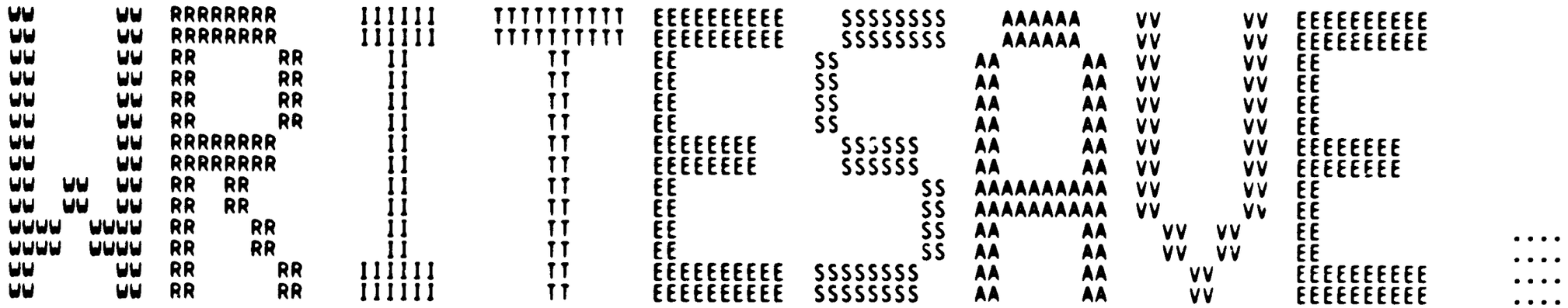
```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         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
```


WW WW RRRRRRRR IIIIII TTTTTTTTTT EEEEEEEEEE SSSSSSSS AAAAAA VV VV EEEEEEEEEE
WW WW RRRRRRRR IIIIII TTTTTTTTTT EEEEEEEEEE SSSSSSSS AAAAAA VV VV EEEEEEEEEE
WW WW RR RR II TT EE SS AA AA VV VV EE
WW WW RR RR II TT EE SS AA AA VV VV EE
WW WW RR RR II TT EE SS AA AA VV VV EE
WW WW RR RR II TT EE SS AA AA VV VV EE
WW WW RRRRRRRR II TT EEEEEEEE SSSSSS AA AA VV VV EEEEEEEE
WW WW RRRRRRRR II TT EEEEEEEE SSSSSS AA AA VV VV EEEEEEEE
WW WW WW RR RR II TT EE SS AAAAAAAAAA VV VV EE
WW WW WW RR RR II TT EE SS AAAAAAAAAA VV VV EE
WWWW WWWW RR RR II TT EE SS AA AA VV VV EE
WWWW WWWW RR RR II TT EE SS AA AA VV VV EE
WW WW RR RR IIIIII TT EEEEEEEEEE SSSSSSSS AA AA VV EEEEEEEEEE
WW WW RR RR IIIIII TT EEEEEEEEEE SSSSSSSS AA AA VV EEEEEEEEEE
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1          0001  0 MODULE WRITESAVE(%TITLE 'Write Save Set'
;     2          0002  0                   IDENT = 'V04-000'
;     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 writing
;    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 !       V03-011 LY0510          Larry Yetto             19-JUL-1984 08:43
;    51          0051  1 !               Add FORWARD ROUTINE definitions.  Backspace 2 tape marks at the
;    52          0052  1 !               end of a save set instead of 3.
;    53          0053  1 !
;    54          0054  1 !       V03-010 ACG0374         Andrew C. Goldstein,    17-Nov-1983  11:36
;    55          0055  1 !               Allow save set to start on full sequential disk
;    56          0056  1 !
;    57          0057  1 !       V03-009 ACG0332         Andrew C. Goldstein,    29-Apr-1983  18:28
```

```
;    58          0058  1 !               Add support for file highwater mark;
;    59          0059  1 !               remove .B32 from BACKDEF require file
;    60          0060  1 !
;    61          0061  1 !       V03-008 ACG0327         Andrew C. Goldstein,    11-Apr-1983  13:53
;    62          0062  1 !               Allow appending to full tape
;    63          0063  1 !
;    64          0064  1 !       V03-007 ACG0313         Andrew C. Goldstein,    10-Feb-1983  23:07
;    65          0065  1 !               Check for minimum space on disk.
;    66          0066  1 !
;    67          0067  1 !       V03-006 ACG0312         Andrew C. Goldstein,    4-Feb-1983  11:46
;    68          0068  1 !               Get best effort volume label if /NOREWIND
;    69          0069  1 !
;    70          0070  1 !       V03-005 ACG0295         Andrew C. Goldstein,    2-Jul-1982  12:04
;    71          0071  1 !               Use volume number to index into label list
;    72          0072  1 !
;    73          0073  1 !       V03-004 ACG0292         Andrew C. Goldstein,    4-Jun-1982  18:29
;    74          0074  1 !               Fix label length in sequential disk labels
;    75          0075  1 !
;    76          0076  1 !       V03-003 ACG0289         Andrew C. Goldstein,    16-Apr-1982  16:33
;    77          0077  1 !               Check for software write lock on tape
;    78          0078  1 !
;    79          0079  1 !       V03-002 ACG0276         Andrew C. Goldstein,    26-Mar-1982  19:07
;    80          0080  1 !               Clean up size checking in seq disk save sets
;    81          0081  1 !
;    82          0082  1 !       V03-001 ACG0269         Andrew C. Goldstein,    22-Mar-1982  16:57
;    83          0083  1 !               Use normal mode write in tape error rewrites
;    84          0084  1 !
;    85          0085  1 !       V02-015 ACG0257         Andrew C. Goldstein,    21-Jan-1982  18:51
;    86          0086  1 !               Add support for lists of volume labels
;    87          0087  1 !
;    88          0088  1 !       V02-014 ACG0256         Andrew C. Goldstein,    19-Jan-1982  22:02
;    89          0089  1 !               Add /PROTECTION and /OWNER to save set
;    90          0090  1 !
;    91          0091  1 !       V02-013 ACG0254         Andrew C. Goldstein,    19-Jan-1982  15:56
;    92          0092  1 !               Use completion AST to detect EOT
;    93          0093  1 !
;    94          0094  1 !       V02-012 ACG0243         Andrew C. Goldstein,    21-Dec-1981  8:31
;    95          0095  1 !               Detect EOT during write retries
;    96          0096  1 !
;    97          0097  1 !       V02-011 MLJ0054         Martin L. Jack, 31-Oct-1981  15:03
;    98          0098  1 !               Implement network save sets.  Move STAACP globals to common.
;    99          0099  1 !
;   100          0100  1 !       V02-010 ACG0218         Andrew C. Goldstein,    1-Oct-1981  13:14
;   101          0101  1 !               Inhibit normal write error recovery on /INTERCHANGE
;   102          0102  1 !
;   103          0103  1 !       V02-009 ACG0216         Andrew C. Goldstein,    4-Sep-1981  16:28
;   104          0104  1 !               Fix spacing over empty files in /NOREWIND
;   105          0105  1 !
;   106          0106  1 !       V02-008 ACG0211         Andrew C. Goldstein,    20-Jul-1981  15:32
;   107          0107  1 !               Implement sequential disk
;   108          0108  1 !
;   109          0109  1 !       V02-007 ACG209          Andrew C. Goldstein,    5-Jun-1981  15:23
;   110          0110  1 !               Stop on excessive output error rate
;   111          0111  1 !
;   112          0112  1 !       V02-006 MLJ0025         Martin L. Jack, 8-May-1981  14:54
;   113          0113  1 !               Reorganize qualifier database.  Move some global variables to
;   114          0114  1 !               common.  Make routines non-global if possible.
```

```
;  115          0115  1 !
;  116          0116  1 !       V02-005 ACG0202         Andrew C. Goldstein,    23-Apr-1981  16:47
;  117          0117  1 !               Fix handling of /SAVE on magtapes
;  118          0118  1 !
;  119          0119  1 !       V02-004 MLJ0020         Martin L. Jack, 20-Apr-1981  21:56
;  120          0120  1 !               Implement /JOURNAL qualifier.
;  121          0121  1 !
;  122          0122  1 !       V02-003 MLJ0010         Martin L. Jack, 25-Mar-1981  15:23
;  123          0123  1 !               Reorganize global storage.  Delete limiting of buffer count, as
;  124          0124  1 !               COMMAND now does this.  Correct some references to OQUA_OUTP_FC
;  125          0125  1 !               to be RWSV_SAVE_FAB.  Change OQUA_F'LE to OQUA_SAVE.
;  126          0126  1 !
;  127          0127  1 !       V02-002 ACG0199         Andrew C. Goldstein,    9-Mar-1981  13:51
;  128          0128  1 !               Fix reference to uninitialized RAB in WRITE_BLOCK
;  129          0129  1 !
;  130          0130  1 !       V02-001 ACG0187         Andrew C. Goldstein,    11-Feb-1981  19:34
;  131          0131  1 !               Make unrecoverable save set write errors fatal
;  132          0132  1 !
;  133          0133  1 !**
;  134          0134  1
;  135          0135  1
;  136          0136  1 LIBRARY 'SYS$LIBRARY:LIB';
;  137          0137  1 REQUIRE 'SRC$:COMMON';
;  138          1243  1 REQUIRE 'LIB$:BACKDEF';
;  139          1693  1
;  140          1694  1 EXTERNAL LITERAL
;  141          1695  1          BACKUP$_OPENOUT,
;  142          1696  1          BACKUP$_FATALERR,
;  143          1697  1          BACKUP$_WRITEERR,
;  144          1698  1          BACKUP$_CLOSEOUT,
;  145          1699  1          BACKUP$_LABELERR,
;  146          1700  1          BACKUP$_MAXVOLS,
;  147          1701  1          BACKUP$_NOTANSI,
;  148          1702  1          BACKUP$_CONTINUED,
;  149          1703  1          BACKUP$_DENSITY,
;  150          1704  1          BACKUP$_WRITERRS,
;  151          1705  1          BACKUP$_SAVSETCLU,
;  152          1706  1          BACKUP$_SOFTWERRS,
;  153          1707  1          BACKUP$_STARTVERIFY,
;  154          1708  1          BACKUP$_RESUME;
;  155          1709  1
;  156          1710  1 FORWARD ROUTINE
;  157          1711  1          WRITE_AST        : NOVALUE ,
;  158          1712  1          WRITE_ERROR      : NOVALUE ,
;  159          1713  1          WRITE_BLOCK      : NOVALUE ,
;  160          1714  1          INIT_SAVE_DISK   : NOVALUE ,
;  161          1715  1          INIT_SAVE_TAPE   : NOVALUE ,
;  162          1716  1          INIT_OUT_SAVE    : NOVALUE ,
;  163          1717  1          WRITE_BUFFER     : NOVALUE ,
;  164          1718  1          FIN_OUT_SAVE     : NOVALUE ;
;  165          1719  1
;  166          1720  1
;  167          1721  1 G$DEFINE();                ! Define global common area
```

```
;  169          1722  1 %SBTTL 'WRITE_AST - handle write completion AST'
;  170          1723  1 ROUTINE WRITE_AST (BCB, R0, R1, PC, PSL) : NOVALUE =
;  171          1724  1
;  172          1725  1 !++
;  173          1726  1 !
;  174          1727  1 ! FUNCTIONAL DESCRIPTION:
;  175          1728  1 !
;  176          1729  1 !       This routine is called as a completion AST when a tape write
;  177          1730  1 !       completes. It turns an EOT status into success and sets the
;  178          1731  1 !       global EOT flag.
;  179          1732  1 !
;  180          1733  1 ! CALLING SEQUENCE:
;  181          1734  1 !       WRITE_AST (BCB, R0, R1, PC, PSL)
;  182          1735  1 !
;  183          1736  1 ! INPUT PARAMETERS:
;  184          1737  1 !       BCB: (AST parameter) - address of BCB for this operation
;  185          1738  1 !       R0: R0 at time of AST
;  186          1739  1 !       R1: R1 at time of AST
;  187          1740  1 !       PC: PC at time of AST
;  188          1741  1 !       PSL: PSL at time of AST
;  189          1742  1 !
;  190          1743  1 ! IMPLICIT INPUTS:
;  191          1744  1 !       NONE
;  192          1745  1 !
;  193          1746  1 ! OUTPUT PARAMETERS:
;  194          1747  1 !       NONE
;  195          1748  1 !
;  196          1749  1 ! IMPLICIT OUTPUTS:
;  197          1750  1 !       COM_FLAGS[COM_EOV]: set if EOT encountered
;  198          1751  1 !
;  199          1752  1 ! ROUTINE VALUE:
;  200          1753  1 !       NONE
;  201          1754  1 !
;  202          1755  1 ! SIDE EFFECTS:
;  203          1756  1 !       NONE
;  204          1757  1 !
;  205          1758  1 !--
;  206          1759  1
;  207          1760  2 BEGIN
;  208          1761  2
;  209          1762  2 MAP
;  210          1763  2         BCB             : REF BBLOCK;   ! BCB argument
;  211          1764  2
;  212          1765  2 ! Check for EOT status. Further qualification is not necessary since
;  213          1766  2 ! this routine is only invoked when writing tape.
;  214          1767  2 !
;  215          1768  2
;  216          1769  2 IF .BCB[BCB_IO_STATUS] EQL SS$_ENDOFTAPE
;  217          1770  2 THEN
;  218          1771  3     BEGIN
;  219          1772  3     COM_FLAGS[COM_EOV] = TRUE;
;  220          1773  3     BCB[BCB_IO_STATUS] = TRUE;
;  221          1774  2     END;
;  222          1775  2
;  223          1776  1 END;                                    ! End of routine WRITE_AST
```

```
                        .TITLE  WRITESAVE Write Save Set
                        .IDENT  \V04-000\

                        .PSECT  COMMON,NOEXE,  OVR,2

00000                GLOBAL_BASE:
                        .BLKB   0
00000                FREE_LIST:
                        .BLKB   8
00008                INPUT_WAIT:
                        .BLKB   8
00010                REREAD_WAIT:
                        .BLKB   8
00018                OUTPUT_WAIT:
                        .BLKB   8
00020                JPI_UIC:.BLKB   4
00024                JPI_USERNAME:
                        .BLKB   12
00030                JPI_DATE:
                        .BLKB   8
00038                JPI_NODE_DESC:
                        .BLKB   8
00040                JPI_CURPRIV:
                        .BLKB   8
00048                SYI_VERSION:
                        .BLKB   4
0004C                SYI_SID:.BLKB   4
00050                RWSV_HOLD_LIST:
                        .BLKB   8
00058                RWSV_CRC16:
                        .BLKB   64
00098                RWSV_AUTODIN:
                        .BLKB   64
000D8                RWSV_FILESET_ID:
                        .BLKB   8
000E0                RWSV_VOLUME_ID:
                        .BLKB   12
000EC                RWSV_VOL_NUMBER:
                        .BLKB   2
000EE                RWSV_SEG_NUMBER:
                        .BLKB   2
000F0                RWSV_FILE_NUMBER:
                        .BLKB   4
000F4                RWSV_SAVE_QUAL:
                        .BLKB   4
000F8                RWSV_SAVE_FAB:
                        .BLKB   4
000FC                RWSV_CHAN:
                        .BLKB   4
00100                RWSV_XOR_BCB:
                        .BLKB   4
00104                RWSV_IN_SEQ:
                        .BLKB   4
00108                RWSV_IN_SEQ_0:
                        .BLKB   4
0010C                RWSV_IN_XOR_SEQ:
                        .BLKB   4
```

```
00110 RWSV_IN_XOR_RFA:
                .BLKB   6
00116 RWSV_LOOKAHEAD.
                .BLKB   1
00117 RWSV_XORSIZE:
                .BLKB   1
00118 RWSV_IN_GROUP_SIZE:
                .BLKB   4
0011C RWSV_IN_ERRORS:
                .BLKB   2
0011E RWSV_IN_XORUSE:
                .BLKB   2
00120 RWSV_IN_ORGERR:
                .BLKB   8
00128 RWSV_IN_VBN:
                .BLKB   4
0012C RWSV_IN_VBN_0:
                .BLKB   4
00130 RWSV_ALLOC:
                .BLKB   4
00134 RWSV_EOF:
                .BLKB   4
00138 RWSV_OUT_SEQ:
                .BLKB   4
0013C RWSV_OUT_VBN:
                .BLKB   4
00140 RWSV_OUT_BLOCK_COUNT:
                .BLKB   4
00144 RWSV_OUT_ERRORS:
                .BLKB   2
00146 RWSV_SEQ_ERRORS:
                .BLKB   2
00148 RWSV_OUT_GROUP_COUNT:
                .BLKB   1
00149 RWSV_PADDING:
                .BLKB   3
0014C QUAL:     .BLKB   112
001BC COM_SSNAME:
                .BLKB   8
001C4 COM_VALID_TYPES:
                .BLKB   2
001C6 COM_FLAGS:
                .BLKB   2
001C8 COM_PADDING:
                .BLKB   1
001C9 COM_BUFF_COUNT:
                .BLKB   1
001CA COM_I_SETCOUNT:
                .BLKB   1
001CB COM_O_SETCOUNT:
                .BLKB   1
001CC COM_I_STRUCNAME:
                .BLKB   12
001D8 COM_O_STRUCNAME:
                .BLKB   12
001E4 COM_O_BSRDATE:
                .BLKB   8
```

```
001EC ALT_SSNAME:
                .BLKB   32
0020C INPUT_FUNC:
                .BLKB   1
0020D INPUT_RTYPE:
                .BLKB   1
0020E OUTPUT_FUNC:
                .BLKB   1
0020F FAST_STRUCLEV:
                .BLKB   1
00210 INPUT_BEG:
                .BLKB   0
00210 INPUT_CHAN:
                .BLKB   4
00214 INPUT_FLAGS:
                .BLKB   2
00216 INPUT_PADDING:
                .BLKB   2
00218 INPUT_FAB:
                .BLKB   4
0021C INPUT_NAM:
                .BLKB   4
00220 INPUT_BCB:
                .BLKB   4
00224 INPUT_QUAL:
                .BLKB   4
00228 INPUT_BAD:
                .BLKB   4
0022C INPUT_BLOCK:
                .BLKB   4
00230 INPUT_MAXBLOCK:
                .BLKB   4
00234 INPUT_MEDIA_ID:
                .BLKB   4
00238 INPUT_NAMEDESC:
                .BLKB   8
00240 INPUT_STATBLK:
                .BLKB   8
00248 INPUT_HDR_BEG:
                .BLKB   0
00248 INPUT_CREDATE:
                .BLKB   8
00250 INPUT_REVDATE:
                .BLKB   8
00258 INPUT_EXPDATE:
                .BLKB   8
00260 INPUT_BAKDATE:
                .BLKB   8
00268 INPUT_FILEOWNER:
                .BLKB   4
0026C INPUT_FILECHAR:
                .BLKB   4
00270 INPUT_RECATTR:
                .BLKB   32
00290 INPUT_HDR_END:
                .BLKB   0
00290 INPUT_END:
```

```
                        .BLKB   0
00290 INPUT_PROC_LIST:
                        .BLKB   4
00294 INPUT_PLACEMENT:
                        .BLKB   8
0029C INPUT_VBN_LIST:
                        .BLKB   8
002A4 INPUT_PLACE_LEN:
                        .BLKB   2
002A6 INPUT_PADDING_2:
                        .BLKB   2
002A8 OUTPUT_BEG:
                        .BLKB   0
002A8 OUTPUT_CHAN:
                        .BLKB   4
002AC OUTPUT_FLAGS:
                        .BLKB   2
002AE OUTPUT_PADDING:
                        .BLKB   2
002B0 OUTPUT_FAB:
                        .BLKB   4
002B4 OUTPUT_NAM:
                        .BLKB   4
002B8 OUTPUT_BCB:
                        .BLKB   4
002BC OUTPUT_QUAL:
                        .BLKB   4
002C0 OUTPUT_BAD:
                        .BLKB   4
002C4 OUTPUT_BLOCK:
                        .BLKB   4
002C8 OUTPUT_MAXBLOCK:
                        .BLKB   4
002CC OUTPUT_DEVGEOM:
                        .BLKB   8
002D4 OUTPUT_ATTBUF:
                        .BLKB   144
00364 OUTPUT_END:
                        .BLKB   0
00364 LIST_TOTFILES:
                        .BLKB   4
00368 LIST_TOTSIZE:
                        .BLKB   4
0036C VERIFY_FAB:
                        .BLKB   4
00370 VERIFY_USE_COUNT:
                        .BLKB   4
00374 VERIFY_QUAL:
                        .BLKB   4
00378 COMPARE_BCB:
                        .BLKB   4
0037C FAST_BUFFER:
                        .BLKB   4
00380 FAST_BUFFER_SIZE:
                        .BLKB   4
00384 FAST_RVN:
                        .BLKB   1
```

```
00385 FAST_PADDING:
                .BLKB   1
00386 DIR_VERLIMIT:
                .BLKB   2
00388 FAST_VOL_BEG:
                .BLKB   0
00388 FAST_IMAP_SIZE:
                .BLKB   4
0038C FAST_IMAP:
                .BLKB   4
00390 FAST_HDR_OFFSET:
                .BLKB   4
00394 FAST_BOOT_LBN:
                .BLKB   4
00398 FAST_VOL_END:
                .BLKB   0
00398 JOUR_BUFFER:
                .BLKB   4
0039C JOUR_DIR:
                .BLKB   4
003A0 JOUR_HIBLK:
                .BLKB   4
003A4 JOUR_EFBLK:
                .BLKB   4
003A8 JOUR_INBLK:
                .BLKB   4
003AC JOUR_FFBYTE:
                .BLKB   2
003AE JOUR_INBYTE:
                .BLKB   2
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$_OPENOUT
                                .EXTRN  BACKUP$_FATALERR
```

```
                                                        .EXTRN  BACKUP$_WRITEERR
                                                        .EXTRN  BACKUP$_CLOSEOUT
                                                        .EXTRN  BACKUP$_LABELERR
                                                        .EXTRN  BACKUP$_MAXVOLS
                                                        .EXTRN  BACKUP$_NOTANSI
                                                        .EXTRN  BACKUP$_CONTINUED
                                                        .EXTRN  BACKUP$_DENSITY
                                                        .EXTRN  BACKUP$_WRITERRS
                                                        .EXTRN  BACKUP$_SAVSETCLU
                                                        .EXTRN  BACKUP$_SOFTWERRS
                                                        .EXTRN  BACKUP$_STARTVERIFY
                                                        .EXTRN  BACKUP$_RESUME

                                                        .PSECT  CODE,NOWRT,2

                                        0000 00000 WRITE_AST:
                                                        .WORD   Save nothing                    ; 1723
                        50      04      AC      D0 00002        MOVL    BCB, R0                 ; 1769
                  0878  8F      18      A0      B1 00006        CMPW    24(R0), #2168           ;
                                        0B      12 0000C        BNEQ    1$                      ;
          00000000'   EF              01      88 0000E        BISB2   #1, COM_FLAGS           ; 1772
                18    A0              01      B0 00015        MOVW    #1, 24(R0)              ; 1773
                                              04 00019 1$:      RET                             ; 1776

; Routine Size:  26 bytes,    Routine Base:  CODE + 0000
```

```
;  225          1777  1 %SBTTL 'WRITE_ERROR - handle write error'
;  226          1778  1 ROUTINE WRITE_ERROR (ERR_BCB) : NOVALUE =
;  227          1779  1
;  228          1780  1 !++
;  229          1781  1 !
;  230          1782  1 ! FUNCTIONAL DESCRIPTION:
;  231          1783  1 !
;  232          1784  1 !       This routine handles an error writing data to the output
;  233          1785  1 !       save set.
;  234          1786  1 !
;  235          1787  1 ! CALLING SEQUENCE:
;  236          1788  1 !       WRITE_ERROR (ERR_BCB)
;  237          1789  1 !
;  238          1790  1 ! INPUT PARAMETERS:
;  239          1791  1 !       ERR_BCB: buffer control block on which the error occurred
;  240          1792  1 !
;  241          1793  1 ! IMPLICIT INPUTS:
;  242          1794  1 !       NONE
;  243          1795  1 !
;  244          1796  1 ! OUTPUT PARAMETERS:
;  245          1797  1 !       NONE
;  246          1798  1 !
;  247          1799  1 ! IMPLICIT OUTPUTS:
;  248          1800  1 !       NONE
;  249          1801  1 !
;  250          1802  1 ! ROUTINE VALUE:
;  251          1803  1 !       NONE
;  252          1804  1 !
;  253          1805  1 ! SIDE EFFECTS:
;  254          1806  1 !       NONE
;  255          1807  1 !
;  256          1808  1 !--
;  257          1809  1
;  258          1810  2 BEGIN
;  259          1811  2
;  260          1812  2 BUILTIN
;  261          1813  2         INSQUE,
;  262          1814  2         REMQUE;
;  263          1815  2
;  264          1816  2 MAP
;  265          1817  2         ERR_BCB         : REF BBLOCK;   ! BCB argument
;  266          1818  2
;  267          1819  2 LITERAL
;  268          1820  2         RETRY_COUNT     = 32;           ! number of times to retry a write
;  269          1821  2
;  270          1822  2 LOCAL
;  271          1823  2         FAB             : REF BBLOCK,   ! FAB of output file
;  272          1824  2         BCB             : REF BBLOCK,   ! random BCB pointer
;  273          1825  2         TEMP1,                          ! temps to hold action routines
;  274          1826  2         TEMP2;
;  275          1827  2
;  276          1828  2 EXTERNAL ROUTINE
;  277          1829  2         WAIT,                           ! wait for I/O completion
;  278          1830  2         WRITE_BLOCK,                    ! write a buffer to output
;  279          1831  2         FREE_BUFFER,                    ! free a block buffer
;  280          1832  2         FILE_ERROR;                     ! signal file related error
;  281          1833  2
```

