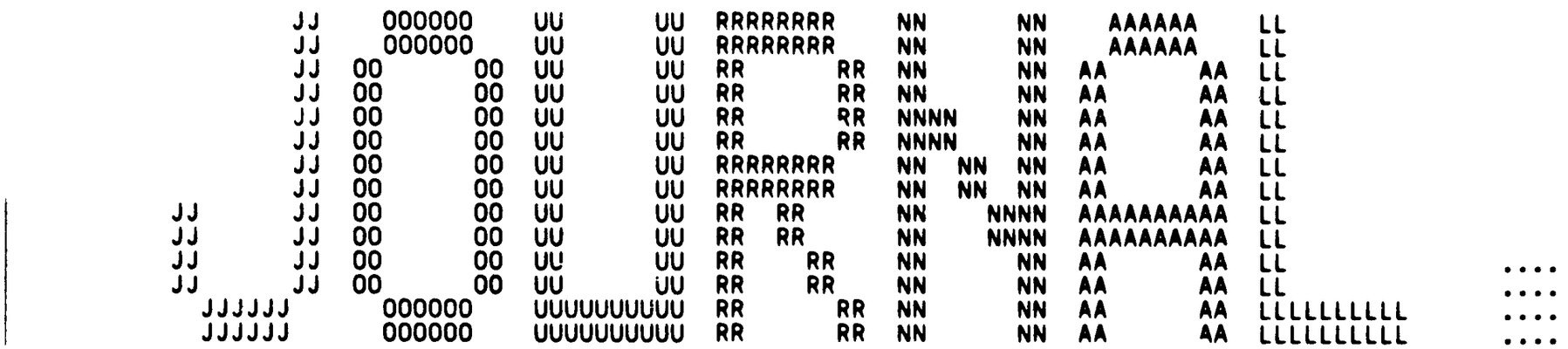
```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB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


JJ 000000 UU UU RRRRRRRR NN NN AAAAAA LL
JJ 000000 UU UU RRRRRRRR NN NN AAAAAA LL
JJ OO OO UU UU RR RR NN NN AA AA LL
JJ OO OO UU UU RR RR NN NN AA AA LL
JJ OO OO UU UU RR RR NNNN NN AA AA LL
JJ OO OO UU UU RR RR NNNN NN AA AA LL
JJ OO OO UU UU RRRRRRRR NN NN NN AA AA LL
JJ OO OO UU UU RRRRRRRR NN NN NN AA AA LL
JJ JJ OO OO UU UU RR RR NN NNNN AAAAAAAAAA LL
JJ JJ OO OO UU UU RR RR NN NNNN AAAAAAAAAA LL
JJ JJ OO OO UU UU RR RR NN NN AA AA LL
JJ JJ OO OO UU UU RR RR NN NN AA AA LL
JJJJJJ 000000 UUUUUUUUUU RR RR NN NN AA AA LLLLLLLLLL
JJJJJJ 000000 UUUUUUUUUU RR RR NN NN AA AA LLLLLLLLLL
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1        0001  0 MODULE JOURNAL  (%TITLE 'Journal file manager'
;     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 !       This module contains the routines that manage the journal file.
;    37        0037  1 !
;    38        0038  1 ! ENVIRONMENT:
;    39        0039  1 !       VAX/VMS user mode.
;    40        0040  1 !--
;    41        0041  1 !
;    42        0042  1 ! AUTHOR: M. Jack, CREATION DATE: 8-Apr-1981
;    43        0043  1 !
;    44        0044  1 ! MODIFIED BY:
;    45        0045  1 !
;    46        0046  1 !       V03-005 LY0510          Larry Yetto             19-JU -1984 08:52
;    47        0047  1 !               Remove "_" from default journal file name
;    48        0048  1 !
;    49        0049  1 !       V03-004 LY0498          Larry Yetto             25-JUN-1984 12:30
;    50        0050  1 !               Save extention size of journal file after opening
;    51        0051  1 !               and use this size for future extends instead of 1.
;    52        0052  1 !
;    53        0053  1 !       V03-003 LY0488          Larry Yetto             21-MAY-1984 16:08
;    54        0054  1 !               Add ASSUME statument to make sre that the directory
;    55        0055  1 !               record size is large enough for long directory names.
;    56        0056  1 !               Fix problem where the journal file ends
;    57        0057  1 !               up with a block of zeros in it if
```

```
;    58          0058  1 !                 a save set ends exactly on a block boundary
;    59          0059  1 !
;    60          0060  1 !       V03-002 LY0463          Larry Yetto             1-FEB-1984 10:36
;    61          0061  1 !               Add support for journal file structure level 2.  This new
;    62          0062  1 !               format add a flag longword to each file entry to enable us
;    63          0063  1 !               to flag the fact that only a file header was saved for a
;    64          0064  1 !               particular file.
;    65          0065  1 !
;    66          0066  1 !       V03-001 ACG0313         Andrew C. Goldstein,    12-Feb-1983  16:17
;    67          0067  1 !               Add routine subtitles
;    68          0068  1 !
;    69          0069  1 !       V02-003 MLJ0054         Martin L. Jack, 22-Nov-1981  22:11
;    70          0070  1 !               Integrate GET_VM and FREE_VM jacket routines.
;    71          0071  1 !
;    72          0072  1 !       V02-002 MLJ0036         Martin L. Jack, 28-Aug-1981  17:53
;    73          0073  1 !               Replace OWN storage with common variables.  Remove explicit
;    74          0074  1 !               routine parameters.
;    75          0075  1 !
;    76          0076  1 !       V02-001 MLJ0025         Martin L. Jack, 8-May-1981  11:48
;    77          0077  1 !               Reorganize OWN storage.  Dynamically allocate directory string
;    78          0078  1 !               buffer.  Don't bother to free dynamic memory.
;    79          0079  1 !
;    80          0080  1 !**
```

```
;    82          0081  1 REQUIRE 'SRC$:COMMON';
;    83          1187  1 LIBRARY 'SYS$LIBRARY:LIB';
;    84          1188  1 REQUIRE 'LIB$:BACKDEF';
;    85          1638  1
;    86          1639  1
;    87          1640  1 FORWARD ROUTINE
;    88          1641  1         OPEN_JOURNAL:   NOVALUE,        ! Open or create journal file
;    89          1642  1         CLOSE_JOURNAL:  NOVALUE,        ! Close journal file
;    90          1643  1         WRITE_BLOCK:    NOVALUE,        ! Write block to journal file
;    91          1644  1         WRITE_BYTE:     NOVALUE,        ! Write byte to journal file
;    92          1645  1         WRITE_RECORD:   NOVALUE,        ! Write record to journal file
;    93          1646  1         WRITE_JOUR_SSNAME:
;    94          1647  1                         NOVALUE,        ! Write SSNAME record
;    95          1648  1         WRITE_JOUR_VOLUME:
;    96          1649  1                         NOVALUE,        ! Write VOLUME record
;    97          1650  1         WRITE_JOUR_FILE:NOVALUE,        ! Write DIRECTORY and FILE record
;    98          1651  1         WRITE_JOUR_DIRECTORY:
;    99          1652  1                         NOVALUE;        ! Write final DIRECTORY record
;   100          1653  1
;   101          1654  1
;   102          1655  1 EXTERNAL ROUTINE
;   103          1656  1         GET_VM,                         ! Allocate virtual memory
;   104          1657  1         GET_ZERO_VM,                    ! Allocate and clear virtual memory
;   105          1658  1         FILE_ERROR:     NOVALUE;        ! Signal file-related error
;   106          1659  1
;   107          1660  1
;   108          1661  1 EXTERNAL LITERAL
;   109          1662  1         BACKUP$_OPENIN,
;   110          1663  1         BACKUP$_OPENOUT,
;   111          1664  1         BACKUP$_READERR,
;   112          1665  1         BACKUP$_WRITEERR,
;   113          1666  1         BACKUP$_READATTR,
;   114          1667  1         BACKUP$_WRITEATTR,
;   115          1668  1         BACKUP$_NOTJOURNAL,
;   116          1669  1         BACKUP$_JOURNAL,
;   117          1670  1         BACKUP$_INVBJLTYP,
;   118          1671  1         BACKUP$_INVBJLSIZ,
;   119          1672  1         BACKUP$_INVBJLSTR,
;   120          1673  1         BACKUP$_INVBJLEOF;
;   121          1674  1
;   122          1675  1
;   123          1676  1 G$DEFINE();                        ! Define global common area
;   124          1677  1
;   125          1678  1
;   126          1679  1 BUILTIN
;   127          1680  1         ROT;
;   128          1681  1
;   129          1682  1
;   130          1683  1 BIND
;   131          1684  1         STRUC_RECORD = UPLIT BYTE (
;   132          1685  1             WORD(BJL$K_LEVEL2));
```

```
;  134          1686  1 %SBTTL 'OPEN_JOURNAL - open journal file'
;  135          1687  1 GLOBAL ROUTINE OPEN_JOURNAL(OUTPUT,REVERSE): NOVALUE=
;  136          1688  1
;  137          1689  1 !++
;  138          1690  1 !
;  139          1691  1 ! FUNCTIONAL DESCRIPTION:
;  140          1692  1 !       This routine opens (or creates) the journal file.
;  141          1693  1 !
;  142          1694  1 ! INPUT PARAMETERS:
;  143          1695  1 !       OUTPUT          - True if the journal file is being written, false if
;  144          1696  1 !                         being read.
;  145          1697  1 !       REVERSE         - True if the journal file is being read backward,
;  146          1698  1 !                         false if being read forward.  Required only if
;  147          1699  1 !                         OUTPUT is false.
;  148          1700  1 !
;  149          1701  1 ! IMPLICIT INPUTS:
;  150          1702  1 !       QUAL[QUAL_JOUR_FC] - Pointer to FAB for journal file.
;  151          1703  1 !
;  152          1704  1 ! OUTPUT PARAMETERS:
;  153          1705  1 !       NONE
;  154          1706  1 !
;  155          1707  1 ! IMPLICIT OUTPUTS:
;  156          1708  1 !       NONE
;  157          1709  1 !
;  158          1710  1 ! ROUTINE VALUE:
;  159          1711  1 !       NONE
;  160          1712  1 !
;  161          1713  1 ! SIDE EFFECTS:
;  162          1714  1 !       NONE
;  163          1715  1 !
;  164          1716  1 !--
;  165          1717  1
;  166          1718  2 BEGIN
;  167          1719  2 LOCAL
;  168          1720  2         FAB:            REF BBLOCK,     ! Pointer to FAB
;  169          1721  2         STATUS,                         ! Status variable
;  170          1722  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;  171          1723  2         ATR_DESC:       BBLOCK[20],     ! ACP attributes list
;  172          1724  2         RECATTR:        BBLOCK[FAT$C_LENGTH],   ! Record attributes area
;  173          1725  2         STATBLK:        BBLOCK[8];      ! Statistics block
;  174          1726  2
;  175          1727  2
;  176          1728  2 ! Initialize the FAB.
;  177          1729  2 !
;  178          1730  2 FAB = .QUAL[QUAL_JOUR_FC];
;  179          1731  2 FAB[FAB$L_DNA] = UPLIT BYTE('.BJL');
;  180          1732  2 FAB[FAB$B_DNS] = %CHARCOUNT('.BJL');
;  181          1733  2 IF .FAB[FAB$B_FNS] EQL 0
;  182          1734  2 THEN
;  183          1735  3     BEGIN
;  184          1736  3     FAB[FAB$L_FNA] = UPLIT BYTE('BACKUP');
;  185          1737  3     FAB[FAB$B_FNS] = %CHARCOUNT('BACKUP');
;  186          1738  2     END;
;  187          1739  2 FAB[FAB$V_UFO] = TRUE;
;  188          1740  2 FAB[FAB$B_RFM] = FAB$C_FIX;
;  189          1741  2 FAB[FAB$W_MRS] = 1;
;  190          1742  2
```

```
;  191          1743  2 ! Issue the OPEN or CREATE service as appropriate.
;  192          1744  2 !
;  193          1745  2 IF .OUTPUT
;  194          1746  2 THEN
;  195          1747  3     BEGIN
;  196          1748  3     FAB[FAB$W_DEQ] = 1 ;
;  197          1749  3     FAB[FAB$V_CIF] = TRUE;
;  198          1750  4     IF NOT $CREATE(FAB=.FAB)
;  199          1751  3     THEN
;  200          1752  3         FILE_ERROR(
;  201          1753  3             BACKUP$_OPENOUT + STS$K_SEVERE,
;  202          1754  3             .FAB,
;  203          1755  3             .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
;  204          1756  3     END
;  205          1757  2 ELSE
;  206          1758  3     BEGIN
;  207          1759  4     IF NOT $OPEN(FAB=.FAB)
;  208          1760  3     THEN
;  209          1761  3         FILE_ERROR(
;  210          1762  3             BACKUP$_OPENIN + STS$K_SEVERE,
;  211          1763  3             .FAB,
;  212          1764  3             .FAB[FAB$L_STS], .FAB[FAB$L_STV]);
;  213          1765  2     END;
;  214          1766  2
;  215          1767  2
;  216          1768  2 ! Get the record attributes and the file size.
;  217          1769  2 !
;  218          1770  2 ATR_DESC[0,0,16,0] = ATR$S_RECATTR;
;  219          1771  2 ATR_DESC[2,0,16,0] = ATR$C_RECATTR;
;  220          1772  2 ATR_DESC[4,0,32,0] = RECATTR;
;  221          1773  2 ATR_DESC[8,0,16,0] = 8;          ! only need 8 bytes of statistics block
;  222          1774  2 ATR_DESC[10,0,16,0] = ATR$C_STATBLK;
;  223          1775  2 ATR_DESC[12,0,32,0] = STATBLK;
;  224          1776  2 ATR_DESC[16,0,32,0] = 0;
;  225        P 1777  2 STATUS = $QIOW(
;  226        P 1778  2     FUNC=IO$_ACCESS,
;  227        P 1779  2     CHAN=.FAB[FAB$L_STV],
;  228        P 1780  2     IOSB=IOSB,
;  229          1781  2     P5=ATR_DESC);
;  230          1782  2 IF .STATUS THEN STATUS = .IOSB[0];
;  231          1783  2 IF NOT .STATUS
;  232          1784  2 THEN
;  233          1785  2     FILE_ERROR(
;  234          1786  2         BACKUP$_READATTR,
;  235          1787  2         .FAB,
;  236          1788  2         .STATUS);
;  237          1789  2
;  238          1790  2
;  239          1791  2 ! Check the record attributes to ensure that this is, in fact, a BACKUP
;  240          1792  2 ! journal file.  If the file is zero length, allow any attributes and then
;  241          1793  2 ! force them correct.  For example, this allows the CREATE command to be
;  242          1794  2 ! used to generate a fresh journal.
