF .STATUS EQL SS$_ENDOFVOLUME
;  1403          2950  4          THEN
;  1404          2951  5              BEGIN
;  1405          2952  5              SKIP_TM (-2);
;  1406          2953  5              IF .INPUT_FLAGS[INPUT_WILDSAVE]
;  1407          2954  5              THEN COM_FLAGS[COM_EOV] = TRUE
;  1408          2955  5              ELSE
;  1409          2956  6                  BEGIN
;  1410          2957  6                  COM_FLAGS[COM_EOV] = FALSE;
;  1411          2958  6                  FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE,
;  1412          2959  6                              .FAB, SS$_NOSUCHFILE);
;  1413          2960  5                  END;
;  1414          2961  5              RETURN;
;  1415          2962  4              END;
;  1416          2963  3          END;
;  1417          2964  2      END;
;  1418          2965  2
;  1419          2966  2 INPUT_FLAGS[INPUT_SSFOUND] = TRUE;              ! Note that save set was found
;  1420          2967  2
;  1421          2968  2 ! Notify of a new save set if requested.
;  1422          2969  2 !
;  1423          2970  2 IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .QUAL[QUAL_LOG]
;  1424          2971  2 THEN SIGNAL (BACKUP$_NEWSAVSET, 2,
;  1425          2972  2                HD1$S_FILEID, LABEL_BUFFER[HD1$T_FILEID]);
;  1426          2973  2
;  1427          2974  2 ! Check other HDR1 fields.
;  1428          2975  2 !
;  1429          2976  2
;  1430          2977  2 CH$MOVE (HD1$S_FILESETID, LABEL_BUFFER[HD1$T_FILESETID], RWSV_FILESET_ID);
;  1431          2978  2 IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], RWSV_FILE_NUMBER)
;  1432          2979  2 THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  1433          2980  2 IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], FILE_SECTION)
;  1434          2981  2 THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  1435          2982  2 IF .FILE_SECTION NEQ 1
;  1436          2983  2 THEN
;  1437          2984  3     BEGIN
;  1438          2985  3     IF .QUAL[QUAL_IMAG]
;  1439          2986  3     THEN FILE_ERROR ((BACKUP$_NOT1STVOL AND NOT $FIELDMASK (STS$V_SEVERITY)) + STS$K_SEVERE, .FAB)
;  1440          2987  3     ELSE FILE_ERROR (BACKUP$_NOT1STVOL, .FAB);
;  1441          2988  2     END;
;  1442          2989  2 RWSV_VOL_NUMBER = .FILE_SECTION;
;  1443          2990  2
;  1444          2991  2 ! Read and check HDR2.
;  1445          2992  2 !
;  1446          2993  2
;  1447          2994  2 STATUS = READ_LABEL (LABEL_BUFFER, 'HDR2');
;  1448          2995  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1449          2996  2 IF .LABEL_BUFFER[HD2$B_RECFORMAT] NEQ 'F'
;  1450          2997  2 OR CH$NEQ (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN],
;  1451          2998  2             HD2$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

                                                OFFC 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+T1, 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
```

```
;  1466          3012  1 %SBTTL 'INIT_IN_SAVE - initialize save set for reading'
;  1467          3013  1 GLOBAL ROUTINE INIT_IN_SAVE (VERIFY) : NOVALUE =
;  1468          3014  1
;  1469          3015  1 !++
;  1470          3016  1 !
;  1471          3017  1 ! FUNCTIONAL DESCRIPTION:
;  1472          3018  1 !
;  1473          3019  1 !       This routine initializes the input save set.
;  1474          3020  1 !
;  1475          3021  1 ! CALLING SEQUENCE:
;  1476          3022  1 !       INIT_IN_SAVE (VERIFY)
;  1477          3023  1 !
;  1478          3024  1 ! INPUT PARAMETERS:
;  1479          3025  1 !       VERIFY: TRUE to indicate setup for verify pass
;  1480          3026  1 !               FALSE to indicate normal initialization
;  1481          3027  1 !
;  1482          3028  1 ! IMPLICIT INPUTS:
;  1483          3029  1 !       NONE
;  1484          3030  1 !
;  1485          3031  1 ! OUTPUT PARAMETERS:
;  1486          3032  1 !       NONE
;  1487          3033  1 !
;  1488          3034  1 ! IMPLICIT OUTPUTS:
;  1489          3035  1 !       NONE
;  1490          3036  1 !
;  1491          3037  1 ! ROUTINE VALUE:
;  1492          3038  1 !       NONE
;  1493          3039  1 !
;  1494          3040  1 ! SIDE EFFECTS:
;  1495          3041  1 !       NONE
;  1496          3042  1 !
;  1497          3043  1 !--
;  1498          3044  1
;  1499          3045  2 BEGIN
;  1500          3046  2
;  1501          3047  2 LOCAL
;  1502          3048  2         FAB             : REF BBLOCK,   ! pointer to input FAB
;  1503          3049  2         RAB             : REF BBLOCK,   ! pointer to input RAB
;  1504          3050  2         STATUS,                         ! the usual status value
;  1505          3051  2         DEVICE_DESC     : VECTOR[2];    ! Input device desc if mounted /FOREIGN
;  1506          3052  2
;  1507          3053  2 EXTERNAL ROUTINE
;  1508          3054  2         FILE_ERROR,                     ! signal file related error
;  1509          3055  2         INIT_BUFFERS,                   ! initialize the buffer pool
;  1510          3056  2         EXTRACT_FILENAME;               ! extract file name, type, and version
;  1511          3057  2
;  1512          3058  2 RWSV_IN_ERRORS = 0;
;  1513          3059  2 RWSV_IN_XORUSE = 0;
;  1514          3060  2 RWSV_IN_SEQ = 1;
;  1515          3061  2 RWSV_IN_SEQ_0 = 1;
;  1516          3062  2 RWSV_IN_XOR_RFA = 1;
;  1517          3063  2 RWSV_IN_VBN = 1;
;  1518          3064  2 RWSV_IN_VBN_0 = 1;
;  1519          3065  2
;  1520          3066  2 IF NOT .VERIFY
;  1521          3067  2 THEN
;  1522          3068  3     BEGIN
```