```
;  282          1834  2 ! Asynchronous output errors are handled for tape and seq disk. What
;  283          1835  2 ! we do is first wait for completion of all pending writes, ignoring
;  284          1836  2 ! completion status, since they are now junk.
;  285          1837  2 !
;  286          1838  2
;  287          1839  2 FAB = .RWSV_SAVE_FAB;
;  288          1840  2 RWSV_OUT_ERRORS = .RWSV_OUT_ERRORS + 1;
;  289          1841  2 IF NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
;  290          1842  2 OR .QUAL[QUAL_INTE]
;  291          1843  2 THEN
;  292          1844  3     BEGIN
;  293          1845  3     FREE_BUFFER (.ERR_BCB);
;  294          1846  3     FILE_ERROR (BACKUP$_WRITEERR+STS$K_SEVFRE, .FAB, .ERR_BCB[BCB_IO_STATUS]);
;  295          1847  3     RETURN;
;  296          1848  2     END;
;  297          1849  2
;  298          1850  2 UNTIL REMQUE (.OUTPUT_WAIT[0], BCB)
;  299          1851  2 DO
;  300          1852  3     BEGIN
;  301          1853  3     TEMP1 = .BCB[BCB_SUCC_ACT];
;  302          1854  3     TEMP2 = .BCB[BCB_FAIL_ACT];
;  303          1855  3     BCB[BCB_SUCC_ACT] = 0;
;  304          1856  3     BCB[BCB_FAIL_ACT] = 0;
;  305          1857  3     WAIT (.BCB);
;  306          1858  3     BCB[BCB_SUCC_ACT] = .TEMP1;
;  307          1859  3     BCB[BCB_FAIL_ACT] = .TEMP2;
;  308          1860  3     INSQUE (.BCB, .RWSV_HOLD_LIST[1]);
;  309          1861  2     END;
;  310          1862  2
;  311          1863  2 ! Iterate in the following loop until either we get the block in question
;  312          1864  2 ! written successfully or the operator gives up.
;  313          1865  2 !
;  314          1866  2 ! Special case medium offline, since it clearly won't go away by itself.
;  315          1867  2 ! Check the output error rate. If it is excessive, complain.
;  316          1868  2 !
;  317          1869  2
;  318          1870  2 WHILE TRUE
;  319          1871  2 DO
;  320          1872  3     BEGIN
;  321          1873  3     IF .RWSV_OUT_ERRORS GTRU 100
;  322          1874  3     AND .RWSV_OUT_BLOCK_COUNT / .RWSV_OUT_ERRORS LSSU 10
;  323          1875  3     AND NOT .COM_FLAGS[COM_CONTINUE]
;  324          1876  3     THEN
;  325          1877  4         BEGIN
;  326          1878  4         INSQUE(.ERR_BCB, RWSV_HOLD_LIST[0]);
;  327          1879  4         FILE_ERROR (BACKUP$_WRITERRS, .FAB, .ERR_BCB[BCB_IO_STATUS]);
;  328          1880  4         COM_FLAGS[COM_CONTINUE] = TRUE;
;  329          1881  4         REMQUE (.ERR_BCB, BCB);
;  330          1882  3         END;
;  331          1883  3
;  332          1884  3 ! Now attempt to rewrite the block in error. Wait for completion and
;  333          1885  3 ! check status; retry until it succeeds or we lose patience.
;  334          1886  3 ! Note that the first rewrite is done with error recovery inhibited,
;  335          1887  3 ! as are normal writes. After the first retry, we enable normal error
;  336          1888  3 ! recovery in the driver. The reason for this is that for some unknown
;  337          1889  3 ! reason, errors sometimes get "stuck" in the tape controller even
;  338          1890  3 ! though the tape is good. The driver's error retry sequence appears
```

```
;  339          1891  3 ! to clear them when simple transfer operations don't.
;  340          1892  3 !
;  341          1893  3
;  342          1894  3     DECR J FROM RETRY_COUNT TO 1
;  343          1895  3     DO
;  344          1896  4         BEGIN
;  345          1897  4         RWSV_OUT_BLOCK_COUNT = .RWSV_OUT_BLOCK_COUNT + 1;
;  346          1898  4         WRITE_BLOCK (.ERR_BCB, .J NEQ RETRY_COUNT);
;  347          1899  4         ERR_BCB[BCB_FAIL_ACT] = 0;
;  348          1900  4         REMQUE (.ERR_BCB, BCB);
;  349          1901  4         WAIT (.ERR_BCB);
;  350          1902  4         IF .ERR_BCB[BCB_IO_STATUS]
;  351          1903  4         THEN EXITLOOP;
;  352          1904  4         RWSV_OUT_ERRORS = .RWSV_OUT_ERRORS + 1;
;  353          1905  4         IF .ERR_BCB[BCB_IO_STATUS] EQL SS$_MEDOFL
;  354          1906  4         OR .ERR_BCB[BCB_IO_STATUS] EQL SS$_VOLINV
;  355          1907  4         THEN EXITLOOP;
;  356          1908  3         END;
;  357          1909  3
;  358          1910  3     IF .ERR_BCB[BCB_IO_STATUS] THEN EXITLOOP;
;  359          1911  3
;  360          1912  3     INSQUE (.ERR_BCB, RWSV_HOLD_LIST[0]);
;  361          1913  3     FILE_ERROR (BACKUP$_FATALERR, .FAB, .ERR_BCB[BCB_IO_STATUS]);
;  362          1914  3     REMQUE (.ERR_BCB, BCB);
;  363          1915  2     END;
;  364          1916  2
;  365          1917  2 ! Now re-issue the writes on all the other buffers that were queued
;  366          1918  2 ! for write. The presumption is that if the above write finally
;  367          1919  2 ! succeeded, these will now, too.
;  368          1920  2 !
;  369          1921  2
;  370          1922  2 UNTIL REMQUE (.RWSV_HOLD_LIST[0], BCB)
;  371          1923  2 DO
;  372          1924  3     BEGIN
;  373          1925  3     RWSV_OUT_BLOCK_COUNT = .RWSV_OUT_BLOCK_COUNT + 1;
;  374          1926  3     WRITE_BLOCK (.BCB);
;  375          1927  2     END;
;  376          1928  2
;  377          1929  1 END;                                     ! End of routine WRITE_ERROR


                                                                .EXTRN  WAIT, WRITE_BLOCK
                                                                .EXTRN  FREE_BUFFER, FILE_ERROR

                                                03FC 00000 WRITE_ERROR:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9            ; 1778
                                    59 00000000G   00   9E 00002        MOVAB   WRITE_BLOCK, R9                 ;
                                    58 00000000G   00   9E 00009        MOVAB   WAIT, R8                        ;
                                    57 00000000G   00   9E 00010        MOVAB   FILE_ERROR, R7                  ;
                                    56 00000000'   EF   9E 00017        MOVAB   RWSV_OUT_ERRORS, R6             ;
                                    55        B4   A6   D0 0001E        MOVL    RWSV_SAVE_FAB, FAB              ; 1839
                                                   66   B6 00022        INCW    RWSV_OUT_ERRORS                 ; 1840
                              05          40   A5        05   E1 00024        BBC     #5, 64(FAB), 1$         ; 1841
                              1D          16   A6        01   E1 00029        BBC     #1, QUAL+14, 2$         ; 1842
                                    52        04   AC   D0 0002E 1$:      MOVL    ERR_BCB, R2                     ; 1845
                                                   52   DD 00032        PUSHL   R2                              ;
```

```
                00000000G  00              01  FB 00034          CALLS   #1, FREE_BUFFER
                           7E        18     A2  3C 0003B          MOVZWL  24(R2), -(SP)                   ;1846
                                            55  DD 0003F          PUSHL   FAB
                                 00000000G  8F  DD 00041          PUSHL   #BACKUP$_WRITEERR+4
                           67               03  FB 00047          CALLS   #3, FILE_ERROR
                                            04  0004A          RET                                     ;1844
                           52      FED4     D6  0F 0004B 2$:     REMQUE  @OUTPUT_WAIT, BCB               ;1850
                                            1F  1D 00050          BVS     3$
                           54        20     A2  D0 00052          MOVL    32(BCB), TEMP1                  ;1853
                           53        24     A2  D0 00056          MOVL    36(BCB), TEMP2                  ;1854
                                     20     A2  7C 0005A          CLRQ    32(BCB)                         ;1855
                                            52  DD 0005D          PUSHL   BCB                             ;1857
                           68               01  FB 0005F          CALLS   #1, WAIT
                        20 A2               54  D0 00062          MOVL    TEMP1, 32(BCB)                  ;1858
                        24 A2               53  D0 00066          MOVL    TEMP2, 36(BCB)                  ;1859
                      FF10 D6               62  0E 0006A          INSQUE  (BCB), @RWSV_HOLD_LIST+4        ;1860
                                            DA  11 0006F          BRB     2$                              ;1850
                      0064 8F               66  B1 00071 3$:     CMPW    RWSV_OUT_ERRORS, #100           ;1873
                                            34  1B 00076          BLEQU   4$
                           50               66  3C 00078          MOVZWL  RWSV_OUT_ERRORS, R0             ;1874
                50        FC A6             50  C7 0007B          DIVL3   R0, RWSV_OUT_BLOCK_COUNT, R0
                           0A               50  D1 00080          CMPL    R0, #10
                                            27  1E 00083          BGEQU   4$
                21    0082 C6               05  E0 00085          BBS     #5, COM_FLAGS, 4$               ;1875
                      FF0C C6        04     BC  0E 0008B          INSQUE  @ERR_BCB, RWSV_HOLD_LIST        ;1878
                           53        04     AC  D0 00091          MOVL    ERR_BCB, R3                     ;1879
                           7E        18     A3  3C 00095          MOVZWL  24(R3), -(SP)
                                            55  DD 00099          PUSHL   FAB
                                 00000000G  8F  DD 0009B          PUSHL   #BACKUP$_WRITERRS
                           67               03  FB 000A1          CALLS   #3, FILE_ERROR
                0082       C6               20  88 000A4          BISB2   #32, COM_FLAGS                  ;1880
                           52               63  0F 000A9          REMQUE  (R3), BCB                       ;1881
                           53        04     AC  D0 000AC 4$:     MOVL    ERR_BCB, R3                     ;1898
                           54               20  D0 000B0          MOVL    #32, J
                                     FC     A6  D6 000B3 5$:     INCL    RWSV_OUT_BLOCK_COUNT            ;1897
                                            7E  D4 000B6          CLRL    -(SP)                           ;1898
                           20               54  D1 000B8          CMPL    J, #32
                                            02  13 000BB          BEQL    6$
                                            6E  D6 000BD          INCL    (SP)
                                            53  DD 000BF 6$:     PUSHL   R3
                           69               02  FB 000C1          CALLS   #2, WRITE_BLOCK
                                     24     A3  D4 000C4          CLRL    36(R3)                          ;1899
                           52               63  0F 000C7          REMQUE  (R3), BCB                       ;1900
                           53        04     AC  D0 000CA          MOVL    ERR_BCB, R3                     ;1901
                                            53  DD 000CE          PUSHL   R3
                           68               01  FB 000D0          CALLS   #1, WAIT
                           15        18     A3  E8 000D3          BLBS    24(R3), 7$                      ;1902
                                            66  B6 000D7          INCW    RWSV_OUT_ERRORS                 ;1904
                0A4        8F        18     A3  B1 000D9          CMPW    24(R3), #420                    ;1905
                                            0B  13 000DF          BEQL    7$
                0254       8F        18     A3  B1 000E1          CMPW    24(R3), #596                    ;1906
                                            03  13 000E7          BEQL    7$
                           C7               54  F5 000E9          SOBGTR  J, 5$                           ;1894
                           50        04     AC  D0 000EC 7$:     MOVL    ERR_BCB, R0                     ;1910
                           1E        18     A0  E8 000F0          BLBS    24(R0), 8$
                FF0C       C6               60  0E 000F4          INSQUE  (R0), RWSV_HOLD_LIST            ;1912
                           53        04     AC  D0 000F9          MOVL    ERR_BCB, R3                     ;1913
```

```
7E        18   A3  3C 000FD          MOVZWL  24(R3), -(SP)             ;
                55  DD 00101          PUSHL   FAB                       ;
     00000000G  8F  DD 00103          PUSHL   #BACKUP$_FATALERR         ;
67              03  FB 00109          CALLS   #3, FILE_ERROR            ;
52              63  0F 0010C          REMQUE  (R3), BCB                 ; 1914
              FF5F  31 0010F          BRW     3$                        ; 1870
52      FF0C    D6  0F 00112 8$:      REMQUE  @RWSV_HOLD_LIST, BCB      ; 1922
                0A  1D 00117          BVS     9$                        ;
          FC    A6  D6 00119          INCL    RWSV_OUT_BLOCK_COUNT      ; 1925
                52  DD 0011C          PUSHL   BCB                       ; 1926
69              01  FB 0011E          CALLS   #1, WRITE_BLOCK           ;
                EF  11 00121          BRB     8$                        ; 1922
                    04 00123 9$:      RET                               ; 1929
```

; Routine Size: 292 bytes,   Routine Base: CODE + 001A

```
;  379          1930  1 %SBTTL 'WRITE_BLOCK - write save set block'
;  380          1931  1 GLOBAL ROUTINE WRITE_BLOCK (BCB, RETRY) : NOVALUE =
;  381          1932  1
;  382          1933  1 !++
;  383          1934  1 !
;  384          1935  1 ! FUNCTIONAL DESCRIPTION:
;  385          1936  1 !
;  386          1937  1 !       This routine issues a write I/O for the specified buffer
;  387          1938  1 !       onto the output save medium.
;  388          1939  1 !
;  389          1940  1 ! CALLING SEQUENCE:
;  390          1941  1 !       WRITE_BLOCK (BCB, RETRY)
;  391          1942  1 !
;  392          1943  1 ! INPUT PARAMETERS:
;  393          1944  1 !       BCB: buffer control block to be written
;  394          1945  1 !       RETRY: (optional) 0 for normal no-retry tape write
;  395          1946  1 !                         1 for write with error recovery enabled
;  396          1947  1 !
;  397          1948  1 ! IMPLICIT INPUTS:
;  398          1949  1 !       NONE
;  399          1950  1 !
;  400          1951  1 ! OUTPUT PARAMETERS:
;  401          1952  1 !       NONE
;  402          1953  1 !
;  403          1954  1 ! IMPLICIT OUTPUTS:
;  404          1955  1 !       NONE
;  405          1956  1 !
;  406          1957  1 ! ROUTINE VALUE:
;  407          1958  1 !       NONE
;  408          1959  1 !
;  409          1960  1 ! SIDE EFFECTS:
;  410          1961  1 !       NONE
;  411          1962  1 !
;  412          1963  1 !--
;  413          1964  1
;  414          1965  2 BEGIN
;  415          1966  2
;  416          1967  2 BUILTIN
;  417          1968  2         ACTUALCOUNT,
;  418          1969  2         CRC,
;  419          1970  2         INSQUE;
;  420          1971  2
;  421          1972  2 MAP
;  422          1973  2         BCB             : REF BBLOCK;   ! buffer control block arg
;  423          1974  2
;  424          1975  2 LOCAL
;  425          1976  2         STATUS,                         ! general status value
;  426          1977  2         HDR_SUM,                        ! CRC of block header
;  427          1978  2         BLOCK_SUM,                      ! CRC of entire block
;  428          1979  2         BUFFER          : REF BBLOCK,   ! data buffer being written
;  429          1980  2         RAB             : REF BBLOCK,   ! output RAB
;  430          1981  2         BLOCK_COUNT,                    ! number of blocks about to be written
;  431          1982  2         BLOCKS_ALLOC;                   ! blocks receiver from allocator
;  432          1983  2
;  433          1984  2 EXTERNAL ROUTINE
;  434          1985  2         FREE_BUFFER,                    ! free an I/O buffer
;  435          1986  2         SWITCH_VOLUME,                  ! switch to desired disk volume
```

```
;  436          1987  2          STA_EXTEND,                     ! extend save set file
;  437          1988  2          FILE_ERROR;                     ! signal file related error
;  438          1989  2
;  439          1990  2 ! Set up pointers and compute the CRC's.
;  440          1991  2 !
;  441          1992  2
;  442          1993  2 BUFFER = .BCB[BCB_BUFFER];
;  443          1994  2 BUFFER[BBH$W_CHECKSUM] = 0;
;  444          1995  2 BUFFER[BBH$L_CRC] = 0;
;  445          1996  2 BLOCK_SUM = -1;
;  446          1997  2 IF .QUAL[QUAL_CRC]
;  447          1998  2 THEN CRC (RWSV_AUTODIN, %REF (-1), BCB[BCB_SIZE], .BUFFER, BLOCK_SUM)
;  448          1999  2 ELSE BUFFER[BBH$V_NOCRC] = TRUE;
;  449          2000  2
;  450          2001  2 CRC (RWSV_CRC16, %REF (0), %REF (BBH$K_LENGTH), .BUFFER, HDR_SUM);
;  451          2002  2 BUFFER[BBH$W_CHECKSUM] = .HDR_SUM;
;  452          2003  2 BUFFER[BBH$L_CRC] = NOT .BLOCK_SUM;
;  453          2004  2
;  454          2005  2 ! Issue the write I/O appropriate to the output medium.
;  455          2006  2 !
;  456          2007  2
;  457          2008  2 BCB[BCB_SUCC_ACT] = 0;
;  458          2009  2 BCB[BCB_FAIL_ACT] = WRITE_ERROR;
;  459          2010  2
;  460          2011  2 ! Write through file system.
;  461          2012  2 !
;  462          2013  2
;  463          2014  2 IF .QUAL[QUAL_SS_FILE]
;  464          2015  2 THEN
;  465          2016  3     BEGIN
;  466          2017  3     RAB = RWSV_SAVE_FAB[FC_RAB];
;  467          2018  3     RAB[RAB$W_RSZ] = .BCB[BCB_SIZE];
;  468          2019  3     RAB[RAB$L_RBF] = .BUFFER;
;  469          2020  3     IF .RWSV_SAVE_FAB[FAB$V_BIO]
;  470          2021  4         THEN STATUS = $WRITE (RAB = .RAB)
;  471          2022  3         ELSE STATUS = $PUT (RAB = .RAB);
;  472          2023  3     IF NOT .STATUS
;  473          2024  3     THEN FILE_ERROR (BACKUP$_WRITEERR+STS$K_SEVERE,
;  474          2025  3                      .RWSV_SAVE_FAB, .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
;  475          2026  3     FREE_BUFFER (.BCB);
;  476          2027  3     END
;  477          2028  3
;  478          2029  2 ELSE
;  479          2030  3     BEGIN
;  480          2031  3
;  481          2032  3 ! Write to tape.
;  482          2033  3 !
;  483          2034  3
;  484          2035  3     IF .BBLOCK [RWSV_SAVE_FAB[FAB$L_DEV], DEV$V_SQD]
;  485          2036  3     THEN
;  486          2037  4         BEGIN
;  487        P 2038  4         STATUS = $QIO  (CHAN = .RWSV_CHAN,
;  488        P 2039  4                         FUNC = (IF .QUAL[QUAL_INTE]
;  489        P 2040  4                                 OR (IF ACTUALCOUNT () GEQU 2
;  490        P 2041  4                                     THEN .RETRY
;  491        P 2042  4                                     ELSE 0)
;  492        P 2043  4                                 THEN IO$_WRITELBLK
```

```
;  493        P 2044  4                                   ELSE IO$_WRITELBLK OR IO$M_INHRETRY),
;  494        P 2045  4                          EFN  = BCB_S_WRITE,
;  495        P 2046  4                          IOSB = BCB[BCB_IOSB],
;  496        P 2047  4                          ASTADR = WRITE_AST,
;  497        P 2048  4                          ASTPRM = .BCB,
;  498        P 2049  4                          P1   = .BUFFER,
;  499        P 2050  4                          P2   = .BCB[BCB_SIZE]
;  500           2051  4                          );
;  501           2052  4          IF NOT .STATUS
;  502           2053  4          THEN FILE_ERROR (BACKUP$_WRITEERR+STS$K_SEVERE,
;  503           2054  4                           .RWSV_SAVE_FAB, .STATUS);
;  504           2055  4          BCB[BCB_STATE] = BCB_S_WRITE;
;  505           2056  4          INSQUE (.BCB, .OUTPUT_QAIT[1]);
;  506           2057  4          END
;  507           2058  4
;  508           2059  4 ! Write to sequential disk. Make sure space is available for both the
;  509           2060  4 ! current block, the next block, and the XOR block to follow. If the
;  510           2061  4 ! allocation fails, set EOV for force a volume switch before the next
;  511           2062  4 ! write.
;  512           2063  4 !
;  513           2064  4
;  514           2065  3     ELSE
;  515           2066  4          BEGIN
;  516           2067  4          CURRENT_MTL = .OUTPUT_MTL;
;  517           2068  4          SWITCH_VOLUME (.RWSV_VOL_NUMBER);
;  518           2069  4          BLOCK_COUNT = (.BCB[BCB_SIZE]+511) / 512;
;  519           2070  4          WHILE .RWSV_ALLOC LSSU
;  520           2071  5              (IF .QUAL[QUAL_GROU_VALUE] NEQ 0
;  521           2072  5               THEN .BLOCK_COUNT * 4
;  522           2073  5               ELSE .BLOCK_COUNT * 3)
;  523           2074  4          DO
;  524           2075  5              BEGIN
;  525           2076  5              STATUS = STA_EXTEND (1^30, BLOCKS_ALLOC);
;  526           2077  5              IF .STATUS
;  527           2078  5              THEN
;  528           2079  5                  RWSV_ALLOC = .RWSV_ALLOC + .BLOCKS_ALLOC
;  529           2080  5              ELSE
;  530           2081  6                  BEGIN
;  531           2082  6                  COM_FLAGS[COM_EOV] = 1;
;  532           2083  6                  EXITLOOP;
;  533           2084  5                  END;
;  534           2085  4              END;
;  535           2086  4
;  536        P 2087  4          STATUS = $$QIO (CHAN = .RWSV_CHAN,
;  537        P 2088  4                          FUNC = IO$_WRITEVBLK,
;  538        P 2089  4                          EFN  = BCB_S_WRITE,
;  539        P 2090  4                          IOSB = BCB[BCB_IOSB],
;  540        P 2091  4                          P1   = .BUFFER,
;  541        P 2092  4                          P2   = .BCB[BCB_SIZE],
;  542        P 2093  4                          P3   = .RWSV_OUT_VBN
;  543           2094  4                          );
;  544           2095  4          IF NOT .STATUS
;  545           2096  4          THEN FILE_ERROR (BACKUP$_WRITEERR+STS$K_SEVERE,
;  546           2097  4                           .RWSV_SAVE_FAB, .STATUS);
;  547           2098  4          BCB[BCB_STATE] = BCB_S_WRITE;
;  548           2099  4          INSQUE (.BCB, .OUTPUT_QAIT[1]);
;  549           2100  4          RWSV_OUT_VBN = .RWSV_OUT_VBN + .BLOCK_COUNT;
```

```
;  550          2101 4            RWSV_ALLOC = .RWSV_ALLOC - .BLOCK_COUNT;
;  551          2102 3            END;
;  552          2103 2        END;
;  553          2104 2
;  554          2105 1 END;                                      ! End of routine WRITE_BLOCK