;  243          1795  2 !
;  244          1796  2 IF .STATBLK[SBK$L_FILESIZE] NEQ 0
;  245          1797  2 THEN
;  246          1798  2     IF
;  247          1799  2         .RECATTR[FAT$B_RTYPE] NEQ FAT$C_FIXED OR
```

```
;  248          1800  2          .RECATTR[FAT$B_RATTRIB] NEQ 0 OR
;  249          1801  2          .RECATTR[FAT$W_RSIZE] NEQ 1
;  250          1802  2      THEN
;  251          1803  2          FILE_ERROR(
;  252          1804  2              BACKUP$_NOTJOURNAL,
;  253          1805  2              .FAB);
;  254          1806  2
;  255          1807  2
;  256          1808  2 ! Save the critical attributes for further processing.
;  257          1809  2 !
;  258          1810  2 JOUR_HIBLK = ROT(.STATBLK[SBK$L_FILESIZE], 16);
;  259          1811  2 JOUR_EFBLK = ROT(.RECATTR[FAT$L_EFBLK], 16);
;  260          1812  2 JOUR_FFBYTE = .RECATTR[FAT$W_FFBYTE];
;  261          1813  2 IF .RECATTR[FAT$W_DEFEXT] NEQ 0
;  262          1814  2     THEN JOUR_EXSZ  = .RECATTR[FAT$W_DEFEXT]
;  263          1815  2     ELSE JOUR_EXSZ  = 1 ;
;  264          1816  2
;  265          1817  2 ! Initialize other context.
;  266          1818  2 !
;  267          1819  2 JOUR_COUNT = 0;
;  268          1820  2 JOUR_BUFFER = GET_VM(512);
;  269          1821  2 JOUR_DIR = GET_ZERO_VM(BJL$C_DIR_LEN+1);
;  270          1822  2
;  271          1823  2
;  272          1824  2 ! Initialize for input or output processing, as appropriate.
;  273          1825  2 !
;  274          1826  2 IF .OUTPUT
;  275          1827  2 THEN
;  276          1828  3     BEGIN
;  277          1829  3
;  278          1830  3     ! Read back last block if it exists and is not full.
;  279          1831  3     !
;  280          1832  3     IF .JOUR_EFBLK LEQU .JOUR_HIBLK AND .JOUR_FFBYTE NEQ 512
;  281          1833  3     THEN
;  282          1834  4         BEGIN
;  283        P 1835  4         STATUS = $QIOW(
;  284        P 1836  4             FUNC=IO$_READVBLK,
;  285        P 1837  4             CHAN=.FAB[FAB$L_STV],
;  286        P 1838  4             IOSB=IOSB,
;  287        P 1839  4             P1=.JOUR_BUFFER,
;  288        P 1840  4             P2=512,
;  289          1841  4             P3=.JOUR_EFBLK);
;  290          1842  4         IF .STATUS THEN STATUS = .IOSB[0];
;  291          1843  4         IF NOT .STATUS
;  292          1844  4         THEN
;  293          1845  4             FILE_ERROR(
;  294          1846  4                 BACKUP$_READERR + STS$K_SEVERE,
;  295          1847  4                 .FAB,
;  296          1848  4                 .STATUS);
;  297          1849  4         END
;  298          1850  3     ELSE
;  299          1851  4         BEGIN
;  300          1852  4         CH$FILL(0, 512, .JOUR_BUFFER);
;  301          1853  4         JOUR_FFBYTE = 0 ;
;  302          1854  3         END ;
;  303          1855  3
;  304          1856  3
```

```
;  305          1857  3      ! Log the file name if requested.
;  306          1858  3      !
;  307          1859  3      IF .QUAL[QUAL_LOG]
;  308          1860  3      THEN
;  309          1861  3          FILE_ERROR(BACKUP$_JOURNAL, .FAB);
;  310          1862  3
;  311          1863  3
;  312          1864  3      ! Write structure-level record.
;  313          1865  3      !
;  314          1866  3      JOUR_COUNT = 0;                         ! Initialize XOR byte count
;  315          1867  3      WRITE_RECORD(
;  316          1868  3          BJL$K_STRUCLEV,
;  317          1869  3          BJL$C_STRUC_LEN,
;  318          1870  3          STRUC_RECORD);
;  319          1871  3      END
;  320          1872  2  ELSE
;  321          1873  3      BEGIN
;  322          1874  3
;  323          1875  3      ! Establish pointer to current input byte.  The pointer always points to
;  324          1876  3      ! the byte that would have been returned on a previous call to READ_BYTE
;  325          1877  3      ! in the established reading direction.
;  326          1878  3      !
;  327          1879  3      JOUR_REVERSE = .REVERSE;
;  328          1880  3      IF .REVERSE
;  329          1881  3      THEN
;  330          1882  4          BEGIN
;  331          1883  4          JOUR_INBLK = .JOUR_EFBLK;
;  332          1884  4          JOUR_INBYTE = .JOUR_FFBYTE;
;  333          1885  4          END
;  334          1886  3      ELSE
;  335          1887  4          BEGIN
;  336          1888  4          JOUR_INBLK = 0;
;  337          1889  4          JOUR_INBYTE = 511;
;  338          1890  3          END;
;  339          1891  3
;  340          1892  3
;  341          1893  3      ! Read first block if it exists.
;  342          1894  3      !
;  343          1895  3      IF .JOUR_INBLK NEQ 0 AND .JOUR_INBLK LEQU .JOUR_HIBLK
;  344          1896  3      THEN
;  345          1897  4          BEGIN
;  346        P 1898  4          STATUS = $QIOW(
;  347        P 1899  4              FUNC=IO$_READVBLK,
;  348        P 1900  4              CHAN=.FAB[FAB$L_STV],
;  349        P 1901  4              IOSB=IOSB,
;  350        P 1902  4              P1=.JOUR_BUFFER,
;  351        P 1903  4              P2=512,
;  352          1904  4              P3=.JOUR_INBLK);
;  353          1905  4          IF .STATUS THEN STATUS = .IOSB[0];
;  354          1906  4          IF NOT .STATUS
;  355          1907  4          THEN
;  356          1908  4              FILE_ERROR(
;  357          1909  4                  BACKUP$_READERR + STS$K_SEVERE,
;  358          1910  4                  .FAB,
;  359          1911  4                  .STATUS);
;  360          1912  3          END;
;  361          1913  2      END;
```

```
;  362          1914  1 END;

                        .TITLE  JOURNAL Journal file manager
                        .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
```

```
                                                  .PSECT   CODE,NOWRT,2

                         0102  00000 P.AAA:       .WORD    258                                          ;
             4C  4A  42   2E  00002 P.AAB:       .ASCII   \.BJL\                                       ;
     50  55  4B  43  41   42  00006 P.AAC:       .ASCII   \BACKUP\                                     ;

                                   STRUC_RECORD=          P.AAA
                                                  .EXTRN   GET_VM, GET_ZERO_VM
                                                  .EXTRN   FILE_ERROR, BACKUP$_OPENIN
                                                  .EXTRN   BACKUP$_OPENOUT
                                                  .EXTRN   BACKUP$_READERR
                                                  .EXTRN   BACKUP$_WRITEERR
                                                  .EXTRN   BACKUP$_READATTR
                                                  .EXTRN   BACKUP$_WRITEATTR
                                                  .EXTRN   BACKUP$_NOTJOURNAL
                                                  .EXTRN   BACKUP$_JOURNAL
                                                  .EXTRN   BACKUP$_INVBJLTYP
                                                  .EXTRN   BACKUP$_INVBJLSIZ
                                                  .EXTRN   BACKUP$_INVBJLSTR
                                                  .EXTRN   BACKUP$_INVBJLEOF
                                                  .EXTRN   SYS$CREATE, SYS$OPEN
                                                  .EXTRN   SYS$QIOW

                          0FFC  00000             .ENTRY   OPEN_JOURNAL, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1687
                                                           R10,R11
         5B 00000000G  8F   D0  00002             MOVL     #BACKUP$_READERR+4, R11                      ;
         5A 00000000G  00   9E  00009             MOVAB    SYS$QIOW, R10                                ;
         59 00000000G  00   9E  00010             MOVAB    FILE_ERROR, R9                               ;
         58 00000000'  EF   9E  00017             MOVAB    JOUR_EFBLK, R8                               ;
         5E        BC  AE   9E  0001E             MOVAB    -68(SP), SP                                  ;
         56      FDDC  C8   D0  00022             MOVL     QUAL+52, FAB                                 ; 1730
  30     A6        CC  AF   9E  00027             MOVAB    P.AAB, 48(FAB)                               ; 1731
  35     A6            04   90  0002C             MOVB     #4, 53(FAB)                                  ; 1732
                   34  A6   95  00030             TSTB     52(FAB)                                      ; 1733
                       09   12  00033             BNEQ     1$                                           ;
  2C     A6        C2  AF   9E  00035             MOVAB    P.AAC, 44(FAB)                               ; 1736
  34     A6            06   90  0003A             MOVB     #6, 52(FAB)                                  ; 1737
  06     A6            02   88  0003E 1$:         BISB2    #2, 6(FAB)                                   ; 1739
  1F     A6            01   90  00042             MOVB     #1, 31(FAB)                                  ; 1740
  36     A6            01   B0  00046             MOVW     #1, 54(FAB)                                  ; 1741
         22        04  AC   E9  0004A             BLBC     OUTPUT, 2$                                   ; 1745
  14     A6            01   B0  0004E             MOVW     #1, 20(FAB)                                  ; 1748
  07     A6            02   88  00052             BISB2    #2, 7(FAB)                                   ; 1749
                       56   DD  00056             PUSHL    FAB                                          ; 1750
00000000G  00          01   FB  00058             CALLS    #1, SYS$CREATE                               ;
           29          50   E8  0005F             BLBS     R0, 4$                                       ;
           7E      08  A6   7D  00062             MOVQ     8(FAB), -(SP)                                ; 1755
                       56   DD  00066             PUSHL    FAB                                          ; 1754
            00000000G  8F   DD  00068             PUSHL    #BACKUP$_OPENOUT+4                           ; 1753
                       18   11  0006E             BRB      3$                                           ;
                       56   DD  00070 2$:         PUSHL    FAB                                          ; 1759
00000000G  00          01   FB  00072             CALLS    #1, SYS$OPEN                                 ;
           0F          50   E8  00079             BLBS     R0, 4$                                       ;
           7E      08  A6   7D  0007C             MOVQ     8(FAB), -(SP)                                ; 1764
                       56   DD  00080             PUSHL    FAB                                          ; 1763
            00000000G  8F   DD  00082             PUSHL    #BACKUP$_OPENIN+4                            ; 1762
```

```
                    69              04  FB 00088 3$:    CALLS   #4, FILE_ERROR                          ;
                28  AE 00040020     8F  D0 0008B 4$:    MOVL    #262176, ATR_DESC                       ; 1770
                2C  AE       08     AE  9E 00093        MOVAB   RECATTR, ATR_DESC+4                     ; 1772
                30  AE 00090008     8F  D0 00098        MOVL    #589832, ATR_DESC+8                     ; 1773
                34  AE             6E  9E 000A0        MOVAB   STATBLK, ATR_DESC+12                    ; 1775
                            38      AE  D4 000A4        CLRL    ATR_DESC+16                             ; 1776
                                    7E  D4 000A7        CLRL    -(SP)                                   ; 1781
                            2C      AE  9F 000A9        PUSHAB  ATR_DESC                                ;
                                    7E  7C 000AC        CLRQ    -(SP)                                   ;
                                    7E  7C 000AE        CLRQ    -(SP)                                   ;
                                    7E  7C 000B0        CLRQ    -(SP)                                   ;
                            5C      AE  9F 000B2        PUSHAB  IOSB                                    ;
                                    32  DD 000B5        PUSHL   #50                                     ;
                            0C      A6  DD 000B7        PUSHL   12(FAB)                                 ;
                                    7E  D4 000BA        CLRL    -(SP)                                   ;
                    6A              0C  FB 000BC        CALLS   #12, SYS$QIOW                           ;
                    57              50  D0 000BF        MOVL    R0, STATUS                              ;
                    07              57  E9 000C2        BLBC    STATUS, 5$                              ; 1782
                    57      3C      AE  3C 000C5        MOVZWL  IOSB, STATUS                            ;
                    0C              57  E8 000C9        BLBS    STATUS, 6$                              ; 1783
                    7E              56  7D 000CC 5$:    MOVQ    FAB, -(SP)                              ; 1787
                        00000000G   8F  DD 000CF        PUSHL   #BACKUP$_READATTR                       ; 1785
                    69              03  FB 000D5        CALLS   #3, FILE_ERROR                          ;
                            04      AE  D5 000D8 6$:    TSTL    STATBLK+4                               ; 1796
                                    1C  13 000DB        BEQL    8$                                      ;
                    01      08      AE  91 000DD        CMPB    RECATTR, #1                             ; 1799
                                    0B  12 000E1        BNEQ    7$                                      ;
                            09      AE  95 000E3        TSTB    RECATTR+1                               ; 1800
                                    06  12 000E6        BNEQ    7$                                      ;
                    01      0A      AE  B1 000E8        CMPW    RECATTR+2, #1                           ; 1801
                                    0B  13 000EC        BEQL    8$                                      ;
                                    56  DD 000EE 7$:    PUSHL   FAB                                     ; 1805
                        00000000G   8F  DD 000F0        PUSHL   #BACKUP$_NOTJOURNAL                     ; 1803
                    69              02  FB 000F6        CALLS   #2, FILE_ERROR                          ;
FC  A8          04  AE              10  9C 000F9 8$:    ROTL    #16, STATBLK+4, JOUR_HIBLK              ; 1810
    68          10  AE              10  9C 000FF        ROTL    #16, RECATTR+8, JOUR_EFBLK              ; 1811
                08  A8      14      AE  B0 00104        MOVW    RECATTR+12, JOUR_FFBYTE                 ; 1812
                            1A      AE  B5 00109        TSTW    RECATTR+18                              ; 1813
                                    07  13 0010C        BEQL    9$                                      ;
                10  A8          1A  AE  B0 0010E        MOVW    RECATTR+18, JOUR_EXSZ                   ; 1814
                                    04  11 00113        BRB     10$                                     ;
                10  A8              01  B0 00115 9$:    MOVW    #1, JOUR_EXSZ                           ; 1815
                            0E      A8  94 00119 10$:    CLRB    JOUR_COUNT                              ; 1819
                    7E      0200    8F  3C 0011C        MOVZWL  #512, -(SP)                             ; 1820
            00000000G       00      01  FB 00121        CALLS   #1, GET_VM                              ;
                    F4      A8      50  D0 00128        MOVL    R0, JOUR_BUFFER                         ;
                    7E      0100    8F  3C 0012C        MOVZWL  #256, -(SP)                             ; 1821
            00000000G       00      01  FB 00131        CALLS   #1, GET_ZERO_VM                         ;
                    F8      A8      50  D0 00138        MOVL    R0, JOUR_DIR                            ;
                    70          04  AC  E9 0013C        BLBC    OUTPUT, T5$                             ; 1826
                    FC      A8      68  D1 00140        CMPL    JOUR_EFBLK, JOUR_HIBLK                  ; 1832
                                    3C  1A 00144        BGTRU   12$                                     ;
                0200    8F      08  A8  B1 00146        CMPW    JOUR_FFBYTE, #512                       ;
                                    34  13 0014C        BEQL    12$                                     ;
                                    7E  7C 0014E        CLRQ    -(SP)                                   ; 1841
                                    7E  D4 00150        CLRL    -(SP)                                   ;
                                    68  DD 00152        PUSHL   JOUR_EFBLK                              ;
```

```
