

```

BBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBBBB  AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBBBB  AAAAAAAAAA      CCCCCCCCCCCCCC      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
BBB      BBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBB      BBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP      PPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBB      BBB  AAAAAAAAAAAAAAAAAA  CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB  AAAAAAAAAAAAAAAAAA  CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB  AAAAAAAAAAAAAAAAAA  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
BBB      BBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBB      BBB  AAA      AAA      CCC      KKK      KKK      UUU      UUU      PPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCCCCCCCCCCCCC  KKK      KKK      UUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCCCCCCCCCCCCC  KKK      KKK      UUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBBBBBB  AAA      AAA      CCCCCCCCCCCCCC  KKK      KKK      UUUUUUUUUUUUUUU  PPP

```

```

LL          IIIIII  SSSSSSSS  TTTTTTTTTT  JJ  000000  UU  UU  RRRRRRRR
LL          IIIIII  SSSSSSSS  TTTTTTTTTT  JJ  000000  UU  UU  RRRRRRRR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SSSSSS     TT          JJ  00    00  UU  UU  RRRRRRRR
LL          II      SSSSSS     TT          JJ  00    00  UU  UU  RRRRRRRR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LL          II      SS          TT          JJ  00    00  UU  UU  RR      RR
LLLLLLLLLLL IIIIII  SSSSSSSS  TT          JJJJJJ  000000  UUUUUUUUU  RR      RR
LLLLLLLLLLL IIIIII  SSSSSSSS  TT          JJJJJJ  000000  UUUUUUUUU  RR      RR

```

```

LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
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
LLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE LISTJOUR (%TITLE 'List a journal file'
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 list a 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: 10-Apr-1981
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 V03-003 LY0490 Larry Yetto 21-MAY-1984 16:12
47 0047 1 Modify code to allow long directory names to be output.
48 0048 1
49 0049 1 V03-002 LY0464 Larry Yetto 1-FEB-1984 10:39
50 0050 1 Add support for journal file structure level 2. This new
51 0051 1 format add a flag longword to each file entry to enable us
52 0052 1 to flag the fact that only a file header was saved for a
53 0053 1 particular file.
54 0054 1
55 0055 1 V03-001 ACG0313 Andrew C. Goldstein, 12-Feb-1983 16:23
56 0056 1 Add routine subtitles
57 0057 1

```

LISTJOUR
V04-000

List a journal file

L 1
16-Sep-1984 00:10:02
14-Sep-1984 11:53:55

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]LISTJOUR.B32;1

Page 2
(1)