                                                             .EXTRN  SWITCH_VOLUME, STA_EXTEND
                                                             .EXTRN  SYS$WRITE, SYS$PUT
                                                             .EXTRN  SYS$QIO, STA_QIO

                                           03FC 00000        .ENTRY  WRITE_BLOCK, Save R2,R3,R4,R5,R6,R7,R8,R9   ; 1931
                 59 00000000G  00       9E 00002             MOVAB   FILE_ERROR, R9                              ;
                 58 00000000G  8F       D0 00009             MOVL    #BACKUP$_WRITEERR+4, R8                     ;
                 57 00000000'  EF       9E 00010             MOVAB   RWSV_SAVE_FAB, R7                           ;
                 5E            04       C2 00017             SUBL2   #4, SP                                      ;
                 54       04   AC       D0 0001A             MOVL    BCB, R4                                     ; 1993
                 55       0C   A4       D0 0001E             MOVL    12(R4), BUFFER                              ;
                        00FE   C5       B4 00022             CLRW    254(BUFFER)                                 ; 1994
                          24   A5       D4 00026             CLRL    36(BUFFER)                                  ; 1995
                 56            01       CE 00029             MNEGL   #1, BLOCK_SUM                               ; 1996
        10       5D            A7    01 E1 0002C             BBC     #1, QUAL+9, 1$                              ; 1997
65    08  A4 FFFFFFFF  8F      A0    A7 0B 00031             CRC     RWSV_AUTODIN, #-1, 8(R4), (BUFFER)          ; 1998
                 56            50       D0 0003C             MOVL    R0, BLOCK_SUM                               ;
                               04       11 0003F             BRB     2$                                          ;
                 2C   A5       01       88 00041 1$:         BISB2   #1, 44(BUFFER)                              ; 1999
65  0100  8F          00  FF60 C7       0B 00045 2$:         CRC     RWSV_CRC16, #0, #256, (BUFFER)              ; 2001
               00FE   C5       50       B0 0004E             MOVW    HDR_SUM, 254(BUFFER)                        ; 2002
                 24   A5       56       D2 00053             MCOML   BLOCK_SUM, 36(BUFFER)                       ; 2003
                          20   A4       D4 00057             CLRL    32(R4)                                      ; 2008
                 24   A4  FE7E CF       9E 0005A             MOVAB   WRITE_ERROR, 36(R4)                         ; 2009
                      50       67       D0 00060             MOVL    RWSV_SAVE_FAB, R0                           ; 2017
        41       63   A7       03       E1 00063             BBC     #3, QUAL+15, 6$                             ; 2014
                      52  50   A0       9E 00068             MOVAB   80(R0), RAB                                 ; 2017
                 22   A2  08   A4       B0 0006C             MOVW    8(R4), 34(RAB)                              ; 2018
                 28   A2       55       D0 00071             MOVL    BUFFER, 40(RAB)                             ; 2019
        0B       16   A0       05       E1 00075             BBC     #5, 22(R0), 3$                              ; 2020
                               52       DD 0007A             PUSHL   RAB                                         ; 2021
         00000000G    00       01       FB 0007C             CALLS   #1, SYS$WRITE                               ;
                               09       11 00083             BRB     4$                                          ;
                               52       DD 00085 3$:         PUSHL   RAB                                         ; 2022
         00000000G    00       01       FB 00087             CALLS   #1, SYS$PUT                                 ;
                      53       50       D0 0008E 4$:         MOVL    R0, STATUS                                  ;
                      0B       53       E8 00091             BLBS    STATUS, 5$                                  ; 2023
                      7E  08   A2       7D 00094             MOVQ    8(RAB), -(SP)                               ; 2025
                               67       DD 00098             PUSHL   RWSV_SAVE_FAB                               ;
                               58       DD 0009A             PUSHL   R8                                          ; 2024
                      69       04       FB 0009C             CALLS   #4, FILE_ERROR                              ;
                               54       DD 0009F 5$:         PUSHL   R4                                          ; 2026
         00000000G    00       01       FB 000A1             CALLS   #1, FREE_BUFFER                             ;
                                        04 000A8             RET                                                 ; 2014
                      56  18   A4       9E 000A9 6$:         MOVAB   24(R4), R6                                  ; 2051
        4C       40   A0       05       E1 000AD             BBC     #5, 64(R0), 11$                             ; 2035
                               7E       7C 000B2             CLRQ    -(SP)                                       ; 2051
                               7E       7C 000B4             CLRQ    -(SP)                                       ;
                      7E  08   A4       3C 000B6             MOVZWL  8(R4), -(SP)                                ;
```

```
                                    30  BB 000BA          PUSHR   #^M<R4,R5>                        ;
                           FE02     CF  9F 000BC          PUSHAB  WRITE_AST                         ;
                                    56  DD 000C0          PUSHL   R6                                ;
  09          62    A7              01  E0 000C2          BBS     #1, QUAL+14, 7$                   ;
                    02              6C  91 000C7          CMPB    (AP), #2                          ;
                                    08  1F 000CA          BLSSU   8$                                ;
                    04       08     AC  E9 000CC          BLBC    RETRY, 8$                         ;
                                    20  DD 000D0 7$:      PUSHL   #32                               ;
                                    05  11 000D2          BRB     9$                                ;
                    7E     8020     8F  3C 000D4 8$:      MOVZWL  #32800, -(SP)                     ;
                             04     A7  DD 000D9 9$:      PUSHL   RWSV_CHAN                         ;
                                    02  DD 000DC          PUSHL   #2                                ;
         00000000G  00              0C  FB 000DE          CALLS   #12, SYS$QIO                      ;
                    53              50  D0 000E5          MOVL    R0, STATUS                        ;
                    09              53  E8 000E8          BLBS    STATUS, 10$                       ; 2052
                                    53  DD 000EB          PUSHL   STATUS                            ; 2054
                                    67  DD 000ED          PUSHL   RWSV_SAVE_FAB                     ;
                                    58  DD 000EF          PUSHL   R8                                ; 2053
                    69              03  FB 000F1          CALLS   #3, FILE_ERROR                    ;
              0A    A4              02  90 000F4 10$:     MOVB    #2, 10(R4)                        ; 2055
            FF24    D7              64  0E 000F8          INSQUE  (R4), @OUTPUT_WAIT+4              ; 2056
                                        04 000FD          RET                                       ; 2035
            06CC    C7     06C8     C7  D0 000FE 11$:     MOVL    OUTPUT_MTL, CURRENT_MTL           ; 2067
                    7E       F4     A7  3C 00105          MOVZWL  RWSV_VOL_NUMBER, -(SP)            ; 2068
         00000000G  00              01  FB 00109          CALLS   #1, SWITCH_VOLUME                 ;
                    50       08     A4  3C 00110          MOVZWL  8(R4), R0                         ; 2069
                    50     01FF     C0  9E 00114          MOVAB   511(R0), R0                       ;
  52                50 00000200     8F  C7 00119          DIVL3   #512, R0, BLOCK_COUNT             ;
                           00A2     C7  95 00121 12$:     TSTB    QUAL+78                           ; 2071
                                    06  13 00125          BEQL    13$                               ;
  50                52              02  78 00127          ASHL    #2, BLOCK_COUNT, R0               ; 2072
                                    04  11 0012B          BRB     14$                               ;
  50                52              03  C5 0012D 13$:     MULL3   #3, BLOCK_COUNT, R0               ; 2073
                    50       38     A7  D1 00131 14$:     CMPL    RWSV_ALLOC, R0                    ; 2071
                                    20  1E 00135          BGEQU   16$                               ;
                                    5E  DD 00137          PUSHL   SP                                ; 2076
                       40000000     8F  DD 00139          PUSHL   #1073741824                       ;
         00000000G  00              02  FB 0013F          CALLS   #2, STA_EXTEND                    ;
                    53              50  D0 00146          MOVL    R0, STATUS                        ;
                    06              53  E9 00149          BLBC    STATUS, 15$                       ; 2077
              38    A7              6E  C0 0014C          ADDL2   BLOCKS_ALLOC, RWSV_ALLOC          ; 2079
                                    CF  11 00150          BRB     12$                               ;
            00CE    C7              01  88 00152 15$:     BISB2   #1, COM_FLAGS                     ; 2082
                                    7E  7C 00157 16$:     CLRQ    -(SP)                             ; 2094
                                    7E  D4 00159          CLRL    -(SP)                             ;
                             44     A7  DD 0015B          PUSHL   RWSV_OUT_VBN                      ;
                    7E       08     A4  3C 0015E          MOVZWL  8(R4), -(SP)                      ;
                                    55  DD 00162          PUSHL   BUFFER                            ;
                                    7E  7C 00164          CLRQ    -(SP)                             ;
                                    56  DD 00166          PUSHL   R6                                ;
                                    30  DD 00168          PUSHL   #48                               ;
                             04     A7  DD 0016A          PUSHL   RWSV_CHAN                         ;
                                    02  DD 0016D          PUSHL   #2                                ;
         00000000G  00              0C  FB 0016F          CALLS   #12, STA_QIO                      ;
                    53              50  D0 00176          MOVL    R0, STATUS                        ;
                    09              53  E8 00179          BLBS    STATUS, 17$                       ; 2095
                                    53  DD 0017C          PUSHL   STATUS                            ; 2097
```

```
                                                67  DD 0017E           PUSHL   RWSV_SAVE_FAB                    ;
                                                58  DD 00180           PUSHL   R8                              ; 2096
                                    69          03  FB 00182           CALLS   #3, FILE_ERROR                  ;
                                  0A    A4      02  90 00185 17$:      MOVB    #2, 10(R4)                      ; 2098
                                FF24    D7      64  0E 00189           INSQUE  (R4), @OUTPUT_WAIT+4            ; 2099
                                  44    A7      52  C0 0018E           ADDL2   BLOCK_COUNT, RWSV_OUT_VBN       ; 2100
                                  38    A7      52  C2 00192           SUBL2   BLOCK_COUNT, RWSV_ALLOC         ; 2101
                                                    04 00196           RET                                     ; 2105

; Routine Size:  407 bytes,    Routine Base:  CODE + 013E
```

```
;  556          2106  1 %SBTTL 'INIT_SAVE_DISK - initialize save set disk'
;  557          2107  1 ROUTINE INIT_SAVE_DISK (CONTINUE) : NOVALUE =
;  558          2108  1
;  559          2109  1 !++
;  560          2110  1 !
;  561          2111  1 ! FUNCTIONAL DESCRIPTION:
;  562          2112  1 !
;  563          2113  1 !       This routine initializes an output disk volume for use as an
;  564          2114  1 !       offline save set disk.
;  565          2115  1 !
;  566          2116  1 ! CALLING SEQUENCE:
;  567          2117  1 !       INIT_SAVE_DISK ()
;  568          2118  1 !
;  569          2119  1 ! INPUT PARAMETERS:
;  570          2120  1 !       NONE
;  571          2121  1 !
;  572          2122  1 ! IMPLICIT INPUTS:
;  573          2123  1 !       NONE
;  574          2124  1 !
;  575          2125  1 ! OUTPUT PARAMETERS:
;  576          2126  1 !       NONE
;  577          2127  1 !
;  578          2128  1 ! IMPLICIT OUTPUTS:
;  579          2129  1 !       NONE
;  580          2130  1 !
;  581          2131  1 ! ROUTINE VALUE:
;  582          2132  1 !       NONE
;  583          2133  1 !
;  584          2134  1 ! SIDE EFFECTS:
;  585          2135  1 !       disk initialized, corresponding VCB altered
;  586          2136  1 !
;  587          2137  1 !--
;  588          2138  1
;  589          2139  2 BEGIN
;  590          2140  2
;  591          2141  2 BUILTIN
;  592          2142  2         FFC;
;  593          2143  2
;  594          2144  2 BIND
;  595          2145  2
;  596          2146  2 ! Home block file format string.
;  597          2147  2 !
;  598          2148  2
;  599          2149  2         FORMAT_DESC     = $DESCRIPTOR ('DECFILE11B') : VECTOR,
;  600          2150  2
;  601          2151  2 ! Template attribute control list for create operation.
;  602          2152  2 !
;  603          2153  2
;  604          2154  2         ATTCTL_TEMPLATE = UPLIT (
;  605          2155  2                                   WORD (ATR$S_SEGNUM, ATR$C_SEGNUM), LONG (0),
;  606          2156  2                                   WORD (ATR$S_UIC, ATR$C_UIC), LONG (0),
;  607          2157  2                                   WORD (ATR$S_FPRO, ATR$C_FPRO), LONG (0),
;  608          2158  2                                   WORD (ATR$S_CREDATE, ATR$C_CREDATE), LONG (0),
;  609          2159  2                                   WORD (ATR$S_REVDATE, ATR$C_REVDATE), LONG (0),
;  610          2160  2                                   WORD (2, ATR$C_ASCDATES), LONG (0),
;  611          2161  2                                   LONG (0)
;  612          2162  2                                   );
```

```
;  613          2163  2
;  614          2164  2 ! Entries in the attribute control vector.
;  615          2165  2 !
;  616          2166  2
;  617          2167  2 MACRO
;  618          2168  2         ATT_POINTER (N) = ATT_CONTROL[N*2+1] %;
;  619          2169  2
;  620          2170  2 LITERAL
;  621          2171  2         ATT_SEGNUM      = 0,
;  622          2172  2         ATT_UIC         = 1,
;  623          2173  2         ATT_PROT        = 2,
;  624          2174  2         ATT_CREDATE     = 3,
;  625          2175  2         ATT_REVDATE     = 4,
;  626          2176  2         ATT_REVNUM      = 5,
;  627          2177  2         ATCTL_LENGTH    = 6*8 + 4;
;  628          2178  2
;  629          2179  2 LOCAL
;  630          2180  2         MOUNT_MODE,                     ! mode in which to mount output disk
;  631          2181  2         STATUS,                         ! general status value
;  632          2182  2         FAO_DESC        : VECTOR [2],   ! string descriptor for FAO
;  633          2183  2         IO_STATUS       : VECTOR [4, WORD], ! I/O status block
;  634          2184  2         FAB             : REF BBLOCK,   ! FAB for save set file
;  635          2185  2         NAM             : REF BBLOCK,   ! NAMe block for same
;  636          2186  2         VCB             : REF BBLOCK,   ! VCB for volume
;  637          2187  2         DEVICE_CHAR     : BBLOCK [DIB$K_LENGTH], ! device characteristics buffer
;  638          2188  2         DEVCHAR_DESC    : VECTOR [2],   ! descriptor for above
;  639          2189  2         LP              : REF BBLOCK,   ! structure pointer
;  640          2190  2         P,                              ! byte pointer into index file bitmap
;  641          2191  2         ,                               ! bit pointer into index file bitmap
;  642          2192  2         FILE_NUMBER,                    ! free file number found
;  643          2193  2         FIB             : BBLOCK [FIB$C_LENGTH], ! FIB for create call
;  644          2194  2         FIB_DESC        : VECTOR [2],   ! descriptor for above
;  645          2195  2         NAME_DESC       : VECTOR [2],   ! file name descriptor
;  646          2196  2         ATT_CONTROL     : VECTOR [ATCTL_LENGTH/4], ! attribute control list
;  647          2197  2         FILE_PROT,                      ! file protection attribute
;  648          2198  2         REVNUM,                         ! file revision number attribute
;  649          2199  2         BLOCK_COUNT,                    ! size in virtual blocks of save set block
;  650          2200  2         BLOCKS_ALLOC;                   ! block count returned from allocator
;  651          2201  2
;  652          2202  2 EXTERNAL ROUTINE
;  653          2203  2         STA_MOUNT,                      ! mount save set disk volume set
;  654          2204  2         READY_DISK,                     ! ready save set disk volume
;  655          2205  2         FILE_ERROR,                     ! signal file related error
;  656          2206  2         SWITCH_VOLUME,                  ! switch to specified volume
;  657          2207  2         INITIALIZE_VOLUME,              ! init volume from summary parameters
;  658          2208  2         STA_ENTER,                      ! make directory entry
;  659          2209  2         READ_HEADER,                    ! read file header block
;  660          2210  2         STA_EXTEND;                     ! extend output file
;  661          2211  2
;  662          2212  2
;  663          2213  2 ! On the initial setup, prepare the mounted volume list and VCB's
;  664          2214  2 ! for the save set volumes.
;  665          2215  2 !
;  666          2216  2
;  667          2217  2 MOUNT_MODE = 1;
;  668          2218  2 IF NOT .CONTINUE
;  669          2219  2 THEN
```

```
;  670          2220  3      BEGIN
;  671          2221  3      RWSV_OUT_VBN = 1;
;  672          2222  3      IF NOT .QUAL[QUAL_INIT] THEN MOUNT_MODE = 3;
;  673          2223  2      END;
;  674          2224  2
;  675          2225  2  ! Find the VCB and set up context
;  676          2226  2  !
;  677          2227  2
;  678          2228  2  VCB = READY_DISK (.MOUNT_MODE);
;  679          2229  2  VCB[VCB_SAVESET] = TRUE;
;  680          2230  2  IF NOT .CONTINUE
;  681          2231  2  AND .QUAL[QUAL_INIT]
;  682          2232  2  THEN VCB[VCB_NOTVOLSET] = TRUE;
;  683          2233  2
;  684          2234  2  ! Get the output volume label. This comes from the disk if we are not
;  685          2235  2  ! initializing it; otherwise it comes from the command (either save set
;  686          2236  2  ! file name or /LABEL). The save set name in raw form also becomes the
;  687          2237  2  ! volume set name. Then append the volume number as two digits to the
;  688          2238  2  ! volume name if it did not come explicitly.
;  689          2239  2  !
;  690          2240  2
;  691          2241  2  FAB = .RWSV_SAVE_FAB;
;  692          2242  2  LP = 0;
;  693          2243  2  IF .QUAL[QUAL_INIT]
;  694          2244  2  OR .CONTINUE
;  695          2245  2  OR .VCB[VCB_NOTVOLSET]
;  696          2246  2  THEN
;  697          2247  3      BEGIN
;  698          2248  3      CH$COPY (.BBLOCK[FAB[FC_NAM], NAM$B_NAME],
;  699          2249  3               .BBLOCK[FAB[FC_NAM], NAM$L_NAME],
;  700          2250  3               ' ', HM2$S_VOLNAME, RWSV_VOLUME_ID);
;  701          2251  3
;  702          2252  3      IF NOT .CONTINUE
;  703          2253  3      THEN
;  704          2254  4          BEGIN
;  705          2255  4          CH$MOVE (HM2$S_VOLNAME, RWSV_VOLUME_ID, OUTPUT_MTL[MTL_STRUCNAME]);
;  706          2256  4          CH$MOVE (HM2$S_VOLNAME, RWSV_VOLUME_ID, COM_O_STRUCNAME);
;  707          2257  3          END;
;  708          2258  3
;  709          2259  3  ! If an explicit label was specified, get it. Use the segment number
;  710          2260  3  ! to pick the right entry from the label list. If we are out of list,
;  711          2261  3  ! use the first entry and append the volume number.
;  712          2262  3  !
;  713          2263  3
;  714          2264  3      IF .QUAL[QUAL_LABE]
;  715          2265  3      THEN
;  716          2266  4          BEGIN
;  717          2267  4          LP = .QUAL[QUAL_LABE_LIST];
;  718          2268  4          DECR J FROM .RWSV_VOL_NUMBER-1 TO 1
;  719          2269  4          DO
;  720          2270  5              BEGIN
;  721          2271  5              LP = .LP[QUAL_NEXT];
;  722          2272  5              IF .LP EQL 0 THEN EXITLOOP;
;  723          2273  4              END;
;  724          2274  4          IF .LP NEQ 0
;  725          2275  4          THEN CH$MOVE (HM2$S_VOLNAME, LP[QUAL_LABE_VALUE], RWSV_VOLUME_ID)
;  726          2276  4          ELSE CH$MOVE (HM2$S_VOLNAME, BBLOCK [.QUA[QUAL_LABE_LIST], QUAL_LABE_VALUE], RWSV_VOLUME_ID);
```

```
  3                 END;
  3             END
  2     ELSE
  3         BEGIN
  3         CH$MOVE (HM2$S_VOLNAME, OUTPUT_MTL[MTL_STRUCNAME], COM_O_STRUCNAME);
  2         END;
  2
  2     IF NOT .QUAL[QUAL_INIT]
  2     AND NOT .CONTINUE
  2     THEN
  3         BEGIN
  3         IF .VCB[VCB_CLUSTER] NEQ 1
  3         THEN SIGNAL (BACKUP$_SAVSETCLU);
  3         CH$MOVE (HM2$S_VOLNAME, VCB[VCB_VOLNAME], RWSV_VOLUME_ID);
  3         RWSV_VOL_NUMBER = .VCB[VCB_RVN];
  2         END;
  2
  2     IF .LP EQL 0
  2     THEN
  3         BEGIN
  3         P = CH$FIND_CH (HM2$S_VOLNAME-2, RWSV_VOLUME_ID, ' ');
  3         IF CH$FAIL (.P)
  3         THEN P = RWSV_VOLUME_ID + HM2$S_VOLNAME - 2;
  3         FAO_DESC[0] = 2;
  3         FAO_DESC[1] = .P;
P 3         $FAO ($DESCRIPTOR ('!2ZL'),
P 3               0,
P 3               FAO_DESC,
P 3               .RWSV_VOL_NUMBER
  3               );
  2         END;
  2
  2     ! Do the remaining setup for initialization.
  2     !
  2
  2     IF .QUAL[QUAL_INIT] OR .CONTINUE
  2     THEN
  3         BEGIN
  3         OUTPUT_MTL[MTL_STRUCLEV] = 2;
  3         VCB[VCB_ODS_2] = TRUE;
  3
  3         DEVCHAR_DESC[0] = DIB$K_LENGTH;
  3         DEVCHAR_DESC[1] = DEVICE_CHAR;
  3         $GETCHN (CHAN = .VCB[VCB_CHAN], PRIBUF = DEVCHAR_DESC[0]);
  3
  3     ! Initialize the attributes buffer and call the initialize routine.
  3     !
  3
  3         CH$FILL (0, VSR_LENGTH, OUTPUT_ATTBUF);
  3         (OUTPUT_ATTBUF[VSR_VOLNAME]) = HM2$S_VOLNAME;
  3         (OUTPUT_ATTBUF[VSR_VOLNAME])+4 = RWSV_VOLUME_ID;
  3         (OUTPUT_ATTBUF[VSR_OWNERNAME]) = 12;
  3         (OUTPUT_ATTBUF[VSR_OWNERNAME])+4 = JPI_USERNAME;
  3         (OUTPUT_ATTBUF[VSR_FORMAT]) = .FORMAT_DESC[0];
  3         (OUTPUT_ATTBUF[VSR_FORMAT])+4 = .FORMAT_DESC[1];
  3         (OUTPUT_ATTBUF[VSR_VOLDATE]) = .JPI_DATE[0];
  3         (OUTPUT_ATTBUF[VSR_VOLDATE])+4 = .JPI_DATE[1];
```