                                7E      0200    8F  3C 00154          MOVZWL  #512, -(SP)                                             ;
                                  F4        A8  DD 00159          PUSHL   JOUR_BUFFER                                             ;
                                            7E  7C 0015C          CLRQ    -(SP)                                                   ;
                                  5C        AE  9F 0015E          PUSHAB  IOSB                                                    ;
                                            31  DD 00161          PUSHL   #49                                                     ;
                                  0C        A6  DD 00163          PUSHL   12(FAB)                                                 ;
                                            7E  D4 00166          CLRL    -(SP)                                                   ;
                                6A          0C  FB 00168          CALLS   #12, SYS$QIOW                                           ;
                                57          50  D0 0016B          MOVL    R0, STATUS                                              ;
                                07          57  E9 0016E          BLBC    STATUS, 11$                                             ; 1842
                                57        3C    AE 00171          MOVZWL  IOSB, STATUS                                            ;
                                16          57  E8 00175          BLBS    STATUS, 13$                                             ; 1843
                                7E          56  7D 00178 11$:     MOVQ    FAB, -(SP)                                              ; 1847
                                            5B  DD 0017B          PUSHL   R11                                                     ; 1846
                                69          03  FB 0017D          CALLS   #3, FILE_ERROR                                          ;
                                            0C  11 00180          BRB     13$                                                     ; 1832
0200    8F          00          6E          00  2C 00182 12$:     MOVC5   #0, (SP), #0, #512, @JOUR_BUFFER                        ; 1852
                                  F4        B8    00189                                                                           ;
                                  08        A8  B4 0018B          CLRW    JOUR_FFBYTE                                             ; 1853
                    0B    FDB3    C8          01  E1 0018E 13$:     BBC     #1, QUAL+11, 14$                                        ; 1859
                                            56  DD 00194          PUSHL   FAB                                                     ; 1861
                                00000000G   8F  DD 00196          PUSHL   #BACKUP$_JOURNAL                                        ;
                                69          02  FB 0019C          CALLS   #2, FILE_ERROR                                          ;
                                  0E        A8  94 0019F 14$:     CLRB    JOUR_COUNT                                              ; 1866
                                FE4E        CF  9F 001A2          PUSHAB  STRUC_RECORD                                            ; 1867
                                            02  DD 001A6          PUSHL   #2                                                      ;
                                            7E  D4 001A8          CLRL    -(SP)                                                   ;
                          0000V     CF          03  FB 001AA          CALLS   #3, WRITE_RECORD                                        ;
                                                04  001AF          RET                                                             ; 1826
                            0F    A8      08    AC  90 001B0 15$:     MOVB    REVERSE, JOUR_REVERSE                                   ; 1879
                            0B    A8      08    AC  E9 001B5          BLBC    REVERSE, 16$                                            ; 1880
                            04    A8              68  D0 001B9          MOVL    JOUR_EFBLK, JOUR_INBLK                                  ; 1883
                            0A    A8      08    A8  B0 001BD          MOVW    JOUR_FFBYTE, JOUR_INBYTE                                ; 1884
                                            09  11  11 001C2          BRB     17$                                                     ; 1880
                                  04        A8  D4 001C4 16$:     CLRL    JOUR_INBLK                                              ; 1888
                            0A    A8    01FF  8F  B0 001C7          MOVW    #511, JOUR_INBYTE                                       ; 1889
                                50      04    A8  D0 001CD 17$:     MOVL    JOUR_INBLK, R0                                          ; 1895
                                            38  13  D1 001D1          BEQL    19$                                                     ;
                            FC    A8              50  D1 001D3          CMPL    R0, JOUR_HIBLK                                          ;
                                            32  1A  D7 001D7          BGTRU   19$                                                     ;
                                            7E  7C  D9 001D9          CLRQ    -(SP)                                                   ; 1904
                                            7E  D4  DB 001DB          CLRL    -(SP)                                                   ;
                                            50  DD  DD 001DD          PUSHL   R0                                                      ;
                                7E      0200    8F  3C 001DF          MOVZWL  #512, -(SP)                                             ;
                                  F4        A8  DD 001E4          PUSHL   JOUR_BUFFER                                             ;
                                            7E  7C 001E7          CLRQ    -(SP)                                                   ;
                                  5C        AE  9F 001E9          PUSHAB  IOSB                                                    ;
                                            31  DD 001EC          PUSHL   #49                                                     ;
                                  0C        A6  DD 001EE          PUSHL   12(FAB)                                                 ;
                                            7E  D4 001F1          CLRL    -(SP)                                                   ;
                                6A          0C  FB 001F3          CALLS   #12, SYS$QIOW                                           ;
                                57          50  D0 001F6          MOVL    R0, STATUS                                              ;
                                07          57  E9 001F9          BLBC    STATUS, 18$                                             ; 1905
                                57        3C    AE 001FC          MOVZWL  IOSB, STATUS                                            ;
                                08          57  E8 00200          BLBS    STATUS, 19$                                             ; 1906
                                7E          56  7D 00203 18$:     MOVQ    FAB, -(SP)                                              ; 1910
                                            5B  DD 00206          PUSHL   R11                                                     ; 1909
```

```
                          69          03  FB 00208          CALLS   #3, FILE_ERROR                                ;
                                      04 0020B 19$:      RET                                                     ; 1914

; Routine Size:  524 bytes,    Routine Base:  CODE + 000C
```

```
;  364          1915  1 %SBTTL 'CLOSE_JOURNAL - close journal file'
;  365          1916  1 GLOBAL ROUTINE CLOSE_JOURNAL(OUTPUT): NOVALUE=
;  366          1917  1
;  367          1918  1 !++
;  368          1919  1 !
;  369          1920  1 ! FUNCTIONAL DESCRIPTION:
;  370          1921  1 !       This routine closes the journal file.
;  371          1922  1 !
;  372          1923  1 ! INPUT PARAMETERS:
;  373          1924  1 !       OUTPUT          - True if the journal file is being written, false if
;  374          1925  1 !                         being read.
;  375          1926  1 !
;  376          1927  1 ! IMPLICIT INPUTS:
;  377          1928  1 !       QUAL[QUAL_JOUR_FC] - Pointer to FAB for journal file.
;  378          1929  1 !
;  379          1930  1 ! OUTPUT PARAMETERS:
;  380          1931  1 !       NONE
;  381          1932  1 !
;  382          1933  1 ! IMPLICIT OUTPUTS:
;  383          1934  1 !       NONE
;  384          1935  1 !
;  385          1936  1 ! ROUTINE VALUE:
;  386          1937  1 !       NONE
;  387          1938  1 !
;  388          1939  1 ! SIDE EFFECTS:
;  389          1940  1 !       NONE
;  390          1941  1 !
;  391          1942  1 !--
;  392          1943  1
;  393          1944  2 BEGIN
;  394          1945  2 LOCAL
;  395          1946  2         FAB:            REF BBLOCK,     ! Pointer to FAB for journal file
;  396          1947  2         STATUS,                         ! Status variable
;  397          1948  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;  398          1949  2         ATR_DESC:       BBLOCK[20],     ! ACP attributes list
;  399          1950  2         RECATTR:        BBLOCK[FAT$C_LENGTH];   ! Record attributes area
;  400          1951  2
;  401          1952  2
;  402          1953  2 FAB = .QUAL[QUAL_JOUR_FC];
;  403          1954  2
;  404          1955  2
;  405          1956  2 ! Write the last buffer if required.
;  406          1957  2 !
;  407          1958  2 IF .OUTPUT
;  408          1959  2 THEN
;  409          1960  3     BEGIN
;  410          1961  3
;  411          1962  3     ! Write structure-level record.
;  412          1963  3     !
;  413          1964  3     WRITE_RECORD(
;  414          1965  3         BJL$K_STRUCLEV,
;  415          1966  3         BJL$C_STRUC_LEN,
;  416          1967  3         STRUC_RECORD);
;  417          1968  3     WRITE_BYTE(.JOUR_COUNT);                    ! Final byte count
;  418          1969  3
;  419          1970  3
;  420          1971  3     ! Write last buffer.
```

```
  3        !
  3        WRITE_BLOCK();
  3
  3
  3        ! Generate record attributes.
  3        !
  3        CH$FILL(0, FAT$C_LENGTH, RECATTR);
  3        RECATTR[FAT$B_RTYPE] = FAT$C_FIXED;
  3        RECATTR[FAT$W_RSIZE] = 1;
  3        RECATTR[FAT$L_HIBLK] = ROT(.JOUR_HIBLK, 16);
  3        RECATTR[FAT$L_EFBLK] = ROT(.JOUR_EFBLK, 16);
  3        RECATTR[FAT$W_FFBYTE] = .JOUR_FFBYTE;
  3        RECATTR[FAT$W_DEFEXT] = .JOUR_EXSZ ;
  3
  3        ! Write record attributes.
  3        !
  3        ATR_DESC[0,0,16,0] = ATR$S_RECATTR;
  3        ATR_DESC[2,0,16,0] = ATR$C_RECATTR;
  3        ATR_DESC[4,0,32,0] = RECATTR;
  3        ATR_DESC[8,0,32,0] = 0;
P 3        STATUS = $QIOW(
P 3            FUNC=IO$_DEACCESS,
P 3            CHAN=.FAB[FAB$L_STV],
P 3            IOSB=IOSB,
  3            P5=ATR_DESC);
  3        IF .STATUS THEN STATUS = .IOSB[0];
  3        IF NOT .STATUS
  3        THEN
  3            FILE_ERROR(
  3                BACKUP$_WRITEATTR,
  3                .FAB,
  3                .STATUS);
  2        END;
  2
  2
  2    ! Deassign the channel.
  2    !
  2    $DASSGN(CHAN=.FAB[FAB$L_STV]);
  1    END;


                                                  .EXTRN   SYS$DASSGN

                                 00FC 00000       .ENTRY   CLOSE_JOURNAL, Save R2,R3,R4,R5,R6,R7     ; 1916
        57 00000000'     EF   9E 00002            MOVAB    QUAL+52, R7                               ;
        5E               3C   C2 00009            SUBL2    #60, SP                                   ;
        56               67   D0 0000C            MOVL     QUAL+52, FAB                              ; 1953
        03        04     AC   E8 0000F            BLBS     OUTPUT, 1$                                ; 1958
                       0089   31 00013            BRW      3$                                        ;
                FDCE     CF   9F 00016 1$:        PUSHAB   STRUC_RECORD                              ; 1964
                         02   DD 0001A            PUSHL    #2                                        ;
                         7E   D4 0001C            CLRL     -(SP)                                     ;
   0000V  CF             03   FB 0001E            CALLS    #3, WRITE_RECORD                          ;
          7E      0232   C7   9A 00023            MOVZBL   JOUR_COUNT, -(SP)                         ; 1968
   0000V  CF             01   FB 00028            CALLS    #1, WRITE_BYTE                            ;
   0000V  CF             00   FB 0002D            CALLS    #0, WRITE_BLOCK                           ; 1973
```

```
                20              00              6E                      00  2C 00032        MOVC5   #0, (SP), #0, #32, RECATTR      ; 1978
                                                                        6E     00037
                                                6E                      01  90 00038        MOVB    #1, RECATTR                     ; 1979
                                        02      AE                      01  B0 0003B        MOVW    #1, RECATTR+2                   ; 1980
                                04      AE      0220    C7              10  9C 0003F        ROTL    #16, JOUR_HIBLK, RECATTR+4      ; 1981
                                08      AE      0224    C7              10  9C 00046        ROTL    #16, JOUR_EFBLK, RECATTR+8      ; 1982
                                        0C      AE      022C    C7      B0  00004D        MOVW    JOUR_FFBYTE, RECATTR+12         ; 1983
                                        12      AE      0234    C7      B0  00053        MOVW    JOUR_EXSZ, RECATTR+18           ; 1984
                                        20      AE  00040020    8F      D0  00059        MOVL    #262176, ATR_DESC               ; 1988
                                        24      AE              6E      9E  00061        MOVAB   RECATTR, ATR_DESC+4             ; 1990
                                                        28      AE      D4  00065        CLRL    ATR_DESC+8                      ; 1991
                                                                7E      D4  00068        CLRL    -(SP)                           ; 1996
                                                        24      AE      9F  0006A        PUSHAB  ATR_DESC
                                                                7E      7C  0006D        CLRQ    -(SP)
                                                                7E      7C  0006F        CLRQ    -(SP)
                                                                7E      7C  00071        CLRQ    -(SP)
                                                        54      AE      9F  00073        PUSHAB  IOSB
                                                                34      DD  00076        PUSHL   #52
                                                        0C      A6      DD  00078        PUSHL   12(FAB)
                                                                7E      D4  0007B        CLRL    -(SP)
                        00000000G       00                      0C      FB  0007D        CALLS   #12, SYS$QIOW
                                        07                      50      E9  00084        BLBC    STATUS, 2$                      ; 1997
                                        50              34      AE      3C  00087        MOVZWL  IOSB, STATUS
                                        11                      50      E8  0008B        BLBS    STATUS, 3$                      ; 1998
                                                                50      DD  0008E 2$:    PUSHL   STATUS                          ; 2003
                                                                56      DD  00090        PUSHL   FAB                             ; 2002
                                                00000000G       8F      DD  00092        PUSHL   #BACKUP$_WRITEATTR              ; 2000
                        00000000G       00                      03      FB  00098        CALLS   #3, FILE_ERROR
                                                        0C      A6      DD  0009F 3$:    PUSHL   12(FAB)                         ; 2009
                        00000000G       00                      01      FB  000A2        CALLS   #1, SYS$DASSGN
                                                                        04  000A9        RET                                     ; 2010

; Routine Size:  170 bytes,    Routine Base:  CODE + 0218
```

```
;  461          2011  1 %SBTTL 'WRITE_BLOCK - write a block to journal'
;  462          2012  1 ROUTINE WRITE_BLOCK: NOVALUE=
;  463          2013  1
;  464          2014  1 !++
;  465          2015  1 !