:	58	0058	1	:	V02-002	MLJ0036	Martin L. Jack, 28-Aug-1981	17:38
:	59	0059	1	:			Change calling sequence to MATCH routine.	
:	60	0060	1	:				
:	61	0061	1	:	V02-001	MLJ0025	Martin L. Jack, 8-May-1981	13:36
:	62	0062	1	:			Implement selective journal listing.	
:	63	0063	1	:				
:	64	0064	1	:				
:	65	0065	1	:*				

```

: 67 0066 1 REQUIRE 'SRC$:COMMON';
: 68 1172 1 LIBRARY 'SYSS$LIBRARY:STARLET';
: 69 1173 1 REQUIRE 'LIB$:BACKDEF';
: 70 1623 1
: 71 1624 1
: 72 1625 1 LINKAGE
: 73 1626 1 L_PS = CALL: GLOBAL(P$=11);
: 74 1627 1
: 75 1628 1
: 76 1629 1 MACRO
: 77 1630 1 L_DECL = EXTERNAL REGISTER P$ = 11: REF VECTOR %;
: 78 1631 1
: 79 1632 1
: 80 1633 1 FORWARD ROUTINE
: 81 1634 1 SELECT_FILE, ! Evaluate selection criteria
: 82 1635 1 LIST_JOURNAL: NOVALUE; ! Driver for journal file listing
: 83 1636 1
: 84 1637 1
: 85 1638 1 EXTERNAL ROUTINE
: 86 1639 1 MATCH, ! Match file specifications
: 87 1640 1 OPEN_JOURNAL: NOVALUE, ! Open a journal file
: 88 1641 1 READ_JOURNAL, ! Read a journal record
: 89 1642 1 CLOSE_JOURNAL: NOVALUE, ! Close a journal file
: 90 1643 1 LIST_FAO: L_PS NOVALUE, ! Add information to line buffer
: 91 1644 1 LIST_EOL: L_PS NOVALUE; ! Write line buffer to listing file
: 92 1645 1
: 93 1646 1
: 94 1647 1 G$DEFINE(); ! Define global common area
: 95 1648 1
: 96 1649 1
: 97 1650 1 MACRO
: 98 M 1651 1 FAO_(A)=
: 99 M 1652 1 -LIST FAO(
: 100 M 1653 1 OPLIT BYTE (%ASCIC A)
: 101 1654 1 %IF NOT %NULL(%REMAINING) %THEN , %F1 %REMAINING) %,
: 102 1655 1
: 103 1656 1
: 104 M 1657 1 EOL_(A)=
: 105 1658 1 -LIST_EOL() %;
: 106 1659 1
: 107 1660 1
: 108 1661 1 MACRO
: 109 1662 1 -LIST_DESC= P$[0] %, ! Descriptor for buffer
: 110 1663 1 -LIST_BUFFER= P$[2] %, ! Listing buffer
: 111 1664 1 [LIST_DESC,N]= VECTOR[_LIST_DESC,N] %;
: 112 1665 1
: 113 1666 1
: 114 1667 1 LITERAL
: 115 1668 1 P$SIZE= 2 + CH$ALLOCATION(LIST_SIZE);
```

```
117 1669 1 %SBTTL 'SELECT_FILE - select file from journal'
118 1670 1 ROUTINE SELECT_FILE(NAME,CREDATE)=
119 1671 1
120 1672 1 !++
121 1673 1
122 1674 1 ! FUNCTIONAL DESCRIPTION:
123 1675 1 ! This routine evaluates the selection criteria for a file specification
124 1676 1 ! retrieved from a journal file.
125 1677 1
126 1678 1 ! INPUT PARAMETERS:
127 1679 1 ! NAME - Descriptor for file specification "[dir]nam.typ;ver".
128 1680 1 ! CREDATE - Pointer to quadword time value.
129 1681 1
130 1682 1 ! IMPLICIT INPUTS:
131 1683 1 ! NONE
132 1684 1
133 1685 1 ! OUTPUT PARAMETERS:
134 1686 1 ! NONE
135 1687 1
136 1688 1 ! IMPLICIT OUTPUTS:
137 1689 1 ! NONE
138 1690 1
139 1691 1 ! ROUTINE VALUE:
140 1692 1 ! True if the file specification is selected, false otherwise.
141 1693 1
142 1694 1 ! SIDE EFFECTS:
143 1695 1 ! NONE
144 1696 1
145 1697 1 ! --
146 1698 1
147 1699 2 BEGIN
148 1700 2 MAP
149 1701 2 CREDATE: REF VECTOR;
150 1702 2
151 1703 2
152 1704 2 ! BEFORE
153 1705 2
154 1706 2 IF .QUAL[QUAL_BEFO] AND NOT .QUAL[QUAL_BEFO_BACK]
155 1707 2 THEN
156 1708 3 BEGIN
157 1709 3 IF .CREDATE[1] GTRU .VECTOR[QUAL[QUAL_BEFO_VALUE], 1] THEN RETURN FALSE;
158 1710 3 IF .CREDATE[1] EQLU .VECTOR[QUAL[QUAL_BEFO_VALUE], 1] AND .CREDATE[0] GEQU .VECTOR[QUAL[QUAL_BEFO_VALUE]
159 1711 2 END;
160 1712 2
161 1713 2
162 1714 2 ! SINCE
163 1715 2
164 1716 2 IF .QUAL[QUAL_SINC] AND NOT .QUAL[QUAL_SINC_BACK]
165 1717 2 THEN
166 1718 3 BEGIN
167 1719 3 IF .CREDATE[1] LSSU .VECTOR[QUAL[QUAL_SINC_VALUE], 1] THEN RETURN FALSE;
168 1720 3 IF .CREDATE[1] EQLU .VECTOR[QUAL[QUAL_SINC_VALUE], 1] AND .CREDATE[0] LSSU .VECTOR[QUAL[QUAL_SINC_VALUE]
169 1721 2 END;
170 1722 2
171 1723 2
172 1724 2 ! EXCLUDE
173 1725 2 !
```

```

: 174      1726  2 IF .QUAL[QUAL_EXCL]
: 175      1727  2 THEN
: 176      1728  3 BEGIN
: 177      1729  3 LOCAL
: 178      1730  3 EXCL:          REF BBLOCK;
: 179      1731  3
: 180      1732  3 EXCL = .QUAL[QUAL_EXCL_LIST];
: 181      1733  3 WHILE .EXCL NEQ 0 DO
: 182      1734  4 BEGIN
: 183      1735  4 IF MATCH(.NAME, EXCL[QUAL_EXCL_DESC]) THEN RETURN FALSE;
: 184      1736  4 EXCL = .EXCL[QUAL_NEXT];
: 185      1737  3 END;
: 186      1738  2 END;
: 187      1739  2
: 188      1740  2 ! SELECT
: 189      1741  2 !
: 190      1742  2 !
: 191      1743  2 IF .QUAL[QUAL_SELE]
: 192      1744  2 THEN
: 193      1745  3 BEGIN
: 194      1746  3 LOCAL
: 195      1747  3 SELE:          REF BBLOCK;
: 196      1748  3
: 197      1749  3 SELE = .QUAL[QUAL_SELE_LIST];
: 198      1750  3 IF
: 199      1751  4 BEGIN
: 200      1752  4 WHILE .SELE NEQ 0 DO
: 201      1753  5 BEGIN
: 202      1754  5 IF MATCH(.NAME, SELE[QUAL_SELE_DESC]) THEN EXITLOOP FALSE;
: 203      1755  5 SELE = .SELE[QUAL_NEXT];
: 204      1756  5 END
: 205      1757  4 END
: 206      1758  3 THEN
: 207      1759  3 RETURN FALSE;
: 208      1760  2 END;
: 209      1761  2
: 210      1762  2
: 211      1763  2 TRUE
: 212      1764  1 END;