```
;   784         2334  4     OUTPUT_ATTBUF[VSR_VOLOWNER] = (IF .QUAL[QUAL_O_OWN_UIC]
;   785         2335  4                                    THEN .QUAL[QUAL_O_OWN_VALU]
;   786         2336  3                                    ELSE .JPI_UIC);
;   787         2337  3     OUTPUT_ATTBUF[VSR_MAXFILES] = 1000;
;   788         2338  4     OUTPUT_ATTBUF[VSR_MAXFILNUM] = (IF .DEVICE_CHAR[DIB$L_MAXBLOCK] LSSU 4096
;   789         2339  4                                     THEN 16
;   790         2340  3                                     ELSE 50);
;   791         2341  3     OUTPUT_ATTBUF[VSR_VOLSTRUCT] = 2^8+1;
;   792         2342  3     OUTPUT_ATTBUF[VSR_RVN] = .RWSV_VOL_NUMBER;
;   793         2343  3     IF .RWSV_VOL_NUMBER EQL 1
;   794         2344  3     THEN OUTPUT_ATTBUF[VSR_RVN] = 0;
;   795         2345  4     OUTPUT_ATTBUF[VSR_PROTECT] = (IF .QUAL[QUAL_PROT]
;   796         2346  4                                   THEN .QUAL[QUAL_PROT_VALUE]
;   797         2347  3                                   ELSE %X'0000');
;   798         2348  3     OUTPUT_ATTBUF[VSR_FILEPROT] = %X'FF00';
;   799         2349  3     OUTPUT_ATTBUF[VSR_EXTEND] = 5;
;   800         2350  3     OUTPUT_ATTBUF[VSR_CLUSTER] = 1;
;   801         2351  3     OUTPUT_ATTBUF[VSR_RESFILES] = 9;
;   802         2352  3     OUTPUT_ATTBUF[VSR_WINDOW] = 7;
;   803         2353  3     OUTPUT_ATTBUF[VSR_LRU_LIM] = 5;
;   804         2354  3
;   805         2355  3     INITIALIZE_VOLUME (.VCB, DEVICE_CHAR);
;   806         2356  2     END;
;   807         2357  2
;   808         2358  2 ! Now create the save set file. Search the index file bitmap for an
;   809         2359  2 ! available file number; we assume 1 for a sequence number.
;   810         2360  2 ! On continuation volumes, we use the continuation file ID instead.
;   811         2361  2 !
;   812         2362  2
;   813         2363  2 NAM = .FAB[FAB$L_NAM];
;   814         2364  2 FAB[FAB$L_STS] = 1;
;   815         2365  2 FAB[FAB$L_STV] = STA_OUT_CHAN;
;   816         2366  2
;   817         2367  2 IF .CONTINUE
;   818         2368  2 THEN
;   819         2369  3     BEGIN
;   820         2370  3     NAM[NAM$W_FID_NUM] = FID$C_CONTIN;
;   821         2371  3     NAM[NAM$W_FID_SEQ] = FID$C_CONTIN;
;   822         2372  3     NAM[NAM$W_FID_RVN] = .RWSV_VOL_NUMBER;
;   823         2373  3     END
;   824         2374  3
;   825         2375  2 ELSE
;   826         2376  3     BEGIN
;   827         2377  3     P = CH$FIND_NOT_CH (.VCB[VCB_MAXFILIDX]/8, .VCB[VCB_IMAP], %X'FF');                    ;
;   828         2378  3     IF CH$FAIL (.P)
;   829         2379  3     THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB, SS$_IDXFILEFULL);
;   830         2380  3     FFC (%REF (0), %REF (8), .P, B);
;   831         2381  3     FILE_NUMBER = (.P - .VCB[VCB_IMAP]) * 8 + .B + 1;
;   832         2382  3     IF .FILE_NUMBER GTRU .VCB[VCB_MAXFILIDX]
;   833         2383  3     THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB, SS$_IDXFILEFULL);
;   834         2384  3
;   835         2385  3 ! Set up the name block and then attempt to read the header block,
;   836         2386  3 ! to make sure it is present. Then make the directory entry.
;   837         2387  3 !
;   838         2388  3
;   839         2389  3     NAM[NAM$W_FID_NUM] = .FILE_NUMBER;
;   840         2390  3     NAM[NAM$W_FID_SEQ] = 1;
```

```
;  841          2391  3      NAM[NAM$B_FID_NMX] = .FILE_NUMBER<16,8>;
;  842          2392  3      NAM[NAM$B_FID_RVN] = .RWSV_VOL_NUMBER;
;  843          2393  2      END;
;  844          2394  2
;  845          2395  2  NAM[NAM$W_DID_NUM] = 4;
;  846          2396  2  NAM[NAM$W_DID_SEQ] = 4;
;  847          2397  2  NAM[NAM$W_DID_RVN] = .RWSV_VOL_NUMBER;
;  848          2398  2
;  849          2399  2  STATUS = READ_HEADER (NAM[NAM$W_FID], .OUTPUT_MTL[MTL_HEADER]);
;  850          2400  2  IF NOT .STATUS
;  851          2401  2  AND .STATUS NEQ SS$_NOSUCHFILE
;  852          2402  2  THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB, SS$_IDXFILEFULL);
;  853          2403  2
;  854          2404  2  STA_ENTER (.FAB);
;  855          2405  2
;  856          2406  2  ! Now set up the FIB, name descriptor, and attribute list and create
;  857          2407  2  ! the file.
;  858          2408  2  !
;  859          2409  2
;  860          2410  2  FIB_DESC[0] = FIB$C_LENGTH;
;  861          2411  2  FIB_DESC[1] = FIB;
;  862          2412  2  CH$FILL (0, FIB$C_LENGTH, FIB);
;  863          2413  2  FIB[FIB$B_WSIZE] = 80;
;  864          2414  2  FIB[FIB$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
;  865          2415  2  FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
;  866          2416  2  FIB[FIB$W_FID_RVN] = .NAM[NAM$W_FID_RVN];
;  867          2417  2  FIB[FIB$W_DID_NUM] = 4;
;  868          2418  2  FIB[FIB$W_DID_SEQ] = 4;
;  869          2419  2  FIB[FIB$W_DID_RVN] = 0;
;  870          2420  2  FIB[FIB$L_EXVBN] = .RWSV_OUT_VBN;
;  871          2421  2
;  872          2422  2  NAME_DESC[0] = .NAM[NAM$B_NAME] + .NAM[NAM$B_TYPE] + .NAM[NAM$B_VER];
;  873          2423  2  NAME_DESC[1] = .NAM[NAM$L_NAME];
;  874          2424  2
;  875          2425  2  CH$MOVE (ATCTL_LENGTH, ATTCTL_TEMPLATE, ATT_CONTROL);
;  876          2426  2  ATT_POINTER[ATT_SEGNUM] = RWSV_SEG_NUMBER;
;  877          2427  3  ATT_POINTER[ATT_UIC] = (IF .QUAL[QUAL_O_OWN_UIC]
;  878          2428  3                           THEN QUAL[QUAL_O_OWN_VALU]
;  879          2429  2                           ELSE JPI_UIC);
;  880          2430  2  ATT_POINTER[ATT_PROT] = FILE_PROT;
;  881          2431  2  ATT_POINTER[ATT_CREDATE] = JPI_DATE;
;  882          2432  2  ATT_POINTER[ATT_REVDATE] = JPI_DATE;
;  883          2433  2  ATT_POINTER[ATT_REVNUM] = REVNUM;
;  884          2434  3  FILE_PROT = (IF .QUAL[QUAL_PROT]
;  885          2435  3                THEN .QUAL[QUAL_PROT_VALUE]
;  886          2436  2                ELSE %X'FA00');
;  887          2437  2  REVNUM = 1;
;  888          2438  2
;  889          2439  2  ! Set up a P6 attribute buffer to specify the file highwater mark. It
;  890          2440  2  ! is initially set to -1 because it is indeterminate; if the save set
;  891          2441  2  ! only occupies one volume it will be corrected on deaccess.
;  892          2442  2  !
;  893          2443  2
;  894          2444  2  CH$FILL (0, FAR_LENGTH, OUTPUT_ATTBUF);
;  895          2445  2  OUTPUT_ATTBUF[FAR_HIGHWATER] = -1;
;  896          2446  2
;  897        P 2447  2  STATUS = $$QIOW (CHAN = STA_OUT_CHAN,
```

```
P 2                     IOSB = IO_STATUS,
P 2                     FUNC = IO$_CREATE OR IO$M_CREATE OR IO$M_ACCESS,
P 2                     P1   = FIB_DESC,
P 2                     P2   = NAME_DESC,
P 2                     P5   = ATT_CONTROL,
P 2                     P6   = OUTPUT_ATTBUF
  2                     );
  2 IF .STATUS THEN STATUS = .IO_STATUS[0];
  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB, .STATUS);
  2
  2 RWSV_ALLOC = 0;
  2 RWSV_IN_VBN_0 = .RWSV_OUT_VBN;
  2
  2 ! Do the initial allocation for the save set file. If there is
  2 ! insufficient space for the minimum number of save set blocks that
  2 ! must go onto one medium, set up for an immediate volume switch.
  2 !
  2
  2 BLOCK_COUNT = (.QUAL[QUAL_BLOC_VALUE]+511) / 512;
  2 WHILE .RWSV_ALLOC LSSU
  3     (IF .QUAL[QUAL_GROU_VALUE] NEQ 0
  3      THEN .BLOCK_COUNT * 4
  3      ELSE .BLOCK_COUNT * 3)
  2 DO
  3     BEGIN
  3     STATUS = STA_EXTEND (1^30, BLOCKS_ALLOC);
  3     IF .STATUS
  3     THEN
  3         RWSV_ALLOC = .RWSV_ALLOC + .BLOCKS_ALLOC
  3     ELSE
  4         BEGIN
  4         COM_FLAGS[COM_EOV] = 1;
  4         EXITLOOP;
  3         END;
  2     END;
  2
  1 END;                                        ! End of routine INIT_SAVE_DISK


42 31 31 45 4C 49 46 43 45 44  002D5 P.AAB:  .ASCII  \DECFILE11B\
                               002DF          .BLKB   1
                    0000000A   002E0 P.AAA:   .LONG   10
                    00000000'  002E4          .ADDRESS P.AAB
                0028  0002     002E8 P.AAC:   .WORD   2, 40
                    00000000   002EC          .LONG   0
                0015  0004     002F0          .WORD   4, 21
                    00000000   002F4          .LONG   0
                0016  0002     002F8          .WORD   2, 22
                    00000000   002FC          .LONG   0
                0011  0008     00300          .WORD   8, 17
                    00000000   00304          .LONG   0
                0012  0008     00308          .WORD   8, 18
                    00000000   0030C          .LONG   0
                000D  0002     00310          .WORD   2, 13
                    00000000   00314          .LONG   0
                    00000000   00318          .LONG   0
```

```
                                4C  5A  32  21   0031C P.AAE:   .ASCII  \!2ZL\                                    ;
                                      00000004   00320 P.AAD:   .LONG   4                                        ;
                                      00000000'  00324         .ADDRESS P.AAE                                   ;

                                                               FORMAT_DESC=        P.AAA
                                                               ATTCTL_TEMPLATE=    P.AAC
                                                               .EXTRN  STA_MOUNT, READY_DISK
                                                               .EXTRN  INITIALIZE_VOLUME
                                                               .EXTRN  STA_ENTER, READ_HEADER
                                                               .EXTRN  SYS$FAO, SYS$GETCHN
                                                               .EXTRN  STA_QIOW

                                                     0FFC 00000 INIT_SAVE_DISK:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 2107
                                    5B 00000000G   8F   D0 00002         MOVL    #BACKUP$_OPENOUT+4, R11          ;
                                    5A 00000000'   EF   9E 00009         MOVAB   RWSV_VOLUME_ID, R10              ;
                                    5E      FEE4   CE   9E 00010         MOVAB   -284(SP), SP                     ;
                                    50             01   D0 00015         MOVL    #1, MOUNT_MODE                   ; 2217
                                    0C       04    AC   E8 00018         BLBS    CONTINUE, 1$                     ; 2218
                                 5C AA             01   D0 0001C         MOVL    #1, RWSV_OUT_VBN                 ; 2221
                              03 76 AA             05   E0 00020         BBS     #5, QUAL+10, 1$                  ; 2222
                                    50             03   D0 00025         MOVL    #3, MOUNT_MODE                   ;
                                                   50   DD 00028 1$:      PUSHL   MOUNT_MODE                       ; 2228
                                 00000000G 00      01   FB 0002A         CALLS   #1, READY_DISK                   ;
                                    57             50   D0 00031         MOVL    R0, VCB                          ;
                                 07 A7             08   88 00034         BISB2   #8, 7(VCB)                       ; 2229
                                    09       04    AC   E8 00038         BLBS    CONTINUE, 2$                     ; 2230
                              04 76 AA             05   E1 0003C         BBC     #5, QUAL+10, 2$                  ; 2231
                                 07 A7             10   88 00041         BISB2   #16, 7(VCB)                      ; 2232
                                    58       18    AA   D0 00045 2$:      MOVL    RWSV_SAVE_FAB, FAB               ; 2241
                                                   56   D4 00049         CLRL    LP                               ; 2242
                              09 76 AA             05   E0 0004B         BBS     #5, QUAL+10, 3$                  ; 2243
                                    05       04    AC   E8 00050         BLBS    CONTINUE, 3$                     ; 2244
                                 4E 07 A7          04   E1 00054         BBC     #4, 7(VCB), 9$                   ; 2245
                                    50     00CF    C8   9A 00059 3$:      MOVZBL  207(FAB), R0                     ; 2248
                           0C       20     00E0    D8                   50   2C 0005E         MOVC5   R0, @224(FAB), #32, #12, RWSV_VOLUME_ID  ;
                                                                         6A   00065                               ;
                                    10       04    AC   E8 00066         BLBS    CONTINUE, 4$                     ; 2252
                                    50     06E0    CA   D0 0006A         MOVL    OUTPUT_MTL, R0                   ; 2255
                                 24 A0             6A   0C 28 0006F      MOVC3   #12, RWSV_VOLUME_ID, 36(R0)      ;
                              00F8 CA              6A   0C 28 00074      MOVC3   #12, RWSV_VOLUME_ID, COM_O_STRUCNAME ; 2256
                                          76 AA    95   0007A 4$:        TSTB    QUAL+10                          ; 2264
                                                   34   18 0007D         BGEQ    10$                              ;
                                    50     00BC    CA   D0 0007F         MOVL    QUAL+80, R0                      ; 2267
                                    56             50   D0 00084         MOVL    R0, LP                           ;
                                    51       0C    AA   3C 00087         MOVZWL  RWSV_VOL_NUMBER, J               ; 2268
                                                   05   11 0008B         BRB     6$                               ;
                                    56             66   D0 0008D 5$:      MOVL    (LP), LP                         ; 2271
                                                   03   13 00090         BEQL    7$                               ; 2272
                                    F8             51   F5 00092 6$:      SOBGTR  J, 5$                            ; 2268
                                                   56   D5 00095 7$:      TSTL    LP                               ; 2274
                                                   07   13 00097         BEQL    8$                               ;
                                 6A       04    A6                   0C   28 00099         MOVC3   #12, 4(LP), RWSV_VOLUME_ID       ; 2275
                                                                         13   11 0009E         BRB     10$              ;
                                 6A       04    A0                   0C   28 000A0 8$:      MOVC3   #12, 4(R0), RWSV_VOLUME_ID       2276
                                                                         0C   11 000A5         BRB     10$              2264
                                    50     06E0    CA   D0 000A7 9$:      MOVL    OUTPUT_MTL, R0                   ; 2281
```

```
         00F8  CA        24  A0            0C  28 000AC          MOVC3   #12, 36(R0), COM_O_STRUCNAME           ;
               21        76  AA            05  E0 000B3 10$:     BBS     #5, QUAL+10, 12$                       ; 2284
                             1D        04  AC  E8 000B8          BLBS    CONTINUE, 12$                          ; 2285
                             01        04  A7  B1 000BC          CMPW    4(VCB), #1                             ; 2288
                                           0D  13 000C0          BEQL    11$                                    ;
                                00000000G  8F  DD 000C2          PUSHL   #BACKUP$_SAVSETCLU                     ; 2289
                 00000000G   00            01  FB 000C8          CALLS   #1, LIB$SIGNAL                         ;
               6A        38  A7            0C  28 000CF 11$:     MOVC3   #12, 56(VCB), RWSV_VOLUME_ID           ; 2290
                         0C  AA        06  A7  9B 000D4          MOVZBW  6(VCB), RWSV_VOL_NUMBER                ; 2291
                                           56  D5 000D9 12$:     TSTL    LP                                     ; 2294
                                           2D  12 000DB          BNEQ    15$                                    ;
               6A            0A            20  3A 000DD          LOCC    #32, #10, RWSV_VOLUME_ID               ; 2297
                                           02  12 000E1          BNEQ    13$                                    ;
                                           51  D4 000E3          CLRL    R1                                     ;
                             59            51  D0 000E5 13$:     MOVL    R1, P                                  ;
                                           04  12 000E8          BNEQ    14$                                    ; 2298
                             59        0A  AA  9E 000EA          MOVAB   RWSV_VOLUME_ID+10, P                   ; 2299
                         F8  AD            02  D0 000EE 14$:     MOVL    #2, FAO_DESC                           ; 2300
                         FC  AD            59  D0 000F2          MOVL    P, FAO_DESC+4                          ; 2301
                         7E        0C  AA  3C 000F6              MOVZWL  RWSV_VOL_NUMBER, -(SP)                 ; 2306
                                   F8  AD  9F 000FA              PUSHAB  FAO_DESC                               ;
                                           7E  D4 000FD          CLRL    -(SP)                                  ;
                                 FEF5  CF  9F 000FF              PUSHAB  P.AAD                                  ;
                 00000000G   00            04  FB 00103          CALLS   #4, SYS$FAO                            ;
               07        76  AA            05  E0 0010A 15$:     BBS     #5, QUAL+10, 16$                       ; 2312
                             03        04  AC  E8 0010F          BLBS    CONTINUE, 16$                          ;
                                     00DF  31 00113              BRW     24$                                    ;
                             50      06E0  CA  D0 00116 16$:     MOVL    OUTPUT_MTL, R0                         ; 2315
                         1E  A0            02  90 0011B          MOVB    #2, 30(R0)                             ;
                         07  A7            02  88 0011F          BISB2   #2, 7(VCB)                             ; 2316
                       FF74  CD        74  8F  9A 00123          MOVZBL  #116, DEVCHAR_DESC                     ; 2318
                       FF78  CD      FF7C  CD  9E 00129          MOVAB   DEVICE_CHAR, DEVCHAR_DESC+4            ; 2319
                                           7E  7C 00130          CLRQ    -(SP)                                  ; 2320
                                     FF74  CD  9F 00132          PUSHAB  DEVCHAR_DESC                           ;
                                           7E  D4 00136          CLRL    -(SP)                                  ;
                             7E        08  A7  3C 00138          MOVZWL  8(VCB), -(SP)                          ;
                 00000000G   00            05  FB 0013C          CALLS   #5, SYS$GETCHN                         ;
 0070    8F            00    6E            00  2C 00143          MOVC5   #0, (SP), #0, #112, OUTPUT_ATTBUF      ; 2325
                                     01F4  CA     0014A                                                         ;
                       01F4  CA            0C  D0 0014D          MOVL    #12, OUTPUT_ATTBUF                     ; 2326
                       01F8  CA            6A  9E 00152          MOVAB   RWSV_VOLUME_ID, OUTPUT_ATTBUF+4        ; 2327
                       01FC  CA            0C  D0 00157          MOVL    #12, OUTPUT_ATTBUF+8                   ; 2328
                       0200  CA      FF44  CA  9E 0015C          MOVAB   JPI_USERNAME, OUTPUT_ATTBUF+12         ; 2329
                       0204  CA      FE51  CF  7D 00163          MOVQ    FORMAT_DESC, OUTPUT_ATTBUF+16          ; 2330
                       020C  CA      FF50  CA  7D 0016A          MOVQ    JPI_DATE, OUTPUT_ATTBUF+24             ; 2332
               07        78  AA            04  E1 00171          BBC     #4, QUAL+12, 17$                       ; 2334
                             50      00AC  CA  D0 00176          MOVL    QUAL+64, R0                            ; 2335
                                           05  11 0017B          BRB     18$                                    ;
                             50      FF40  CA  D0 0017D 17$:     MOVL    JPI_UIC, R0                            ; 2336
                       021C  CA            50  D0 00182 18$:     MOVL    R0, OUTPUT_ATTBUF+40                   ; 2334
                       0228  CA      03E8  8F  3C 00187          MOVZWL  #1000, OUTPUT_ATTBUF+52                ; 2337
                   00001000  8F        EC  AD  D1 0018E          CMPL    DEVICE_CHAR+112, #4096                 ; 2338
                                           05  1E 00196          BGEQU   19$                                    ;
                             50            10  D0 00198          MOVL    #16, R0                                ;
                                           03  11 0019B          BRB     20$                                    ;
                             50            32  D0 0019D 19$:     MOVL    #50, R0                                ;
                       022C  CA            50  D0 001A0 20$:     MOVL    R0, OUTPUT_ATTBUF+56                   ;
```

```
                0234   CA      0201   8F   B0 001A5          MOVW    #513, OUTPUT_ATTBUF+64                          ; 2341
                0236   CA        0C   AA   B0 001AC          MOVW    RWSV_VOL_NUMBER, OUTPUT_ATTBUF+66               ; 2342
                       01        0C   AA   B1 001B2          CMPW    RWSV_VOL_NUMBER, #1                             ; 2343
                                      04   12 001B6          BNEQ    21$                                             ;
                                0236   CA   B4 001B8          CLRW    OUTPUT_ATTBUF+66                                ; 2344
          07      7A   AA              03   E1 001BC 21$:     BBC     #3, QUAL+14, 22$                                ; 2345
                       50      00C0   CA   3C 001C1          MOVZWL  QUAL+84, R0                                     ; 2346
                                      02   11 001C6          BRB     23$                                             ;
                                      50   D4 001C8 22$:     CLRL    R0                                              ; 2345
                0238   CA              50   B0 001CA 23$:     MOVW    R0, OUTPUT_ATTBUF+68                            ;
                023A   CA        FF00   8F   B0 001CF          MOVW    #-256, OUTPUT_ATTBUF+70                         ; 2348
                0240   CA  00010005    8F   D0 001D6          MOVL    #65541, OUTPUT_ATTBUF+76                        ; 2349
                0244   CA  03070009    8F   D0 001DF          MOVL    #50790409, OUTPUT_ATTBUF+80                     ; 2351
                           FF7C        CD   9F 001E8          PUSHAB  DEVICE_CHAR                                     ; 2355
                                      57   DD 001EC          PUSHL   VCB                                             ;
     00000000G   00                   02   FB 001EE          CALLS   #2, INITIALIZE_VOLUME                           ;
                       56        28   A8   D0 001F5 24$:     MOVL    40(FAB), NAM                                    ; 2363
                  08   A8              01   D0 001F9          MOVL    #1, 8(FAB)                                      ; 2364
                  0C   A8  0002FFFF    8F   D0 001FD          MOVL    #196607, 12(FAB)                                ; 2365
                       0F        04   AC   E9 00205          BLBC    CONTINUE, 25$                                   ; 2372
                  24   A6  00070007    8F   D0 00209          MOVL    #458759, 36(NAM)                                ; 2370
                  28   A6        0C   AA   B0 00211          MOVW    RWSV_VOL_NUMBER, 40(NAM)                        ; 2372
                                      5F   11 00216          BRB     29$                                             ; 2367
          50      1C   A7              08   C7 00218 25$:     DIVL3   #8, 28(VCB), R0                                 ; 2377
    10    B7              50        FF   8F   3B 0021D          SKPC    #255, R0, @16(VCB)                              ;
                                      02   12 00223          BNEQ    26$                                             ;
                                      51   D4 00225          CLRL    R1                                              ;
                       59                 51   D0 00227 26$:     MOVL    R1, P                                           ;
                                      10   12 0022A          BNEQ    27$                                             ; 2378
                       7E        08D0   8F   3C 0022C          MOVZWL  #2256, -(SP)                                    ; 2379
                                      58   DD 00231          PUSHL   FAB                                             ;
                                      5B   DD 00233          PUSHL   R11                                             ;
     00000000G   00                   03   FB 00235          CALLS   #3, FILE_ERROR                                  ;
50        69     08                   00   EB 0023C 27$:     FFC     #0, #8, (P), B                                  ; 2380
51              59        10   A7      C3   00241          SUBL3   16(VCB), P, R1                                  ; 2381
                52        01  A041     7E   00246          MOVAQ   1(B)[R1], FILE_NUMBER                           ;
                  1C   A7              52   D1 0024B          CMPL    FILE_NUMBER, 28(VCB)                            ; 2382
                                      10   1B 0024F          BLEQU   28$                                             ;
                       7E        08D0   8F   3C 00251          MOVZWL  #2256, -(SP)                                    ; 2383
                                      58   DD 00256          PUSHL   FAB                                             ;
                                      5B   DD 00258          PUSHL   R11                                             ;
     00000000G   00                   03   FB 0025A          CALLS   #3, FILE_ERROR                                  ;
                  24   A6              52   B0 00261 28$:     MOVW    FILE_NUMBER, 36(NAM)                            ; 2389
                  26   A6              01   B0 00265          MOVW    #1, 38(NAM)                                     ; 2390
50        52     08                   10   EF 00269          EXTZV   #16, #8, FILE_NUMBER, R0                        ; 2391
                  29   A6              50   90 0026E          MOVB    R0, 41(NAM)                                     ;
                  28   A6        0C   AA   90 00272          MOVB    RWSV_VOL_NUMBER, 40(NAM)                        ; 2392
                  2A   A6  00040004    8F   D0 00277 29$:     MOVL    #262148, 42(NAM)                                ; 2395
                  2E   A6        0C   AA   B0 0027F          MOVW    RWSV_VOL_NUMBER, 46(NAM)                        ; 2397
                       50      06E0   CA   D0 00284          MOVL    OUTPUT_MTL, R0                                  ; 2399
                                0C   A0   DD 00289          PUSHL   12(R0)                                          ;
                                24   A6   9F 0028C          PUSHAB  36(NAM)                                         ;
     00000000G   00                   02   FB 0028F          CALLS   #2, READ_HEADER                                 ;
                                      57   D0 00296          MOVL    R0, STATUS                                      ;
                                      19   E8 00299          BLBS    STATUS, 30$                                     ; 2400
     00000910   8F                   57   D1 0029C          CMPL    STATUS, #2320                                   ; 2401
                                      10   13 002A3          BEQL    30$                                             ;
```