;  466          2016  1 ! FUNCTIONAL DESCRIPTION:
;  467          2017  1 !       This routine writes a block to the journal file, extending it if
;  468          2018  1 !       necessary.
;  469          2019  1 !
;  470          2020  1 ! INPUT PARAMETERS:
;  471          2021  1 !       NONE
;  472          2022  1 !
;  473          2023  1 ! IMPLICIT INPUTS:
;  474          2024  1 !       NONE
;  475          2025  1 !
;  476          2026  1 ! OUTPUT PARAMETERS:
;  477          2027  1 !       NONE
;  478          2028  1 !
;  479          2029  1 ! IMPLICIT OUTPUTS:
;  480          2030  1 !       NONE
;  481          2031  1 !
;  482          2032  1 ! ROUTINE VALUE:
;  483          2033  1 !       NONE
;  484          2034  1 !
;  485          2035  1 ! SIDE EFFECTS:
;  486          2036  1 !       NONE
;  487          2037  1 !
;  488          2038  1 !--
;  489          2039  1
;  490          2040  2 BEGIN
;  491          2041  2 LOCAL
;  492          2042  2         FAB:            REF BBLOCK,     ! Pointer to FAB for journal file
;  493          2043  2         STATUS,                         ! Status variable
;  494          2044  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;  495          2045  2         FIB:            BBLOCK[FIB$C_LENGTH],   ! FIB
;  496          2046  2         FIB_DESC:       VECTOR[2];      ! Descriptor for FIB
;  497          2047  2
;  498          2048  2
;  499          2049  2 ! Find the FAB.
;  500          2050  2 !
;  501          2051  2 FAB = .QUAL[QUAL_JOUR_FC];
;  502          2052  2
;  503          2053  2
;  504          2054  2 ! If there is insufficient space in the file, extend it before writing.
;  505          2055  2 !
;  506          2056  2 IF .JOUR_HIBLK LSSU .JOUR_EFBLK
;  507          2057  2 THEN
;  508          2058  3     BEGIN
;  509          2059  3     FIB_DESC[0] = FIB$C_LENGTH;
;  510          2060  3     FIB_DESC[1] = FIB;
;  511          2061  3     CH$FILL(0, FIB$C_LENGTH, FIB);
;  512          2062  3     FIB[FIB$V_EXTEND] = TRUE;
;  513          2063  3     FIB[FIB$L_EXSZ] = .JOUR_EXSZ;
;  514        P 2064  3     STATUS = $QIOW(
;  515        P 2065  3         FUNC=IO$_MODIFY,
;  516        P 2066  3         CHAN=.FAB[FAB$L_STV],
;  517        P 2067  3         IOSB=IOSB,
```

```
;  518          2068  3          P1=FIB_DESC);
;  519          2069  3      IF .STATUS THEN STATUS = .IOSB[0];
;  520          2070  3      IF NOT .STATUS
;  521          2071  3      THEN
;  522          2072  3          FILE_ERROR(
;  523          2073  3              BACKUP$_WRITEERR + STS$K_SEVERE,
;  524          2074  3              .FAB,
;  525          2075  3              .STATUS);
;  526          2076  3      JOUR_HIBLK = .JOUR_HIBLK + .FIB[FIB$L_EXSZ];
;  527          2077  2      END;
;  528          2078  2
;  529          2079  2
;  530          2080  2  ! Write the block.
;  531          2081  2  !
;  532        P 2082  2  STATUS = $QIOW(
;  533        P 2083  2      FUNC=IO$_WRITEVBLK,
;  534        P 2084  2      CHAN=.FAB[FAB$L_STV],
;  535        P 2085  2      IOSB=IOSB,
;  536        P 2086  2      P1=.JOUR_BUFFER,
;  537        P 2087  2      P2=512,
;  538          2088  2      P3=.JOUR_EFBLK);
;  539          2089  2  IF .STATUS THEN STATUS = .IOSB[0];
;  540          2090  2  IF NOT .STATUS
;  541          2091  2  THEN
;  542          2092  2      FILE_ERROR(
;  543          2093  2          BACKUP$_WRITEERR + STS$K_SEVERE,
;  544          2094  2          .FAB,
;  545          2095  2          .STATUS);
;  546          2096  1  END;


                                        07FC 00000 WRITE_BLOCK:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10        ; 2012
                                 5A 000C0000G   00  9E 00002    MOVAB   FILE_ERROR, R10                ;
                                 59 00000000G   8F  D0 00009    MOVL    #BACKUP$_WRITEERR+4, R9        ;
                                 58 00000000G   00  9E 00010    MOVAB   SYS$QIOW, R8                   ;
                                 57 00000000'   EF  9E 00017    MOVAB   JOUR_HIBLK, R7                 ;
                                 5E       B0    AE  9E 0001E    MOVAB   -80(SP), SP                    ;
                                 56     FDE0    C7  D0 00022    MOVL    QUAL+52, FAB                   ; 2051
                           04    A7             67  D1 00027    CMPL    JOUR_HIBLK, JOUR_EFBLK         ; 2056
                                                4E  1E 0002B    BGEQU   3$                             ;
                                 6E       40    8F  9A 0002D    MOVZBL  #64, FIB_DESC                  ; 2059
                           04    AE       08    AE  9E 00031    MOVAB   FIB, FIB_DESC+4                ; 2060
0040   8F                00      6E             00  2C 00036    MOVC5   #0, (SP), #0, #64, FIB         ; 2061
                                             08    AE  0003D
                           1E    AE       80    8F  88 0003F    BISB2   #128, FIB+22                   ; 2062
                           20    AE       14    A7  3C 00044    MOVZWL  JOUR_EXSZ, FIB+24              ; 2063
                                                7E  7C 00049    CLRQ    -(SP)                          ; 2068
                                                7E  7C 0004B    CLRQ    -(SP)                          ;
                                                7E  D4 0004D    CLRL    -(SP)                          ;
                                             14    AE  9F 0004F    PUSHAB  FIB_DESC                       ;
                                                7E  7C 00052    CLRQ    -(SP)                          ;
                                             68    AE  9F 00054    PUSHAB  IOSB                           ;
                                                36  DD 00057    PUSHL   #54                            ;
```

```
                    0C   A6  DD 00059          PUSHL   12(FAB)                          ;
                         7E  D4 0005C          CLRL    -(SP)                           ;
                68       0C  FB 0005E          CALLS   #12, SYS$QIOW                   ;
                52           50  D0 00061      MOVL    R0, STATUS                      ;
                07           52  E9 00064      BLBC    STATUS, 1$                      ; 2069
                52       48  AE  3C 00067      MOVZWL  IOSB, STATUS                    ;
                09           52  E8 0006B      BLBS    STATUS, 2$                      ; 2070
                             52  DD 0006E 1$:   PUSHL   STATUS                          ; 2075
                             56  DD 00070      PUSHL   FAB                             ; 2074
                             59  DD 00072      PUSHL   R9                              ; 2073
                6A           03  FB 00074      CALLS   #3, FILE_ERROR                  ;
                67       20  AE  C0 00077 2$:   ADDL2   FIB+24, JOUR_HIBLK             ; 2076
                         7E  7C 0007B 3$:   CLRQ    -(SP)                           ; 2088
                         7E  D4 0007D       CLRL    -(SP)                           ;
                    04   A7  DD 0007F      PUSHL   JOUR_EFBLK                      ;
                7E      0200 8F  3C 00082      MOVZWL  #512, -(SP)                     ;
                        F8   A7  DD 00087      PUSHL   JOUR_BUFFER                     ;
                         7E  7C 0008A      CLRQ    -(SP)                           ;
                    68   AE  9F 0008C      PUSHAB  IOSB                            ;
                    30       DD 0008F      PUSHL   #48                             ;
                    0C   A6  DD 00091      PUSHL   12(FAB)                         ;
                         7E  D4 00094      CLRL    -(SP)                           ;
                68       0C  FB 00096      CALLS   #12, SYS$QIOW                   ;
                52           50  D0 00099      MOVL    R0, STATUS                      ;
                07           52  E9 0009C      BLBC    STATUS, 4$                      ; 2089
                52       48  AE  3C 0009F      MOVZWL  IOSB, STATUS                    ;
                09           52  E8 000A3      BLBS    STATUS, 5$                      ; 2090
                             52  DD 000A6 4$:   PUSHL   STATUS                          ; 2095
                             56  DD 000A8      PUSHL   FAB                             ; 2094
                             59  DD 000AA      PUSHL   R9                              ; 2093
                6A           03  FB 000AC      CALLS   #3, FILE_ERROR                  ;
                             04  000AF 5$:   RET                                     ; 2096

; Routine Size:  176 bytes,    Routine Base:  CODE + 02C2
```

```
;  548          2097  1 %SBTTL 'WRITE_BYTE - write a byte to the journal file'
;  549          2098  1 ROUTINE WRITE_BYTE(VALUE): NOVALUE=
;  550          2099  1
;  551          2100  1 !++
;  552          2101  1 !
;  553          2102  1 ! FUNCTIONAL DESCRIPTION:
;  554          2103  1 !       This routine writes one byte to the journal file.
;  555          2104  1 !
;  556          2105  1 ! INPUT PARAMETERS:
;  557          2106  1 !       VALUE           - Value to be written.
;  558          2107  1 !
;  559          2108  1 ! IMPLICIT INPUTS:
;  560          2109  1 !       QUAL[QUAL_JOUR_FC] - Pointer to FAB for journal file.
;  561          2110  1 !
;  562          2111  1 ! OUTPUT PARAMETERS:
;  563          2112  1 !       NONE
;  564          2113  1 !
;  565          2114  1 ! IMPLICIT OUTPUTS:
;  566          2115  1 !       NONE
;  567          2116  1 !
;  568          2117  1 ! ROUTINE VALUE:
;  569          2118  1 !       NONE
;  570          2119  1 !
;  571          2120  1 ! SIDE EFFECTS:
;  572          2121  1 !       NONE
;  573          2122  1 !
;  574          2123  1 !--
;  575          2124  1
;  576          2125  2 BEGIN
;  577          2126  2
;  578          2127  2 ! If there is insufficient space in the current buffer, write it and initialize
;  579          2128  2 ! another.
;  580          2129  2 !
;  581          2130  2 IF .JOUR_FFBYTE GEQU 512
;  582          2131  2 THEN
;  583          2132  3     BEGIN
;  584          2133  3     WRITE_BLOCK();
;  585          2134  3     CH$FILL(0, 512, .JOUR_BUFFER);
;  586          2135  3     JOUR_EFBLK = .JOUR_EFBLK + 1;
;  587          2136  3     JOUR_FFBYTE = 0;
;  588          2137  2     END;
;  589          2138  2
;  590          2139  2
;  591          2140  2 ! Write the byte into the buffer.
;  592          2141  2 !
;  593          2142  2 JOUR_BUFFER[.JOUR_FFBYTE] = .VALUE;
;  594          2143  2 JOUR_FFBYTE = .JOUR_FFBYTE + 1;
;  595          2144  1 END;


                                007C 00000 WRITE_BYTE:
                                           .WORD   Save R2,R3,R4,R5,R6                                     ; 2098
                 56 00000000'   EF  9E 00002        MOVAB   JOUR_FFBYTE, R6                                 ;
       0200      8F             66  B1 00009        CMPW    JOUR_FFBYTE, #512                               ; 2130
```

```
                                   13  1F 0000E         BLSSU   1$                                   ;
                      FF3B   CF    00  FB 00010         CALLS   #0, WRITE_BLOCK                      ; 2133
   0200   8F      00         6E    00  2C 00015         MOVC5   #0, (SP), #0, #512, @JCUR_BUFFER     ; 2134
                                EC B6     0001C                                                      ;
                                F8 A6  D6 0001E         INCL    JOUR_EFBLK                           ; 2135
                                   66  B4 00021         CLRW    JOUR_FFBYTE                          ; 2136
                             50    66  3C 00023 1$:     MOVZWL  JOUR_FFBYTE, R0                      ; 2142
                             50 EC A6  C0 00026         ADDL2   JOUR_BUFFER, R0                      ;
                             60 04 AC  90 0002A         MOVB    VALUE, (R0)                          ;
                                   66  B6 0002E         INCW    JOUR_FFBYTE                          ; 2143
                                       04 00030         RET                                          ; 2144

; Routine Size:  49 bytes,    Routine Base:  CODE + 0372
```

```
;  597          2145  1 %SBTTL 'WRITE_RECORD - write a record to the journal file'
;  598          2146  1 ROUTINE WRITE_RECORD(TYPE,LENGTH,ADDRESS): NOVALUE=
;  599          2147  1
;  600          2148  1 !++
;  601          2149  1 !
;  602          2150  1 ! FUNCTIONAL DESCRIPTION:
;  603          2151  1 !       This routine writes a record in the journal file.
;  604          2152  1 !
;  605          2153  1 ! INPUT PARAMETERS:
;  606          2154  1 !       TYPE            - Record type.
;  607          2155  1 !       LENGTH          - Length of data portion.
;  608          2156  1 !       ADDRESS         - Address of data portion.
;  609          2157  1 !
