

```

BBBBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAAAAAAAAA      CCCCCCCCCCCCCC      KKK      KKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      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
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      UUU      UUU      PPPPPPPPPPPP
BBBBBBBBBBBBBB      AAA      AAA      CCC      KKKKKKKKKK      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
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUUU      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUUU      PPP
BBBBBBBBBBBBBB      AAA      AAA      CCCCCCCCCCCCCC      KKK      KKK      UUUUUUUUUUUUUUUU      PPP

```

LL	IIIIII	SSSSSSSS	TTTTTTTT	
LL	IIIIII	SSSSSSSS	TTTTTTTT	
LL	II	SS	TT	
LL	II	SS	TT	
LL	II	SS	TT	
LL	II	SS	TT	
LL	II	SSSSSS	TT	
LL	II	SSSSSS	TT	
LL	II		TT	
LL	II		TT	
LL	II		TT	
LL	II		TT	
LL	II		TT	
LL	II		TT	
LL	II		TT	
LLLLLLLLLL	IIIIII	SSSSSSSS	TT
LLLLLLLLLL	IIIIII	SSSSSSSS	TT

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
LL	II		SS
LLLLLLLLLL	IIIIII	SSSSSSSS	
LLLLLLLLLL	IIIIII	SSSSSSSS	

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

0001 0 MODULE LIST      (%TITLE 'List a save set'
0002 0                   IDENT = 'V04-000'
0003 0                   ) =
0004 1 BEGIN
0005 1
0006 1
0007 1 *****
0008 1 *
0009 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 *  ALL RIGHTS RESERVED.
0012 1 *
0013 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 *  TRANSFERRED.
0019 1 *
0020 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 *  CORPORATION.
0023 1 *
0024 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 **
0032 1 FACILITY:
0033 1     Backup/Restore
0034 1
0035 1 ABSTRACT:
0036 1     This module contains the routines that list a save set.
0037 1
0038 1 ENVIRONMENT:
0039 1     VAX/VMS user mode.
0040 1 --
0041 1
0042 1 AUTHOR: M. Jack, CREATION DATE: 03-Sep-1980
0043 1
0044 1 MODIFIED BY:
0045 1
0046 1     V03-008 LY0510          Larry Yetto          19-JUL-1984 08:45
0047 1     Add support for the new longword format of the DEVTYP in
0048 1     the physical volume attribute record. This longword is the
0049 1     same format as UCBSL_MEDIA_ID
0050 1
0051 1     V03-007 LY0483          Larry Yetto          27-APR-1984 08:39
0052 1     FT1 QAR # 2088 - If save set being read is encrypted
0053 1     and /ENCRYPT was not specified then signal an error
0054 1
0055 1     V03-006 JEP0003          J. Eric Pollack,      23-Apr-1983 10:53
0056 1     Add support for encrypted savesets.
0057 1

```

58	0058	1	V03-005	ACG0313	Andrew C. Goldstein,	12-Feb-1983	16:20
59	0059	1			Add routine subtitles		
60	0060	1					
61	0061	1	V03-004	MLJ50891	Martin L. Jack,	05-Nov-1982	16:37
62	0062	1			Correct save set volume number in listing produced during save		
63	0063	1			operation.		
64	0064	1					
65	0065	1	V03-003	LMP0032	L. Mark Pilant,	22-Jun-1982	10:35
66	0066	1			Add support for wildcard save sets on a LIST or RESTORE		
67	0067	1			operation.		
68	0068	1					
69	0069	1	V03-002	MLJ0084	Martin L. Jack,	26-Mar-1982	3:07
70	0070	1			Use FAT\$W_MAXREC rather than FAT\$W_RSIZE to report maximum		
71	0071	1			record size. This corrects a discrepancy with DIRECTORY/FULL.		
72	0072	1			Add new device types to track \$DCDEF.		
73	0073	1					
74	0074	1	V03-001	MLJ0083	Martin L. Jack,	22-Mar-1982	13:37
75	0075	1			Track change to DIRECTORY/FULL.		
76	0076	1					
77	0077	1	V02-005	MLJ0081	Martin L. Jack,	26-Feb-1982	16:19
78	0078	1			Implement RETAINMIN and RETAINMAX attributes to support new		
79	0079	1			home block fields.		
80	0080	1					
81	0081	1	V02-004	MLJ0043	Martin L. Jack,	3-Sep-1981	21:34
82	0082	1			Add stream record types to /FULL listing.		
83	0083	1					
84	0084	1	V02-003	MLJ0036	Martin L. Jack,	28-Aug-1981	17:44
85	0085	1			Implement reel restart. Track new device type codes. Add		
86	0086	1			backup date to track DIRECTORY/FULL. Print command line on		
87	0087	1			multiple lines if necessary to avoid truncation.		
88	0088	1					
89	0089	1	V02-002	MLJ0020	Martin L. Jack,	20-Apr-1981	22:09
90	0090	1			Implement /JOURNAL qualifier.		
91	0091	1					
92	0092	1	V02-001	MLJ0010	Martin L. Jack,	25-Mar-1981	16:02
93	0093	1			Reorganize global storage. Replace OWN storage by LOCAL.		
94	0094	1			Add capability for listing concurrently with other operations.		
95	0095	1					
96	0096	1					**