```
; 1523          3069  3     RWSV_VOL_NUMBER = 1;
; 1524          3070  3     RWSV_SEG_NUMBER = 0;
; 1525          3071  3
; 1526          3072  3 ! Do a UFO open on the input FAB and save away the channel.
; 1527          3073  3 !
; 1528          3074  3
; 1529          3075  3     RWSV_SAVE_QUAL = .QUAL[QUAL_INPU_LIST];
; 1530          3076  3     RWSV_SAVE_FAB = FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
; 1531          3077  3     FAB[FAB$V_NAM] = TRUE;
; 1532          3078  3     IF .QUAL[QUAL_SS_FILE]
; 1533          3079  3     THEN
; 1534          3080  4         BEGIN
; 1535          3081  4         IF .BBLOCK[FAB[FAB$L_DEV], DEV$V_NET]
; 1536          3082  4         THEN
; 1537          3083  4             FAB[FAB$V_BIO] = FAB[FAB$V_SQO] = TRUE;
; 1538          3084  4
; 1539          3085  4         FAB[FAB$V_GET] = TRUE;
; 1540          3086  4         END
; 1541          3087  3     ELSE
; 1542          3088  3         FAB[FAB$V_UFO] = TRUE;
; 1543          3089  3
; 1544          3090  3     IF .QUAL[QUAL_SS_FILE]
; 1545          3091  3     OR .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
; 1546          3092  3     THEN
; 1547          3093  4         BEGIN
; 1548          3094  4
; 1549          3095  4 ! If the device is not mounted foreign do a $OPEN; otherwise do a $ASSIGN
; 1550          3096  4 !
; 1551          3097  4
; 1552          3098  4         IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
; 1553          3099  5         THEN STATUS = $OPEN (FAB = .FAB)
; 1554          3100  4         ELSE
; 1555          3101  5             BEGIN
; 1556          3102  5             DEVICE_DESC[0] = .BBLOCK[.FAB[FAB$L_NAM], NAM$B_DEV];
; 1557          3103  5             DEVICE_DESC[1] = .BBLOCK[.FAB[FAB$L_NAM], NAM$L_DEV];
; 1558        P 3104  5             STATUS = FAB[FAB$L_STS] = $ASSIGN (DEVNAM = DEVICE_DESC,
; 1559          3105  5                                                CHAN = FAB[FAB$L_STV]);
; 1560          3106  4             END;
; 1561          3107  4         IF NOT .STATUS
; 1562          3108  4         THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
; 1563          3109  4                          .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
; 1564          3110  4         END
; 1565          3111  3     ELSE
; 1566          3112  4         BEGIN
; 1567          3113  4         INIT_SAVE_DISK (0);
; 1568          3114  3         END;
; 1569          3115  3
; 1570          3116  3     IF .QUAL[QUAL_SS_FILE]
; 1571          3117  3     THEN
; 1572          3118  4         BEGIN
; 1573          3119  4         RAB = FAB[FC_RAB];
; 1574          3120  4         STATUS = $CONNECT (RAB = .RAB);
; 1575          3121  4         IF NOT .STATUS
; 1576          3122  4         THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
; 1577          3123  4                          .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
; 1578          3124  4         IF .FAB[FAB$B_RFM] NEQ FAB$C_FIX
; 1579          3125  4         OR .FAB[FAB$B_ORG] NEQ FAB$C_SEQ
```

```
; 1580          3126  4          OR .BBLOCK[FAB[FAB$L_DEV], DEV$V_NET] AND .(FAB[FAB$W_MRS])<0,9> NEQ 0
; 1581          3127  4          THEN FILE_ERROR (BACKUP$_NOTSAVESET, .FAB);
; 1582          3128  4          QUAL[QUAL_BLOC_VALUE] = .FAB[FAB$W_MRS];
; 1583          3129  4          CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
; 1584          3130  4          END
; 1585          3131  3      ELSE
; 1586          3132  3          RWSV_CHAN = .FAB[FAB$L_STV];
; 1587          3133  3
; 1588          3134  3      EXTRACT_FILENAME (.FAB, COM_SSNAME);
; 1589          3135  3      END
; 1590          3136  3
; 1591          3137  3  ! Otherwise do setup for input verification.
; 1592          3138  3  !
; 1593          3139  3
; 1594          3140  2  ELSE
; 1595          3141  3      BEGIN
; 1596          3142  3      FAB = .RWSV_SAVE_FAB;
; 1597          3143  3      RAB = FAB[FB_RAB];
; 1598          3144  3      IF .QUAL[QUAL_SS_FILE]
; 1599          3145  3      THEN
; 1600          3146  4          BEGIN
; 1601          3147  4          STATUS = $REWIND (RAB = .RAB);
; 1602          3148  4          IF NOT .STATUS
; 1603          3149  4          THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
; 1604          3150  4                           .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
; 1605          3151  4          CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
; 1606          3152  3          END;
; 1607          3153  2      END;
; 1608          3154  2  ALT_SSNAME[0] = 0;
; 1609          3155  2
; 1610          3156  2  ! Set up the input tape.
; 1611          3157  2  !
; 1612          3158  2
; 1613          3159  2  IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
; 1614          3160  2  AND NOT .QUAL[QUAL_SS_FILE]
; 1615          3161  2  THEN
; 1616          3162  3      BEGIN
; 1617          3163  3      INIT_SAVE_TAPE (.VERIFY);
; 1618          3164  3      IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .COM_FLAGS[COM_EOV] THEN RETURN;
; 1619          3165  2      END;
; 1620          3166  2
; 1621          3167  2  ! Set up the buffer pool.
; 1622          3168  2  !
; 1623          3169  2
; 1624          3170  2  IF NOT .VERIFY
; 1625          3171  2  THEN
; 1626          3172  3      BEGIN
; 1627          3173  3      INIT_BUFFERS (.QUAL[QUAL_BUFF_VALUE], .QUAL[QUAL_BLOC_VALUE]);
; 1628          3174  3      RWSV_IN_GROUP_SIZE = 1^3T-1;
; 1629          3175  2      END;
; 1630          3176  2
; 1631          3177  1  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
```

```
;  1633          3178  1 %SBTTL 'READ_BUFFER - read a save set buffer'
;  1634          3179  1 GLOBAL ROUTINE READ_BUFFER =
;  1635          3180  1
;  1636          3181  1 !++
;  1637          3182  1 !
;  1638          3183  1 ! FUNCTIONAL DESCRIPTION:
;  1639          3184  1 !
;  1640          3185  1 !       This routine reads a corrected buffer from the input save
;  1641          3186  1 !       set, applying XOR recovery if necessary.
;  1642          3187  1 !
;  1643          3188  1 ! CALLING SEQUENCE:
;  1644          3189  1 !       READ_BUFFER ()
;  1645          3190  1 !
;  1646          3191  1 ! INPUT PARAMETERS:
;  1647          3192  1 !       NONE
;  1648          3193  1 !
;  1649          3194  1 ! IMPLICIT INPUTS:
;  1650          3195  1 !       NONE
;  1651          3196  1 !
;  1652          3197  1 ! OUTPUT PARAMETERS:
;  1653          3198  1 !       NONE
;  1654          3199  1 !
;  1655          3200  1 ! IMPLICIT OUTPUTS:
;  1656          3201  1 !       NONE
;  1657          3202  1 !
;  1658          3203  1 ! ROUTINE VALUE:
;  1659          3204  1 !       BCB address of buffer read
;  1660          3205  1 !
;  1661          3206  1 ! SIDE EFFECTS:
;  1662          3207  1 !       NONE
;  1663          3208  1 !
;  1664          3209  1 !--
;  1665          3210  1
;  1666          3211  2 BEGIN
;  1667          3212  2
;  1668          3213  2 LOCAL
;  1669          3214  2         K,                              ! count of blocks missed
;  1670          3215  2         RAB             : REF BBLOCK,   ! RAB to read input file
;  1671          3216  2         BCB             : REF BBLOCK,   ! BCB of block read
;  1672          3217  2         BUFFER          : REF BBLOCK;   ! data buffer read
;  1673          3218  2
;  1674          3219  2 EXTERNAL ROUTINE
;  1675          3220  2         FILE_ERROR,                     ! signal file related error
;  1676          3221  2         FREE_BUFFER;                    ! release buffer to free list
;  1677          3222  2
;  1678          3223  2 ! Loop, reading buffers until we get an acceptable data block. Others
;  1679          3224  2 ! are treated appropriately.
;  1680          3225  2 !
;  1681          3226  2
;  1682          3227  2 IF .RWSV_XORSIZE NEQ 0 THEN RWSV_IN_GROUP_SIZE = .RWSV_XORSIZE;
;  1683          3228  2 WHILE TRUE
;  1684          3229  2 DO
;  1685          3230  3     BEGIN
;  1686          3231  3     RWSV_IN_ORGERR[0] = TRUE;
;  1687          3232  3     RWSV_IN_ORGERR[1] = 0;
;  1688          3233  3     BCB = READ_SEQ_BLOCK (TRUE);
;  1689          3234  3     IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
```