```
                                    7E      08D0    8F  3C 002A5           MOVZWL  #2256, -(SP)                    ; 2402
                                                    58  DD 002AA           PUSHL   FAB                             ;
                                                    5B  DD 002AC           PUSHL   R11                             ;
                          00000000G 00              03  FB 002AE           CALLS   #3, FILE_ERROR                  ;
                                                    58  DD 002B5 30$:      PUSHL   FAB                             ; 2404
                          00000000G 00              01  FB 002B7           CALLS   #1, STA_ENTER                   ;
                                48  AE        40    8F  9A 002BE           MOVZBL  #64, FIB_DESC                   ; 2410
                                4C  AE        50    AE  9E 002C3           MOVAB   FIB, FIB_DESC+4                 ; 2411
0040  8F                     00                 6E      00  2C 002C8       MOVC5   #0, (SP), #0, #64, FIB          ; 2412
                                                   50   AE    002CF       ;                                       ;
                                53  AE        50    8F  90 002D1           MOVB    #80, FIB+3                      ; 2413
                                54  AE        24    A6  D0 002D6           MOVL    36(NAM), FIB+4                  ; 2414
                                58  AE        28    A6  B0 002DB           MOVW    40(NAM), FIB+8                  ; 2416
                                5A  AE  00040004    8F  D0 002E0           MOVL    #262148, FIB+10                 ; 2417
                                                    5E  AE  B4 002E8       CLRW    FIB+14                          ; 2419
                                6C  AE        5C    AA  D0 002EB           MOVL    RWSV_OUT_VBN, FIB+28            ; 2420
                                50        3B        A6  9A 002F0           MOVZBL  59(NAM), R0                     ; 2422
                                51        3C        A6  9A 002F4           MOVZBL  60(NAM), R1                     ;
                                50                  51  C0 002F8           ADDL2   R1, R0                          ;
                                52        3D        A6  9A 002FB           MOVZBL  61(NAM), R2                     ;
                          40    AE                  50      52  C1 002FF   ADDL3   R2, R0, NAME_DESC              ;
                                44  AE        4C    A6  D0 00304           MOVL    76(NAM), NAME_DESC+4            ; 2423
                          0C    AE      FCB2  CF        34  28 00309       MOVC3   #52, ATTCTL_TEMPLATE, ATT_CONTROL ; 2425
                                10  AE        0E    AA  9E 00310           MOVAB   RWSV_SEG_NUMBER, ATT_CONTROL+4  ; 2426
                             07     78  AA              04  E1 00315       BBC     #4, QUAL+12, 31$                ; 2427
                                50      00AC    CA  9E 0031A               MOVAB   QUAL+64, R0                     ; 2428
                                                    05  11 0031F           BRB     32$                             ;
                                50      FF40    CA  9E 00321 31$:          MOVAB   JPI_UIC, R0                     ; 2427
                                18  AE                  50  D0 00326 32$:  MOVL    R0, ATT_CONTROL+12              ;
                                20  AE                  6E  9E 0032A       MOVAB   FILE_PROT, ATT_CONTROL+20       ; 2430
                                28  AE        FF50    CA  9E 0032E           MOVAB   JPI_DATE, ATT_CONTROL+28        ; 2431
                                30  AE        FF50    CA  9E 00334           MOVAB   JPI_DATE, ATT_CONTROL+36        ; 2432
                                38  AE          04    AE  9E 0033A           MOVAB   REVNUM, ATT_CONTROL+44          ; 2433
                             07     7A  AA              03  E1 0033F       BBC     #3, QUAL+14, 33$                ; 2434
                                50      00C0    CA  3C 00344               MOVZWL  QUAL+84, R0                     ; 2435
                                                    05  11 00349           BRB     34$                             ;
                                50      FA00    8F  3C 0034B 33$:          MOVZWL  #64000, R0                      ; 2434
                                6E                  50  D0 00350 34$:      MOVL    R0, FILE_PROT                   ;
                                04  AE                  01  D0 00353       MOVL    #1, REVNUM                      ; 2437
0090  8F                     00                 6E      00  2C 00357       MOVC5   #0, (SP), #0, #144, OUTPUT_ATTBUF ; 2444
                                        01F4    CA      0035E           ;                                       ;
                              027C      CA              01  CE 00361       MNEGL   #1, OUTPUT_ATTBUF+136           ; 2445
                                        01F4    CA  9F  00366              PUSHAB  OUTPUT_ATTBUF                   ; 2454
                                          10    AE  9F  0036A              PUSHAB  ATT_CONTROL                     ;
                                                    7E  7C  0036D          CLRQ    -(SP)                           ;
                                          50    AE  9F  0036F              PUSHAB  NAME_DESC                       ;
                                          5C    AE  9F  00372              PUSHAB  FIB_DESC                        ;
                                                    7E  7C  00375          CLRQ    -(SP)                           ;
                                          F0    AD  9F  00377              PUSHAB  IO_STATUS                       ;
                                    7E    F3    8F  9A  0037A              MOVZBL  #243, -(SP)                     ;
                                0002FFFF        8F  DD  0037E              PUSHL   #196607                         ;
                                                    7E  D4  00384          CLRL    -(SP)                           ;
                          00000000G 00              0C  FB  00386          CALLS   #12, STA_QIOW                   ;
                                    57              50  D0  0038D          MOVL    R0, STATUS                      ;
                                    07              57  E9  00390          BLBC    STATUS, 35$                     ; 2455
                                    57      F0      AD  3C  00393          MOVZWL  IO_STATUS, STATUS               ;
                                    0D              57  E8  00397          BLBS    STATUS, 36$                     ; 2456
```

```
                                        57  DD 0039A 35$:   PUSHL   STATUS                          ;
                                        58  DD 0039C        PUSHL   FAB                             ;
                                        5B  DD 0039E        PUSHL   R11                             ;
                0000000G  00            03  FB 003A0        CALLS   #3, FILE_ERROR                  ;
                                  50    AA  D4 003A7 36$:   CLRL    RWSV_ALLOC                      ; 2458
                      4C  AA          5C    AA  D0 003AA        MOVL    RWSV_OUT_VBN, RWSV_IN_VBN_0     ; 2459
                          50        00B4    CA  3C 003AF        MOVZWL  QUAL+72, R0                     ; 2466
                          50        01FF    C0  9E 003B4        MOVAB   511(R0), R0                     ;
        52                50 00000200       8F  C7 003B9        DIVL3   #512, R0, BLOCK_COUNT           ;
                                    00BA    CA  95 003C1 37$:   TSTB    QUAL+78                         ; 2468
                                        06  13 003C5        BEQL    38$                             ;
        50                52            02  78 003C7        ASHL    #2, BLOCK_COUNT, R0             ; 2469
                                        04  11 003CB        BRB     39$                             ;
        50                52            03  C5 003CD 38$:   MULL3   #3, BLOCK_COUNT, R0             ; 2470
                          50          50    AA  D1 003D1 39$:   CMPL    RWSV_ALLOC, R0                  ; 2468
                                        22  1E 003D5        BGEQU   41$                             ;
                                      08    AE  9F 003D7        PUSHAB  BLOCKS_ALLOC                    ; 2473
                                40000000    8F  DD 003DA        PUSHL   #1073741824                     ;
                0000000G  00            02  FB 003E0        CALLS   #2, STA_EXTEND                  ;
                          57            50  D0 003E7        MOVL    R0, STATUS                      ;
                          07            57  E9 003EA        BLBC    STATUS, 40$                     ; 2474
                      50  AA          08    AE  C0 003ED        ADDL2   BLOCKS_ALLOC, RWSV_ALLOC        ; 2476
                                        CD  11 003F2        BRB     37$                             ;
                    00E6  CA            01  88 003F4 40$:   BISB2   #1, COM_FLAGS                   ; 2479
                                            04 003F9 41$:   RET                                     ; 2484

; Routine Size:  1018 bytes,    Routine Base:  CODE + 0328
```

```
;  936          2485  1 %SBTTL 'INIT_SAVE_TAPE - initialize save set tape'
;  937          2486  1 ROUTINE INIT_SAVE_TAPE (CONTINUE) : NOVALUE =
;  938          2487  1
;  939          2488  1 !++
;  940          2489  1 !
;  941          2490  1 ! FUNCTIONAL DESCRIPTION:
;  942          2491  1 !
;  943          2492  1 !       This routine initializes a magtape for an output save set.
;  944          2493  1 !
;  945          2494  1 ! CALLING SEQUENCE:
;  946          2495  1 !       INIT_SAVE_TAPE (CONTINUE)
;  947          2496  1 !
;  948          2497  1 ! INPUT PARAMETERS:
;  949          2498  1 !       CONTINUE: FALSE if this is a new save set
;  950          2499  1 !                 TRUE if this is for the next continuation volume
;  951          2500  1 !
;  952          2501  1 ! IMPLICIT INPUTS:
;  953          2502  1 !       NONE
;  954          2503  1 !
;  955          2504  1 ! OUTPUT PARAMETERS:
;  956          2505  1 !       NONE
;  957          2506  1 !
;  958          2507  1 ! IMPLICIT OUTPUTS:
;  959          2508  1 !       NONE
;  960          2509  1 !
;  961          2510  1 ! ROUTINE VALUE:
;  962          2511  1 !       NONE
;  963          2512  1 !
;  964          2513  1 ! SIDE EFFECTS:
;  965          2514  1 !       NONE
;  966          2515  1 !
;  967          2516  1 !--
;  968          2517  1
;  969          2518  2 BEGIN
;  970          2519  2
;  971          2520  2 LOCAL
;  972          2521  2         FAB             : REF BBLOCK,   ! pointer to output FAB
;  973          2522  2         STATUS,                         ! the usual status value
;  974          2523  2         TAPE_CHAR       : BBLOCK [4],   ! magtape characteristics longword
;  975          2524  2         LABEL_BUFFER    : BBLOCK [90];  ! buffer for tape labels
;  976          2525  2
;   7           2526  2 EXTERNAL ROUTINE
;  978          2527  2         READY_TAPE,                     ! make tape ready for I/O
;  979          2528  2         REWIND,                         ! rewind tape
;  980          2529  2         SENSE_CHAR,                     ! sense magtape characteristics
;  981          2530  2         SET_CHAR,                       ! set magtape characteristics
;  982          2531  2         SKIP_TM,                        ! skip tape marks
;  983          2532  2         SKIP_RECORD,                    ! skip tape record
;  984          2533  2         READ_LABEL,                     ! read file header label
;  985          2534  2         MAKE_VOL1,                      ! generate volume header label
;  986          2535  2         MAKE_HDR1,                      ! generate file header label 1
;  987          2536  2         MAKE_HDR2,                      ! generate file header label 2
;  988          2537  2         WRITE_LABEL,                    ! write file header label
;  989          2538  2         WRITE_TM,                       ! write tape mark
;  990          2539  2         FILE_ERROR,                     ! signal file related error
;  991          2540  2         LIB$CVT_DTB     : ADDRESSING_MODE (GENERAL);
;  992          2541  2                                         ! convert decimal string to binary
```

```
;  993          2542  2
;  994          2543  2 ! Set up the output tape.
;  995          2544  2 !
;  996          2545  2
;  997          2546  2 FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
;  998          2547  2 TAPE_CHAR = READY_TAPE (TRUE);
;  999          2548  2
; 1000          2549  2 ! If rewind is requested, do so and create the volume header label.
; 1001          2550  2 !
; 1002          2551  2
; 1003          2552  2 IF .QUAL[QUAL_REWI] OR .CONTINUE
; 1004          2553  2 THEN
; 1005          2554  3     BEGIN
; 1006          2555  3     REWIND ();
; 1007          2556  3     IF NOT .CONTINUE THEN RWSV_FILE_NUMBER = 1;
; 1008          2557  3     IF .QUAL[QUAL_DENS] OR .CONTINUE
; 1009          2558  3     THEN
; 1010          2559  4         BEGIN
; 1011          2560  4         TAPE_CHAR = SENSE_CHAR ();
; 1012          2561  4         TAPE_CHAR[MT$V_DENSITY] = .QUAL[QUAL_DENS_VALUE];
; 1013          2562  4         SET_CHAR (.TAPE_CHAR);
; 1014          2563  3         END;
; 1015          2564  3
; 1016          2565  3     MAKE_VOL1 (LABEL_BUFFER);
; 1017          2566  3     CH$COPY(VL1$S_VOLLBL, LABEL_BUFFER[VL1$T_VOLLBL], %C' ', 12, RWSV_VOLUME_ID);
; 1018          2567  3     WRITE_LABEL (LABEL_BUFFER);
; 1019          2568  3
; 1020          2569  3     IF .QUAL[QUAL_DENS] OR .CONTINUE
; 1021          2570  3     THEN
; 1022          2571  4         BEGIN
; 1023          2572  4         TAPE_CHAR = SENSE_CHAR ();
; 1024          2573  4         IF .TAPE_CHAR[MT$V_DENSITY] NEQ .QUAL[QUAL_DENS_VALUE]
; 1025          2574  4         THEN FILE_ERROR (BACKUP$_DENSITY, .FAB);
; 1026          2575  3         END;
; 1027          2576  3     END
; 1028          2577  3
; 1029          2578  3 ! Otherwise space to the end of tape and verify that we can actually
; 1030          2579  3 ! append to it. We first construct a best effort volume label for the
; 1031          2580  3 ! journal, by reading the volume header label if the tape is at BOT,
; 1032          2581  3 ! or just getting it from the command if a label is specified. Note
; 1033          2582  3 ! that the "infinite" skip call will terminate on two consecutive
; 1034          2583  3 ! tape marks, which could be either real EOV or an empty file. We tell
; 1035          2584  3 ! what's what by the labels. Note that a read is necessary to advance
; 1036          2585  3 ! over the double tape marks of an empty file.
; 1037          2586  3 !
; 1038          2587  3
; 1039          2588  2 ELSE
; 1040          2589  3     BEGIN
; 1041          2590  3     IF .TAPE_CHAR[MT$V_BOT]
; 1042          2591  3     THEN
; 1043          2592  4         BEGIN
; 1044          2593  4         STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
; 1045          2594  4         IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
; 1046          2595  4         CH$COPY(VL1$S_VOLLBL, LABEL_BUFFER[VL1$T_VOLLBL], %C' ', 12, RWSV_VOLUME_ID);
; 1047          2596  4         END
; 1048          2597  3     ELSE IF .QUAL[QUAL_LABE]
; 1049          2598  3     THEN CH$MOVE (12, @BBLOCK [.QUAL[QUAL_LABE_LIST], QUAL_LABE_VALUE], RWSV_VOLUME_ID);
```

```
;  1050          2599  3
;  1051          2600  3      WHILE TRUE
;  1052          2601  3      DO
;  1053          2602  4          BEGIN
;  1054          2603  4          SKIP_TM (32767);
;  1055          2604  4
;  1056          2605  4          SKIP_TM (-2);
;  1057          2606  4          TAPE_CHAR = SENSE_CHAR ();
;  1058          2607  4          IF NOT .TAPE_CHAR[MT$V_BOT]
;  1059          2608  4          THEN
;  1060          2609  5              BEGIN
;  1061          2610  5              SKIP_RECORD (1);
;  1062          2611  5              STATUS = READ_LABEL (LABEL_BUFFER);
;  1063          2612  5              IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1064          2613  5              IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'EOF1'
;  1065          2614  5              THEN EXITLOOP;
;  1066          2615  5              IF .LABEL_BUFFER[HD1$L_HD1LID] EQL 'EOV1'
;  1067          2616  5              THEN FILE_ERROR (BACKUP$_CONTINUED, .FAB);
;  1068          2617  5              IF .LABEL_BUFFER[HD1$L_HD1LID] NEQ 'HDR1'
;  1069          2618  5              THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  1070          2619  4              END;
;  1071          2620  4          SKIP_TM (1);
;  1072          2621  4          SKIP_RECORD (1);
;  1073          2622  3          END;                                ! end of EOV search loop
;  1074          2623  3
;  1075          2624  3 ! Get the file set ID and file number from the EOT label, so
;  1076          2625  3 ! we can correctly generate the new header label. Then position to the
;  1077          2626  3 ! end of the label set. If the file number in the label is zero, this
;  1078          2627  3 ! is a freshly initialized tape. In this case, rewind and read VOL1
;  1079          2628  3 ! to position to where we will create HDR1, so the dummy file will
;  1080          2629  3 ! be overwritten.
;  1081          2630  3 !
;  1082          2631  3
;  1083          2632  3      CH$MOVE (HD1$S_FILESETID, LABEL_BUFFER[HD1$T_FILESETID], RWSV_FILESET_ID);
;  1084          2633  3      IF NOT LIB$CVT_DTB (4, LABEL_BUFFER[HD1$T_FILESEQNO], RWSV_FILE_NUMBER)
;  1085          2634  3      THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, BACKUP$_NOTANSI);
;  1086          2635  3
;  1087          2636  3      IF .RWSV_FILE_NUMBER EQL 0
;  1088          2637  3      THEN
;  1089          2638  4          BEGIN
;  1090          2639  4          REWIND ();
;  1091          2640  4          STATUS = READ_LABEL (LABEL_BUFFER, 'VOL1');
;  1092          2641  4          IF NOT .STATUS THEN FILE_ERROR (BACKUP$_LABELERR, .FAB, .STATUS);
;  1093          2642  4          END
;  1094          2643  3      ELSE
;  1095          2644  3          SKIP_TM (1);
;  1096          2645  3      RWSV_FILE_NUMBER = .RWSV_FILE_NUMBER + 1;
;  1097          2646  2      END;
;  1098          2647  2
;  1099          2648  2 QUAL[QUAL_DENS_VALUE] = .TAPE_CHAR[MT$V_DENSITY];
;  1100          2649  2
;  1101          2650  2 ! Now write file header labels.
;  1102          2651  2 !
;  1103          2652  2
;  1104          2653  2 MAKE_HDR1 (LABEL_BUFFER);
;  1105          2654  2 WRITE_LABEL (LABEL_BUFFER);
;  1106          2655  2
```

```
;  1107          2656  2 MAKE_HDR2 (LABEL_BUFFER);
;  1108          2657  2 WRITE_LABEL (LABEL_BUFFER);
;  1109          2658  2
;  1110          2659  2 WRITE_TM ();
;  1111          2660  2
;  1112          2661  2 ! See if the created label set runs past EOT. If it does, set up for
;  1113          2662  2 ! an immediate reel switch.
;  1114          2663  2 !
;  1115          2664  2
;  1116          2665  2 TAPE_CHAR = SENSE_CHAR ();
;  1117          2666  2 IF .TAPE_CHAR[MT$V_EOT]
;  1118          2667  2 THEN COM_FLAGS[COM_EOV] = TRUE;
;  1119          2668  2
;  1120          2669  1 END;                                    ! End of routine INIT_SAVE_TAPE


                                                        .EXTRN  READY_TAPE, REWIND
                                                        .EXTRN  SENSE_CHAR, SET_CHAR
                                                        .EXTRN  SKIP_TM, SKIP_RECORD
                                                        .EXTRN  READ_LABEL, MAKE_VOL1
                                                        .EXTRN  MAKE_HDR1, MAKE_HDR2
                                                        .EXTRN  WRITE_LABEL, WRITE_TM
                                                        .EXTRN  LIB$CVT_DTB

                                    0FFC 00000 INIT_SAVE_TAPE:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 2486
                          5B 00000000G   8F   D0 00002  MOVL    #BACKUP$_LABELERR, R11
                          5A 00000000G   00   9E 00009  MOVAB   FILE_ERROR, R10
                          59 00000000'   EF   9E 00010  MOVAB   RWSV_FILE_NUMBER, R9
                          5E        A4   AE   9E 00017  MOVAB   -92(SP), SP
                          50        04   A9   D0 0001B  MOVL    RWSV_SAVE_QUAL, R0                      ; 2546
                          56        04   A0   D0 0001F  MOVL    4(R0), FAB
                                         01   DD 00023  PUSHL   #1                                       ; 2547
                             00000000G   00             01   FB 00025  CALLS   #1, READY_TAPE
                                         58             50   D0 0002C  MOVL    R0, TAPE_CHAR
                                         04        69   A9   E8 0002F  BLBS    QUAL+13, 1$             ; 2552
                                         78        04   AC   E9 00033  BLBC    CONTINUE, 7$
                             00000000G   00             00   FB 00037 1$:    CALLS   #0, REWIND      ; 2555
                                         03        04   AC   E8 0003E  BLBS    CONTINUE, 2$            ; 2556
                                         69             01   D0 00042  MOVL    #1, RWSV_FILE_NUMBER
                                   04        65   A9        03   E0 00045 2$:    BBS     #3, QUAL+9, 3$  ; 2557
                                         1A        04   AC   E9 0004A  BLBC    CONTINUE, 4$
                             00000000G   00             00   FB 0004E 3$:    CALLS   #0, SENSE_CHAR  ; 2560
                                         58             50   D0 00055  MOVL    R0, TAPE_CHAR
                58                 05                  08 00A9 C9   F0 00058  INSV    QUAL+77, #8, #5, TAPE_CHAR      ; 2561
                                                       58   DD 0005F  PUSHL   TAPE_CHAR               ; 2562
                             00000000G   00             01   FB 00061  CALLS   #1, SET_CHAR
                                                       5E   DD 00068 4$:    PUSHL   SP              ; 2565
                             00000000G   00             01   FB 0006A  CALLS   #1, MAKE_VOL1
                0C                 20        04   AE   06   2C 00071  MOVC5   #6, LABEL_BUFFER+4, #32, #12, -  ; 2566
                                                  F0   A9        00077          RWSV_VOLUME_ID
                                                       5E   DD 00079  PUSHL   SP                      ; 2567
                             00000000G   00             01   FB 0007B  CALLS   #1, WRITE_LABEL
                                   04        65   A9        03   E0 00082  BBS     #3, QUAL+9, 5$  ; 2569
                                         21        04   AC   E9 00087  BLBC    CONTINUE, 6$
                             00000000G   00             00   FB 0008B 5$:    CALLS   #0, SENSE_CHAR  ; 2572
```