;  610          2158  1 ! IMPLICIT INPUTS:
;  611          2159  1 !       NONE
;  612          2160  1 !
;  613          2161  1 ! OUTPUT PARAMETERS:
;  614          2162  1 !       NONE
;  615          2163  1 !
;  616          2164  1 ! IMPLICIT OUTPUTS:
;  617          2165  1 !       NONE
;  618          2166  1 !
;  619          2167  1 ! ROUTINE VALUE:
;  620          2168  1 !       NONE
;  621          2169  1 !
;  622          2170  1 ! SIDE EFFECTS:
;  623          2171  1 !       NONE
;  624          2172  1 !
;  625          2173  1 !--
;  626          2174  1
;  627          2175  2 BEGIN
;  628          2176  2 MAP
;  629          2177  2         ADDRESS:        REF VECTOR[,BYTE];      ! Pointer to record
;  630          2178  2
;  631          2179  2
;  632          2180  2 WRITE_BYTE(.JOUR_COUNT XOR (.LENGTH + 1));      ! XOR byte count
;  633          2181  2 WRITE_BYTE(.TYPE);                              ! Type code
;  634          2182  2 INCR I FROM 0 TO .LENGTH-1 DO WRITE_BYTE(.ADDRESS[.I]);
;  635          2183  2 JOUR_COUNT = .LENGTH + 1;
;  636          2184  1 END;


                                        003C 00000 WRITE_RECORD:
                                                           .WORD   Save R2,R3,R4,R5                        ; 2146
                                55          CA   AF  9E 00002          MOVAB   WRITE_BYTE, R5                  ;
                                54 00000000'    EF  9E 00006          MOVAB   JOUR_COUNT, R4                  ;
                     53     08  AC              01  C1 0000D          ADDL3   #1, LENGTH, R3                  ; 2180
                                50              64  9A 00012          MOVZBL  JOUR_COUNT, R0                  ;
                     7E         50              53  CD 00015          XORL3   R3, R0, -(SP)                   ;
                                65              01  FB 00019          CALLS   #2, WRITE_BYTE                  ;
                                            04  AC  DD 0001C          PUSHL   TYPE                            ; 2181
                                65              01  FB 0001F          CALLS   #1, WRITE_BYTE                  ;
                                52              01  CE 00022          MNEGL   #1, I                           ; 2182
                                                08  11 00025          BRB     2$                              ;
```

```
                          7E      0C BC42  9A 00027 1$:    MOVZBL  @ADDRESS[I], -(SP)                ;
                          65          01   FB 0002C       CALLS   #1, WRITE_BYTE                    ;
                    F3    52      08   AC   F2 0002F 2$:  AOBLSS  LENGTH, I, 1$                    ;
                          64          53   90 00034       MOVB    R3, JOUR_COUNT                    ; 2183
                                           04 00037       RET                                       ; 2184

; Routine Size:  56 bytes,    Routine Base:  CODE + 03A3
```

```
;  638          2185  1 %SBTTL 'WRITE_JOUR_SSNAME - write save set name to journal'
;  639          2186  1 GLOBAL ROUTINE WRITE_JOUR_SSNAME: NOVALUE=
;  640          2187  1
;  641          2188  1 !++
;  642          2189  1 !
;  643          2190  1 ! FUNCTIONAL DESCRIPTION:
;  644          2191  1 !       This routine writes the save set name record in the journal file.
;  645          2192  1 !
;  646          2193  1 ! INPUT PARAMETERS:
;  647          2194  1 !       NONE
;  648          2195  1 !
;  649          2196  1 ! IMPLICIT INPUTS:
;  650          2197  1 !       JPI_DATE        - Current date
;  651          2198  1 !       COM_SSNAME      - Save set name
;  652          2199  1 !
;  653          2200  1 ! OUTPUT PARAMETERS:
;  654          2201  1 !       NONE
;  655          2202  1 !
;  656          2203  1 ! IMPLICIT OUTPUTS:
;  657          2204  1 !       NONE
;  658          2205  1 !
;  659          2206  1 ! ROUTINE VALUE:
;  660          2207  1 !       NONE
;  661          2208  1 !
;  662          2209  1 ! SIDE EFFECTS:
;  663          2210  1 !       NONE
;  664          2211  1 !
;  665          2212  1 !--
;  666          2213  1
;  667          2214  2 BEGIN
;  668          2215  2 LOCAL
;  669          2216  2         BUFFER:         BBLOCK[BJL$C_SSNAME_LEN];       ! Area to build record
;  670          2217  2
;  671          2218  2
;  672          2219  2 ! Assemble the record.
;  673          2220  2 !
;  674          2221  2 (BUFFER[BJL$Q_CREDATE]) = .JPI_DATE[0];
;  675          2222  2 (BUFFER[BJL$Q_CREDATE]+4) = .JPI_DATE[1];
;  676          2223  2 CH$MOVE(
;  677          2224  2     .COM_SSNAME[DSC$W_LENGTH],
;  678          2225  2     .COM_SSNAME[DSC$A_POINTER],
;  679          2226  2     BUFFER[BJL$T_SSNAME]);
;  680          2227  2
;  681          2228  2
;  682          2229  2 ! Write the record.
;  683          2230  2 !
;  684          2231  2 WRITE_RECORD(
;  685          2232  2     BJL$K_SSNAME,
;  686          2233  2     $BYTEOFFSET(BJL$T_SSNAME) + .COM_SSNAME[DSC$W_LENGTH],
;  687          2234  2     BUFFER);
;  688          2235  1 END;


                                007C 00000          .ENTRY  WRITE_JOUR_SSNAME, Save R2,R3,R4,R5,R6      ; 2186
```

```
                          56 00000000'   EF  9E 00002         MOVAB   COM_SSNAME, R6                                          ;
                          5E             24  C2 00009         SUBL2   #36, SP                                                 ;
                                 FE74    C6  DD 0000C         PUSHL   JPI_DATE                                                ; 2221
                       04    AE  FE78    C6  D0 00010         MOVL    JPI_DATE+4, BUFFER+4                                    ; 2222
                 08    AE    04  B6      66  28 00016         MOVC3   COM_SSNAME, @COM_SSNAME+4, BUFFER+8                     ; 2226
                                         5E  DD 0001C         PUSHL   SP                                                      ; 2231
                          7E             66  3C 0001E         MOVZWL  COM_SSNAME, -(SP)                                       ; 2233
                          6E             08  C0 00021         ADDL2   #8, (SP)                                                ;
                                         01  DD 00024         PUSHL   #1                                                      ; 2231
                       9E    AF          03  FB 00026         CALLS   #3, WRITE_RECORD                                        ;
                                             04 0002A         RET                                                             ; 2235

; Routine Size:  43 bytes,    Routine Base:  CODE + 03DB
```

```
;  690          2236  1 %SBTTL 'WRITE_JOUR_VOLUME - write volume ID record to journal'
;  691          2237  1 GLOBAL ROUTINE WRITE_JOUR_VOLUME: NOVALUE=
;  692          2238  1
;  693          2239  1 !++
;  694          2240  1 !
;  695          2241  1 ! FUNCTIONAL DESCRIPTION:
;  696          2242  1 !       This routine writes the volume identification record
;  697          2243  1 !       in the journal file.
;  698          2244  1 !
;  699          2245  1 ! INPUT PARAMETERS:
;  700          2246  1 !       NONE
;  701          2247  1 !
;  702          2248  1 ! IMPLICIT INPUTS:
;  703          2249  1 !       RWSV_VOLUME_ID  - Current volume label
;  704          2250  1 !       RWSV_VOL_NUMBER - Current volume number
;  705          2251  1 !
;  706          2252  1 ! OUTPUT PARAMETERS:
;  707          2253  1 !       NONE
;  708          2254  1 !
;  709          2255  1 ! IMPLICIT OUTPUTS:
;  710          2256  1 !       NONE
;  711          2257  1 !
;  712          2258  1 ! ROUTINE VALUE:
;  713          2259  1 !       NONE
;  714          2260  1 !
;  715          2261  1 ! SIDE EFFECTS:
;  716          2262  1 !       NONE
;  717          2263  1 !
;  718          2264  1 !--
;  719          2265  1
;  720          2266  2 BEGIN
;  721          2267  2 LOCAL
;  722          2268  2         BUFFER:         BBLOCK[BJL$C_VOLUME_LEN];       ! Area to build record
;  723          2269  2
;  724          2270  2
;  725          2271  2 ! Assemble the record.
;  726          2272  2 !
;  727          2273  2 CH$COPY(
;  728          2274  2     %ALLOCATION(RWSV_VOLUME_ID), RWSV_VOLUME_ID,
;  729          2275  2     %C' ',
;  730          2276  2     BJL$S_VOLNAME, BUFFER[BJL$T_VOLNAME]);
;  731          2277  2 BUFFER[BJL$W_VOLNUMBER] = .RWSV_VOL_NUMBER;
;  732          2278  2
;  733          2279  2
;  734          2280  2 ! Write the record.
;  735          2281  2 !
;  736          2282  2 WRITE_RECORD(
;  737          2283  2     BJL$K_VOLUME,
;  738          2284  2     BJL$C_VOLUME_LEN,
;  739          2285  2     BUFFER);
;  740          2286  1 END;


                                003C 00000          .ENTRY  WRITE_JOUR_VOLUME, Save R2,R3,R4,R5          ; 2237
```

```
                          5E              10  C2 00002        SUBL2   #16, SP                                                 ;
                6E 00000000'  EF          0C  28 00005        MOVC3   #12, RWSV_VOLUME_ID, BUFFER                      ; 2276
                          0C  AE 00000000'  EF  B0 0000D        MOVW    RWSV_VOL_NUMBER, BUFFER+12                       ; 2277
                                            5E  DD 00015        PUSHL   SP                                               ; 2282
                                            0E  DD 00017        PUSHL   #14                                              ;
                                            02  DD 00019        PUSHL   #2                                               ;
                            FF7D  CF        03  FB 0001B        CALLS   #3, WRITE_RECORD                                 ;
                                                04 00020        RET                                                      ; 2286

; Routine Size:  33 bytes,    Routine Base:  CODE + 0406
```

```
;  742          2287  1 %SBTTL 'WRITE_JOUR_FILE - write file name record to journal'
;  743          2288  1 GLOBAL ROUTINE WRITE_JOUR_FILE ( HEADONLY ) : NOVALUE=
;  744          2289  1
;  745          2290  1 !++
;  746          2291  1 !
;  747          2292  1 ! FUNCTIONAL DESCRIPTION:
;  748          2293  1 !       This routine writes the file name record (and directory name record,
;  749          2294  1 !       if required) in the journal file.
;  750          2295  1 !
;  751          2296  1 ! INPUT PARAMETERS:
;  752          2297  1 !       HEADONLY        - Flag specifing if the file is marked as NOBACKUP
;  753          2298  1 !                         and therefore only the file header is saved.
;  754          2299  1 !
;  755          2300  1 ! IMPLICIT INPUTS:
;  756          2301  1 !       INPUT_NAM       - File specification in resultant string
;  757          2302  1 !
;  758          2303  1 ! OUTPUT PARAMETERS:
;  759          2304  1 !       NONE
;  760          2305  1 !
;  761          2306  1 ! IMPLICIT OUTPUTS:
;  762          2307  1 !       NONE
;  763          2308  1 !
;  764          2309  1 ! ROUTINE VALUE:
;  765          2310  1 !       NONE
;  766          2311  1 !
;  767          2312  1 ! SIDE EFFECTS:
;  768          2313  1 !       NONE
;  769          2314  1 !
;  770          2315  1 !--
;  771          2316  1
;  772          2317  2 BEGIN
;  773          2318  2
;  774          2319  2 LOCAL
;  775          2320  2     NAME_LENGTH : WORD,
;  776          2321  2     REC_BUFF    : VECTOR [BJL$C_FILE_LEN,BYTE] ;  ! Record buffer
;  777          2322  2
;  778          2323  2 ! Write the directory record if necessary.
;  779          2324  2 !
;  780          2325  2 IF CH$NEQ(
;  781          2326  2     .INPUT_NAM[NAM$B_DIR]-2, .INPUT_NAM[NAM$L_DIR]+1,
;  782          2327  2     BJL$C_DIR_LEN, JOUR_DIR[1],
;  783          2328  2     0)
;  784          2329  2 THEN
;  785          2330  3     BEGIN
;  786          2331  3     LOCAL
;  787          2332  3         P,                                  ! Temp
;  788          2333  3         REC_LENGTH;                         ! Length of directory record
;  789          2334  3
;  790          2335  3
;  791          2336  3     ! Establish the XOR of the new and old directory strings.
;  792          2337  3     !
;  793          2338  3     INCR I FROM 0 TO .INPUT_NAM[NAM$B_DIR]-2-1 DO
;  794          2339  3         JOUR_DIR[.I+1] = .JOUR_DIR[.I+1] XOR
;  795          2340  3             .VECTOR[.INPUT_NAM[NAM$L_DIR]+1, .I ;,BYTE];
;  796          2341  3
;  797          2342  3
;  798          2343  3     ! Strip leading nulls from the XOR.
```

```
;  799     2344  3     !
;  800     2345  3     P = CH$FIND_NOT_CH(BJL$C_DIR_LEN, JOUR_DIR[1], 0);  ! Find non-null
;  801     2346  3     IF .P EQL 0 THEN P = JOUR_DIR[1];                   ! Entire string is null
;  802     2347  3     P = .P - 1;                                         ! Point to previous byte
;  803     2348  3     (.P)<0,8> = .P - JOUR_DIR[0];                       ! Set count of leading nulls
;  804     2349  3
;  805     2350  3
;  806     2351  3     ! Strip trailing nulls from the XOR.
;  807     2352  3     !
;  808     2353  3     REC_LENGTH = 1;
;  809     2354  3     DECRA Q FROM JOUR_DIR[BJL$C_DIR_LEN] TO .P+1 DO
;  810     2355  4         BEGIN
;  811     2356  4         IF .(.Q)<0,8> NEQ 0
;  812     2357  4         THEN
;  813     2358  5             BEGIN
;  814     2359  5             REC_LENGTH = .Q - .P + 1;
;  815     2360  5             EXITLOOP;
;  816     2361  4             END;
;  817     2362  3         END;
;  818     2363  3
;  819     2364  3
;  820     2365  3     ! Write the directory record.
;  821     2366  3     !