```

```

.TITLE LISTJOUR List a journal file
.IDENT \V04-000\
.PSECT COMMON,NOEXE, OVR,2

```

```

00000 GLOBAL_BASE:
      .BLKB 0
^0000 FREE_LIST:
      .BLKB 8
00008 INPUT_WAIT:
      .BLKB 8
00010 REREAD_WAIT:
      .BLKB 8
00018 OUTPUT_WAIT:
      .BLKB 8
00020 JPI_JIC: .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_XOR_SIZE:	.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:	
	.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_CRDATE:	
	.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:	

002AE	OUTPUT_PADDING:	.BLKB	2
002B0	OUTPUT_FAB:	.BLKB	2
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_DEVGEO:	.BLKB	8
002D4	OUTPUT_ATTBUF:	.BLKB	144
00364	OUTPUT_END:	.BLKB	0
00364	LIST_TOTFILES:	.BLKB	4
00368	LIST_TOTSIZE:	.BLKB	4
0036C	VERIFY_FAB:	.BLKB	4
00370	VERIFY_USE_COUNT:	.BLKB	4
00374	VERIFY_QUAL:	.BLKB	4
00378	COMPARE_BCB:	.BLKB	4
0037C	FAST_BUFFER:	.BLKB	4
00380	FAST_BUFFER_SIZE:	.BLKB	4
00384	FAST_RVN:	.BLKB	1
00385	FAST_PADDING:	.BLKB	1
00386	DIR_VERLIMIT:	.BLKB	2
00388	FAST_VOL_BEG:	.BLKB	0
00388	FAST_IMAP_SIZE:	.BLKB	4
0038C	FAST_IMAP:	.BLKB	4
00390	FAST_HDR_OFFSET:	.BLKB	4
00394	FAST_BOOT_LBN:	.BLKB	4
00398	FAST_VOL_END:	.BLKB	0

00398	JOUR_BUFFER:	
	.BLKB	4
0039C	JOUR_DIR:	
	.BLKB	4
003A0	JOUR_HIBLK:	
	.BLKB	4
003A4	JOUR_EFBLK:	
	.BLKB	4
003A8	JOUR_INBLK:	
	.BLKB	4
003AC	JOUR_FFBYTE:	
	.BLKB	2
003AE	JOUR_INBYTE:	
	.BLKB	2
003B0	JOUR_STRUCT_LEV:	
	.BLKB	2
003B2	JOUR_COUNT:	
	.BLKB	1
003B3	JOUR_REVERSE:	
	.BLKB	1
003B4	JOUR_EXSZ:	
	.BLKB	2
003B6	JOUR_PADDING:	
	.BLKB	2
003B8	CHKPT_HIGH_SP:	
	.BLKB	4
003BC	CHKPT_LOW_SP:	
	.BLKB	4
003C0	CHKPT_STACK:	
	.BLKB	4
003C4	CHKPT_VARS:	
	.BLKB	4
003C8	CHKPT_STATUS:	
	.BLKB	4
003CC	DIR_BEG:	.BLKB 0
003CC	DIR_CHAN:	
	.BLKB	4
003D0	DIR_NAM:	.BLKB 4
003D4	DIR_DEV_DESC:	
	.BLKB	4
003D8	DIR_SEL_DIR:	
	.BLKB	8
003E0	DIR_SEL_NTV:	
	.BLKB	8
003E8	DIR_STRUCLEV:	
	.BLKB	1
003E9	DIR_LEVELS:	
	.BLKB	1
003EA	DIR_FLAGS:	
	.BLKB	1
003EB	DIR_STATUS:	
	.BLKB	1
003EC	DIR_STRING:	
	.BLKB	320
0052C	DIR_STACK:	
	.BLKB	612
00790	DIR_SP:	.BLKB 4

```

00794 DIR_SEL_LATEST:
      .BLKB 4
00798 DIR_END: .BLKB 0
00798 DIR_SCANLIMIT:
      .BLKB 36
007BC INPUT_MTL:
      .BLKB 4
007C0 OUTPUT_MTL:
      .BLKB 4
007C4 CURRENT_MTL:
      .BLKB 4
007C8 CURRENT_VCB:
      .BLKB 4
007CC CURRENT_WCB:
      .BLKB 4
007D0 ACL_FIB_DESCR:
      .BLKB 8
007D8 ACL_FIB: .BLKB 64
00818 ACL_LENGTH:
      .BLKB 4
0081C ACL_BUFFER:
      .BLKB 4
00820 CRYP_IN_CONTEXT:
      .BLKB 4
00824 CRYP_OU_CONTEXT:
      .BLKB 4
00828 CRYP_DA_CONTEXT:
      .BLKB 4
0082C CRYP_DATA_ENCIV:
      .BLKB 8
00834 CRYP_DATA_CODE:
      .BLKB 4
00838 CRYP_DATA_KEY:
      .BLKB 8
00840 CRYP_DATA_IV:
      .BLKB 8
00848 CRYP_DATA_CKSM:
      .BLKB 4