```
                        58          50  D0 00092          MOVL    R0, TAPE_CHAR                    ;
                        50    00A9  C9  9A 00095          MOVZBL  QUAL+77, R0                      ; 2573
   50           58      05          08  ED 0009A          CMPZV   #8, #5, TAPE_CHAR, R0            ;
                                    0B  13 0009F          BEQL    6$                               ;
                                    56  DD 000A1          PUSHL   FAB                              ; 2574
                     00000000G      8F  DD 000A3          PUSHL   #BACKUP$_DENSITY                 ;
                        6A          02  FB 000A9          CALLS   #2, FILE_ERROR                   ;
                                  0122  31 000AC  6$:     BRW     18$                              ; 2552
           28           58          10  E1 000AF  7$:     BBC     #16, TAPE_CHAR, 9$               ; 2590
                      314C4F56      8F  DD 000B3          PUSHL   #827084630                       ; 2593
                            04      AE  9F 000B9          PUSHAB  LABEL_BUFFER                     ;
             00000000G  00          02  FB 000BC          CALLS   #2, READ_LABEL                   ;
                        57          50  D0 000C3          MOVL    R0, STATUS                       ;
                        08          57  E8 000C6          BLBS    STATUS, 8$                       ; 2594
                        7E          56  7D 000C9          MOVQ    FAB, -(SP)                       ;
                                    5B  DD 000CC          PUSHL   R11                              ;
                        6A          03  FB 000CE          CALLS   #3, FILE_ERROR                   ;
   0C           20      04    AE    06  2C 000D1  8$:     MOVC5   #6, LABEL_BUFFER+4, #32, #12, -  ; 2595
                              F0    A9     000D7                  RWSV_VOLUME_ID                   ;
                                    10  11 000D9          BRB     10$                              ; 2590
                              66    A9  95 000DB  9$:     TSTB    QUAL+10                          ; 2597
                                    0B  18 000DE          BGEQ    10$                              ;
                        50    00AC  C9  D0 000E0          MOVL    QUAL+80, R0                      ; 2598
        F0    A9        04    A0    0C  28 000E5          MOVC3   #12, 4(R0), RWSV_VOLUME_ID       ;
                        7E    7FFF  8F  3C 000EB  10$:    MOVZWL  #32767, -(SP)                    ; 2603
             00000000G  00          01  FB 000F0          CALLS   #1, SKIP_TM                      ;
                        7E          02  CE 000F7          MNEGL   #2, -(SP)                        ; 2605
             00000000G  00          01  FB 000FA          CALLS   #1, SKIP_TM                      ;
             00000000G  00          00  FB 00101          CALLS   #0, SENSE_CHAR                   ; 2606
                        58          50  D0 00108          MOVL    R0, TAPE_CHAR                    ;
           53           58          10  E0 0010B          BBS     #16, TAPE_CHAR, 13$              ; 2607
                                    01  DD 0010F          PUSHL   #1                               ; 2610
             00000000G  00          01  FB 00111          CALLS   #1, SKIP_RECORD                  ;
                                    5E  DD 00118          PUSHL   SP                               ; 2611
             00000000G  00          01  FB 0011A          CALLS   #1, READ_LABEL                   ;
                        57          50  D0 00121          MOVL    R0, STATUS                       ;
                        08          57  E8 00124          BLBS    STATUS, 11$                      ; 2612
                        7E          56  7D 00127          MOVQ    FAB, -(SP)                       ;
                                    5B  DD 0012A          PUSHL   R11                              ;
                        6A          03  FB 0012C          CALLS   #3, FILE_ERROR                   ;
             31464F45   8F          6E  D1 0012F  11$:    CMPL    LABEL_BUFFER, #826691397         ; 2613
                                    3F  13 00136          BEQL    14$                              ;
             31564F45   8F          6E  D1 00138          CMPL    LABEL_BUFFER, #827739973         ; 2615
                                    0B  12 0013F          BNEQ    12$                              ;
                                    56  DD 00141          PUSHL   FAB                              ; 2616
                     00000000G      8F  DD 00143          PUSHL   #BACKUP$_CONTINUED               ;
                        6A          02  FB 00149          CALLS   #2, FILE_ERROR                   ;
             31524448   8F          6E  D1 0014C  12$:    CMPL    LABEL_BUFFER, #827475016         ; 2617
                                    0D  13 00153          BEQL    13$                              ;
                     00000000G      8F  DD 00155          PUSHL   #BACKUP$_NOTANSI                 ; 2618
                                    56  DD 0015B          PUSHL   FAB                              ;
                                    5B  DD 0015D          PUSHL   R11                              ;
                        6A          03  FB 0015F          CALLS   #3, FILE_ERROR                   ;
                                    01  DD 00162  13$:    PUSHL   #1                               ; 2620
             00000000G  00          01  FB 00164          CALLS   #1, SKIP_TM                      ;
                                    01  DD 0016B          PUSHL   #1                               ; 2621
             00000000G  00          01  FB 0016D          CALLS   #1, SKIP_RECORD                  ;
```

```
                                   FF74  31 00174          BRW     10$                                  ; 2600
              E8   A9       15   AE        06  28 00177 14$:     MOVC3   #6, LABEL_BUFFER+21, RWSV_FILESET_ID ; 2632
                                           59  DD 0017D          PUSHL   R9                                   ; 2633
                                   23     AE  9F 0017F          PUSHAB  LABEL_BUFFER+31                      ;
                                           04  DD 00182          PUSHL   #4                                   ;
                   00000000G  00           03  FB 00184          CALLS   #3, LIB$CVT_DTB                      ;
                              0D           50  E8 0018B          BLBS    R0, 15$                              ;
                                00000000G  8F  DD 0018E          PUSHL   #BACKUP$_NOTANSI                     ; 2634
                                           56  DD 00194          PUSHL   FAB                                  ;
                                           5B  DD 00196          PUSHL   R11                                  ;
                              6A           03  FB 00198          CALLS   #3, FILE_ERROR                       ;
                                           69  D5 0019B 15$:     TSTL    RWSV_FILE_NUMBER                     ; 2636
                                           27  12 0019D          BNEQ    16$                                  ;
                   00000000G  00           00  FB 0019F          CALLS   #0, REWIND                           ; 2639
                                 314C4F56  8F  DD 001A6          PUSHL   #827084630                           ; 2640
                                       04  AE  9F 001AC          PUSHAB  LABEL_BUFFER                         ;
                   00000000G  00           02  FB 001AF          CALLS   #2, READ_LABEL                       ;
                              57           50  D0 001B6          MOVL    R0, STATUS                           ;
                              13           57  E8 001B9          BLBS    STATUS, 17$                          ; 2641
                              7E           56  7D 001BC          MOVQ    FAB, -(SP)                           ;
                                           5B  DD 001BF          PUSHL   R11                                  ;
                              6A           03  FB 001C1          CALLS   #3, FILE_ERROR                       ;
                                           09  11 001C4          BRB     17$                                  ; 2636
                                           01  DD 001C6 16$:     PUSHL   #1                                   ; 2644
                   00000000G  00           01  FB 001C8          CALLS   #1, SKIP_TM                          ;
                                           69  D6 001CF 17$:     INCL    RWSV_FILE_NUMBER                     ; 2645
         50             58            05   08  EF 001D1 18$:     EXTZV   #8, #5, TAPE_CHAR, R0                ; 2648
                      00A9  C9             50  90 001D6          MOVB    R0, QUAL+77                          ;
                                           5E  DD 001DB          PUSHL   SP                                   ; 2653
                   00000000G  00           01  FB 001DD          CALLS   #1, MAKE_HDR1                        ;
                                           5E  DD 001E4          PUSHL   SP                                   ; 2654
                   00000000G  00           01  FB 001E6          CALLS   #1, WRITE_LABEL                      ;
                                           5E  DD 001ED          PUSHL   SP                                   ; 2656
                   00000000G  00           01  FB 001EF          CALLS   #1, MAKE_HDR2                        ;
                                           5E  DD 001F6          PUSHL   SP                                   ; 2657
                   00000000G  00           01  FB 001F8          CALLS   #1, WRITE_LABEL                      ;
                   00000000G  00           00  FB 001FF          CALLS   #0, WRITE_TM                         ; 2659
                   00000000G  00           00  FB 00206          CALLS   #0, SENSE_CHAR                       ; 2665
                              58           50  D0 0020D          MOVL    R0, TAPE_CHAR                        ;
                    05        58           12  E1 00210          BBC     #18, TAPE_CHAR, 19$                  ; 2666
                      00D6  C9             01  88 00214          BISB2   #1, COM_FLAGS                        ; 2667
                                               04 00219 19$:     RET                                          ; 2669

; Routine Size:  538 bytes,    Routine Base:  CODE + 0722
```

```
;  1122          2670  1 %SBTTL 'INIT_OUT_SAVE - initialize save set for output'
;  1123          2671  1 GLOBAL ROUTINE INIT_OUT_SAVE (CONTINUE) : NOVALUE =
;  1124          2672  1
;  1125          2673  1 !++
;  1126          2674  1 !
;  1127          2675  1 ! FUNCTIONAL DESCRIPTION:
;  1128          2676  1 !
;  1129          2677  1 !       This routine initializes the output save set.
;  1130          2678  1 !
;  1131          2679  1 ! CALLING SEQUENCE:
;  1132          2680  1 !       INIT_OUT_SAVE (CONTINUE)
;  1133          2681  1 !
;  1134          2682  1 ! INPUT PARAMETERS:
;  1135          2683  1 !       CONTINUE: FALSE if this is a new save set
;  1136          2684  1 !                 TRUE if this is for the next continuation volume
;  1137          2685  1 !
;  1138          2686  1 ! IMPLICIT INPUTS:
;  1139          2687  1 !       NONE
;  1140          2688  1 !
;  1141          2689  1 ! OUTPUT PARAMETERS:
;  1142          2690  1 !       NONE
;  1143          2691  1 !
;  1144          2692  1 ! IMPLICIT OUTPUTS:
;  1145          2693  1 !       NONE
;  1146          2694  1 !
;  1147          2695  1 ! ROUTINE VALUE:
;  1148          2696  1 !       NONE
;  1149          2697  1 !
;  1150          2698  1 ! SIDE EFFECTS:
;  1151          2699  1 !       NONE
;  1152          2700  1 !
;  1153          2701  1 !--
;  1154          2702  1
;  1155          2703  2 BEGIN
;  1156          2704  2
;  1157          2705  2 LOCAL
;  1158          2706  2         FAB             : REF BBLOCK,   ! pointer to output FAB
;  1159          2707  2         RAB             : REF BBLOCK,   ! pointer to output RAB
;  1160          2708  2         PROT_XAB        : $XABPRO_DECL, ! protection XAB for file creates
;  1161          2709  2         STATUS;                         ! the usual status value
;  1162          2710  2
;  1163          2711  2 EXTERNAL ROUTINE
;  1164          2712  2         EXTRACT_FILENAME,               ! extract file name from file string
;  1165          2713  2         FILE_ERROR;                     ! signal file related error
;  1166          2714  2
;  1167          2715  2 ! Do common setup. Get the next device spec in the list. If first
;  1168          2716  2 ! time or end of list, start at the beginning.
;  1169          2717  2 !
;  1170          2718  2
;  1171          2719  2 IF .RWSV_SAVE_QUAL NEQ 0 THEN RWSV_SAVE_QUAL = .RWSV_SAVE_QUAL[QUAL_NEXT];
;  1172          2720  2 IF .RWSV_SAVE_QUAL EQL 0 THEN RWSV_SAVE_QUAL = .QUAL[QUAL_OUTP_LIST];
;  1173          2721  2 RWSV_SAVE_FAB = FAB = .RWSV_SAVE_QUAL[QUAL_PARA_FC];
;  1174          2722  2
;  1175          2723  2 IF NOT .CONTINUE
;  1176          2724  2 THEN
;  1177          2725  3     BEGIN
;  1178          2726  3     RWSV_SEG_NUMBER = -1;
```

```
;  1179          2727  3      RWSV_VOL_NUMBER = 0;
;  1180          2728  3      RWSV_IN_GROUP_SIZE = RWSV_XORSIZE = .QUAL[QUAL_GROU_VALUE];
;  1181          2729  2      END;
;  1182          2730  2  RWSV_OUT_ERRORS = 0;
;  1183          2731  2  RWSV_OUT_BLOCK_COUNT = 0;
;  1184          2732  2  RWSV_OUT_GROUP_COUNT = 0;
;  1185          2733  2  RWSV_VOL_NUMBER = .RWSV_VOL_NUMBER + 1;
;  1186          2734  2  RWSV_SEG_NUMBER = .RWSV_SEG_NUMBER + 1;
;  1187          2735  2
;  1188          2736  2  ! Open the save set file through RMS.
;  1189          2737  2  !
;  1190          2738  2
;  1191          2739  2  FAB[FAB$V_NAM] = TRUE;
;  1192          2740  2  IF .QUAL[QUAL_SS_FILE]
;  1193          2741  2  THEN
;  1194          2742  3      BEGIN
;  1195          2743  3      FAB[FAB$V_PUT] = TRUE;
;  1196          2744  3      FAB[FAB$V_SUP] = TRUE;
;  1197          2745  3      FAB[FAB$W_MRS] = .QUAL[QUAL_BLOC_VALUE];
;  1198          2746  3      FAB[FAB$B_ORG] = FAB$C_SEQ;
;  1199          2747  3      FAB[FAB$B_RAT] = 0;
;  1200          2748  3      FAB[FAB$B_RFM] = FAB$C_FIX;
;  1201          2749  3      IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  1202          2750  3      THEN FAB[FAB$W_BLS] = .QUAL[QUAL_BLOC_VALUE];
;  1203          2751  3
;  1204          2752  3      $XABPRO_INIT (XAB = PROT_XAB);
;  1205          2753  3      PROT_XAB[XAB$W_PRO] = -1;
;  1206          2754  3      IF .QUAL[QUAL_O_OWN_UIC]
;  1207          2755  3      THEN PROT_XAB[XAB$L_UIC] = .QUAL[QUAL_O_OWN_VALU];
;  1208          2756  3      IF .QUAL[QUAL_PROT]
;  1209          2757  3      THEN PROT_XAB[XAB$W_PRO] = .QUAL[QUAL_PROT_VALUE];
;  1210          2758  3      FAB[FAB$L_XAB] = PROT_XAB;
;  1211          2759  3      END
;  1212          2760  2  ELSE
;  1213          2761  2      FAB[FAB$V_UFO] = TRUE;
;  1214          2762  2
;  1215          2763  2  IF .QUAL[QUAL_SS_FILE]
;  1216          2764  2  OR .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  1217          2765  2  THEN
;  1218          2766  3      BEGIN
;  1219          2767  3      IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_NET]
;  1220          2768  3      THEN
;  1221          2769  3          FAB[FAB$V_BIO] = FAB[FAB$V_SQO] = TRUE;
;  1222          2770  3
;  1223          2771  3      STATUS = $CREATE (FAB = .FAB);
;  1224          2772  3      FAB[FAB$L_XAB] = 0;
;  1225          2773  3      IF NOT .STATUS
;  1226          2774  3      THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB,
;  1227          2775  3                       .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
;  1228          2776  3      IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SWL]
;  1229          2777  3      THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB, SS$_WRITLCK);
;  1230          2778  3      END
;  1231          2779  2  ELSE
;  1232          2780  2      INIT_SAVE_DISK (.CONTINUE);
;  1233          2781  2
;  1234          2782  2  IF .QUAL[QUAL_SS_FILE]
;  1235          2783  2  THEN
```

```
; 1236      2784 3      BEGIN
; 1237      2785 3      RAB = FAB[FC_RAB];
; 1238      2786 3      STATUS = $CONNECT (RAB = .RAB);
; 1239      2787 3      IF NOT .STATUS
; 1240      2788 3      THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .FAB,
; 1241      2789 3                       .RAB[RAB$L_STS], .RAB[RAB$L_STV]);
; 1242      2790 3      END
; 1243      2791 2  ELSE
; 1244      2792 2      RWSV_CHAN = .FAB[FAB$L_STV];
; 1245      2793 2
; 1246      2794 2  IF NOT .CONTINUE
; 1247      2795 2  THEN EXTRACT_FILENAME (.FAB, COM_SSNAME);
; 1248      2796 2
; 1249      2797 2  ! Set up the output if it is a  tape.
; 1250      2798 2  !
; 1251      2799 2
; 1252      2800 2  IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
; 1253      2801 2  AND NOT .QUAL[QUAL_SS_FILE]
; 1254      2802 2  THEN
; 1255      2803 2      INIT_SAVE_TAPE (.CONTINUE);
; 1256      2804 2
; 1257      2805 1  END;                                        ! End of routine INIT_OUT_SAVE


                                                          .EXTRN  EXTRACT_FILENAME
                                                          .EXTRN  SYS$CREATE, SYS$CONNECT

                                         03FC 00000       .ENTRY  INIT_OUT_SAVE, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 2671
               59 00000000G  00    9E 00002               MOVAB   FILE_ERROR, R9                                ;
               58 00000000G  8F    D0 00009               MOVL    #BACKUP$_OPENOUT+4, R8                        ;
               57 00000000'  EF    9E 00010               MOVAB   QUAL+12, R7                                   ;
               5E        A8  AE    9E 00017               MOVAB   -88(SP), SP                                   ;
               50        9C  A7    D0 0001B               MOVL    RWSV_SAVE_QUAL, R0                            ; 2719
                             06    13 0001F               BEQL    1$                                            ;
         9C    A7            60    D0 00021               MOVL    (R0), RWSV_SAVE_QUAL                          ;
                             05    12 00025               BNEQ    2$                                            ; 2720
         9C    A7        F8  A7    D0 00027 1$:           MOVL    QUAL+4, RWSV_SAVE_QUAL                        ;
               50        9C  A7    D0 0002C 2$:           MOVL    RWSV_SAVE_QUAL, R0                            ; 2721
               56        04  A0    D0 00030               MOVL    4(R0), FAB                                    ;
         A0    A7            56    D0 00034               MOVL    FAB, RWSV_SAVE_FAB                            ;
               14        04  AC    E8 00038               BLBS    CONTINUE, 3$                                  ; 2723
         94    A7 FFFF0000   8F    D0 0003C               MOVL    #-65536, RWSV_VOL_NUMBER                      ; 2727
               50        42  A7    9A 00044               MOVZBL  QUAL+78, R0                                   ; 2728
         BF    A7            50    90 00048               MOVB    R0, RWSV_XORSIZE                              ;
         C0    A7            50    D0 0004C               MOVL    R0, RWSV_IN_GROUP_SIZE                        ;
                         EC  A7    B4 00050 3$:           CLRW    RWSV_OUT_ERRORS                               ; 2730
                         E8  A7    D4 00053               CLRL    RWSV_OUT_BLOCK_COUNT                          ; 2731
                         F0  A7    94 00056               CLRB    RWSV_OUT_GROUP_COUNT                          ; 2732
                         94  A7    B6 00059               INCW    RWSV_VOL_NUMBER                               ; 2733
                         96  A7    B6 0005C               INCW    RWSV_SEG_NUMBER                               ; 2734
         07    A6            01    88 0005F               BISB2   #1, 7(FAB)                                    ; 2739
    48   03    A7            03    E1 00063               BBC     #3, QUAL+15, 7$                               ; 2740
         16    A6            01    88 00068               BISB2   #1, 22(FAB)                                   ; 2743
         04    A6            04    88 0006C               BISB2   #4, 4(FAB)                                    ; 2744
         36    A6        3C  A7    B0 00070               MOVW    QUAL+72, 54(FAB)                              ; 2745
                         1D  A6    B4 00075               CLRW    29(FAB)                                       ; 2746
```

```
                                 1F  A6       01  90 00078        MOVB    #1, 31(FAB)                          ; 2748
                   05            40  A6       05  E1 0007C        BBC     #5, 64(FAB), 4$                      ; 2749
                                 3C  A6   3C  A7  B0 00081        MOVW    QUAL+72, 60(FAB)                     ; 2750
   0058   8F       00                6E       00  2C 00086 4$:    MOVC5   #0, (SP), #0, #88, $RMS_PTR          ; 2752
                                              6E     0008D                                                     ;
                                 6E     5813  8F  B0 0008E        MOVW    #22547, $RMS_PTR                     ;
                                 08  AE       01  AE 00093        MNEGW   #1, PROT_XAB+8                       ; 2753
                   05            67           04  E1 00097        BBC     #4, QUAL+12, 5$                      ; 2754
                                 0C  AE   34  A7  D0 0009B        MOVL    QUAL+64, PROT_XAB+12                 ; 2755
                   05            02  A7       03  E1 000A0 5$:    BBC     #3, QUAL+14, 6$                      ; 2756
                                 08  AE   48  A7  B0 000A5        MOVW    QUAL+84, PROT_XAB+8                  ; 2757
                                 24  A6       6E  9E 000AA 6$:    MOVAB   PROT_XAB, 36(FAB)                    ; 2758
                                              04  11 000AE        BRB     8$                                   ; 2740
                                 06  A6       02  88 000B0 7$:    BISB2   #2, 6(FAB)                           ; 2761
                   05            03  A7       03  E0 000B4 8$:    BBS     #3, QUAL+15, 9$                      ; 2763
                   3E            40  A6       05  E1 000B9        BBC     #5, 64(FAB), 12$                     ; 2764
                   09            41  A6       05  E1 000BE 9$:    BBC     #5, 65(FAB), 10$                     ; 2767
                                 04  A6   40  8F  88 000C3        BISB2   #64, 4(FAB)                          ; 2769
                                 16  A6       20  88 000C8        BISB2   #32, 22(FAB)                         ;
                                              56  DD 000CC 10$:   PUSHL   FAB                                  ; 2771
                        00000000G    00       01  FB 000CE        CALLS   #1, SYS$CREATE                       ;
                                     53       50  D0 000D5        MOVL    R0, STATUS                           ;
                                          24  A6  D4 000D8        CLRL    36(FAB)                              ; 2772
                                     0B       53  E8 000DB        BLBS    STATUS, 11$                          ; 2773
                                     7E   08  A6  7D 000DE        MOVQ    8(FAB), -(SP)                        ; 2775
                                              56  DD 000E2        PUSHL   FAB                                  ; 2774
                                              58  DD 000E4        PUSHL   R8                                   ;
                                     69       04  FB 000E6        CALLS   #4, FILE_ERROR                       ;
                   16            43  A6       01  E1 000E9 11$:   BBC     #1, 67(FAB), 13$                     ; 2776
                                     7E 025C  8F  3C 000EE        MOVZWL  #604, -(SP)                          ; 2777
                                              56  DD 000F3        PUSHL   FAB                                  ;
                                              58  DD 000F5        PUSHL   R8                                   ;
                                     69       03  FB 000F7        CALLS   #3, FILE_ERROR                       ;
                                              08  11 000FA        BRB     13$                                  ; 2763
                                          04  AC  DD 000FC 12$:   PUSHL   CONTINUE                             ; 2780
                             F8E8    CF       01  FB 000FF        CALLS   #1, INIT_SAVE_DISK                   ;
                   20            03  A7       03  E1 00104 13$:   BBC     #3, QUAL+15, 14$                     ; 2782
                                 52       50  A6  9E 00109        MOVAB   80(R6), RAB                          ; 2785
                                              52  DD 0010D        PUSHL   RAB                                  ; 2786
                        00000000G    00       01  FB 0010F        CALLS   #1, SYS$CONNECT                      ;
                                     53       50  D0 00116        MOVL    R0, STATUS                           ;
                                     12       53  E8 00119        BLBS    STATUS, 15$                          ; 2787
                                     7E   08  A2  7D 0011C        MOVQ    8(RAB), -(SP)                        ; 2789
                                              56  DD 00120        PUSHL   FAB                                  ; 2788
                                              58  DD 00122        PUSHL   R8                                   ;
                                     69       04  FB 00124        CALLS   #4, FILE_ERROR                       ;
                                              05  11 00127        BRB     15$                                  ; 2782
                                 A4  A7   0C  A6  D0 00129 14$:   MOVL    12(FAB), RWSV_CHAN                   ; 2792
                                     0C   04  AC  E8 0012E 15$:   BLBS    CONTINUE, 16$                        ; 2794
                                          64  A7  9F 00132        PUSHAB  COM_SSNAME                           ; 2795
                                              56  DD 00135        PUSHL   FAB                                  ;
                        00000000G    00       02  FB 00137        CALLS   #2, EXTRACT_FILENAME                 ;
                   0D            40  A6       05  E1 0013E 16$:   BBC     #5, 64(FAB), 17$                     ; 2800
                   08            03  A7       03  E0 00143        BBS     #3, QUAL+15, 17$                     ; 2801
                                          04  AC  DD 00148        PUSHL   CONTINUE                             ; 2803
                             FC96    CF       01  FB 0014B        CALLS   #1, INIT_SAVE_TAPE                   ;
                                              04 00150 17$:       RET                                          ; 2805
```

; Routine Size: 337 bytes, Routine Base: CODE + 093C

```
;  1259          2806  1 %SBTTL 'WRITE_BUFFER - write a save set buffer'
;  1260          2807  1 GLOBAL ROUTINE WRITE_BUFFER (BCB) : NOVALUE =
;  1261          2808  1
;  1262          2809  1 !++
;  1263          2810  1 !
;  1264          2811  1 ! FUNCTIONAL DESCRIPTION:
;  1265          2812  1 !
;  1266          2813  1 !       This routine writes the indicated buffer to the output save set.
;  1267          2814  1 !
;  1268          2815  1 ! CALLING SEQUENCE:
;  1269          2816  1 !       WRITE_BUFFER (BCB)
;  1270          2817  1 !
;  1271          2818  1 ! INPUT PARAMETERS:
;  1272          2819  1 !       BCB: address of buffer control block
;  1273          2820  1 !
;  1274          2821  1 ! IMPLICIT INPUTS:
;  1275          2822  1 !       NONE
;  1276          2823  1 !
;  1277          2824  1 ! OUTPUT PARAMETERS:
;  1278          2825  1 !       NONE
;  1279          2826  1 !
;  1280          2827  1 ! IMPLICIT OUTPUTS:
;  1281          2828  1 !       NONE
;  1282          2829  1 !
;  1283          2830  1 ! ROUTINE VALUE:
;  1284          2831  1 !       NONE
;  1285          2832  1 !
;  1286          2833  1 ! SIDE EFFECTS:
;  1287          2834  1 !       NONE
;  1288          2835  1 !
;  1289          2836  1 !--
;  1290          2837  1
;  1291          2838  2 BEGIN
;  1292          2839  2
;  1293          2840  2 BUILTIN
;  1294          2841  2         INSQUE,
;  1295          2842  2         CRC;                            ! compute CRC instruction
;  1296          2843  2
;  1297          2844  2 MAP
;  1298          2845  2         BCB             : REF BBLOCK;   ! buffer control block arg
;  1299          2846  2
;  1300          2847  2 LOCAL
;  1301          2848  2         STATUS,                         ! general status value
;  1302          2849  2         RAB             : REF BBLOCK,   ! output RAB
;  1303          2850  2         BUFFER          : REF BBLOCK,   ! I/O buffer
;  1304          2851  2         XOR_BUFFER      : REF BBLOCK,   ! XOR accumulation buffer
;  1305          2852  2         P1,                             ! buffer pointer
;  1306          2853  2         P2;                             ! buffer pointer
;  1307          2854  2
;  1308          2855  2 EXTERNAL ROUTINE
;  1309          2856  2         GET_BUFFER,                     ! get an I/O buffer
;  1310          2857  2         FREE_BUFFER;                    ! free an I/O buffer
;  1311          2858  2
;  1312          2859  2 ! Do the block preamble formatting.
;  1313          2860  2 !
;  1314          2861  2
;  1315          2862  2 BUFFER = .BCB[BCB_BUFFER];
```