```

: 98 0097 1 REQUIRE 'SRCS:COMMON';
: 99 1203 1 LIBRARY 'SYSS$LIBRARY:LIB';
100 1204 1 REQUIRE 'LIBS:BACKDEF';
101 1654 1
102 1655 1
103 1656 1 LINKAGE
104 1657 1 L_PS = CALL: GLOBAL(PS=11);
105 1658 1
106 1659 1
107 1660 1 MACRO
108 1661 1 L_DECL= EXTERNAL REGISTER PS = 11: REF VECTOR %;
109 1662 1
110 1663 1
111 1664 1 FORWARD ROUTINE
112 1665 1 LIST_FAO: L_PS NOVALUE, ! Format information to line buffer
113 1666 1 LIST_EOL: L_PS NOVALUE, ! Write line buffer to listing file
114 1667 1 LIST_PROTECTION:L_PS NOVALUE, ! List protection code
115 1668 1 FULL_LISTING: L_PS NOVALUE, ! Produce /FULL listing
116 1669 1 LIST_ONE_RECORD:L_PS NOVALUE, ! List one save set record
117 1670 1 INIT_LIST: NOVALUE, ! Initialize listing
118 1671 1 FIN_LIST: NOVALUE, ! Finish listing
119 1672 1 CONCURRENT_LIST:NOVALUE, ! List one buffer concurrently with
120 1673 1 ! another operation
121 1674 1 LIST: NOVALUE; ! Driver for list-only execution
122 1675 1
123 1676 1
124 1677 1 EXTERNAL ROUTINE
125 1678 1 DEBLOCK: L_PS NOVALUE, ! Deblock a save set buffer
126 1679 1 DECODE_DEVTYP: NOVALUE, ! Convert DEVTYP to ASCII string
127 1680 1 FILE_ERROR: NOVALUE, ! Signal file-related error
128 1681 1 FIN_IN_SAVE: NOVALUE, ! Finish input save set processing
129 1682 1 GET_SUMMARY_ATTRIBUTES:
130 1683 1 NOVALUE, ! Get BACKUP summary attributes
131 1684 1 GET_VOLUME_ATTRIBUTES:
132 1685 1 NOVALUE, ! Get volume summary attributes
133 1686 1 GET_FILE_ATTRIBUTES:
134 1687 1 NOVALUE, ! Get file attributes
135 1688 1 GET_PHYSVOL_ATTRIBUTES:
136 1689 1 NOVALUE, ! Get physical volume attributes
137 1690 1 INIT_IN_SAVE: NOVALUE, ! Initialize input save set processing
138 1691 1 READ_BUFFER, ! Get one buffer from save set
139 1692 1 RESTORE_HANDLER, ! Handler for RESTORE, LIST, ANALYZE
140 1693 1 CRYPTO_INIDEC: WEAK, ! Setup for reading encrypted saveset
141 1694 1 CRYPTO_CHKSAV, ! Check for encrypted save set
142 1695 1 CRYPTO_DECR_BLOCK:
143 1696 1 NOVALUE WEAK; ! Decrypt one block
144 1697 1
145 1698 1
146 1699 1 EXTERNAL LITERAL
147 1700 1 BACKUP$_OPENOUT,
148 1701 1 BACKUP$_WRITEERR,
149 1702 1 BACKUP$_BACNOTENC,
150 1703 1 BACKUP$_CLOSEOUT,
151 1704 1 BACKUP$_ENCSAVSET;
152 1705 1
153 1706 1
154 1707 1 G$DEFINE(); ! Define global common area

```