;  822     2367  3     WRITE_RECORD(
;  823     2368  3         BJL$K_DIRECTORY,
;  824     2369  3         .REC_LENGTH,
;  825     2370  3         .P);
;  826     2371  3
;  827     2372  3
;  828     2373  3     ! Save the new directory string.
;  829     2374  3     !
;  830     2375  3     CH$COPY(
;  831     2376  3         .INPUT_NAM[NAM$B_DIR]-2, .INPUT_NAM[NAM$L_DIR]+1,
;  832     2377  3         0,
;  833     2378  3         BJL$C_DIR_LEN, JOUR_DIR[1]);
;  834     2379  2     END;
;  835     2380  2
;  836     2381  2 ! Compute length of name, extention etc.
;  837     2382  2 !
;  838     2383  2 NAME_LENGTH = .INPUT_NAM[NAM$L_RSA] +
;  839     2384  2                .INPUT_NAM[NAM$B_RSL] -
;  840     2385  2                .INPUT_NAM[NAM$L_NAME] ;
;  841     2386  2
;  842     2387  2 ! Move the name string into local buffer
;  843     2388  2 !
;  844     2389  2 CH$COPY ( .NAME_LENGTH, .INPUT_NAM[NAM$L_NAME], %C' ',
;  845     2390  2           BJL$S_FILENAME, REC_BUFF) ;
;  846     2391  2
;  847     2392  2 ! Concatenate the flags to the end of the name
;  848     2393  2 !
;  849     2394  3 BEGIN
;  850     2395  3 BIND
;  851     2396  3     FLAGS = REC_BUFF[.NAME_LENGTH] : LONG ;
;  852     2397  3 IF .HEADONLY
;  853     2398  3 THEN
;  854     2399  3     FLAGS = BJL$M_HEADONLY
;  855     2400  3 ELSE
```

```
;  856          2401  3     FLAGS = 0 ;
;  857          2402  2 END;
;  858          2403  2
;  859          2404  2 ! Write the file record.
;  860          2405  2 !
;  861          2406  2 WRITE_RECORD(
;  862          2407  2     BJL$K_FILE,
;  863          2408  2     .NAME_LENGTH + 4 ,
;  864          2409  2     REC_BUFF ) ;
;  865          2410  1 END;
```

```
                                                00FC 00000          .ENTRY  WRITE_JOUR_FILE, Save R2,R3,R4,R5,R6,R7     ; 2288
                                        57 00000000'   EF  9E 00002 MOVAB   INPUT_NAM, R7                               ;
                                        5E     FF7C    CE  9E 00009 MOVAB   -132(SP), SP                               ;
                                        50             67  D0 0000E MOVL    INPUT_NAM, R0                            ; 2326
                                        56       3A    A0  9A 00011 MOVZBL  58(R0), R6                               ;
                                        56             02  C2 00015 SUBL2   #2, R6                                   ;
                                        55       48    A0  D0 00018 MOVL    72(R0), R5                               ;
                                        54     0180    C7  D0 0001C MOVL    JOUR_DIR, R4                             ; 2327
    00FF    8F          00      01    A5              56  2D 00021 CMPC5   R6, T(R5), #0, #255, 1(R4)               ;
                                              01    A4      00029       ;                                             ;
                                                    74  13 0002B BEQL    8$                                       ;
                                        50              01  CE 0002D MNEGL   #1, I                                   ; 2338
                                                    07  11 00030 BRB     2$                                       ;
                                      01 A044         01 A540   8C 00032 1$:  XORB2   1(R5)[I], 1(I)[R4]              ; 2340
                                F5          50            56  F2 00039 2$:  AOBLSS  R6, I, 1$                       ; 2339
                          01    A4      00FF    8F        00  3B 0003D       SKPC    #0, #255, 1(R4)                  ; 2345
                                                    02  12 00044 BNEQ    3$                                       ;
                                                    51  D4 00046 CLRL    R1                                       ;
                                                    51  D5 00048 3$:  TSTL    P                                    ; 2346
                                                    04  12 0004A BNEQ    4$                                       ;
                                        51       01    A4  9E 0004C MOVAB   1(R4), P                               ;
                                                    51  D7 00050 4$:  DECL    P                                    ; 2347
                61                      51            54  83 00052 SUBB3   R4, P, (P)                             ; 2348
                                        52            01  D0 00056 MOVL    #1, REC_LENGTH                           ; 2353
                                        53       01   A1  9E 00059 MOVAB   1(R1), R3                               ; 2354
                                        50     0100   C4  9E 0005D MOVAB   256(R4), Q                               ;
                                                    0E  11 00062 BRB     6$                                       ;
                                                    60  95 00064 5$:  TSTB    (Q)                                  ; 2356
                                                    0A  13 00066 BEQL    6$                                       ;
                54                      50            51  C3 00068 SUBL3   P, Q, R4                               ; 2359
                                        52       01   A4  9E 0006C MOVAB   1(R4), REC_LENGTH                       ;
                                                    07  11 00070 BRB     7$                                       ; 2358
                                                    50  D7 00072 6$:  DECL    Q                                    ; 2354
                                        53          50  D1 00074 CMPL    Q, R3                                    ;
                                                    EB  1E 00077 BGEQU   5$                                       ;
                                                    51  DD 00079 7$:  PUSHL   P                                    ; 2370
                                                    52  DD 0007B PUSHL   REC_LENGTH                                ; 2369
                                                    03  DD 0007D PUSHL   #3                                        ; 2367
                      FEF8      CF                  03  FB 0007F CALLS   #3, WRITE_RECORD                          ;
                                50                  67  D0 00084 MOVL    INPUT_NAM, R0                            ; 2376
                                52       3A         A0  9A 00087 MOVZBL  58(R0), R2                               ;
                                52                  02  C2 0008B SUBL2   #2, R2                                   ;
```

```
                                  51      48    A0  D0 0008E           MOVL    72(R0), R1                                      ;
                                  50    0180    C7  D0 00092           MOVL    JOUR_DIR, R0                                    ; 2378
    00FF    8F          00    01    A1          52  2C 00097           MOVC5   R2, T(R1), #0, #255, 1(R0)                      ;
                                              01  A0     0009F                                                                ;
                                  50          67  D0 000A1 8$:        MOVL    INPUT_NAM, R0                                   ; 2383
                                  51    03    A0  9A 000A4           MOVZBL  3(R0), R1                                       ; 2384
                                  51    04    A0  C0 000A8           ADDL2   4(R0), R1                                       ;
                        52        51    4C    A0  A3 000AC           SUBW3   76(R0), R1, NAME_LENGTH                         ; 2385
                                  56          52  3C 000B1           MOVZWL  NAME_LENGTH, R6                                 ; 2389
    0080    8F          20    4C    B0          56  2C 000B4           MOVC5   R6, @76(R0), #32, #128, REC_BUFF                ;
                                              6E  000BC                                                                       ;
                                  08    04    AC  E9 000BD           BLBC    HEADONLY, 9$                                    ; 2397
                                            6E46  9F 000C1           PUSHAB  REC_BUFF[R6]                                    ; 2399
                                  9E          01  D0 000C4           MOVL    #1, @(SP)+                                      ;
                                              05  11 000C7           BRB     10$                                             ;
                                            6E46  9F 000C9 9$:       PUSHAB  REC_BUFF[R6]                                    ; 2401
                                              9E  D4 000CC           CLRL    @(SP)+                                          ;
                                              5E  DD 000CE 10$:      PUSHL   SP                                              ; 2406
                                        04    A6  9F 000D0           PUSHAB  4(R6)                                           ; 2408
                                              04  DD 000D3           PUSHL   #4                                              ; 2406
                              FEA2    CF          03  FB 000D5           CALLS   #3, WRITE_RECORD                                ;
                                                  04  000DA           RET                                                     ; 2410

; Routine Size:  219 bytes,    Routine Base:  CODE + 0427
```

```
;  867          2411  1 %SBTTL 'WRITE_JOUR_DIRECTORY - write directory name record to journal'
;  868          2412  1 GLOBAL ROUTINE WRITE_JOUR_DIRECTORY: NOVALUE=
;  869          2413  1
;  870          2414  1 !++
;  871          2415  1 !
;  872          2416  1 ! FUNCTIONAL DESCRIPTION:
;  873          2417  1 !       This routine writes the final directory name record
;  874          2418  1 !       in the journal file.
;  875          2419  1 !
;  876          2420  1 ! INPUT PARAMETERS:
;  877          2421  1 !       NONE
;  878          2422  1 !
;  879          2423  1 ! IMPLICIT INPUTS:
;  880          2424  1 !       NONE
;  881          2425  1 !
;  882          2426  1 ! OUTPUT PARAMETERS:
;  883          2427  1 !       NONE
;  884          2428  1 !
;  885          2429  1 ! IMPLICIT OUTPUTS:
;  886          2430  1 !       NONE
;  887          2431  1 !
;  888          2432  1 ! ROUTINE VALUE:
;  889          2433  1 !       NONE
;  890          2434  1 !
;  891          2435  1 ! SIDE EFFECTS:
;  892          2436  1 !       NONE
;  893          2437  1 !
;  894          2438  1 !--
;  895          2439  1
;  896          2440  2 BEGIN
;  897          2441  2
;  898          2442  2 $ASSUME ( BJL$C_DIR_LEN GEQ NAM$C_MAXRSS )
;  899          2443  2
;  900          2444  2 LOCAL
;  901          2445  2         P,                                  ! Temp
;  902          2446  2         REC_LENGTH;                         ! Length of directory record
;  903          2447  2
;  904          2448  2
;  905          2449  2 ! Note that there are no leading nulls in the directory string.
;  906          2450  2 ! Strip trailing nulls from the directory string.  Since the directory string
;  907          2451  2 ! is printable ASCII, there can be no embedded nulls.
;  908          2452  2 !
;  909          2453  2 JOUR_DIR[0] = 0;                                ! Zero leading nulls
;  910          2454  2 REC_LENGTH = BJL$C_DIR_LEN + 1;                ! Assume no null found
;  911          2455  2 P = CH$FIND_CH(BJL$C_DIR_LEN, JOUR_DIR[1], 0); ! Find first null
;  912          2456  2 IF .P NEQ 0 THEN REC_LENGTH = .P - JOUR_DIR[0]; ! Establish count
;  913          2457  2
;  914          2458  2
;  915          2459  2 ! Write the directory record.
;  916          2460  2 !
;  917          2461  2 WRITE_RECORD(
;  918          2462  2     BJL$K_DIRECTORY,
;  919          2463  2     .REC_LENGTH,
;  920          2464  2     JOUR_DIR[0]);
;  921          2465  1 END;
```

```
                                                 000C 00000          .ENTRY  WRITE_JOUR_DIRECTORY, Save R2,R3                ; 2412
                                  52 00000000'   EF   D0 00002       MOVL    JOUR_DIR, R2                                    ; 2453
                                                 62   94 00009       CLRB    (R2)                                            ;
                                  53      0100   8F   3C 0000B       MOVZWL  #256, REC_LENGTH                                ; 2454
                01    A2    00FF   8F           00   3A 00010       LOCC    #0, #255, 1(R2)                                 ; 2455
                                                 02   12 00017       BNEQ    1$                                              ;
                                                 51   D4 00019       CLRL    R1                                              ;
                                                 51   D5 0001B 1$:    TSTL    P                                               ; 2456
                                                 04   13 0001D       BEQL    2$                                              ;
                     53                 51       52   C3 0001F       SUBL3   R2, P, REC_LENGTH                               ;
                                                 52   DD 00023 2$:    PUSHL   R2                                              ; 2464
                                                 53   DD 00025       PUSHL   REC_LENGTH                                      ;
                                                 03   DD 00027       PUSHL   #3                                              ;
                          FE73   CF              03   FB 00029       CALLS   #3, WRITE_RECORD                                ;
                                                      04 0002E       RET                                                     ; 2465

; Routine Size:  47 bytes,    Routine Base:  CODE + 0502
```

```
;  923          2466  1 %SBTTL 'READ_BYTE - read a byte from journal'
;  924          2467  1 ROUTINE READ_BYTE(BUFFER)=
;  925          2468  1
;  926          2469  1 !++
;  927          2470  1 !
;  928          2471  1 ! FUNCTIONAL DESCRIPTION:
;  929          2472  1 !       This routine reads one byte from the journal file, reading a new
;  930          2473  1 !       block if necessary.
;  931          2474  1 !
;  932          2475  1 ! INPUT PARAMETERS:
;  933          2476  1 !       BUFFER          - Pointer to where byte will be returned.
;  934          2477  1 !
;  935          2478  1 ! IMPLICIT INPUTS:
;  936          2479  1 !       NONE
;  937          2480  1 !
;  938          2481  1 ! OUTPUT PARAMETERS:
;  939          2482  1 !       NONE
;  940          2483  1 !
;  941          2484  1 ! IMPLICIT OUTPUTS:
;  942          2485  1 !       NONE
;  943          2486  1 !
;  944          2487  1 ! ROUTINE VALUE:
;  945          2488  1 !       SS$_NORMAL      - Byte successfully returned.
;  946          2489  1 !       SS$_ENDOFFILE   - End (beginning) of file, no byte returned.
;  947          2490  1 !
;  948          2491  1 ! SIDE EFFECTS:
;  949          2492  1 !       NONE
;  950          2493  1 !
;  951          2494  1 !--
;  952          2495  1
;  953          2496  2 BEGIN
;  954          2497  2 LOCAL
;  955          2498  2         ORIG_INBLK;             ! Original block
;  956          2499  2
;  957          2500  2
;  958          2501  2 ! Establish the new file position and test for end of file.
;  959          2502  2 !
;  960          2503  2 ORIG_INBLK = .JOUR_INBLK;                ! Save position to check for change
;  961          2504  2 IF .JOUR_REVERSE                         ! Reading in reverse?
;  962          2505  2 THEN
;  963          2506  3     BEGIN
;  964          2507  3     IF .JOUR_INBYTE EQL 0                ! Any more bytes in this block?