```
;  1316          2863  2 CH$FILL (0, $BYTEOFFSET (BBH$W_FID), .BUFFER);
;  1317          2864  2 CH$FILL (0, BBH$K_LENGTH - $BYTEOFFSET (BBH$T_RESERVED2), BUFFER[BBH$T_RESERVED2]);
;  1318          2865  2 BUFFER[BBH$W_SIZE] = BBH$K_LENGTH;
;  1319          2866  2 BUFFER[BBH$W_OPSYS] = BACKUP$K_OPSYS;
;  1320          2867  2 BUFFER[BBH$W_SUBSYS] = BACKUP$K_BACKUP;
;  1321          2868  2 BUFFER[BBH$W_APPLIC] = BACKUP$K_DATABLOCK;
;  1322          2869  2 RWSV_OUT_SEQ = .RWSV_OUT_SEQ + 1;
;  1323          2870  2 RWSV_OUT_BLOCK_COUNT = .RWSV_OUT_BLOCK_COUNT + 1;
;  1324          2871  2 BUFFER[BBH$L_NUMBER] = .RWSV_OUT_SEQ;
;  1325          2872  2 BUFFER[BBH$W_STRUCLEV] = BBH$K_LEVEL1;
;  1326          2873  2 BUFFER[BBH$W_VOLNUM] = .RWSV_VOL_NUMBER;
;  1327          2874  2 BUFFER[BBH$L_BLOCKSIZE] = .BCB[BCB_SIZE];
;  1328          2875  2 (BUFFER[BBH$T_SSNAME])<0,8> =.COM_SSNAME[DSC$W_LENGTH];
;  1329          2876  2 CH$COPY (.COM_SSNAME[DSC$W_LENGTH],
;  1330          2877  2         .COM_SSNAME[DSC$A_POINTER],
;  1331          2878  2         0,
;  1332          2879  2         BBH$S_SSNAME-1,
;  1333          2880  2         BUFFER[BBH$T_SSNAME]+1);
;  1334          2881  2
;  1335          2882  2 ! Accumulate the block into the running XOR. If the is the first block
;  1336          2883  2 ! of a group, we have to allocate a new buffer.
;  1337          2884  2 !
;  1338          2885  2
;  1339          2886  2 IF .QUAL[QUAL_GROU_VALUE] NEQ 0
;  1340          2887  2 THEN
;  1341          2888  3     BEGIN
;  1342          2889  3     IF .RWSV_OUT_GROUP_COUNT EQL 0
;  1343          2890  3     THEN
;  1344          2891  4         BEGIN
;  1345          2892  4         RWSV_XOR_BCB = GET_BUFFER ();
;  1346          2893  4         XOR_BUFFER = .RWSV_XOR_BCB[BCB_BUFFER];
;  1347          2894  4         CH$MOVE (.BCB[BCB_SIZE], .BUFFER, .XOR_BUFFER);
;  1348          2895  4         XOR_BUFFER[BBH$W_APPLIC] = BACKUP$K_XORBLOCK;
;  1349          2896  4         END
;  1350          2897  3     ELSE
;  1351          2898  4         BEGIN
;  1352          2899  4         XOR_BUFFER = .RWSV_XOR_BCB[BCB_BUFFER];
;  1353          2900  4         P1 = .BUFFER + BBH$K_COMMON;
;  1354          2901  4         P2 = .XOR_BUFFER + BBH$K_COMMON;
;  1355          2902  4         DECR J FROM (.BCB[BCB_SIZE]-BBH$K_COMMON)/16 TO 1
;  1356          2903  4         DO
;  1357          2904  5             BEGIN
;  1358          2905  5             .P2 = ..P2 XOR ..P1;
;  1359          2906  5             P1 = .P1 + 4;
;  1360          2907  5             P2 = .P2 + 4;
;  1361          2908  5             .P2 = ..P2 XOR ..P1;
;  1362          2909  5             P1 = .P1 + 4;
;  1363          2910  5             P2 = .P2 + 4;
;  1364          2911  5             .P2 = ..P2 XOR ..P1;
;  1365          2912  5             P1 = .P1 + 4;
;  1366          2913  5             P2 = .P2 + 4;
;  1367          2914  5             .P2 = ..P2 XOR ..P1;
;  1368          2915  5             P1 = .P1 + 4;
;  1369          2916  5             P2 = .P2 + 4;
;  1370          2917  4             END;
;  1371          2918  3         END;
;  1372          2919  2     END;
```

```
2
2 ! Finally write the block.
2 !
2
2 WRITE_BLOCK (.BCB);
2
2 ! Bump the group count; if the group is full, write the XOR block.
2 ! We do not write the XOR block if we are into EOV, since one is
2 ! about to be written by FIN_OUT_SAVE anyway. In the boundary case,
2 ! this means that we can end up with an XOR group of n+1 blocks,
2 ! rather than another XOR group of one block. This gambit reduces by
2 ! one tle maximum number of blocks written after COM_EOV is raised.
2 !
2
2 IF .QUAL[QUAL_GROU_VALUE] NEQ 0
2 THEN
3     BEGIN
3     RWSV_OUT_GROUP_COUNT = .RWSV_OUT_GROUP_COUNT + 1;
3     IF .RWSV_OUT_GROUP_COUNT GEQU .QUAL[QUAL_GROU_VALUE]
3     AND NOT .COM_FLAGS[COM_EOV]
3     THEN
4         BEGIN
4         RWSV_OUT_SEQ = .RWSV_OUT_SEQ + 1;
4         RWSV_OUT_BLOCK_COUNT = .RWSV_OUT_BLOCK_COUNT + 1;
4         XOR_BUFFER[BBH$L_NUMBER] = .RWSV_OUT_SEQ;
4         WRITE_BLOCK (.RWSV_XOR_BCB);
4         RWSV_OUT_GROUP_COUNT = 0;
4         RWSV_XOR_BCB = 0;
3         END;
2     END;
1 END;                                          ! End of routine WRITE_BUFFER


                                                                  .EXTRN  GET_BUFFER

                                       03FC 00000                 .ENTRY  WRITE_BUFFER, Save R2,R3,R4,R5,R6,R7,R8,R9   ; 2807
                        59 00000000'  EF   9E 00002               MOVAB   RWSV_OUT_SEQ, R9                                   ;
                        58        04  AC   D0 00009               MOVL    BCB, R8                                            ; 2862
                        57        0C  A8   D0 0000D               MOVL    12(R8), BUFFER                                     ;
 0050   8F        00        6E         00  2C 00011               MOVC5   #0, (SP), #0, #80, (BUFFER)                        ; 2863
                                       67     00018                                                                          ;
        18        00        6E         00  2C 00019               MOVC5   #0, (SP), #0, #24, 232(BUFFER)                     ; 2864
                              00E8     C7     0001E                                                                          ;
                        67 04000100    8F  D0 00021               MOVL    #67109120, (BUFFER)                                ; 2865
                  04    A7 00010001    8F  D0 00028               MOVL    #65537, 4(BUFFER)                                  ; 2867
                                       69  D6 00030               INCL    RWSV_OUT_SEQ                                       ; 2869
                                  08   A9  D6 00032               INCL    RWSV_OUT_BLOCK_COUNT                               ; 2870
                  08    A7             69  D0 00035               MOVL    RWSV_OUT_SEQ, 8(BUFFER)                            ; 2871
                  20    A7      0101   8F  B0 00039               MOVW    #257, 32(BUFFER)                                   ; 2872
                  22    A7        B4   A9  B0 0003F               MOVW    RWSV_VOL_NUMBER, 34(BUFFER)                        ; 2873
                  28    A7        08   A8  3C 00044               MOVZWL  8(R8), 40(BUFFER)                                  ; 2874
                  30    A7      0084   C9  90 00049               MOVB    COM_SSNAME, 48(BUFFER)                             ; 2875
        1F        00  0088  D9  0084   C9  2C 0004F               MOVC5   COM_SSNAME, @COM_SSNAME+4, #0, #31, -              ; 2880
                                  31   A7     00058                       49(BUFFER)                                         ;
                                  62   A9  95 0005A               TSTB    QUAL+78                                            ; 2886
```

```
                                 4C  13 0005D        BEQL    4$                                          ;
                           10    A9  95 0005F        TSTB    RWSV_OUT_GROUP_COUNT                        ; 2889
                                 1A  12 00062        BNEQ    1$                                          ;
             00000000G  00       00  FB 00064        CALLS   #0, GET_BUFFER                              ; 2892
                    C8  A9       50  D0 0006B        MOVL    R0, RWSV_XOR_BCB                            ;
                        56  0C   A0  D0 0006F        MOVL    12(R0), XOR_BUFFER                          ; 2893
        66              67  08   A8  28 00073        MOVC3   8(R8), (BUFFER), (XOR_BUFFER)               ; 2894
                06      A6       02  B0 00078        MOVW    #2, 6(XOR_BUFFER)                           ; 2895
                                 2D  11 0007C        BRB     4$                                          ; 2889
                        50  C8   A9  D0 0007E 1$:    MOVL    RWSV_XOR_BCB, R0                            ; 2899
                        56  0C   A0  D0 00082        MOVL    12(R0), XOR_BUFFER                          ;
                        51  20   A7  9E 00086        MOVAB   32(BUFFER), P1                              ; 2900
                        50  20   A6  9E 0008A        MOVAB   32(R6), P2                                  ; 2901
                        52  08   A8  3C 0008E        MOVZWL  8(R8), R2                                   ; 2902
                        52       20  C2 00092        SUBL2   #32, R2                                     ;
                        52       10  C6 00095        DIVL2   #16, R2                                     ;
                                 52  D6 00098        INCL    J                                           ;
                                 0C  11 0009A        BRB     3$                                          ;
                        80       81  CC 0009C 2$:    XORL2   (P1)+, (P2)+                                ; 2905
                        80       81  CC 0009F        XORL2   (P1)+, (P2)+                                ; 2908
                        80       81  CC 000A2        XORL2   (P1)+, (P2)+                                ; 2911
                        80       81  CC 000A5        XORL2   (P1)+, (P2)+                                ; 2914
                        F1       52  F5 000A8 3$:    SOBGTR  J, 2$                                       ; 2902
                                 58  DD 000AB 4$:    PUSHL   R8                                          ; 2924
                  F5FF  CF       01  FB 000AD        CALLS   #1, WRITE_BLOCK                             ;
                        50  62   A9  9A 000B2        MOVZBL  QUAL+78, R0                                 ; 2934
                                 25  13 000B6        BEQL    5$                                          ;
                            10   A9  96 000B8        INCB    RWSV_OUT_GROUP_COUNT                        ; 2937
                        50  10   A9  91 000BB        CMPB    RWSV_OUT_GROUP_COUNT, R0                    ; 2938
                                 1C  1F 000BF        BLSSU   5$                                          ;
                        17 008E  C9  E8 000C1        BLBS    COM_FLAGS, 5$                               ; 2939
                                 69  D6 000C6        INCL    RWSV_OUT_SEQ                                ; 2942
                            08   A9  D6 000C8        INCL    RWSV_OUT_BLOCK_COUNT                        ; 2943
                08      A6       69  D0 000CB        MOVL    RWSV_OUT_SEQ, 8(XOR_BUFFER)                 ; 2944
                            C8   A9  DD 000CF        PUSHL   RWSV_XOR_BCB                                ; 2945
                  F5DA  CF       01  FB 000D2        CALLS   #1, WRITE_BLOCK                             ;
                            10   A9  94 000D7        CLRB    RWSV_OUT_GROUP_COUNT                        ; 2946
                            C8   A9  D4 000DA        CLRL    RWSV_XOR_BCB                                ; 2947
                                 04     000DD 5$:    RET                                                 ; 2951

; Routine Size:  222 bytes,    Routine Base:  CODE + 0A8D
```

```
;  1406          2952  1 %SBTTL 'FIN_OUT_SAVE - finish writing save set'
;  1407          2953  1 GLOBAL ROUTINE FIN_OUT_SAVE (CONTINUE) : NOVALUE =
;  1408          2954  1
;  1409          2955  1 !++
;  1410          2956  1 !
;  1411          2957  1 ! FUNCTIONAL DESCRIPTION:
;  1412          2958  1 !
;  1413          2959  1 !       This routine completes the writing of the output save set.
;  1414          2960  1 !
;  1415          2961  1 ! CALLING SEQUENCE:
;  1416          2962  1 !       FIN_OUT_SAVE (CONTINUE)
;  1417          2963  1 !
;  1418          2964  1 ! INPUT PARAMETERS:
;  1419          2965  1 !       CONTINUE: FALSE if this is the real end of the save set
;  1420          2966  1 !                 TRUE if this is the end of a save set volume
;  1421          2967  1 !
;  1422          2968  1 ! IMPLICIT INPUTS:
;  1423          2969  1 !       NONE
;  1424          2970  1 !
;  1425          2971  1 ! OUTPUT PARAMETERS:
;  1426          2972  1 !       NONE
;  1427          2973  1 !
;  1428          2974  1 ! IMPLICIT OUTPUTS:
;  1429          2975  1 !       NONE
;  1430          2976  1 !
;  1431          2977  1 ! ROUTINE VALUE:
;  1432          2978  1 !       NONE
;  1433          2979  1 !
;  1434          2980  1 ! SIDE EFFECTS:
;  1435          2981  1 !       NONE
;  1436          2982  1 !
;  1437          2983  1 !--
;  1438          2984  1
;  1439          2985  2 BEGIN
;  1440          2986  2
;  1441          2987  2 BUILTIN
;  1442          2988  2         ROT,
;  1443          2989  2         INSQUE,
;  1444          2990  2         REMQUE;
;  1445          2991  2
;  1446          2992  2 LOCAL
;  1447          2993  2         STATUS,                         ! general status value
;  1448          2994  2         IO_STATUS       : VECTOR [4, WORD], ! I/O status block
;  1449          2995  2         BCB             : REF BBLOCK,   ! pointer to buffer control block
;  1450          2996  2         XOR_BUFFER      : REF BBLOCK,   ! buffer holding XOR block
;  1451          2997  2         FAB             : REF BBLOCK,   ! output FAB
;  1452          2998  2         LABEL_BUFFER    : BBLOCK [90],  ! buffer for tape labels
;  1453          2999  2         DESCRIPTOR      : VECTOR [2],   ! descriptor for above
;  1454          3000  2         BLOCKS_ALLOC,                   ! blocks allocated or returned
;  1455          3001  2         VCB             : REF BBLOCK,   ! VCB of current disk volume
;  1456          3002  2         RECATTR         : BBLOCK [FAT$C_LENGTH], ! record attributes buffer
;  1457          3003  2         CONTINUE_FID    : BBLOCK [FID$C_LENGTH], ! save set extension file ID
;  1458          3004  2         ATT_CONTROL     : BBLOCK [20];  ! attribute control list
;  1459          3005  2
;  1460          3006  2 BIND
;  1461          3007  2         ATT_CONTROL1    = ATT_CONTROL + 0 : BBLOCK,
;  1462          3008  2         ATT_CONTROL2    = ATT_CONTROL + 8 : BBLOCK,
```

```
; 1463          3009  2          ATTR_END        = ATT_CONTROL + 16;
; 1464          3010  2
; 1465          3011  2 EXTERNAL ROUTINE
; 1466          3012  2          WAIT,                          ! wait for I/O completion                                  P
; 1467          3013  2          FILE_ERROR,                    ! signal file related error                                -
; 1468          3014  2          FREE_BUFFER,                   ! free an I/O buffer
; 1469          3015  2          GET_BUFFER,                    ! allocate an I/O buffer                                   $
; 1470          3016  2          SWITCH_VOLUME,                 ! switch to desired disk volume
; 1471          3017  2          STA_EXTEND,                    ! extend (or truncate) save set file
; 1472          3018  2          STA_DISMOUNT_OUTPUT,           ! dismount output disk volume                              $
; 1473          3019  2          STA_DISMOUNT,                  ! complete dismount
; 1474          3020  2          INIT_IN_SAVE,                  ! initialize input save set
; 1475          3021  2          READ_NEXT_VOLUME,              ! set up continuation volume for input
; 1476          3022  2          SAVE_VERIFY_REEL,              ! verify a save set volume
; 1477          3023  2          FIN_IN_SAVE,                   ! finish input save set
; 1478          3024  2          UNLOAD,                        ! unload a tape volume
; 1479          3025  2          SKIP_TM,                       ! skip tape marks                                          -
; 1480          3026  2          MAKE_HDR1,                     ! generate file header label 1
; 1481          3027  2          MAKE_HDR2,                     ! generate file header label 2
; 1482          3028  2          WRITE_LABEL,                   ! write file header label                                  $
; 1483          3029  2          WRITE_TM;                      ! write tape mark
; 1484          3030  2
; 1485          3031  2 ! Write the final XOR block.
; 1486          3032  2 !
; 1487          3033  2
; 1488          3034  2 IF .RWSV_OUT_GROUP_COUNT NEQ 0
; 1489          3035  2 THEN
; 1490          3036  3     BEGIN
; 1491          3037  3     XOR_BUFFER = .RWSV_XOR_BCB[BCB_BUFFER];                                                           $
; 1492          3038  3     RWSV_OUT_SEQ = .RWSV_OUT_SEQ + 1;
; 1493          3039  3     RWSV_OUT_BLOCK_COUNT = .RWSV_OUT_BLOCK_COUNT + 1;
; 1494          3040  3     XOR_BUFFER[BBH$L_NUMBER] = .RWSV_OUT_SEQ;
; 1495          3041  3     WRITE_BLOCK (.RWSV_XOR_BCB);                                                                      |
; 1496          3042  3     RWSV_XOR_BCB = 0;                                                                                 -
; 1497          3043  3     RWSV_OUT_GROUP_COUNT = 0;
; 1498          3044  2     END;
; 1499          3045  2
; 1500          3046  2 ! Wait for all pending writes to complete.                                                           -
; 1501          3047  2 !
; 1502          3048  2
; 1503          3049  2 UNTIL REMQUE (.OUTPUT_WAIT[0], BCB)                                                                -
; 1504          3050  2 DO
; 1505          3051  3     BEGIN
; 1506          3052  3     WAIT (.BCB);                                                                                      P
; 1507          3053  3     FREE_BUFFER (.BCB);
; 1508          3054  2     END;
; 1509          3055  2
; 1510          3056  2 ! If the output is a file, close it.                                                               P
; 1511          3057  2 !
; 1512          3058  2
; 1513          3059  2 FAB = .RWSV_SAVE_FAB;                                                                             P
; 1514          3060  2 IF .QUAL[QUAL_SS_FILE]
; 1515          3061  2 THEN                                                                                               P
; 1516          3062  3     BEGIN
; 1517          3063  3     IF NOT .QUAL[QUAL_VERI]
; 1518          3064  3     THEN
; 1519          3065  4         BEGIN
```

```
;  1520          3066  4          STATUS = $CLOSE (FAB = .FAB);
;  1521          3067  4          IF NOT .STATUS
;  1522          3068  4          THEN FILE_ERROR (BACKUP$_CLOSEOUT+STS$K_SEVERE, .FAB,
;  1523          3069  4                           .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
;  1524          3070  3          END;
;  1525          3071  3      END
;  1526          3072  3
;  1527          3073  3  ! Generate and write the trailer label set for tape output.
;  1528          3074  3  !
;  1529          3075  3
;  1530          3076  2  ELSE
;  1531          3077  3      BEGIN
;  1532          3078  3      IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  1533          3079  3      THEN
;  1534          3080  4          BEGIN
;  1535          3081  4          WRITE_TM ();
;  1536          3082  4
;  1537          3083  4          MAKE_HDR1 (LABEL_BUFFER);
;  1538          3084  4          IF .CONTINUE
;  1539          3085  4          THEN LABEL_BUFFER[HD1$L_HD1LID] = 'EOV1'
;  1540          3086  4          ELSE LABEL_BUFFER[HD1$L_HD1LID] = 'EOF1';
;  1541          3087  4          DESCRIPTOR[0] = HD1$S_BLOCKCNT;
;  1542          3088  4          DESCRIPTOR[1] = LABEL_BUFFER[HD1$T_BLOCKCNT];
;  1543        P 3089  4          $FAO ($DESCRIPTOR ('!6ZL'),
;  1544        P 3090  4                  0,
;  1545        P 3091  4                  DESCRIPTOR,
;  1546        P 3092  4                  .RWSV_OUT_BLOCK_COUNT
;  1547          3093  4                  );
;  1548          3094  4          WRITE_LABEL (LABEL_BUFFER);
;  1549          3095  4
;  1550          3096  4          MAKE_HDR2 (LABEL_BUFFER);
;  1551          3097  4          IF .CONTINUE
;  1552          3098  4          THEN LABEL_BUFFER[HD1$L_HD1LID] = 'EOV2'
;  1553          3099  4          ELSE LABEL_BUFFER[HD1$L_HD1LID] = 'EOF2';
;  1554          3100  4          WRITE_LABEL (LABEL_BUFFER);
;  1555          3101  4
;  1556          3102  4  ! Mark the end of tape with two tape marks.
;  1557          3103  4  !
;  1558          3104  4
;  1559          3105  4          WRITE_TM ();
;  1560          3106  4          WRITE_TM ();
;  1561          3107  4          SKIP_TM (-2);
;  1562          3108  4
;  1563          3109  4          IF NOT .QUAL[QUAL_VERI]
;  1564          3110  4          THEN
;  1565          3111  5              BEGIN
;  1566          3112  5              IF .CONTINUE
;  1567          3113  5              THEN UNLOAD ();
;  1568          3114  5              $DASSGN (CHAN = .RWSV_CHAN);
;  1569          3115  5              RWSV_CHAN = 0;
;  1570          3116  4              END;
;  1571          3117  4          END
;  1572          3118  4
;  1573          3119  4  ! Finish up sequential disk. Return unused blocks, set up the record
;  1574          3120  4  ! attributes, and deaccess the file.
;  1575          3121  4  !
;  1576          3122  4
```