```
; 1690          3235  3     THEN
; 1691          3236  4         BEGIN
; 1692          3237  4         BCB = NEXT_VOLUME (.BCB);
; 1693          3238  4         IF .BCB EQL 0
; 1694          3239  4         THEN RETURN 0;
; 1695          3240  3         END;
; 1696          3241  3
; 1697          3242  3     IF .(RWSV_IN_XOR_RFA)<0,32> EQL 0
; 1698          3243  3     AND .(RWSV_IN_XOR_RFA+4)<0,16> EQL 0
; 1699          3244  3     THEN
; 1700          3245  4         BEGIN
; 1701          3246  4         RAB = RWSV_SAVE_FAB[FC_RAB];
; 1702          3247  4         CH$MOVE (RAB$S_RFA, RAB[RAB$W_RFA], RWSV_IN_XOR_RFA);
; 1703          3248  3         END;
; 1704          3249  3     BUFFER = .BCB[BCB_BUFFER];
; 1705          3250  3
; 1706          3251  3 ! Check if a block has been skipped. If exactly one has, then apply XOR
; 1707          3252  3 ! recovery to get it back. If recovery fails, report the error unless the
; 1708          3253  3 ! missed block was an XOR block. We do not count or report block skip
; 1709          3254  3 ! errors if we're trying to read the first save set block, and there was
; 1710          3255  3 ! no I/O error; this is the condition of being handed a continuation
; 1711          3256  3 ! volume to start with.
; 1712          3257  3 !
; 1713          3258  3
; 1714          3259  3     IF .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ EQL 1
; 1715          3260  4     AND (.RWSV_IN_SEQ NEQ .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE + 1
; 1716          3261  4          OR .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_XORBLOCK)
; 1717          3262  4     AND (.RWSV_IN_SEQ NEQ 1 OR NOT .RWSV_IN_ORGERR[0])
; 1718          3263  3     THEN
; 1719          3264  4         BEGIN
; 1720          3265  4         BCB = XORCUE (.RWSV_IN_SEQ, .BCB);
; 1721          3266  4         BUFFER = .BCB[BCB_BUFFER];
; 1722          3267  3         END;
; 1723          3268  3
; 1724          3269  3     K = .BUFFER[BBH$L_NUMBER] - .RWSV_IN_SEQ;
; 1725          3270  3     IF .K NEQ 0
; 1726          3271  3     THEN
; 1727          3272  4         BEGIN
; 1728          3273  5         IF (.RWSV_IN_SEQ NEQ 1 OR NOT .RWSV_IN_ORGERR[0])
; 1729          3274  4         THEN
; 1730          3275  5             BEGIN
; 1731          3276  6             IF (.RWSV_IN_SEQ NEQ .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE
; 1732          3277  6                 OR .K NEQ 1)
; 1733          3278  5             THEN
; 1734          3279  6                 BEGIN
; 1735          3280  6                 FILE_ERROR (BACKUP$_BLOCKLOST, .RWSV_SAVE_FAB,
; 1736          3281  6                             .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
; 1737          3282  6                 RWSV_IN_ERRORS = .RWSV_IN_ERRORS - 1;
; 1738          3283  5                 END;
; 1739          3284  5             END
; 1740          3285  4         ELSE
; 1741          3286  4             RWSV_IN_ERRORS = .RWSV_IN_ERRORS - 1;
; 1742          3287  4         IF .RWSV_IN_GROUP_SIZE NEQ 0
; 1743          3288  4         THEN
; 1744          3289  5             BEGIN
; 1745          3290  5             IF .BUFFER[BBH$L_NUMBER] GEQU .RWSV_IN_XOR_SEQ + .RWSV_IN_GROUP_SIZE
; 1746          3291  5             THEN RWSV_IN_XOR_SEQ = .RWSV_IN_XOR_SEQ
```

```
;  1747          3292  6                                   + (.BUFFER[BBH$L_NUMBER] - .RWSV_IN_XOR_SEQ)
;  1748          3293  5                                   / .RWSV_IN_GROUP_SIZE * .RWSV_IN_GROUP_SIZE;
;  1749          3294  4               END;
;  1750          3295  3           END;
;  1751          3296  3
;  1752          3297  3       ! Do basic validation checks to make sure we can understand this block.
;  1753          3298  3       ! Note that this is done even if the block reads with an error, since
;  1754          3299  3       ! the header data is protected by the header CRC.
;  1755          3300  3       !
;  1756          3301  3
;  1757          3302  3           IF .BUFFER[BBH$W_SIZE] LSSU BBH$K_LENGTH
;  1758          3303  3           THEN SIGNAL (BACKUP$_INVBLKHDR)
;  1759          3304  3
;  1760          3305  3       ! Process the data block. If there has been a read error, attempt to
;  1761          3306  3       ! recover it. If the error persists, report it.
;  1762          3307  3       !
;  1763          3308  3
;  1764          3309  3           ELSE
;  1765          3310  4               BEGIN
;  1766          3311  4               IF .BUFFER[BBH$W_APPLIC] EQL BACKUP$K_DATABLOCK
;  1767          3312  4               THEN
;  1768          3313  5                   BEGIN
;  1769          3314  5                   IF .BUFFER[BBH$W_STRUCLEV] NEQ BBH$K_LEVEL1
;  1770          3315  5                   THEN SIGNAL (BACKUP$_INVSTRUCT)
;  1771          3316  5                   ELSE
;  1772          3317  6                       BEGIN
;  1773          3318  6                       IF NOT .BCB[BCB_STATUS]
;  1774          3319  6                       THEN
;  1775          3320  7                           BEGIN
;  1776          3321  7                           BCB = XORCIZE (.RWSV_IN_SEQ, .BCB);
;  1777          3322  7                           BUFFER = .BCB[BCB_BUFFER];
;  1778          3323  6                           END;
;  1779          3324  6
;  1780          3325  6                       IF NOT .BCB[BCB_STATUS]
;  1781          3326  6                       THEN
;  1782          3327  7                           BEGIN
;  1783          3328  7                           FILE_ERROR (BACKUP$_READERR+STS$K_ERROR,
;  1784          3329  7                                       .RWSV_SAVE_FAB, .RWSV_IN_ORGERR[0], .RWSV_IN_ORGERR[1]);
;  1785          3330  7                           RWSV_IN_ERRORS = .RWSV_IN_ERRORS + 1;
;  1786          3331  6                           END;
;  1787          3332  6
;  1788          3333  6       ! Check that the save set name in the block header matches the labels.
;  1789          3334  6       ! A mismatch indicates running off the end of an incomplete tape.
;  1790          3335  6       ! If we are instructed to continue, use the new save set name.
;  1791          3336  6       !
;  1792          3337  6
;  1793          3338  6                       IF CH$NEQ (.ALT_SSNAME[0]+1, ALT_SSNAME,
;  1794          3339  6                                  .(BUFFER[BBH$T_SSNAME])<0,8>+1, BUFFER[BBH$T_SSNAME], 255)
;  1795          3340  6                       THEN
;  1796          3341  7                           BEGIN
;  1797          3342  7                           IF .ALT_SSNAME[0] NEQ 0
;  1798          3343  7                           THEN FILE_ERROR (BACKUP$_SSCHANGE, .RWSV_SAVE_FAB);
;  1799          3344  7                           CH$COPY (.(BUFFER[BBH$T_SSNAME])<0,8>+1, BUFFER[BBH$T_SSNAME],
;  1800          3345  7                                    0, BBH$S_SSNAME, ALT_SSNAME);
;  1801          3346  6                           END;
;  1802          3347  6
;  1803          3348  6                       IF .BUFFER[BBH$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                EXIT[OOP;
;  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$L_NUMBER];
;  1817          3362  5              RWSV_IN_XOR_RFA = .BCB[BCB_BLOCKNUM];
;  1818          3363  5              IF .QUA[QUAL_SS_FILE]
;  1819          3364  5              THEN
;  1820          3365  6                  BEGIN
;  1821          3366  6                  RAB = RWSV_SAVE_FAB[FC_RAB];
;  1822          3367  6                  CH$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
```