;  965          2508  3     THEN
;  966          2509  4         BEGIN                            ! No more bytes in current block
;  967          2510  4         JOUR_INBLK = .JOUR_INBLK - 1;    ! Decrease block number
;  968          2511  4         IF .JOUR_INBLK EQL 0             ! If block number has reached 0
;  969          2512  4             THEN RETURN SS$_ENDOFFILE;   ! then return end of file
;  970          2513  4         JOUR_INBYTE = 512;               ! Reset to byte following end of block
;  971          2514  3         END;
;  972          2515  3     JOUR_INBYTE = .JOUR_INBYTE - 1;      ! Decrease byte number
;  973          2516  3     END
;  974          2517  2 ELSE
;  975          2518  3     BEGIN
;  976          2519  3     JOUR_INBYTE = .JOUR_INBYTE + 1;      ! Increase byte number
;  977          2520  3     IF .JOUR_INBYTE GEQU 512             ! Any more bytes in this block?
;  978          2521  3     THEN
;  979          2522  4         BEGIN                            ! No more bytes in current block
```

```
;  980          2523  4          JOUR_INBLK = .JOUR_INBLK + 1;    ! Increase block number
;  981          2524  4          JOUR_INBYTE = 0;                 ! Reset to first byte of block
;  982          2525  3          END;
;  983          2526  3      IF .JOUR_INBLK GEQU .JOUR_EFBLK AND .JOUR_INBYTE GEQU .JOUR_FFBYTE
;  984          2527  3          THEN RETURN SS$_ENDOFFILE;       ! Check for end of file
;  985          2528  2      END;
;  986          2529  2
;  987          2530  2
;  988          2531  2  ! Read next block if necessary.
;  989          2532  2  !
;  990          2533  2  IF .ORIG_INBLK NEQ .JOUR_INBLK
;  991          2534  2  THEN
;  992          2535  3      BEGIN
;  993          2536  3      LOCAL
;  994          2537  3          STATUS,                          ! Status variable
;  995          2538  3          IOSB:           VECTOR[4,WORD],  ! I/O status block
;  996          2539  3          FAB:            REF BBLOCK;      ! Pointer to FAB for journal file
;  997          2540  3
;  998          2541  3      FAB = .QUAL[QUAL_JOUR_FC];
;  999        P 2542  3      STATUS = $QIOW(
; 1000        P 2543  3          FUNC=IO$_READVBLK,
; 1001        P 2544  3          CHAN=.FAB[FAB$L_STV],
; 1002        P 2545  3          IOSB=IOSB,
; 1003        P 2546  3          P1=.JOUR_BUFFER,
; 1004        P 2547  3          P2=512,
; 1005          2548  3          P3=.JOUR_INBLK);
; 1006          2549  3      IF .STATUS THEN STATUS = .IOSB[0];
; 1007          2550  3      IF NOT .STATUS
; 1008          2551  3      THEN
; 1009          2552  3          FILE_ERROR(
; 1010          2553  3              BACKUP$_READERR + STS$K_SEVERE,
; 1011          2554  3              .FAB,
; 1012          2555  3              .STATUS);
; 1013          2556  2      END;
; 1014          2557  2
; 1015          2558  2
; 1016          2559  2  ! Return the required byte, and success.
; 1017          2560  2  !
; 1018          2561  2  (.BUFFER)<0,8> = .JOUR_BUFFER[.JOUR_INBYTE];
; 1019          2562  2  SS$_NORMAL
; 1020          2563  1  END;


                                000C 00000 READ_BYTE:
                                                .WORD   Save R2,R3                              ; 2467
                53 00000000'    EF  9E 00002            MOVAB   JOUR_INBYTE, R3                 ;
                5E              08  C2 00009            SUBL2   #8, SP                          ;
                50        FA    A3  D0 0000C            MOVL    JOUR_INBLK, ORIG_INBLK          ; 2503
                12        05    A3  E9 00010            BLBC    JOUR_REVERSE, 2$                ; 2504
                                63  B5 00014            TSTW    JOUR_INBYTE                     ; 2507
                                0A  12 00016            BNEQ    1$                              ;
                          FA    A3  D7 00018            DECL    JOUR_INBLK                      ; 2510
                                24  13 0001B            BEQL    4$                              ; 2511
                63      0200    8F  B0 0001D            MOVW    #512, JOUR_INBYTE               ; 2513
```

```
                                63  B7 00022 1$:    DECW    JOUR_INBYTE                             ; 2515
                                21  11 00024        BRB     5$                                      ; 2504
                                63  B6 00026 2$:    INCW    JOUR_INBYTE                             ; 2519
                          0200  8F    63  B1 00028        CMPW    JOUR_INBYTE, #512                       ; 2520
                                05  1F 0002D        BLSSU   3$                                      ;
                                   FA   A3  D6 0002F        INCL    JOUR_INBLK                              ; 2523
                                63  B4 00032        CLRW    JOUR_INBYTE                             ; 2524
                           F6   A3    FA   A3  D1 00034 3$:    CMPL    JOUR_INBLK, JOUR_EFBLK                  ; 2526
                                0C  1F 00039        BLSSU   5$                                      ;
                           FE   A3         63  B1 0003B        CMPW    JOUR_INBYTE, JOUR_FFBYTE                ;
                                06  1F 0003F        BLSSU   5$                                      ;
                                50    0870  8F  3C 00041 4$:    MOVZWL  #2160, R0                       ; 2527
                                      04 00046        RET                                             ;
                           FA   A3         50  D1 00047 5$:    CMPL    ORIG_INBLK, JOUR_INBLK                  ; 2533
                                41  13 0004B        BEQL    7$                                      ;
                                52    FDD2  C3  D0 0004D        MOVL    QUAL+52, FAB                            ; 2541
                                7E  7C 00052        CLRQ    -(SP)                                   ; 2548
                                7E  D4 00054        CLRL    -(SP)                                   ;
                                   FA   A3  DD 00056        PUSHL   JOUR_INBLK                              ;
                                7E    0200  8F  3C 00059        MOVZWL  #512, -(SP)                     ;
                                   EA   A3  DD 0005E        PUSHL   JOUR_BUFFER                             ;
                                7E  7C 00061        CLRQ    -(SP)                                   ;
                                20  AE 9F 00063        PUSHAB  IOSB                                    ;
                                31  DD 00066        PUSHL   #49                                     ;
                                0C  A2 DD 00068        PUSHL   12(FAB)                                 ;
                                7E  D4 0006B        CLRL    -(SP)                                   ;
                00000000G  00   0C  FB 0006D        CALLS   #12, SYS$QIOW                           ;
                           06   50  E9 00074        BLBC    STATUS, 6$                              ; 2549
                           50   6E  3C 00077        MOVZWL  IOSB, STATUS                            ;
                           11   50  E8 0007A        BLBS    STATUS, 7$                              ; 2550
                                50  DD 0007D 6$:    PUSHL   STATUS                                  ; 2555
                                52  DD 0007F        PUSHL   FAB                                     ; 2554
                                      00000000G  8F  DD 00081        PUSHL   #BACKUP$_READERR+4              ; 2553
                     00000000G  00   03  FB 00087        CALLS   #3, FILE_ERROR                  ;
                           50   63  3C 0008E 7$:    MOVZWL  JOUR_INBYTE, R0                 ; 2561
                           50      EA   A3  C0 00091        ADDL2   JOUR_BUFFER, R0                 ;
                      04   BC   60  90 00095        MOVB    (R0), @BUFFER                   ;
                           50   01  D0 00099        MOVL    #1, R0                                  ; 2563
                                     04 0009C        RET                                             ;

; Routine Size:  157 bytes,    Routine Base:  CODE + 0531
```

```
;  1022     2564  1 %SBTTL 'READ_JOURNAL - read record from journal'
;  1023     2565  1 GLOBAL ROUTINE READ_JOURNAL(BUFFER)=
;  1024     2566  1
;  1025     2567  1 !++
;  1026     2568  1 !
;  1027     2569  1 ! FUNCTIONAL DESCRIPTION:
;  1028     2570  1 !       This routine reads a record from the journal file and validity checks
;  1029     2571  1 !       various fields.
;  1030     2572  1 !
;  1031     2573  1 ! INPUT PARAMETERS:
;  1032     2574  1 !       BUFFER          - Pointer to input buffer, where record is returned in
;  1033     2575  1 !                         ASCIC format.
;  1034     2576  1 !
;  1035     2577  1 ! IMPLICIT INPUTS:
;  1036     2578  1 !       NONE
;  1037     2579  1 !
;  1038     2580  1 ! OUTPUT PARAMETERS:
;  1039     2581  1 !       NONE
;  1040     2582  1 !
;  1041     2583  1 ! IMPLICIT OUTPUTS:
;  1042     2584  1 !       NONE
;  1043     2585  1 !
;  1044     2586  1 ! ROUTINE VALUE:
;  1045     2587  1 !       SS$_NORMAL      - Record successfully returned.
;  1046     2588  1 !       SS$_ENDOFFILE   - End (beginning) of file, no record returned.
;  1047     2589  1 !
;  1048     2590  1 ! SIDE EFFECTS:
;  1049     2591  1 !       NONE
;  1050     2592  1 !
;  1051     2593  1 !--
;  1052     2594  1
;  1053     2595  2 BEGIN
;  1054     2596  2 MAP
;  1055     2597  2         BUFFER:         REF BBLOCK;
;  1056     2598  2
;  1057     2599  2
;  1058     2600  2 ! Read byte count and convert to true byte count using XOR context.  If the
;  1059     2601  2 ! byte count is zero, in a well-formed journal file, it means that we have
;  1060     2602  2 ! reached the end of the data for a save set, and the next byte, if it
;  1061     2603  2 ! exists, is the initial byte count for the next save set.
;  1062     2604  2 !
;  1063     2605  2 DO
;  1064     2606  3     BEGIN
;  1065     2607  3     IF NOT READ_BYTE(BUFFER[BJL$B_SIZE]) THEN RETURN SS$_ENDOFFILE;
;  1066     2608  3     JOUR_COUNT = .JOUR_COUNT XOR .BUFFER[BJL$B_SIZE];
;  1067     2609  3     END
;  1068     2610  2 WHILE .JOUR_COUNT EQL 0;
;  1069     2611  2 BUFFER[BJL$B_SIZE] = .JOUR_COUNT;
;  1070     2612  2
;  1071     2613  2
;  1072     2614  2 ! Read remainder of record.
;  1073     2615  2 !
;  1074     2616  2 IF .JOUR_REVERSE
;  1075     2617  2 THEN
;  1076     2618  2     DECR I FROM .JOUR_COUNT TO 1 DO
;  1077     2619  3         BEGIN
;  1078     2620  3         IF NOT READ_BYTE(BUFFER[.I,0,8,0]) THEN SIGNAL(BACKUP$_INVBJLEOF);
```

```
;  1079     2621  3          END
;  1080     2622  2 ELSE
;  1081     2623  2      INCR I FROM 1 TO .JOUR_COUNT DO
;  1082     2624  3          BEGIN
;  1083     2625  3          IF NOT READ_BYTE(BUFFER[.I,0,8,0]) THEN SIGNAL(BACKUP$_INVBJLEOF);
;  1084     2626  2          END;
;  1085     2627  2
;  1086     2628  2
;  1087     2629  2 ! Do type-specific checking of the record.
;  1088     2630  2 !
;  1089     2631  2 CASE .BUFFER[BJL$B_TYPE] FROM BJL$K_STRUCLEV TO BJL$K_FILE OF
;  1090     2632  2      SET
;  1091     2633  2
;  1092     2634  2
;  1093     2635  2      [BJL$K_STRUCLEV]:
;  1094     2636  3          BEGIN
;  1095     2637  3
;  1096     2638  3          ! The record length must be exactly correct, and the structure level
;  1097     2639  3          ! must be correct.
;  1098     2640  3          !
;  1099     2641  3          IF .BUFFER[BJL$B_SIZE] NEQ BJL$C_STRUC_LEN+1
;  1100     2642  3              THEN SIGNAL(BACKUP$_INVBJLSIZ);
;  1101     2643  3          IF .BBLOCK[BUFFER[BJL$B_DATA], BJL$W_STRUCLEV] NEQ BJL$K_LEVEL1 AND
;  1102     2644  3             .BBLOCK[BUFFER[BJL$B_DATA], BJL$W_STRUCLEV] NEQ BJL$K_LEVEL2
;  1103     2645  3              THEN SIGNAL(BACKUP$_INVBJLSTR);
;  1104     2646  3          JOUR_STRUCT_LEV = .BBLOCK[BUFFER[BJL$B_DATA], BJL$W_STRUCLEV] ;
;  1105     2647  2          END;
;  1106     2648  2
;  1107     2649  2
;  1108     2650  2      [BJL$K_SSNAME]:
;  1109     2651  3          BEGIN
;  1110     2652  3
;  1111     2653  3          ! The record must be long enough to cover the creation
;  1112     2654  3          ! date, but not more than the maximum size.
;  1113     2655  3          !
;  1114     2656  3          IF .BUFFER[BJL$B_SIZE] LSSU $BYTEOFFSET(BJL$T_SSNAME)+1
;  1115     2657  3          OR .BUFFER[BJL$B_SIZE] GTRU BJL$C_SSNAME_LEN+1
;  1116     2658  3              THEN SIGNAL(BACKUP$_INVBJLSIZ);
;  1117     2659  2          END;
;  1118     2660  2
;  1119     2661  2
;  1120     2662  2      [BJL$K_VOLUME]:
;  1121     2663  3          BEGIN
;  1122     2664  3
;  1123     2665  3          ! The record must be exactly the correct size.
;  1124     2666  3          !
;  1125     2667  3          IF .BUFFER[BJL$B_SIZE] NEQ BJL$C_VOLUME_LEN+1
;  1126     2668  3              THEN SIGNAL(BACKUP$_INVBJLSIZ);
;  1127     2669  2          END;
;  1128     2670  2
;  1129     2671  2
;  1130     2672  2      [BJL$K_DIRECTORY]:
;  1131     2673  3          BEGIN
;  1132     2674  3          LOCAL
;  1133     2675  3              P,                          ! Temp
;  1134     2676  3              LEADING_NULLS;              ! Count of leading nulls in XOR
;  1135     2677  3
```

```
; 1136     2678  3
; 1137     2679  3          ! The record must be long enough to cover the leading-null count,
; 1138     2680  3          ! but not longer than the maximum.  The count of real data plus the
; 1139     2681  3          ! count of leading nulls must not exceed the maximum.
; 1140     2682  3          !