```
;  1577          3123  3     ELSE
;  1578          3124  4         BEGIN
;  1579          3125  4         CURRENT_MTL = .OUTPUT_MTL;
;  1580          3126  4         SWITCH_VOLUME (.RWSV_VOL_NUMBER);
;  1581          3127  4         STATUS = STA_EXTEND (.RWSV_ALLOC, BLOCKS_ALLOC);
;  1582          3128  4         IF NOT .STATUS
;  1583          3129  4         THEN FILE_ERROR (BACKUP$_CLOSEOUT+STS$K_SEVERE, .FAB, .STATUS);
;  1584          3130  4         RWSV_EOF = .RWSV_OUT_VBN - 1;
;  1585          3131  4
;  1586          3132  4         CH$FILL (0, FAT$C_LENGTH, RECATTR);
;  1587          3133  4         RECATTR[FAT$B_RTYPE] = FAT$C_FIXED;
;  1588          3134  4         RECATTR[FAT$W_RSIZE] = .QUAL[QUAL_BLOC_VALUE];
;  1589          3135  4         RECATTR[FAT$W_MAXREC] = .QUAL[QUAL_BLOC_VALUE];
;  1590          3136  4
;  1591          3137  4         IF .CONTINUE
;  1592          3138  4         AND .RWSV_VOL_NUMBER LSSU 255
;  1593          3139  4         THEN
;  1594          3140  5             BEGIN
;  1595          3141  5             RECATTR[FAT$L_EFBLK] = 1^31 - 1;
;  1596          3142  5             RECATTR[FAT$L_HIBLK] = 1^31 - 1;
;  1597          3143  5             CONTINUE_FID[FID$W_NUM] = FID$C_CONTIN;
;  1598          3144  5             CONTINUE_FID[FID$W_SEQ] = FID$C_CONTIN;
;  1599          3145  5             CONTINUE_FID[FID$W_RVN] = .RWSV_VOL_NUMBER + 1;
;  1600          3146  5             END
;  1601          3147  4         ELSE
;  1602          3148  5             BEGIN
;  1603          3149  5             RECATTR[FAT$L_EFBLK] = ROT (.RWSV_OUT_VBN, 16);
;  1604          3150  5             RECATTR[FAT$L_HIBLK] = ROT (.RWSV_OUT_VBN+.RWSV_ALLOC+.BLOCKS_ALLOC-1, 16);
;  1605          3151  5             CONTINUE_FID[FID$W_NUM] = 0;
;  1606          3152  5             CONTINUE_FID[FID$W_SEQ] = 0;
;  1607          3153  5             CONTINUE_FID[FID$W_RVN] = 0;
;  1608          3154  4             END;
;  1609          3155  4
;  1610          3156  4         ATT_CONTROL1[ATR$W_SIZE] = ATR$S_RECATTR;
;  1611          3157  4         ATT_CONTROL1[ATR$W_TYPE] = ATR$C_RECATTR;
;  1612          3158  4         ATT_CONTROL1[ATR$L_ADDR] = RECATTR;
;  1613          3159  4         ATT_CONTROL2[ATR$W_SIZE] = ATR$S_EXTFID;
;  1614          3160  4         ATT_CONTROL2[ATR$W_TYPE] = ATR$C_EXTFID;
;  1615          3161  4         ATT_CONTROL2[ATR$L_ADDR] = CONTINUE_FID;
;  1616          3162  4         ATTR_END = 0;
;  1617          3163  4
;  1618        P 3164  4         STATUS = $$QIOW (CHAN = .RWSV_CHAN,
;  1619        P 3165  4                          IOSB = IO_STATUS,
;  1620        P 3166  4                          FUNC = IO$_MODIFY,
;  1621        P 3167  4                          P5   = ATT_CONTROL
;  1622          3168  4                          );
;  1623          3169  4         IF .STATUS THEN STATUS = .IO_STATUS[0];
;  1624          3170  4         IF NOT .STATUS
;  1625          3171  4         THEN FILE_ERROR (BACKUP$_CLOSEOUT+STS$K_SEVERE, .FAB, .STATUS);
;  1626          3172  4         STA_DISMOUNT_OUTPUT (.RWSV_VOL_NUMBER, .CONTINUE);
;  1627          3173  4
;  1628          3174  4         IF NOT .QUAL[QUAL_VERI]
;  1629          3175  4         THEN
;  1630          3176  5             BEGIN
;  1631        P 3177  5             $$QIOW (CHAN = .RWSV_CHAN,
;  1632        P 3178  5                     IOSB = IO_STATUS,
;  1633        P 3179  5                     FUNC = IO$_DEACCESS
```

```
;  1634          3180  5                    );
;  1635          3181  5                STA_DISMOUNT (.RWSV_VOL_NUMBER);
;  1636          3182  5                IF .CONTINUE
;  1637          3183  5                THEN
;  1638          3184  6                    BEGIN
;  1639        P 3185  6                    $QIOW (CHAN = .CURRENT_VCB[VCB_CHAN],
;  1640        P 3186  6                           FUNC = IO$_UNLOAD
;  1641          3187  6                           );
;  1642          3188  5                    END;
;  1643          3189  4                END;
;  1644          3190  3            END;
;  1645          3191  2        END;
;  1646          3192  2
;  1647          3193  2 ! Report the number of soft write errors.
;  1648          3194  2 !
;  1649          3195  2
;  1650          3196  2 IF .RWSV_OUT_ERRORS NEQ 0
;  1651          3197  2 THEN FILE_ERROR (BACKUP$_SOFTWERRS, .FAB, .RWSV_OUT_ERRORS);
;  1652          3198  2
;  1653          3199  2 ! Now do the verify pass, if requested.
;  1654          3200  2 !
;  1655          3201  2
;  1656          3202  2 IF .QUAL[QUAL_VERI]
;  1657          3203  2 THEN
;  1658          3204  3     BEGIN
;  1659          3205  3     SIGNAL (BACKUP$_STARTVERIFY);
;  1660          3206  3     COM_FLAGS[COM_VERIFYING] = TRUE;
;  1661          3207  3     IF .RWSV_SEG_NUMBER EQL 0
;  1662          3208  3     THEN INIT_IN_SAVE (TRUE)
;  1663          3209  3     ELSE
;  1664          3210  3         IF .BBLOCK [FAB[FAB$L_DEV], DEV$V_SQD]
;  1665          3211  3         THEN READY_NEXT_VOLUME ();
;  1666          3212  3     RWSV_IN_VBN = .RWSV_IN_VBN_0;
;  1667          3213  3     RWSV_IN_ERRORS = 0;
;  1668          3214  3     RWSV_IN_XORUSE = 0;
;  1669          3215  3     SAVE_VERIFY_REEL ();
;  1670          3216  3     FIN_IN_SAVE (.CONTINUE);
;  1671          3217  3     COM_FLAGS[COM_VERIFYING] = FALSE;
;  1672          3218  2     END;
;  1673          3219  2
;  1674          3220  2 IF .RWSV_VOL_NUMBER GEQU 255
;  1675          3221  2 THEN FILE_ERROR (BACKUP$_MAXVOLS, .FAB);
;  1676          3222  2
;  1677          3223  2 IF .CONTINUE THEN SIGNAL (BACKUP$_RESUME, 1, .RWSV_VOL_NUMBER+1);
;  1678          3224  2
;  1679          3225  1 END;                                  ! End of routine FIN_OUT_SAVE


                                  4C  5A  36  21  00B6B P.AAG:  .ASCII  \!6ZL\                          ;
                                                  00B6F         .BLKB   1
                                    00000004      00B70 P.AAF:  .LONG   4                               ;
                                    00000000'     00B74         .ADDRESS P.AAG                        ;

                                                                .EXTRN  STA_DISMOUNT_OUTPUT
                                                                .EXTRN  STA_DISMOUNT_INIT_IN_SAVE
                                                                .EXTRN  READY_NEXT_VOLUME
```

```
                                                        .EXTRN  SAVE_VERIFY_REEL
                                                        .EXTRN  FIN_IN_SAVE, UNLOAD
                                                        .EXTRN  SYS$CLOSE, SYS$DASSGN
                                                        .EXTRN  SYS$QIOW

                                        0FFC 00000      .ENTRY  FIN_OUT_SAVE, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  2953
                                                                R10,R11
                    5B 00000000G    00  9E 00002       MOVAB   FILE_ERROR, R11                                 ;
                    5A 00000000'    EF  9E 00009       MOVAB   RWSV_VOL_NUMBER, R10                            ;
                    5E        FF54  CE  9E 00010       MOVAB   -172(SP), SP                                    ;
                              5C    AA  95 00015       TSTB    RWSV_OUT_GROUP_COUNT                            ;  3034
                              20    13  00018       BEQL    1$                                              ;
                    50        14    AA  D0 0001A       MOVL    RWSV_XOR_BCB, R0                                ;  3037
                    51        0C    A0  D0 0001E       MOVL    12(R0), XOR_BUFFER                              ;
                              4C    AA  D6 00022       INCL    RWSV_OUT_SEQ                                    ;  3038
                              54    AA  D6 00025       INCL    RWSV_OUT_BLOCK_COUNT                            ;  3039
                 08   A1        4C    AA  D0 00028       MOVL    RWSV_OUT_SEQ, 8(XOR_BUFFER)                     ;  3040
                              50    DD  0002D       PUSHL   R0                                              ;  3041
               F592   CF              01  FB 0002F       CALLS   #1, WRITE_BLOCK                                 ;
                              14    AA  D4 00034       CLRL    RWSV_XOR_BCB                                    ;  3042
                              5C    AA  94 00037       CLRB    RWSV_OUT_GROUP_COUNT                            ;  3043
                    52        FF2C  DA  0F 0003A 1$:   REMQUE  @OUTPUT_WAIT, BCB                               ;  3049
                              14    1D  0003F       BVS     2$                                              ;
                              52    DD  00041       PUSHL   BCB                                             ;  3052
          00000000G   00              01  FB 00043       CALLS   #1, WAIT                                        ;
                              52    DD  0004A       PUSHL   BCB                                             ;  3053
          00000000G   00              01  FB 0004C       CALLS   #1, FREE_BUFFER                                 ;
                              E5    11  00053       BRB     1$                                              ;  3049
                    56        0C    AA  D0 00055 2$:   MOVL    RWSV_SAVE_FAB, FAB                              ;  3059
              26        6F    AA      03  E1 00059       BBC     #3, QUAL+T5, 4$                                 ;  3060
                              6D    AA  95 0005E       TSTB    QUAL+13                                         ;  3063
                              1E    19  00061       BLSS    3$                                              ;
                              56    DD  00063       PUSHL   FAB                                             ;  3066
          00000000G   00              01  FB 00065       CALLS   #1, SYS$CLOSE                                   ;
                    58              50  D0 0006C       MOVL    R0, STATUS                                      ;
                    0F              58  E8 0006F       BLBS    STATUS, 3$                                      ;  3067
                    7E        08    A6  7D 00072       MOVQ    8(FAB), -(SP)                                   ;  3069
                              56    DD  00076       PUSHL   FAB                                             ;  3068
                 00000000G    8F    DD  00078       PUSHL   #BACKUP$_CLOSEOUT+4                             ;
                    6B              04  FB 0007E       CALLS   #4, FILE_ERROR                                  ;
                          01F3  31  00081 3$:      BRW     18$                                             ;  3060
                    59        04    AC  D0 00084 4$:   MOVL    CONTINUE, R9                                    ;  3084
              03        40    A6      05  E0 00088       BBS     #5, 64(FAB), 5$                                 ;  3078
                          00AC  31  0008D       BRW     12$                                             ;
          00000000G   00              00  FB 00090 5$:   CALLS   #0, WRITE_TM                                    ;  3081
                              48    AE  9F 00097       PUSHAB  LABEL_BUFFER                                    ;  3083
          00000000G   00              01  FB 0009A       CALLS   #1, MAKE_HDR1                                   ;
                    0A              59  E9 000A1       BLBC    R9, 6$                                          ;  3084
                 48   AE  31564F45    8F  D0 000A4       MOVL    #827739973, LABEL_BUFFER                        ;  3085
                              08    11  000AC       BRB     7$                                              ;
                 48   AE  31464F45    8F  D0 000AE 6$:   MOVL    #826691397, LABEL_BUFFER                        ;  3086
                 40   AE              06  D0 000B6 7$:   MOVL    #6, DESCRIPTOR                                  ;  3087
                 44   AE        7E    AE  9E 000BA       MOVAB   LABEL_BUFFER+54, DESCRIPTOR+4                   ;  3088
                              54    AA  DD 000BF       PUSHL   RWSV_OUT_BLOCK_COUNT                            ;  3093
                              44    AE  9F 000C2       PUSHAB  DESCRIPTOR                                      ;
                                    7E  D4 000C5       CLRL    -(SP)                                           ;
                          FF2D      CF  9F 000C7       PUSHAB  P.AAF                                           ;
```

```
             00000000G  00                04  FB 000CB        CALLS   #4, SYS$FAO                           ;
                              48          AE  9F 000D2        PUSHAB  LABEL_BUFFER                          ; 3094
             00000000G  00                01  FB 000D5        CALLS   #1, WRITE_LABEL                       ;
                              48          AE  9F 000DC        PUSHAB  LABEL_BUFFER                          ; 3096
             00000000G  00                01  FB 000DF        CALLS   #1, MAKE_HDR2                         ;
                        0A                59  E9 000E6        BLBC    R9, 8$                                ; 3097
                   48   AE 32564F45       8F  D0 000E9        MOVL    #844517189, LABEL_BUFFER              ; 3098
                                          08  11 000F1        BRB     9$                                    ;
                   48   AE 32464F45       8F  D0 000F3 8$:    MOVL    #843468613, LABEL_BUFFER              ; 3099
                              48          AE  9F 000FB 9$:    PUSHAB  LABEL_BUFFER                          ; 3100
             00000000G  00                01  FB 000FE        CALLS   #1, WRITE_LABEL                       ;
             00000000G  00                00  FB 00105        CALLS   #0, WRITE_TM                          ; 3105
             00000000G  00                00  FB 0010C        CALLS   #0, WRITE_TM                          ; 3106
                        7E                02  CE 00113        MNEGL   #2, -(SP)                             ; 3107
             00000000G  00                01  FB 00116        CALLS   #1, SKIP_TM                           ;
                              6D          AA  95 0011D        TSTB    QUAL+13                               ; 3109
                                          17  19 00120        BLSS    11$                                   ;
                        07                59  E9 00122        BLBC    R9, 10$                               ; 3112
             00000000G  00                00  FB 00125        CALLS   #0, UNLOAD                            ; 3113
                              10          AA  DD 0012C 10$:   PUSHL   RWSV_CHAN                             ; 3114
             00000000G  00                01  FB 0012F        CALLS   #1, SYS$DASSGN                        ;
                              10          AA  D4 00136        CLRL    RWSV_CHAN                             ; 3115
                                        013B  31 00139 11$:   BRW     18$                                   ; 3078
                 06D8   CA        06D4    CA  D0 0013C 12$:   MOVL    OUTPUT_MTL, CURRENT_MTL               ; 3125
                        7E                6A  3C 00143        MOVZWL  RWSV_VOL_NUMBER, -(SP)                ; 3126
             00000000G  00                01  FB 00146        CALLS   #1, SWITCH_VOLUME                     ;
                                          5E  DD 0014D        PUSHL   SP                                    ; 3127
                        7E        44      AA  CE 0014F        MNEGL   RWSV_ALLOC, -(SP)                     ;
             00000000G  00                02  FB 00153        CALLS   #2, STA_EXTEND                        ;
                        58                50  D0 0015A        MOVL    R0, STATUS                            ;
                        0D                58  E8 0015D        BLBS    STATUS, 13$                           ; 3128
                              0140        8F  BB 00160        PUSHR   #^M<R6,R8>                            ; 3129
                          00000000G       8F  DD 00164        PUSHL   #BACKUP$_CLOSEOUT+4                   ;
                        6B                03  FB 0016A        CALLS   #3, FILE_ERROR                        ;
                        57        50      AA  D0 0016D 13$:   MOVL    RWSV_OUT_VBN, R7                      ; 3130
                   48   AA        FF      A7  9E 00171        MOVAB   -1(R7), RWSV_EOF                      ;
 20          00         6E                00  2C 00176        MOVC5   #0, (SP), #0, #32, RECATTR            ; 3132
                              20          AE     0017B                                                  ;
                   20   AE                01  90 0017D        MOVB    #1, RECATTR                           ; 3133
                   22   AE        00A8    CA  B0 00181        MOVW    QUAL+72, RECATTR+2                    ; 3134
                   30   AE        00A8    CA  B0 00187        MOVW    QUAL+72, RECATTR+16                   ; 3135
                        26                59  E9 0018D        BLBC    R9, 14$                               ; 3137
                 00FF   8F                6A  B1 00190        CMPW    RWSV_VOL_NUMBER, #255                 ; 3138
                                          1F  1E 00195        BGEQU   14$                                   ;
                   28   AE 7FFFFFFF       8F  D0 00197        MOVL    #2147483647, RECATTR+8                ; 3141
                   24   AE 7FFFFFFF       8F  D0 0019F        MOVL    #2147483647, RECATTR+4                ; 3142
                   18   AE 00070007       8F  D0 001A7        MOVL    #458759, CONTINUE_FID                 ; 3143
       1C   AE          6A                01  A1 001AF        ADDW3   #1, RWSV_VOL_NUMBER, CONTINUE_FID+4   ; 3145
                                          1A  11 001B4        BRB     15$                                   ; 3137
       28   AE          57                10  9C 001B6 14$:   ROTL    #16, R7, RECATTR+8                    ; 3149
            50          57        44      AA  C1 001BB        ADDL3   RWSV_ALLOC, R7, R0                    ; 3150
                        50                6E  C0 001C0        ADDL2   BLOCKS_ALLOC, R0                      ;
                                          50  D7 001C3        DECL    R0                                    ;
       24   AE          50                10  9C 001C5        ROTL    #16, R0, RECATTR+4                    ;
                              18          AE  D4 001CA        CLRL    CONTINUE_FID                          ; 3151
                              1C          AE  B4 001CD        CLRW    CONTINUE_FID+4                        ; 3153
                   04   AE 00040020       8F  D0 001D0 15$:   MOVL    #262176, ATT_CONTROL1                 ; 3156
```

```
                    08   AE        20   AE   9E 001D8          MOVAB   RECATTR, ATT_CONTROL1+4                 ; 3158
                    0C   AE 00270006   8F   D0 001DD          MOVL    #2555910, ATT_CONTROL2                  ; 3159
                    10   AE        18   AE   9E 001E5          MOVAB   CONTINUE_FID, ATT_CONTROL2+4            ; 3161
                                   14   AE   D4 001EA          CLRL    ATTR_END                                ; 3162
                                        7E   D4 001ED          CLRL    -(SP)                                   ; 3168
                                   08   AE   9F 001EF          PUSHAB  ATT_CONTROL                             ;
                                        7E   7C 001F2          CLRQ    -(SP)                                   ;
                                        7E   7C 001F4          CLRQ    -(SP)                                   ;
                                        7E   7C 001F6          CLRQ    -(SP)                                   ;
                                   F8   AD   9F 001F8          PUSHAB  IO_STATUS                               ;
                                        36   DD 001FB          PUSHL   #54                                     ;
                                   10   AA   DD 001FD          PUSHL   RWSV_CHAN                               ;
                                        7E   D4 00200          CLRL    -(SP)                                   ;
              00000000G   00             0C   FB 00202          CALLS   #12, STA_QIOW                           ;
                          58             50   D0 00209          MOVL    R0, STATUS                              ;
                          07             58   E9 0020C          BLBC    STATUS, 16$                             ; 3169
                          58   F8        AD   3C 0020F          MOVZWL  IO_STATUS, STATUS                       ;
                          0D             58   E8 00213          BLBS    STATUS, 17$                             ; 3170
                               0140      8F   BB 00216 16$:     PUSHR   #^M<R6,R8>                              ; 3171
                          00000000G      8F   DD 0021A          PUSHL   #BACKUP$_CLOSEOUT+4                     ;
                          6B             03   FB 00220          CALLS   #3, FILE_ERROR                          ;
                                         59   DD 00223 17$:     PUSHL   R9                                      ; 3172
                          7E             6A   3C 00225          MOVZWL  RWSV_VOL_NUMBER, -(SP)                  ;
              00000000G   00             02   FB 00228          CALLS   #2, STA_DISMOUNT_OUTPUT                 ;
                                   6D   AA   95 0022F          TSTB    QUAL+13                                 ; 3174
                                        43   19 00232          BLSS    18$                                     ; 3180
                                        7E   7C 00234          CLRQ    -(SP)                                   ;
                                        7E   7C 00236          CLRQ    -(SP)                                   ;
                                        7E   7C 00238          CLRQ    -(SP)                                   ;
                                        7E   7C 0023A          CLRQ    -(SP)                                   ;
                                   F8   AD   9F 0023C          PUSHAB  IO_STATUS                               ;
                                        34   DD 0023F          PUSHL   #52                                     ;
                                   10   AA   DD 00241          PUSHL   RWSV_CHAN                               ;
                                        7E   D4 00244          CLRL    -(SP)                                   ;
              00000000G   00             0C   FB 00246          CALLS   #12, STA_QIOW                           ;
                          7E             6A   3C 0024D          MOVZWL  RWSV_VOL_NUMBER, -(SP)                  ; 3181
              00000000G   00             01   FB 00250          CALLS   #1, STA_DISMOUNT                        ;
                          1D             59   E9 00257          BLBC    R9, 18$                                 ; 3182
                                        7E   7C 0025A          CLRQ    -(SP)                                   ; 3187
                                        7E   7C 0025C          CLRQ    -(SP)                                   ;
                                        7E   7C 0025E          CLRQ    -(SP)                                   ;
                                        7E   7C 00260          CLRQ    -(SP)                                   ;
                          7E             01   7D 00262          MOVQ    #1, -(SP)                               ;
                          50        06DC   CA   D0 00265          MOVL    CURRENT_VCB, R0                         ;
                          7E        08   A0   3C 0026A          MOVZWL  8(R0), -(SP)                            ;
                                        7E   D4 0026E          CLRL    -(SP)                                   ;
              00000000G   00             0C   FB 00270          CALLS   #12, SYS$QIOW                           ;
                          50        58   AA   3C 00277 18$:     MOVZWL  RWSV_OUT_ERRORS, R0                     ; 3196
                                        0D   13 0027B          BEQL    19$                                     ;
                                        50   DD 0027D          PUSHL   R0                                      ; 3197
                                        56   DD 0027F          PUSHL   FAB                                     ;
                               00000000G   8F   DD 00281          PUSHL   #BACKUP$_SOFTWERRS                      ;
                          6B             03   FB 00287          CALLS   #3, FILE_ERROR                          ;
                                   6D   AA   95 0028A 19$:     TSTB    QUAL+13                                 ; 3202
                                        4C   18 0028D          BGEQ    22$                                     ;
                               00000000G   8F   DD 0028F          PUSHL   #BACKUP$_STARTVERIFY                    ; 3205
              00000000G   00             01   FB 00295          CALLS   #1, LIB$SIGNAL                          ;
```

```
                    00DA    CA                08  88 0029C         BISB2   #8, COM_FLAGS                    ; 3206
                                      02      AA  B5 002A1         TSTW    RWSV_SEG_NUMBER                  ; 3207
                                              0B  12 00???         BNEQ    20$                              ;
                                              01  DD 002A6         PUSHL   #1                               ; 3208
              00000000G     00                01  FB 002A8         CALLS   #1, INIT_IN_SAVE                 ;
                                              0C  11 002AF         BRB     21$                              ;
          07        40      A6                05  E1 002B1 20$:    BBC     #5, 64(FAB), 21$                 ; 3210
              00000000G     00                00  FB 002B6         CALLS   #0, READY_NEXT_VOLUME            ; 3211
                    3C      AA        40      AA  D0 002BD 21$:    MOVL    RWSV_IN_VBN_0, RWSV_IN_VBN       ; 3212
                                      30      AA  D4 002C2         CLRL    RWSV_IN_ERRORS                   ; 3213
              00000000G     00                00  FB 002C5         CALLS   #0, SAVE_VERIFY_REEL             ; 3215
                                      04      AC  DD 002CC         PUSHL   CONTINUE                         ; 3216
              00000000G     00                01  FB 002CF         CALLS   #1, FIN_IN_SAVE                  ;
                    00DA    CA                08  8A 002D6         BICB2   #8, COM_FLAGS                    ; 3217
                    00FF    8F                6A  B1 002DB 22$:    CMPW    RWSV_VOL_NUMBER, #255            ; 3220
                                              0B  1F 002E0         BLSSU   23$                              ;
                                              56  DD 002E2         PUSHL   FAB                              ; 3221
                               00000000G      8F  DD 002E4         PUSHL   #BACKUP$_MAXVOLS                 ;
                            6B                02  FB 002EA         CALLS   #2, FILE_ERROR                   ;
                            14        04      AC  E9 002ED 23$:    BLBC    CONTINUE, 24$                    ; 3223
                            7E                6A  3C 002F1         MOVZWL  RWSV_VOL_NUMBER, -(SP)           ;
                                              6E  D6 002F4         INCL    (SP)                             ;
                                              01  DD 002F6         PUSHL   #1                               ;
                               00000000G      8F  DD 002F8         PUSHL   #BACKUP$_RESUME                  ;
              00000000G     00                03  FB 002FE         CALLS   #3, LIB$SIGNAL                   ;
                                                  04 00305 24$:    RET                                      ; 3225

; Routine Size:  774 bytes,    Routine Base:  CODE + 0B78


; 1680          3226  1
; 1681          3227  1 END
; 1682          3228  0 ELUDOM



                                                                   .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY
;
;       Name                        Bytes                    Attributes
;
;  COMMON                            2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                              3710  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       145         0         1000        00:01.8
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         3614 code + 2220 data bytes
; Run Time:       01:10.1
; Elapsed Time:   03:41.0
; Lines/CPU Min:   2764
; Lexemes/CPU-Min: 27888
; Memory Used:  498 pages
; Compilation Complete
```

BADBLOCKS
LIS

BAD

STABAD
MAP

BADDATA
LIS

ANALYZBAD
MAP

ANALYZFMD
CLD

BADDEF
SDL

WRITESAVE
LIS

BADINIT
LIS

ANALYZCMD
LIS