```
;1835     3379  1 %SBTTL 'FIN_IN_SAVE - finish reading save set'
;1836     3380  1 GLOBAL ROUTINE FIN_IN_SAVE (CONTINUE) : NOVALUE =
;1837     3381  1
;1838     3382  1 !++
;1839     3383  1 !
;1840     3384  1 ! FUNCTIONAL DESCRIPTION:
;1841     3385  1 !
;1842     3386  1 !       This routine completes the reading of the input save set.
;1843     3387  1 !
;1844     3388  1 ! CALLING SEQUENCE:
;1845     3389  1 !       FIN_IN_SAVE (CONTINUE)
;1846     3390  1 !
;1847     3391  1 ! INPUT PARAMETERS:
;1848     3392  1 !       CONTINUE: TRUE if there is a continuation tape
;1849     3393  1 !                 FALSE if this is the last reel
;1850     3394  1 !
;1851     3395  1 ! IMPLICIT INPUTS:
;1852     3396  1 !       NONE
;1853     3397  1 !
;1854     3398  1 ! OUTPUT PARAMETERS:
;1855     3399  1 !       NONE
;1856     3400  1 !
;1857     3401  1 ! IMPLICIT OUTPUTS:
;1858     3402  1 !       NONE
;1859     3403  1 !
;1860     3404  1 ! ROUTINE VALUE:
;1861     3405  1 !       NONE
;1862     3406  1 !
;1863     3407  1 ! SIDE EFFECTS:
;1864     3408  1 !       NONE
;1865     3409  1 !
;1866     3410  1 !--
;1867     3411  1
;1868     3412  2 BEGIN
;1869     3413  2
;1870     3414  2 BUILTIN
;1871     3415  2         REMQUE;
;1872     3416  2
;1873     3417  2 LOCAL
;1874     3418  2         STATUS,                         ! general status value
;1875     3419  2         TM_COUNT,                       ! count of tape marks crossed
;1876     3420  2         BCB             : REF BBLOCK,   ! pointer to current BCB
;1877     3421  2         FAB             : REF BBLOCK;   ! input FAB
;1878     3422  2
;1879     3423  2 EXTERNAL ROUTINE
;1880     3424  2         WAIT,                           ! wait for I/O completion
;1881     3425  2         FREE_BUFFER,                    ! free an I/O buffer
;1882     3426  2         SKIP_TM,                        ! skip tape marks
;1883     3427  2         UNLOAD,                         ! rewind and unload tape
;1884     3428  2         STA_DISMOUNT,                   ! dismount save set disk
;1885     3429  2         RESTORE_VERIFY_REEL,            ! verify input save set volume
;1886     3430  2         FILE_ERROR;                     ! signal file related error
;1887     3431  2
;1888     3432  2 ! If the input is a file, close it.
;1889     3433  2 !
;1890     3434  2
;1891     3435  2 FAB = .RWSV_SAVE_FAB;
```

```
; 1892          3436  2 IF .QUAL[QUAL_SS_FILE]
; 1893          3437  2 THEN
; 1894          3438  3     BEGIN
; 1895          3439  3     IF NOT .QUAL[QUAL_VERI]
; 1896          3440  3     OR .COM_FLAGS[COM_VERIFYING]
; 1897          3441  3     THEN
; 1898          3442  4         BEGIN
; 1899          3443  4         STATUS = $CLOSE (FAB = .FAB);
; 1900          3444  4         IF NOT .STATUS
; 1901          3445  4         THEN FILE_ERROR (BACKUP$_CLOSEIN+STS$K_ERROR, .FAB,
; 1902          3446  4                          .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
; 1903          3447  3         END;
; 1904          3448  3     END
; 1905          3449  3
; 1906          3450  3 ! For a tape, wait out the pending read aheads, and backspace the tape
; 1907          3451  3 ! over the trailer label set so that appending won't get lost.
; 1908          3452  3 !
; 1909          3453  3
; 1910          3454  2 ELSE
; 1911          3455  3     BEGIN
; 1912          3456  3     IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
; 1913          3457  3     THEN
; 1914          3458  4         BEGIN
; 1915          3459  4         TM_COUNT = 0;
; 1916          3460  4         UNTIL REMQUE (.INPUT_WAIT[0], BCB)
; 1917          3461  4         DO
; 1918          3462  5             BEGIN
; 1919          3463  5             WAIT (.BCB);
; 1920          3464  5             FREE_BUFFER (.BCB);
; 1921          3465  5             IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFFILE
; 1922          3466  5             THEN TM_COUNT = .TM_COUNT + 1;
; 1923          3467  4             END;
; 1924          3468  4
; 1925          3469  4         SKIP_TM (-.TM_COUNT);
; 1926          3470  4
; 1927          3471  4         IF NOT .QUAL[QUAL_VERI]
; 1928          3472  4         OR .COM_FLAGS[COM_VERIFYING]
; 1929          3473  4         THEN
; 1930          3474  5             BEGIN
; 1931          3475  5             IF .CONTINUE
; 1932          3476  5             THEN UNLOAD ();
; 1933          3477  5
; 1934          3478  5             $DASSGN (CHAN = .RWSV_CHAN);
; 1935          3479  5             RWSV_CHAN = 0;
; 1936          3480  4             END;
; 1937          3481  4         END
; 1938          3482  4
; 1939          3483  4 ! For a save set disk, wait out any pending reads. Then deaccess the file
; 1940          3484  4 ! and dismount the volume.
; 1941          3485  4 !
; 1942          3486  4
; 1943          3487  3     ELSE
; 1944          3488  4         BEGIN
; 1945          3489  4         UNTIL REMQUE (.INPUT_WAIT[0], BCB)
; 1946          3490  4         DO
; 1947          3491  5             BEGIN
; 1948          3492  5             WAIT (.BCB);
```