      .EXTRN MATCH, OPEN_JOURNAL
      .EXTRN READ_JOURNAL, CLOSE_JOURNAL
      .EXTRN LIST_FAO, LIST_EOL

      .PSECT CODE, NOWRT, 2
  
```

```

17
13
      54 00000000G 00 9E 00002      .WORD Save R2,R3,R4
      53 00000000' EF 9E 00009      MOVAB MATCH, R4
      63          02 E1 00010      MOVAB QUAL+8, R3
      63          03 E0 00014      BBC #2, QUAL+8, 1$
      50          08 AC D0 00018      BBS #3, QUAL+8, 1$
      OC A3 04 A0 D1 0001C      MOVL CREDATE, R0
      5D 1A 00021      CMPL 4(R0), QUAL+20
      06 12 00023      BGTRU 6$
      08 A3 60 D1 00025      BNEQ 1$
      55 1E 00029      CMPL (R0), QUAL+16
      BGEQU 6$
  
```

: 1670
: 1706
: 1709
: 1710
:

LISTJOUR
V04-000

List a journal file
SELECT_FILE - select file from journal

1 2
16-Sep-1984 00:10:02
14-Sep-1984 11:53:55

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]LISTJOUR.B32;1

Page 12
(3)

18	05	A3	04	E1	0002B	1\$:	BBC	#4, QUAL+13, 2\$	1716	
13	05	A3	05	E0	00030		BBS	#5, QUAL+13, 2\$	1719	
		50	08	AC	D0	00035	MOVL	CRÉDATE, R0		
	1C	A3	04	A0	D1	00039	CMPL	4(R0), QUAL+36		
				40	1F	0003E	BLSSU	6\$		
				06	12	00040	BNEQ	2\$	1720	
	18	A3		60	D1	00042	CMPL	(R0), QUAL+32		
				38	1F	00046	BLSSU	6\$		
17	01	A3		04	E1	00048	2\$:	BBC	#4, QUAL+9, 4\$	1726
		52	28	A3	D0	0004D	MOVL	QUAL+48, EXCL	1732	
				11	13	00051	3\$:	BEQL	4\$	1733
			04	A2	9F	00053	PUSHAB	4(EXCL)	1735	
			04	AC	DD	00056	PUSHL	NAME		
	64			02	FB	00059	CALLS	#2, MATCH		
	21			50	EB	0005C	BLBS	R0, 5\$		
	52			62	D0	0005F	MOVL	(EXCL), EXCL	1736	
				ED	11	00062	BRB	3\$	1733	
1A	05	A3		03	E1	00064	4\$:	BBC	#3, QUAL+13, 7\$	1743
		52	3C	A3	D0	00069	MOVL	QUAL+68, SELE	1749	
				11	13	0006D	5\$:	BEQL	6\$	1752
			04	A2	9F	0006F	PUSHAB	4(SELE)	1754	
			04	AC	DD	00072	PUSHL	NAME		
	64			02	FB	00075	CALLS	#2, MATCH		
	08			50	EB	00078	BLBS	R0, 7\$		
	52			62	D0	0007B	MOVL	(SELE), SELE	1755	
				ED	11	0007E	BRB	5\$	1752	
				50	D4	00080	6\$:	CLRL	R0	1759
				04	00082		RET			
		50		01	D0	00083	7\$:	MOVL	#1, R0	1764
				04	00086		RET			

; Routine Size: 135 bytes, Routine Base: CODE + 0000

```
.. 214 1765 1 %SBTTL 'LIST JOURNAL - main journal list routine'
.. 215 1766 1 GLOBAL ROUTINE LIST_JOURNAL: NOVALUE=
.. 216 1767 1
.. 217 1768 1 |**
.. 218 1769 1 |
.. 219 1770 1 | FUNCTIONAL DESCRIPTION:
.. 220 1771 1 | This routine is the driver for listing of a journal file.
.. 221 1772 1 |
.. 222 1773 1 | INPUT PARAMETERS:
.. 223 1774 1 | NONE
.. 224 1775 1 |
.. 225 1776 1 | IMPLICIT INPUTS:
.. 226 1777 1 | NONE
.. 227 1778 1 |
.. 228 1779 1 | OUTPUT PARAMETERS:
.. 229 1780 1 | NONE
.. 230 1781 1 |
.. 231 1782 1 | IMPLICIT OUTPUTS:
.. 232 1783 1 | NONE
.. 233 1784 1 |
.. 234 1785 1 | ROUTINE VALUE:
.. 235 1786 1 | NONE
.. 236 1787 1 |
.. 237 1788 1 | SIDE EFFECTS:
.. 238 1789 1 | NONE
.. 239 1790 1 |
.. 240 1791 1 | --
.. 241 1792 1 |
.. 242 1793 2 BEGIN
.. 243 1794 2 LITERAL
.. 244 1795 2 STR_LENGTH = %CHARCOUNT(' file header only'),
.. 245 1796 2 BUFFER_LENGTH = BJL$C_DIR_LEN + BJL$S_FILENAME + STR_LENGTH ;
.. 246 1797 2
.. 247 1798 2
.. 248 1799 2 LOCAL
.. 249 1800 2 STR: LONG INITIAL(UPLIT BYTE (' file header only')),
.. 250 1801 2 FAB: REF BBLOCK, ! Pointer to FAB for journal file
.. 251 1802 2 BUFFER: BBLOCK[BUFFER_LENGTH], ! Record buffer
.. 252 1803 2 SSNAME_LEN: LONG INITIAL(0), ! Length of SSNAME
.. 253 1804 2 SSNAME: BBLOCK[BJL$C_SSNAME_LEN], ! SSNAME buffer
.. 254 1805 2 VOLUME: BBLOCK[BJL$C_VOLUME_LEN], ! VOLUME buffer
.. 255 1806 2 DIR_LEN: LONG INITIAL(0), ! Length of DIRECTORY
.. 256 1807 2 DIRECTORY: BBLOCK[BJL$C_DIR_LEN], ! DIRECTORY buffer
.. 257 1808 2 CHANGE, ! True if SSNAME or VOLUME changed
.. 258 1809 2 PSAREA: VECTOR[P$SIZE]; ! Impure area
.. 259 1810 2 GLOBAL REGISTER
.. 260 1811 2 P$ = 11: REF VECTOR; ! Impure area base register
.. 261 1812 2
.. 262 1813 2
.. 263 1814 2 ! Initialize impure area.
.. 264 1815 2 !
.. 265 1816 2 P$ = PSAREA;
.. 266 1817 2 LIST_DESC[0] = LIST_SIZE;
.. 267 1818 2 LIST_DESC[1] = _LIST_BUFFER;
.. 268 1819 2
.. 269 1820 2
.. 270 1821 2 ! Do the listing.
```