; 1141     2683  3          IF .BUFFER[BJL$B_SIZE] LSSU 2
; 1142     2684  3          OR .BUFFER[BJL$B_SIZE] GTRU BJL$C_DIR_LEN+1
; 1143     2685  3          OR .BUFFER[BJL$B_SIZE]+.BUFFER[BJ[$B_DATA] GTRU BJL$C_DIR_LEN+2
; 1144     2686  3              THEN SIGNAL(BACKUP$_INVBJLSIZ);
; 1145     2687  3
; 1146     2688  3
; 1147     2689  3          ! Pick up count of implicit leading nulls.
; 1148     2690  3          !
; 1149     2691  3          LEADING_NULLS = .BUFFER[BJL$B_DATA];
; 1150     2692  3
; 1151     2693  3
; 1152     2694  3          ! Compute XOR of bytes in record and bytes of context to develop
; 1153     2695  3          ! true value.
; 1154     2696  3          !
; 1155     2697  3          INCR I FROM $BYTEOFFSET(BJL$B_DATA)+1 TO .JOUR_COUNT DO
; 1156     2698  4              BEGIN
; 1157     2699  4              BUFFER[.I,0,8,0] = .BUFFER[.I,0,8,0] XOR
; 1158     2700  4                  .JOUR_DIR[.I + .LEADING_NULLS - $BYTEOFFSET(BJL$B_DATA)];
; 1159     2701  3              END;
; 1160     2702  3
; 1161     2703  3
; 1162     2704  3          ! Move non-null data in record to appropriate offset.
; 1163     2705  3          !
; 1164     2706  3          IF .LEADING_NULLS NEQ 1
; 1165     2707  3          THEN
; 1166     2708  3              CH$MOVE(
; 1167     2709  3                  .JOUR_COUNT-2,
; 1168     2710  3                  BUFFER[BJL$B_DATA]+1,
; 1169     2711  3                  BUFFER[BJL$B_DATA]+.LEADING_NULLS);
; 1170     2712  3
; 1171     2713  3
; 1172     2714  3          ! Establish leading portion of directory string from context value.
; 1173     2715  3          !
; 1174     2716  3          IF .LEADING_NULLS NEQ 0
; 1175     2717  3          THEN
; 1176     2718  3              CH$MOVE(
; 1177     2719  3                  .LEADING_NULLS,
; 1178     2720  3                  JOUR_DIR[1],
; 1179     2721  3                  BUFFER[BJL$B_DATA]);
; 1180     2722  3
; 1181     2723  3
; 1182     2724  3          ! Establish trailing portion of directory string
; 1183     2725  3          ! from context value.
; 1184     2726  3          !
; 1185     2727  3          CH$MOVE(
; 1186     2728  3              BJL$C_DIR_LEN - (.LEADING_NULLS + .JOUR_COUNT - 2),
; 1187     2729  3              JOUR_DIR[.LEADING_NULLS + .JOUR_COUNT - 1],
; 1188     2730  3              BUFFER[BJL$B_DATA] + .LEADING_NULLS + .JOUR_COUNT - 2);
; 1189     2731  3
; 1190     2732  3
; 1191     2733  3          ! Save true value for next directory record.
; 1192     2734  3          !
```

```
;  1193          2735  3          CH$MOVE(BJL$C_DIR_LEN, BUFFER[BJL$B_DATA], JOUR_DIR[1]);
;  1194          2736  3
;  1195          2737  3
;  1196          2738  3          ! Strip trailing zero bytes from true value to develop actual
;  1197          2739  3          ! record.  The value is printable ASCII and thus can contain
;  1198          2740  3          ! no embedded nulls.
;  1199          2741  3          !
;  1200          2742  3          BUFFER[BJL$B_SIZE] = BJL$C_DIR_LEN + 1;          ! Assume no non-null
;  1201          2743  3          P = CH$FIND_CH(BJL$C_DIR_LEN, BUFFER[BJL$B_DATA], 0);
;  1202          2744  3          IF .P NEQ 0 THEN BUFFER[BJL$B_SIZE] = .P - BUFFER[BJL$B_TYPE];
;  1203          2745  2          END;
;  1204          2746  2
;  1205          2747  2
;  1206          2748  2      [BJL$K_FILE]:
;  1207          2749  3          BEGIN
;  1208          2750  3
;  1209          2751  3          ! The record must not exceed the maximum length.
;  1210          2752  3          !
;  1211          2753  3          IF .BUFFER[BJL$B_SIZE] GTRU BJL$C_FILE_LEN+1
;  1212          2754  3              THEN SIGNAL(BACKUP$_INVBJLSIZ);
;  1213          2755  2          END;
;  1214          2756  2
;  1215          2757  2
;  1216          2758  2      [OUTRANGE]:
;  1217          2759  2          SIGNAL(BACKUP$_INVBJLTYP);       ! Undefined type code
;  1218          2760  2
;  1219          2761  2
;  1220          2762  2      TES;
;  1221          2763  2
;  1222          2764  2
;  1223          2765  2  ! Indicate success.
;  1224          2766  2  !
;  1225          2767  2  SS$_NORMAL
;  1226          2768  1  END;


                                    0FFC 00000          .ENTRY  READ_JOURNAL, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 2565
                                                                R10,R11
                5B 00000000G   00    9E 00002          MOVAB   LIB$SIGNAL, R11                               ;
                5A 00000000'   EF    9E 00009          MOVAB   JOUR_COUNT, R10                               ;
                56        04   AC    D0 00010          MOVL    BUFFER, R6                                    ; 2607
                                   56    DD 00014 1$:    PUSHL   R6                                            ;
            FF48   CF             01    FB 00016          CALLS   #1, READ_BYTE                                 ;
                06                 50    E8 0001B          BLBS    R0, 2$                                        ;
                50      08D0       8F    3C 0001E          MOVZWL  #2160, R0                                     ;
                                         04 00023          RET                                                   ;
                6A                 66    8C 00024 2$:      XORB2   (R6), JOUR_COUNT                              ; 2608
                53                 6A    9A 00027          MOVZBL  JOUR_COUNT, R3                                ; 2610
                                   E8    13 0002A          BEQL    1$                                            ;
                66                 53    90 0002C          MOVB    R3, (R6)                                      ; 2611
                1F         01      AA    E9 0002F          BLBC    JOUR_REVERSE, 5$                              ; 2616
                52         01      A3    9E 00033          MOVAB   1(R3), I                                      ; 2618
                                   14    11 00037          BRB     4$                                            ;
                                 6246    9F 00039 3$:      PUSHAB  (I)[R6]                                       ; 2620
```

```
                                FF22    CF              01    FB 0003C          CALLS   #1, READ_BYTE                           ;
                                        09              50    E8 00041          BLBS    R0, 4$                                  ;
                                    00000000G           8F    DD 0C044          PUSHL   #BACKUP$_INVBJLEOF                      ;
                                        6B              01    FB 0004A          CALLS   #1, LIB$SIGNAL                          ;
                                        E9              52    F5 0004D 4$:      SOBGTR  I, 3$                                   ; 2618
                                                        1C    11 00050          BRB     8$                                      ;
                                                        52    D4 00052 5$:      CLRL    I                                       ; 2623
                                                        14    11 00054          BRB     7$                                      ;
                                                      6246    9F 00056 6$:      PUSHAB  (I)[R6]                                 ; 2625
                                FF05    CF              01    FB 00059          CALLS   #1, READ_BYTE                           ;
                                        09              50    E8 0005E          BLBS    R0, 7$                                  ;
                                    00000000G           8F    DD 00061          PUSHL   #BACKUP$_INVBJLEOF                      ;
                                        6B              01    FB 00067          CALLS   #1, LIB$SIGNAL                          ;
                                    E8                  52    53 F3 0006A 7$:   AOBLEQ  R3, I, 6$                              ; 2623
                                    04                  00    01 A6 8F 0006E 8$:CASEB   1(R6), #0, #4                           ; 2631
                0055                004D                0042      0013    00073 9$:   .WORD  10$-9$,-                           ;
                                                              00E9    0007B             14$-9$,-                                ;
                                                                                        15$-9$,-                                ;
                                                                                        17$-9$,-                                ;
                                                                                        25$-9$                                  ;
                                    00000000G           8F    DD 0007D          PUSHL   #BACKUP$_INVBJLTYP                      ; 2759
                                                      00E2    31 00083          BRW     28$                                     ;
                                        03              66    91 00086 10$:     CMPB    (R6), #3                                ; 2641
                                                        09    13 00089          BEQL    11$                                     ;
                                    00000000G           8F    DD 0008B          PUSHL   #BACKUP$_INVBJLSIZ                      ; 2642
                                        6B              01    FB 00091          CALLS   #1, LIB$SIGNAL                          ;
                                    0101    8F      02      A6    B1 00094 11$:     CMPW    2(R6), #257                             ; 2643
                                                        11    13 0009A          BEQL    12$                                     ;
                                    0102    8F      02      A6    B1 0009C          CMPW    2(R6), #258                             ; 2644
                                                        09    13 000A2          BEQL    12$                                     ;
                                    00000000G           8F    DD 000A4          PUSHL   #BACKUP$_INVBJLSTR                      ; 2645
                                        6B              01    FB 000AA          CALLS   #1, LIB$SIGNAL                          ;
                                FE      AA      02      A6    B0 000AD 12$:     MOVW    2(R6), JOUR_STRUCT_LEV                  ; 2646
                                                      00B6    31 000B2 13$:     BRW     29$                                     ; 2631
                                        09              66    91 000B5 14$:     CMPB    (R6), #9                                ; 2656
                                                        0B    1F 000B8          BLSSU   16$                                     ;
                                        29              66    91 000BA          CMPB    (R6), #41                               ; 2657
                                                      00A0    31 000BD          BRW     26$                                     ;
                                        0F              66    91 000C0 15$:     CMPB    (R6), #15                               ; 2667
                                                        ED    13 000C3          BEQL    13$                                     ;
                                                      009A    31 000C5 16$:     BRW     27$                                     ; 2668
                                        02              66    91 000C8 17$:     CMPB    (R6), #2                                ; 2684
                                                        13    1F 000CB          BLSSU   18$                                     ;
                                        50              66    9A 000CD          MOVZBL  (R6), R0                                ; 2685
                                        51      02      A6    9A 000D0          MOVZBL  2(R6), R1                               ;
                                        50              51    C0 000D4          ADDL2   R1, R0                                  ;
                            00000101    8F              50    D1 000D7          CMPL    R0, #257                                ;
                                                        09    1B 000DE          BLEQU   19$                                     ;
                                    00000000G           8F    DD 000E0 18$:     PUSHL   #BACKUP$_INVBJLSIZ                      ; 2686
                                        6B              01    FB 000E6          CALLS   #1, LIB$SIGNAL                          ;
                                        58      02      A6    9A 000E9 19$:     MOVZBL  2(R6), LEADING_NULLS                    ; 2691
                                        59              6A    9A 000ED          MOVZBL  JOUR_COUNT, R9                          ; 2697
                                        57      EA      AA    D0 000F0          MOVL    JOUR_DIR, R7                            ; 2700
                                        50              02    D0 000F4          MOVL    #2, I                                   ;
                                                        0A    11 000F7          BRB     21$                                     ;
                        51              50              58    C1 000F9 20$:     ADDL3   LEADING_NULLS, I, R1                    ;
                                    6046        FE  A147    8C 000FD          XORB2   -2(R1)[R7], (I)[R6]                     ;
```

```
                F2              50              59  F3 00103 21$:   AOBLEQ  R9, I, 20$                                ; 2697
                                01              58  D1 C0107        CMPL    LEADING_NULLS, #1                         ; 2706
                                                0B  13 0010A        BEQL    22$                                       ;
                                50        FE    A9  9E 0010C        MOVAB   -2(R9), R0                                ; 2709
            02 A846         03    A6          50  28 00110        MOVC3   R0, 3(R6), 2(LEADING_NULLS)[R6]           ; 2711
                                                58  D5 00117 22$:   TSTL    LEADING_NULLS                             ; 2716
                                                06  13 00119        BEQL    23$                                       ;
            02    A6        01    A7          58  28 0011B        MOVC3   LEADING_NULLS, 1(R7), 2(R6)               ; 2721
                                52      FEFF C948 9E 00121 23$:   MOVAB   -257(R9)[LEADING_NULLS], R2               ; 2728
                                52              52  CE 00127        MNEGL   R2, R2                                    ;
                51              58              59  C1 0012A        ADDL3   R9, LEADING_NULLS, R1                     ; 2729
                50              56              58  C1 0012E        ADDL3   LEADING_NULLS, R6, R0                     ; 2730
              6940        FF A147               52  28 00132        MOVC3   R2, -1(R1)[R7], (R9)[R0]                  ;
            01    A7        02    A6      00FF  8F  28 00139        MOVC3   #255, 2(R6), 1(R7)                        ; 2735
                                                66  94 00141        CLRB    (R6)                                      ; 2742
            02    A6      00FF    8F            00  3A 00143        LOCC    #0, #255, 2(R6)                           ; 2743
                                                02  12 0014A        BNEQ    24$                                       ;
                                                51  D4 0014C        CLRL    R1                                        ;
                                                51  D5 0014E 24$:   TSTL    P                                         ; 2744
                                                19  13 00150        BEQL    29$                                       ;
                                50        01    A6  9E 00152        MOVAB   1(R6), R0                                 ;
                66              51              50  83 00156        SUBB3   R0, P, (R6)                               ;
                                                0F  11 0015A        BRB     29$                                       ; 2631
                        85      8F              66  91 0015C 25$:   CMPB    (R6), #133                                ; 2753
                                                09  1B 00160 26$:   BLEQU   29$                                       ;
                                00000000G       8F  DD 00162 27$:   PUSHL   #BACKUP$_INVBJLSIZ                        ; 2754
                            6B                  01  FB 00168 28$:   CALLS   #1, LIB$SIGNAL                          ;
                            50                  01  D0 0016B 29$:   MOVL    #1, R0                                    ; 2768
                                                04  0016E        RET                                               ;
```

; Routine Size: 367 bytes,   Routine Base: CODE + 05CE

```
; 1228          2769  1 END
; 1229          2770  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  COMMON                         2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           1853  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        53         0      1000          00:02.2


;                                       COMMAND QUALIFIERS

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

; Size:          1841 code + 2136 data bytes
; Run Time:         00:45.5
; Elapsed Time:     02:34.4
; Lines/CPU Min:     3655
; Lexemes/CPU-Min: 32145
; Memory Used:  370 pages
; Compilation Complete
```

LIST
LIS

JOURNAL
LIS

FASTSCAN
LIS