```
;  1949          3493  5              FREE_BUFFER (.BCB);
;  1950          3494  4              END;
;  1951          3495  4
;  1952          3496  4          IF NOT .QUAL[QUAL_VERI]
;  1953          3497  4          OR .COM_FLAGS[COM_VERIFYING]
;  1954          3498  4          THEN
;  1955          3499  5              BEGIN
;  1956        P 3500  5              $$QIOW (CHAN = .RWSV_CHAN,
;  1957        P 3501  5                      FUNC = IO$_DEACCESS
;  1958          3502  5                      );
;  1959          3503  5              IF .CURRENT_MTL[MTL_SEQ_DISK]
;  1960          3504  5              THEN
;  1961          3505  6                  BEGIN
;  1962          3506  6                  STA_DISMOUNT (.RWSV_VOL_NUMBER);
;  1963        P 3507  6                  $QIOW (CHAN = .CURRENT_VCB[VCB_CHAN],
;  1964        P 3508  6                         FUNC = IO$_UNLOAD
;  1965          3509  6                         );
;  1966          3510  5                  END;
;  1967          3511  4              END;
;  1968          3512  3          END;
;  1969          3513  2      END;
;  1970          3514  2
;  1971          3515  2 ! Inform the user of any input errors that we might have graciously
;  1972          3516  2 ! recovered.
;  1973          3517  2 !
;  1974          3518  2
;  1975          3519  2 IF .RWSV_IN_ERRORS NEQ 0
;  1976          3520  2 THEN FILE_ERROR (BACKUP$_SOFTRERRS, .FAB, .RWSV_IN_ERRORS);
;  1977          3521  2 IF .RWSV_IN_XORUSE NEQ 0
;  1978          3522  2 THEN FILE_ERROR (BACKUP$_XORERRS, .FAB, .RWSV_IN_XORUSE);
;  1979          3523  2
;  1980          3524  2
;  1981          3525  2 ! Run a verify pass if requested (and if this is not already a verify pass).
;  1982          3526  2 !
;  1983          3527  2
;  1984          3528  2 IF NOT .COM_FLAGS[COM_VERIFYING]
;  1985          3529  2 THEN
;  1986          3530  3      BEGIN
;  1987          3531  3      IF .QUAL[QUAL_VERI]
;  1988          3532  3      THEN
;  1989          3533  4          BEGIN
;  1990          3534  4          SIGNAL (BACKUP$_STARTVERIFY);
;  1991          3535  4          COM_FLAGS[COM_VERIFYING] = TRUE;
;  1992          3536  4          IF .RWSV_SEG_NUMBER EQL 0
;  1993          3537  4          THEN INIT_IN_SAVE (TRUE)
;  1994          3538  4          ELSE
;  1995          3539  4              IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  1996          3540  4              THEN READY_NEXT_VOLUME ();
;  1997          3541  4          RWSV_IN_SEQ = .RWSV_IN_SEQ_0;
;  1998          3542  4          RWSV_IN_VBN = .RWSV_IN_VBN_0;
;  1999          3543  4          RWSV_IN_ERRORS = 0;
;  2000          3544  4          RWSV_IN_XORUSE = 0;
;  2001          3545  4          RESTORE_VERIFY_REEL ();
;  2002          3546  4          FIN_IN_SAVE (.CONTINUE);
;  2003          3547  4          COM_FLAGS[COM_VERIFYING] = FALSE;
;  2004          3548  3          END;
;  2005          3549  3      IF .CONTINUE THEN SIGNAL (BACKUP$_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
```

```
;  2010          3553  1 %SBTTL 'NEXTVOL_HANDLER - handler for next volume processing'
;  2011          3554  1 ROUTINE NEXTVOL_HANDLER (SIGNAL, MECHANISM) =
;  2012          3555  1
;  2013          3556  1 !++
;  2014          3557  1 !
;  2015          3558  1 ! FUNCTIONAL DESCRIPTION:
;  2016          3559  1 !
;  2017          3560  1 !       This routine is the condition handler for the TRY_NEXT_VOLUME
;  2018          3561  1 !       routine. It resignals the error with informational severity and then
;  2019          3562  1 !       returns failure, causing a retry.
;  2020          3563  1 !
;  2021          3564  1 ! CALLING SEQUENCE:
;  2022          3565  1 !       NEXTVOL_HANDLER (SIGNAL, MECHANISM)
;  2023          3566  1 !
;  2024          3567  1 ! INPUT PARAMETERS:
;  2025          3568  1 !       SIGNAL: signal argument list
;  2026          3569  1 !       MECHANISM: mechanism arg list
;  2027          3570  1 !
;  2028          3571  1 ! IMPLICIT INPUTS:
;  2029          3572  1 !       NONE
;  2030          3573  1 !
;  2031          3574  1 ! OUTPUT PARAMETERS:
;  2032          3575  1 !       NONE
;  2033          3576  1 !
;  2034          3577  1 ! IMPLICIT OUTPUTS:
;  2035          3578  1 !       NONE
;  2036          3579  1 !
;  2037          3580  1 ! ROUTINE VALUE:
;  2038          3581  1 !       TRUE
;  2039          3582  1 !
;  2040          3583  1 ! SIDE EFFECTS:
;  2041          3584  1 !       NONE
;  2042          3585  1 !
;  2043          3586  1 !--
;  2044          3587  1
;  2045          3588  2 BEGIN
;  2046          3589  2
;  2047          3590  2 BUILTIN
;  2048          3591  2         CALLG;
;  2049          3592  2
;  2050          3593  2 MAP
;  2051          3594  2         SIGNAL          : REF BBLOCK,   ! signal arg
;  2052          3595  2         MECHANISM       : REF BBLOCK;   ! mechanism arg
;  2053          3596  2
;  2054          3597  2 EXTERNAL ROUTINE
;  2055          3598  2         STA_DISMOUNT,           ! dismount volume
;  2056          3599  2         LIB$SIGNAL              : ADDRESSING_MODE (GENERAL);
;  2057          3600  2
;  2058          3601  2 ! Fix up the severity of the error being signalled. Then resignal it
;  2059          3602  2 ! and unwind.
;  2060          3603  2 !
;  2061          3604  2
;  2062          3605  2 IF .SIGNAL[CHF$L_SIG_NAME] NEQ SS$_UNWIND
;  2063          3606  2 AND NOT .SIGNAL[CHF$[_SIG_NAME]
;  2064          3607  2 THEN
;  2065          3608  3     BEGIN
;  2066          3609  3     IF NOT .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
```