```
271 1822 2 !
272 1823 2 ! OPEN_JOURNAL(FALSE, FALSE);
273 1824 2
274 1825 2
275 1826 2 ! Heading.
276 1827 2
277 1828 2 FAB = .QUAL[QUAL_JOUR_FC];
278 P 1829 2 FAO_('Journal file !AD on !XD',
279 P 1830 2 .BBLOCK[FAB[FC_NAM], NAM$B_RSL],
280 P 1831 2 .BBLOCK[FAB[FC_NAM], NAM$L_RSA],
281 1832 2 0);
282 1833 2 EOL_();
283 1834 2
284 1835 2
285 1836 2 ! Read all records of the journal file to produce the listing.
286 1837 2
287 1838 2 CHANGE = FALSE;
288 1839 2 WHILE READ_JOURNAL(BUFFER) DO
289 1840 3 BEGIN
290 1841 3 CASE .BUFFER[BJL$B_TYPE] FROM BJL$K_SSNAME TO BJL$K_FILE OF
291 1842 3 SET
292 1843 3
293 1844 3
294 1845 3 [BJL$K_SSNAME]:
295 1846 4 BEGIN
296 1847 4
297 1848 4 ! Save the SSNAME record in a local.
298 1849 4
299 1850 4 IF .BUFFER[BJL$B_SIZE] LEQU BJL$C_SSNAME_LEN+1
300 1851 4 THEN
301 1852 5 BEGIN
302 1853 5 SSNAME_LEN = .BUFFER[BJL$B_SIZE] - 1;
303 1854 5 CH$MOVE(.SSNAME_LEN, BUFFER[BJL$B_DATA], SSNAME);
304 1855 5 CHANGE = TRUE;
305 1856 4 END;
306 1857 3 END;
307 1858 3
308 1859 3 [BJL$K_VOLUME]:
309 1860 4 BEGIN
310 1861 4
311 1862 4 ! Save the VOLUME record in a local.
312 1863 4
313 1864 4
314 1865 4 IF .BUFFER[BJL$B_SIZE] EQLU BJL$C_VOLUME_LEN+1
315 1866 4 THEN
316 1867 5 BEGIN
317 1868 5 CH$MOVE(BJL$C_VOLUME_LEN, BUFFER[BJL$B_DATA], VOLUME);
318 1869 5 CHANGE = TRUE;
319 1870 4 END;
320 1871 3 END;
321 1872 3
322 1873 3 [BJL$K_DIRECTORY]:
323 1874 4 BEGIN
324 1875 4
325 1876 4 ! Save the DIRECTORY record in a local.
326 1877 4
327 1878 4
```