```
  3      THEN
  4          BEGIN
P 4          SS$QIOW (CHAN = STA_IN_CHAN,
P 4                   FUNC = IO$_DEACCESS
  4                   );
  4          STA_DISMOUNT (.RWSV_VOL_NUMBER);
  3          END;
  3      BBLOCK [SIGNAL[CHF$L_SIG_NAME], STS$V_SEVERITY] = STS$K_INFO;
  3      SIGNAL[CHF$L_SIG_ARGS] = .SIGNAL[CHF$L_SIG_ARGS] - 2;
  3      CALLG (.SIGNAL, LIB$SIGNAL);
  3      MECHANISM[CHF$L_MCH_SAVR0] = FALSE;
  3      $UNWIND ();
  2      END;
  2
  2 SS$_RESIGNAL
  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
```

```
;2084           3626  1 %SBTTL 'READY_NEXT_VOLUME - set up next volume'
;2085           3627  1 GLOBAL ROUTINE READY_NEXT_VOLUME (FILE_ID) =
;2086           3628  1
;2087           3629  1 !++
;2088           3630  1 !
;2089           3631  1 ! FUNCTIONAL DESCRIPTION:
;2090           3632  1 !
;2091           3633  1 !       This routine reads and verifies the labels on the candidate
;2092           3634  1 !       continuation volume.
;2093           3635  1 !
;2094           3636  1 ! CALLING SEQUENCE:
;2095           3637  1 !       READY_NEXT_VOLUME ()
;2096           3638  1 !
;2097           3639  1 ! INPUT PARAMETERS:
;2098           3640  1 !       FILE_ID: address of continuation file ID if disk
;2099           3641  1 !
;2100           3642  1 ! IMPLICIT INPUTS:
;2101           3643  1 !       NONE
;2102           3644  1 !
;2103           3645  1 ! OUTPUT PARAMETERS:
;2104           3646  1 !       NONE
;2105           3647  1 !
;2106           3648  1 ! IMPLICIT OUTPUTS:
;2107           3649  1 !       NONE
;2108           3650  1 !
;2109           3651  1 ! ROUTINE VALUE:
;2110           3652  1 !       TRUE
;2111           3653  1 !
;2112           3654  1 ! SIDE EFFECTS:
;2113           3655  1 !       NONE
;2114           3656  1 !
;2115           3657  1 !--
;2116           3658  1
;2117           3659  2 BEGIN
;2118           3660  2
;2119           3661  2 MAP
;2120           3662  2         FILE_ID         : REF BBLOCK;   ! file ID arg
;2121           3663  2
;2122           3664  2 LOCAL
;2123           3665  2         FAB             : REF BBLOCK,   ! pointer to input FAB
;2124           3666  2         STATUS,                         ! the usual status value
;2125           3667  2         BUFFER_SIZE,                    ! size of I/O buffers
;2126           3668  2         NUMBER,                         ! output for decimal convert
;2127           3669  2         LABEL_BUFFER    : BBLOCK [90];  ! buffer for tape labels
;2128           3670  2
;2129           3671  2 EXTERNAL ROUTINE
;2130           3672  2         FILE_ERROR,                     ! signal file related error
;2131           3673  2         READY_TAPE,                     ! prepare tape for I/O
;2132           3674  2         REWIND,                         ! rewind tape
;2133           3675  2         SKIP_TM,                        ! skip tape marks
;2134           3676  2         READ_LABEL,                     ! read and check tape label
;2135           3677  2         LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL);
;2136           3678  2                                         ! convert decimal to binary
;2137           3679  2
;2138           3680  2 FAB = .RWSV_SAVE_FAB;
;2139           3681  2 RWSV_IN_ERRORS = 0;
;2140           3682  2 RWSV_IN_XORUSE = 0;
```

```
;  2141          3683  2
;  2142          3684  2 ! Set up the input tape. Rewind it if called for.                                          \
;  2143          3685  2 !
;  2144          3686  2
;  2145          3687  2 IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  2146          3688  2 THEN
;  2147          3689  2     BEGIN
;  2148          3690  2     READY_TAPE (FALSE);
;  2149          3691  2     REWIND ();
;  2150          3692  2
;  2151          3693  2 ! Read and verify the volume header label. Then scan for HDR1.
;  2152          3694  2 !
;  2153          3695  2
;  2154          3696  2     STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
;  2155          3697  2     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  2156          3698  2
;  2157          3699  2     WHILE TRUE
;  2158          3700  3     DO
;  2159          3701  4         BEGIN
;  2160          3702  4         STATUS = READ_LABEL (LABEL_BUFFER);
;  2161          3703  4         IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  2162          3704  4         IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'HDR1' THEN EXITLOOP;
;  2163          3705  3         END;
;  2164          3706  3
;  2165          3707  3 ! Check HDR1 for the correct save set name, file set ID, and file
;  2166          3708  3 ! & section numbers.
;  2167          3709  3 !
;  2168          3710  3
;  2169          3711  3     IF NOT MATCH_SSNAME (LABEL_BUFFER)
;  2170          3712  3
;  2171          3713  3     OR CH$NEQ (HD1$S_FILESETID, LABEL_BUFFER[HD1$T_FILESETID],
;  2172          3714  3                HD1$S_FILESETID, RWSV_FILESET_ID)
;  2173          3715  3
;  2174          3716  3     OR
;  2175          3717  4         BEGIN
;  2176          3718  4         IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], NUMBER)
;  2177          3719  4         THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  2178          3720  4         .NUMBER NEQ .RWSV_FILE_NUMBER
;  2179          3721  4         END
;  2180          3722  4
;  2181          3723  3     OR
;  2182          3724  4         BEGIN
;  2183          3725  4         IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESECNO], NUMBER)
;  2184          3726  4         THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  2185          3727  4         .NUMBER NEQ .RWSV_VOL_NUMBER
;  2186          3728  4         END
;  2187          3729  4
;  2188          3730  3     THEN FILE_ERROR (BACKUP$_WRONGVOL, .FAB);
;  2189          3731  3
;  2190          3732  3 ! Read and check HDR2.
;  2191          3733  3 !
;  2192          3734  3
;  2193          3735  3     STATUS = READ_LABEL (LABEL_BUFFER, 'HDR2');
;  2194          3736  3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  2195          3737  3     IF .LABEL_BUFFER[HD2$B_RECFORMAT] NEQ 'F'
;  2196          3738  3     OR CH$NEQ (HD2$S_RECLEN, LABEL_BUFFER[HD2$T_RECLEN],
;  2197          3739  3                HD2$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 00039          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$:     CMPL    LABEL_BUFFER, #82,475016                        : 3704
                                          E3  12 00082         BNEQ    2$                                              :
                                      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
                                      28  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
```

```
;  2220          3761  1 %SBTTL 'TRY_NEXT_VOLUME - set up next volume under handler'
;  2221          3762  1 GLOBAL ROUTINE TRY_NEXT_VOLUME (FILE_ID) =
;  2222          3763  1
;  2223          3764  1 !++
;  2224          3765  1 !
;  2225          3766  1 ! FUNCTIONAL DESCRIPTION:
;  2226          3767  1 !
;  2227          3768  1 !       This routine reads and verifies the labels on the candidate
;  2228          3769  1 !       continuation volume.
;  2229          3770  1 !
;  2230          3771  1 ! CALLING SEQUENCE:
;  2231          3772  1 !       TRY_NEXT_VOLUME ()
;  2232          3773  1 !
;  2233          3774  1 ! INPUT PARAMETERS:
;  2234          3775  1 !       FILE_ID: address of continuation file ID if disk
;  2235          3776  1 !
;  2236          3777  1 ! IMPLICIT INPUTS:
;  2237          3778  1 !       NONE
;  2238          3779  1 !
;  2239          3780  1 ! OUTPUT PARAMETERS:
;  2240          3781  1 !       NONE
;  2241          3782  1 !
;  2242          3783  1 ! IMPLICIT OUTPUTS:
;  2243          3784  1 !       NONE
;  2244          3785  1 !
;  2245          3786  1 ! ROUTINE VALUE:
;  2246          3787  1 !       TRUE if volume is OK
;  2247          3788  1 !       FALSE if not; retry requested
;  2248          3789  1 !
;  2249          3790  1 ! SIDE EFFECTS:
;  2250          3791  1 !       NONE
;  2251          3792  1 !
;  2252          3793  1 !--
;  2253          3794  1
;  2254          3795  2 BEGIN
;  2255          3796  2
;  2256          3797  2 MAP
;  2257          3798  2         FILE_ID         : REF BBLOCK;   ! file ID arg
;  2258          3799  2
;  2259          3800  2 ! This routine simply wraps a handler around the READY_NEXT_VOLUME
;  2260          3801  2 ! routine so that errors cause a non-fatal message and error return,
;  2261          3802  2 ! rather than an exit.
;  2262          3803  2 !
;  2263          3804  2
;  2264          3805  2 ENABLE NEXTVOL_HANDLER;
;  2265          3806  2 READY_NEXT_VOLUME (.FILE_ID)
;  2266          3807  2
;  2267          3808  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, NEXT_VOL_HANDLER       ;
                                    04 0001F         RET                                 ;