```

328      1879  4      IF .BUFFER[BJL$B_SIZE] LEQU BJL$C_DIR_LEN+1
329      1880  4      THEN
330      1881  5          BEGIN
331      1882  5          DIR_LEN = .BUFFER[BJL$B_SIZE] - 1;
332      1883  5          CH$MOVE(.DIR_LEN, BUFFER[BJL$B_DATA], DIRECTORY);
333      1884  4          END;
INFO#252      L1:1879
Test expression is always true
334      1885  3      END;
335      1886  3
336      1887  3
337      1888  3      [BJL$K_FILE]:
338      1889  4      BEGIN
339      1890  4      LOCAL
340      1891  4          BUFF_SIZE      : WORD
341      1892  4          BUFF_FLAGS      : BBLOCK [4] ,
342      1893  4          NAME_DESC      : VECTOR[2];      ! Descriptor for name string
343      1894  4
344      1895  4
345      1896  4      BUFF_SIZE = .BUFFER [ BJL$B_SIZE ] ;
346      1897  4
347      1898  4      IF .JOUR_STRUCT_LEV GTRU BJL$K_LEVEL1
348      1899  4      THEN
349      1900  5          BEGIN
350      1901  5          ! Adjust the name size to remove the flag longword
351      1902  5          !
352      1903  5          BUFF_SIZE = .BUFF_SIZE - 4 ;
353      1904  5
354      1905  5          ! Get the flags longword ( size +1 to skip size field )
355      1906  5          !
356      1907  5          BUFF_FLAGS = .BUFFER [.BUFF_SIZE+1,0,32,0] ;
357      1908  5          END
358      1909  4      ELSE
359      1910  4          BUFF_FLAGS = 0 ;
360      1911  4
361      1912  4          ! Assemble the complete name string.
362      1913  4          !
363      1914  4          NAME_DESC[1] = BUFFER[0,0,0,0];
364      1915  4          NAME_DESC[0] = .BUFF_SIZE
365      1916  4          - 1      ! Adjust for size field
366      1917  4          + .DIR_LEN ! Add in directory name length
367      1918  4          + 2 ;      ! Adjust for '['
368      1919  4
369      1920  4      CH$COPY(
370      1921  4          .BUFF_SIZE - 1,
371      1922  4          BUFFER[BJL$B_DATA],
372      1923  4          %C' ',
373      1924  4          BUFFER + BUFFER_LENGTH - BUFFER[.DIR_LEN+2,0,0,0],
374      1925  4          BUFFER[.DIR_LEN+2,0,0,0]);
375      1926  4      CH$MOVE(
376      1927  4          .DIR_LEN,
377      1928  4          DIRECTORY,
378      1929  4          BUFFER[1,0,0,0]);
379      1930  4      BUFFER[0,0,8,0] = %C'[';
380      1931  4      BUFFER[.DIR_LEN+1,0,8,0] = %C']';
381      1932  4
382      1933  4      IF .BUFF_FLAGS[BJL$V_HEADONLY]

```

```

383 1934 4
384 1935 5
385 1936 5
386 1937 5
387 1938 5
388 1939 5
389 1940 5
390 1941 5
391 1942 4
392 1943 4
393 1944 4
394 1945 4
395 1946 4
396 1947 4
397 1948 5
398 1949 5
399 1950 5
400 1951 5
401 1952 5
402 1953 5
403 1954 5
404 1955 6
405 1956 6
406 P 1957 6
407 P P 1958 6
408 P P 1959 6
409 P 1960 6
410 1961 6
411 1962 6
412 P 1963 6
413 P P 1964 6
414 P P 1965 6
415 P P 1966 6
416 P P 1967 6
417 P 1968 6
418 1969 6
419 1970 6
420 1971 6
421 1972 5
422 1973 5
423 1974 5
424 1975 5
425 1976 5
426 1977 5
427 1978 5
428 1979 4
429 1980 5
430 1981 5
431 1982 5
432 1983 5
433 1984 5
434 1985 5
435 1986 5
436 1987 5
437 1988 2
438 1989 2
439 1990 1

```

```

THEN
  BEGIN
    CH$COPY ( STR_LENGTH,
              .STR,
              %C' ',
              BUFFER_LENGTH - .NAME_DESC[0],
              BUFFER[.NAME_DESC[0],0,8,0]);
    NAME_DESC[0] = .NAME_DESC[0] + STR_LENGTH ;
  END;

! Write the name string, if the selection criteria are satisfied.
IF SELECT_FILE(NAME_DESC, SSNAME[BJL$Q_CREDATE])
THEN
  BEGIN
    ! If the save set or the volume number has changed since the
    ! last line, write the new information.
    IF .CHANGE
    THEN
      BEGIN
        EOL ();
        FAO_ (
          'Save set !AD created on !XD',
          .SSNAME_LEN - $BYTEOFFSET(BJL$T_SSNAME),
          SSNAME[BJL$T_SSNAME],
          SSNAME[BJL$Q_CREDATE]);
        EOL ();
        FAO_ (
          'Volume number !UW, volume label !AD',
          .VOLUME[BJL$W_VOLNUMBER],
          BJL$S_VOLNAME,
          (IF .(VOLUME[BJL$T_VOLNAME])<0,8> GTRU %C' '
            THEN VOLUME[BJL$T_VOLNAME]
            ELSE UPLIT BYTE ('unknown ')));
        EOL ();
        CHARGE = FALSE;
      END;

    ! Write the file specification.
    FAO_ (' !AS', NAME_DESC);
    EOL ();
  END;
END;

[INRANGE, OUTRANGE]:
0;