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

```
;  2269     3809  1 %SBTTL 'NEXT_VOLUME - switch to continuation volume'
;  2270     3810  1 ROUTINE NEXT_VOLUME (CUR_BCB) =
;  2271     3811  1
;  2272     3812  1 !++
;  2273     3813  1 !
;  2274     3814  1 ! FUNCTIONAL DESCRIPTION:
;  2275     3815  1 !
;  2276     3816  1 !       This routine is invoked when end of medium is encountered on
;  2277     3817  1 !       the input save set. It determines if there is a continuation
;  2278     3818  1 !       medium, and if so, readies it for use and returns the first block.
;  2279     3819  1 !
;  2280     3820  1 ! CALLING SEQUENCE:
;  2281     3821  1 !       NEXT_VOLUME (CUR_BCB)
;  2282     3822  1 !
;  2283     3823  1 ! INPUT PARAMETERS:
;  2284     3824  1 !       CUR_BCB: BCB of last read (containing end of file status)
;  2285     3825  1 !
;  2286     3826  1 ! IMPLICIT INPUTS:
;  2287     3827  1 !       NONE
;  2288     3828  1 !
;  2289     3829  1 ! OUTPUT PARAMETERS:
;  2290     3830  1 !       NONE
;  2291     3831  1 !
;  2292     3832  1 ! IMPLICIT OUTPUTS:
;  2293     3833  1 !       NONE
;  2294     3834  1 !
;  2295     3835  1 ! ROUTINE VALUE:
;  2296     3836  1 !       BCB of first block on new volume, or 0 if no more volumes
;  2297     3837  1 !
;  2298     3838  1 ! SIDE EFFECTS:
;  2299     3839  1 !       NONE
;  2300     3840  1 !
;  2301     3841  1 !--
;  2302     3842  1
;  2303     3843  2 BEGIN
;  2304     3844  2
;  2305     3845  2 MAP
;  2306     3846  2         CUR_BCB         : REF BBLOCK;   ! BCB input arg
;  2307     3847  2
;  2308     3848  2 LOCAL
;  2309     3849  2         STATUS,                         ! the usual status value
;  2310     3850  2         IO_STATUS       : VECTOR [4, WORD], ! I/O status block
;  2311     3851  2         FAB             : REF BBLOCK,   ! pointer to input FAB
;  2312     3852  2         BCB             : REF BBLOCK,   ! BCB of buffer in use
;  2313     3853  2         BUFFER          : REF BBLOCK,   ! current I/O buffer
;  2314     3854  2         NUMBER,                         ! output for decimal convert
;  2315     3855  2         ATT_CONTROL     : BBLOCK [12],  ! attribute control list
;  2316     3856  2         FILE_ID         : BBLOCK [FID$C_LENGTH], ! extension file ID
;  2317     3857  2         DEVICE_DESC     : VECTOR[2];    ! Input device desc if mounted /FOREIGN
;  2318     3858  2
;  2319     3859  2 EXTERNAL ROUTINE
;  2320     3860  2         FILE_ERROR,                     ! signal file related error
;  2321     3961  2         UNLOAD,                         ! rewind and unload tape
;  2322     3862  2         FREE_BUFFER;                    ! free an I/O buffer
;  2323     3863  2
;  2324     3864  2
;  2325     3865  2 ! Discard the BCB that reported the EOF. Return 0 (no continuation) if the
```

```
; 2326          3866  2 ! save set is being handled through the file system.
; 2327          3867  2 !
; 2328          3868  2
; 2329          3869  2 FAB = .RWSV_SAVE_FAB;
; 2330          3870  2 RWSV_IN_VBN = .CUR_BCB[BCB_BLOCKNUM];
; 2331          3871  2 FREE_BUFFER (.CUR_BCB);
; 2332          3872  2
; 2333          3873  2 IF .QUAL[QUAL_SS_FILE]
; 2334          3874  2 THEN RETURN 0;
; 2335          3875  2
; 2336          3876  2 ! For tape, read the trailer labels and see if a continuation exists.
; 2337          3877  2 !
; 2338          3878  2
; 2339          3879  2 IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
; 2340          3880  2 THEN
; 2341          3881  3     BEGIN
; 2342          3882  3     BCB = READ_BLOCK (FALSE, FALSE);
; 2343          3883  3     FREE_BUFFER (.BCB);
; 2344          3884  3
; 2345          3885  3     IF NOT .BCB[BCB_IO_STATUS]
; 2346          3886  3     THEN
; 2347          3887  4         BEGIN
; 2348          3888  4         IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFVOLUME
; 2349          3889  4         THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI)
; 2350          3890  4         ELSE FILE_ERROR (BACKUP$_LABELERR, .FAB, .BCB[BCB_IO_STATUS]);
; 2351          3891  4         END
; 2352          3892  4
; 2353          3893  3     ELSE IF .BCB[BCB_IO_BCOUNT] NEQ 80
; 2354          3894  3     THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
; 2355          3895  3
; 2356          3896  3     BUFFER = .BCB[BCB_BUFFER];
; 2357          3897  3     IF .BUFFER[HD1$L_HD1LID] EQL 'EOF1' OR .COM_FLAGS[COM_VERIFYING]
; 2358          3898  3     THEN RETURN 0;
; 2359          3899  3     IF .BUFFER[HD1$L_HD1LID] NEQ 'EOV1'
; 2360          3900  3     THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
; 2361          3901  3     END
; 2362          3902  3
; 2363          3903  3 ! For sequential disk, read the extension file ID and sequence number.
; 2364          3904  3 !
; 2365          3905  3
; 2366          3906  2 ELSE
; 2367          3907  3     BEGIN
; 2368          3908  3     IF .COM_FLAGS[COM_VERIFYING] THEN RETURN 0;
; 2369          3909  3     IF NOT .INPUT_MTL[MTL_SEQ_DISK] THEN RETURN 0;
; 2370          3910  3     ATT_CONTROL[ATR$W_SIZE] = ATR$S_EXTFID;
; 2371          3911  3     ATT_CONTROL[ATR$W_TYPE] = ATR$C_EXTFID;
; 2372          3912  3     ATT_CONTROL[ATR$L_ADDR] = FILE_ID;
; 2373          3913  3     ATT_CONTROL+8 = 0;
; 2374        P 3914  3     STATUS = $SQIOW (CHAN = .RWSV_CHAN,
; 2375        P 3915  3                      FUNC = IO$_ACCESS,
; 2376        P 3916  3                      IOSB = IO_STATUS,
; 2377        P 3917  3                      P5   = ATT_CONTROL
; 2378          3918  3                      );
; 2379          3919  3     IF .STATUS THEN STATUS = .IO_STATUS[0];
; 2380          3920  3     IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENIN, .FAB, .STATUS);
; 2381          3921  3     IF .FILE_ID[FID$W_NUM] EQL 0
; 2382          3922  3     AND .FILE_ID[FID$B_RVN] EQL 0
```

```
;  2383          3923  3     THEN RETURN 0;
;  2384          3924  3     IF .INPUT_MTL[MTL_SEQ_DISK]
;  2385          3925  3     AND .FILE_ID[FID$B_RVN] NEQ .RWSV_VOL_NUMBER + 1
;  2386          3926  3     THEN FILE_ERROR (BACKUP$_INVFILEXT, .FAB);
;  2387          3927  2     END;
;  2388          3928  2
;  2389          3929  2 ! Finish processing of the current volume. Then set up the new input
;  2390          3930  2 ! volume. Loop until the operator gets it right.
;  2391          3931  2 !
;  2392          3932  2
;  2393          3933  2 FIN_IN_SAVE (TRUE);
;  2394          3934  2 RWSV_SEG_NUMBER = .RWSV_SEG_NUMBER + 1;
;  2395          3935  2
;  2396          3936  2 ! Do a UFO open on the input FAB and save away the channel.
;  2397          3937  2 !
;  2398          3938  2
;  2399          3939  2 RWSV_SAVE_QUAL = .RWSV_SAVE_QUAL[QUAL_NEXT];
;  2400          3940  2 IF .RWSV_SAVE_QUAL EQL 0 THEN RWSV_SAVE_QUAL = .QUAL[QUAL_INPU_LIST];
;  2401          3941  2 RWSV_SAVE_FAB = FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
;  2402          3942  2
;  2403          3943  2 IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  2404          3944  2 THEN
;  2405          3945  3     BEGIN
;  2406          3946  3     RWSV_VOL_NUMBER = .RWSV_VOL_NUMBER + 1;
;  2407          3947  3     FAB[FAB$V_NAM] = TRUE;
;  2408          3948  3     FAB[FAB$V_UFO] = TRUE;
;  2409          3949  3
;  2410          3950  3 ! If the device is not mounted foreign do a $OPEN; otherwise do a $ASSIGN
;  2411          3951  3 !
;  2412          3952  3
;  2413          3953  3     IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
;  2414          3954  4     THEN STATUS = $OPEN (FAB = .FAB)
;  2415          3955  3     ELSE
;  2416          3956  4         BEGIN
;  2417          3957  4         DEVICE_DESC[0] = .BBLOCK[.FAB[FAB$L_NAM], NAM$B_DEV];
;  2418          3958  4         DEVICE_DESC[1] = .BBLOCK[.FAB[FAB$L_NAM], NAM$L_DEV];
;  2419        P 3959  4         STATUS = FAB[FAB$L_STS] = $ASSIGN (DEVNAM = DEVICE_DESC,
;  2420          3960  4                                            CHAN = FAB[FAB$L_STV]);
;  2421          3961  3         END;
;  2422          3962  3     IF NOT .STATUS
;  2423          3963  3     THEN FILE_ERROR (BACKUP$_OPENIN+STS$K_SEVERE, .FAB,
;  2424          3964  3                      .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
;  2425          3965  3     RWSV_CHAN = .FAB[FAB$L_STV];
;  2426          3966  3
;  2427          3967  3     UNTIL TRY_NEXT_VOLUME ()
;  2428          3968  3     DO UNLOAD ();
;  2429          3969  3     END
;  2430          3970  3
;  2431          3971  2 ELSE
;  2432          3972  3     BEGIN
;  2433          3973  3     RWSV_VOL_NUMBER = .FILE_ID[FID$B_RVN];
;  2434          3974  3     UNTIL TRY_NEXT_VOLUME (FILE_ID)
;  2435          3975  3     DO
;  2436          3976  4         BEGIN
;  2437          3977  4         IF NOT .INPUT_MTL[MTL_SEQ_DISK]
;  2438          3978  4         THEN FILE_ERROR (BACKUP$_READERR+STS$K_SEVERE, .FAB);
;  2439        P 3979  4         $QIOW (CHAN = .CURRENT_VCB[VCB_CHAN],
```

```
; 2440      P 3980  4                   FUNC = IO$_UNLOAD
; 2441        3981  4                   );
; 2442        3982  3           END;
; 2443        3983  2       END;
; 2444        3984  2
; 2445        3985  2   ! Read the first block from the input volume.
; 2446        3986  2   !
; 2447        3987  2
; 2448        3988  2   RWSV_IN_SEQ_0 = .RWSV_IN_SEQ;
; 2449        3989  2   RWSV_IN_VBN_0 = .RWSV_IN_VBN;
; 2450        3990  2   READ_SEQ_BLOCK (TRUE)
; 2451        3991  2
; 2452        3992  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                                               ; 3871
                        66              01    FB 00037        CALLS   #1, FREE_BUFFER                                  ;
      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$OPF.                          ;
```

```
                              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
```

RESTART
LIS

MAIN
LIS

READSAVE
LIS

LISTJOUR
LIS

OTHERMSG
LIS

MATCH
LIS