TES;
END;
(CLOSE_JOURNAL (FALSE));
END;

```

6E	6F	20	72	65	64	61	65	68	20	65	6C	69	66	20	00087	P.AAA:	.ASCII	\ file header only\	:		
21	20	65	6C	69	66	20	6C	61	6E	72	75	6F	4A	17	00098	P.AAB:	.ASCII	<23>\Journal file !AD on !%D\	:		
63	20	44	41	21	20	74	65	73	20	65	76	61	53	18	000A7	P.AAC:	.ASCII	<27>\Save set !AD created on !%D\	:		
20	72	44	25	21	20	6E	6F	20	64	65	74	61	65	72	000B0	P.AAD:	.ASCII	\#Volume number !UW, volume label !AD\	:		
62	61	6C	20	65	6D	75	6E	20	65	6D	75	6C	6F	56	23	000CC	P.AAD:	.ASCII	\#Volume number !UW, volume label !AD\	:	
			20	20	20	20	20	6E	44	41	21	20	6C	65	000DB					:	
							53	41	21	20	6E	68	6E	75	000EA	P.AAE:	.ASCII	\unknown \		:	
											20	20	20	07	000FO	P.AAF:	.ASCII	<7>\ !AS\		:	
															000FC					:	
															OFFC	00000		.ENTRY	LIST_JOURNAL, Save R2,R3,R4,R5,R6,R7,R8,R9,-;	1766	
								5E	FC28	CE	9E	00002					MOVAB	R10,R11			
								5A	FF78	CF	9E	00007					MOVAB	-984(SP), SP			
										59	D4	0000C					MOVAB	P.AAA, STR		1793	
										56	D4	0000E					CLRL	SSNAME_LEN			
								5B	08	AE	9E	00010					CLRL	DIR_LEN			
								6B	0100	8F	3C	00014					MOVAB	PSAREA, PS		1816	
						04	AB	08		AB	9E	00019					MOVZWL	#256, (PS)		1817	
										7E	7C	0001E					MOVAB	8(R11), 4(PS)		1818	
										02	FB	00020					CLRQ	-(SP)		1823	
										50	00000000'	EF	D0	00027			CALLS	#2, OPEN_JOURNAL			
										7E	D4	0002E					MOVL	QUAL+52, FAB		1828	
										0098	CO	DD	00030				CLRL	-(SP)		1832	
								7E	0097	CO	9A	00034					PUSHL	152(FAB)			
									FF57	CF	9F	00039					MOVZBL	151(FAB), -(SP)			
										04	FB	0003D					PUSHAB	P.AAB			
										00	FB	00044					CALLS	#4, LIST_FAO			
										00	FB	00044					CALLS	#0, LIST_EOL		1833	
										57	D4	0004B					CLRL	CHANGE		1838	
										FE70	CD	9F	0004D	1\$:			PUSHAB	BUFFER		1839	
										01	FB	00051					CALLS	#1, READ_JOURNAL			
										50	E8	00058					BLBS	R0, 2\$			
										0140	31	0005B					BRW	16\$			
										FE71	CD	8F	0005E	2\$:			CASEB	BUFFER+1, #1, #3		1841	
										000A		00064	3\$:				.WORD	4\$-3\$,-			
																		5\$-3\$,-			
																		7\$-3\$,-			
																		8\$-3\$			
																		BRB	1\$		
										29	FE70	CD	91	0006E	4\$:		CMPB	BUFFER, #41		1850	
																	BGTRU	1\$			
										59	FE70	CD	9A	00075			MOVZBL	BUFFER, SSNAME_LEN		1853	
																	DECL	SSNAME_LEN			
								FE48	CD	FE72	CD	59	D7	0007A			MOVZBL	SSNAME_LEN, BUFFER+2, SSNAME		1854	
																	BRB	6\$		1855	
										OF	FE70	CD	91	00086	5\$:		CMPB	BUFFER, #15		1865	
																	BNEQ	1\$			
																	MOVZBL	#14, BUFFER+2, VOLUME		1868	
																	MOVZBL	#1, CHANGE		1869	
																	BRB	1\$		1841	

			56	FE70	CD	9A	0009A	7\$:	MOVZBL	BUFFER, DIR_LEN	1882	
						56	D7	0009F	DECL	DIR_LEN	1883	
	0110	CE	FE72	CD		56	28	000A1	MOVCS	DIR_LEN, BUFFER+2, DIRECTORY	1841	
						A2	11	000A9	BRB	1\$	1896	
			51	FE70	CD	9B	000AB	8\$:	MOVZBW	BUFFER, BUFF_SIZE	1898	
			0101	8F	00000000	EF	B1	000B0	CMPW	JOUR_STRUCT_EV, #257	1903	
						10	1B	000B9	BLEQU	9\$	1907	
			51			04	A2	000BB	SUBW2	#4, BUFF_SIZE	1907	
			50			51	3C	000BE	MOVZWL	BUFF_SIZE, R0	1898	
						FE71	CD	40	9F	000C1	1910	
			58			9E	D0	000C6	PUSHAB	BUFFER+1[R0]	1914	
						02	11	000C9	MOVL	@(SP)+, BUFF_FLAGS	1916	
						58	D4	000CB	BRB	10\$	1918	
			04	AE	FE70	CD	9E	000CD	9\$:	CLRL	BUFF_FLAGS	1921
			50			51	3C	000D3	10\$:	MOVAB	BUFFER, NAME_DESC+4	1924
			6E		01	A6	40	9E	000D6	MOVZWL	BUFF_SIZE, R0	1925
						50	D7	000DB	MOVAB	1(DIR_LEN)[R0], NAME_DESC	1929	
			51			6D	9E	000DD	DECL	R0	1930	
			52	FE72	CD	46	9E	000E0	MOVAB	BUFFER+400, R1	1931	
			51			52	C2	000E6	MOVAB	BUFFER+2[DIR_LEN], R2	1933	
51		20	FE72	CD		50	2C	000E9	SUBL2	R2, R1	1939	
						62		000F0	MOVCS	R0, BUFFER+2, #32, R1, (R2)	1940	
	FE71	CD	0110	CE		56	28	000F1	MOVCS	DIR_LEN, DIRECTORY, BUFFER+1	1941	
			FE70	CD		5B	8F	90	000F9	MOVAB	#91, BUFFER	1946
			FE71	CD	46	5D	8F	90	000FF	MOVAB	#93, BUFFER+1[DIR_LEN]	1953
			18			58	E9	00106	BLBC	BUFF_FLAGS, 11\$	1956	
			51	00000190	8F	6E	C3	00109	SUBL3	NAME_DESC, #400, R1	1961	
			50			FE70	CD	9E	00111	MOVAB	BUFFER, R0	1962
51		20	6A			11	2C	00116	MOVCS	#17, (STR), #32, R1, @NAME_DESC[R0]	1969	
						00	BE	40	0011B		1970	
			6E			11	C0	0011E	ADDL2	#17, NAME_DESC	1971	
						FE48	CD	9F	00121	11\$:	1977	
						04	AE	9F	00125	PUSHAB	SSNAME	1977
							02	FB	00128	PUSHAB	NAME_DESC	1977
			FDCF	CF		50	E9	0012D	CALLS	#2, SELECT_FILE	1977	
						6B			BLBC	R0, 15\$	1977	
						54	E9	00130	BLBC	CHANGE, 14\$	1977	
			00000000G	00		00	FB	00133	CALLS	#0, LIST_EOL	1977	
						FE48	CD	9F	0013A	PUSHAB	SSNAME	1977
						FE50	CD	9F	0013E	PUSHAB	SSNAME+8	1977
						F8	A9	9F	00142	PUSHAB	-8(SSNAME_LEN)	1977
						FE63	CF	9F	00145	PUSHAB	P.AAC	1977
			00000000G	00		04	FB	00149	CALLS	#4, LIST_FAO	1977	
			00000000G	00		00	FB	00150	CALLS	#0, LIST_EOL	1977	
						20	FE38	CD	91	00157	1977	
						07	1B	0015C	CMPB	VOLUME, #32	1977	
			50	FE38	CD	9E	0015E		BLEQU	12\$	1977	
			50	FE83	CF	9E	00165	12\$:	MOVAB	VOLUME, R0	1977	
						50	DD	0016A	13\$:	BRB	13\$	1977
						0C	DD	0016C	MOVAB	P.AAE, R0	1977	
			7E	FE44	CD	3C	0016E		PUSHL	R0	1977	
				FE51	CF	9F	00173		PUSHL	#12	1977	
			00000000G	00		04	FB	00177	MOVZWL	VOLUME+12, -(SP)	1977	
			00000000G	00		00	FB	0017E	PUSHAB	P.AAD	1977	
						57	D4	00185	CALLS	#4, LIST_FAO	1977	
						5E	DD	00187	CALLS	#0, LIST_EOL	1977	
						FE6B	CF	9F	00189	14\$:	1977	
									CLRL	CHANGE	1977	
									PUSHL	SP	1977	
									PUSHAB	P.AAF	1977	

LISTJOUR
V04-000

List a journal file
LIST_JOURNAL - main journal list routine

C 3
16-Sep-1984 00:10:02
14-Sep-1984 11:53:55

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]LISTJOUR.B32;1

Page 19
(4)

00000000G 00
00000000G 00

02 FB 0018D
00 FB 00194
FEAF 31 0019B 158:
7E D4 0019E 168:
01 FB 001A0
04 001A7

CALLS #2, LIST_FA0
CALLS #0, LIST_EOL
BRW 18
CLRL -(SP)
CALLS #1, CLOSE_JOURNAL
RET

:
: 1978
: 1839
: 1989
:
: 1990

; Routine Size: 424 bytes, Routine Base: CODE + 0104

LISTJOUR
V04-000

List a journal file
LIST_JOURNAL - main journal list routine

D 3
16-Sep-1984 00:10:02
14-Sep-1984 11:53:55

VAX-11 Bliss-32 V4.0-742
[BACKUP.SRC]LISTJOUR.B32;1

Page 20
(5)

: 441 1991 1 END
: 442 1992 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	684	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.1

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

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:LISTJOUR/OBJ=OBJ\$:LISTJOUR MSRC\$:LISTJOUR/UPDATE=(ENHS:LISTJOUR)

: Size: 559 code + 2249 data bytes
: Run Time: 00:23.3
: Elapsed Time: 01:21.8
: Lines/CPU Min: 5120
: Lexemes/CPU-Min: 45974
: Memory Used: 313 pages
: Compilation Complete