```
155 1708 1
156 1709 1
157 1710 1 BIND
158 1711 1 OUTPUT_FILENAME=OUTPUT_ATTBUF[FAR_FILENAME]: BBLOCK,
159 1712 1 OUTPUT_RECATTR= OUTPUT_ATTBUF[FAR_RECATTR]: BBLOCK,
160 1713 1 OUTPUT_FILECHAR=OUTPUT_ATTBUF[FAR_UCHAR]: BBLOCK;
161 1714 1
162 1715 1
163 1716 1 MACRO
M 164 1717 1 FAO_(A)=
M 165 1718 1 -LIST FAO(
M 166 1719 1 OPLIT BYTE (%ASCII A)
167 1720 1 %IF NOT %NULL(%REMAINING) %THEN , %I %REMAINING) %,
168 1721 1
169 1722 1
M 170 1723 1 EOL_(A)=
171 1724 1 -LIST_EOL() %;
172 1725 1
173 1726 1
174 1727 1 BUILTIN
175 1728 1 ROT;
176 1729 1
177 1730 1
178 1731 1 MACRO
179 1732 1 -LIST_DESC= PS[0] %, ! Descriptor for buffer
180 1733 1 -LIST_BUFFER= PS[2] %, ! Listing buffer
181 1734 1 [LIST_DESC(N)= VECTOR[_LIST_DESC,N] %;
182 1735 1
183 1736 1
184 1737 1 LITERAL
185 1738 1 P$SIZE= 2 + CH$ALLOCATION/LIST_SIZE);
```

```

187 1739 1 %SBTTL 'LIST FAO - call FAO to format data'
188 1740 1 GLOBAL ROUTINE LIST_FAO(CTRL,PARAM): L_PS NOVALUE=
189 1741 1
190 1742 1 |++
191 1743 1 |
192 1744 1 | FUNCTIONAL DESCRIPTION:
193 1745 1 |     This routine interfaces to FAO to format information into the line.
194 1746 1 |
195 1747 1 | INPUT PARAMETERS:
196 1748 1 |     CTRL           - ASCII control string
197 1749 1 |     PARAM...      - Parameters required by the control string (if any)
198 1750 1 |
199 1751 1 | IMPLICIT INPUTS:
200 1752 1 |     NONE
201 1753 1 |
202 1754 1 | OUTPUT PARAMETERS:
203 1755 1 |     NONE
204 1756 1 |
205 1757 1 | IMPLICIT OUTPUTS:
206 1758 1 |     Information formatted into the line buffer.
207 1759 1 |
208 1760 1 | ROUTINE VALUE:
209 1761 1 |     NONE
210 1762 1 |
211 1763 1 | SIDE EFFECTS:
212 1764 1 |     NONE
213 1765 1 |
214 1766 1 | --
215 1767 1 |
216 1768 2 BEGIN
217 1769 2 MAP
218 1770 2     CTRL:           REF VECTOR[,BYTE];           ! Pointer to ASCII string
219 1771 2 LOCAL
220 1772 2     OUTLEN:         WORD,                          ! Length of FAO result
221 1773 2     DESC:           VECTOR[2];                       ! Descriptor for control string
222 1774 2 L_DECL;
223 1775 2
224 1776 2
225 1777 2 ! Build a descriptor for the control string.
226 1778 2
227 1779 2 DESC[0] = .CTRL[0];
228 1780 2 DESC[1] = CTRL[1];
229 1781 2
230 1782 2
231 1783 2 ! Use FAO to edit the information.
232 1784 2
233 1785 2 $FAOL(CTRSTR=DESC, OUTLEN=OUTLEN, OUTBUF=_LIST_DESC, PRMLST=PARAM);
234 1786 2
235 1787 2
236 1788 2 ! Adjust the line buffer descriptor to account for the information
237 1789 2 ! added to the line.
238 1790 2
239 1791 2 LIST_DESC[0] = .LIST_DESC[0] - .OUTLEN;
240 1792 2 LIST_DESC[1] = .LIST_DESC[1] + .OUTLEN;
241 1793 1 END;

```

LIST
V04-000

List a save set
LIST_FA0 - call FA0 to format data

I 13
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 6 (3)

```
.TITLE LIST List a save set
.IDENT \V04-000\
.PSECT COMMON,NOEXE, OVR,2

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


00110 RWSV_IN_XOR_RFA:
 .BLKB 6
00116 RWSV_LOOKAHEAD:
 .BLKB 1
00117 RWSV_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_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

LIST
V04-000

List a save set
LIST_FAO - call FAO to format data

K 13
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742 Page 8
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B52;1 (3)

001EC	ALT_SSNAME:	
	.BLKB	32
0020C	INPUT_FUNC:	
	.BLKB	1
0020D	INPUT_RTTYPE:	
	.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:	

LIST
V04-000

List a save set
LIST_FA0 - call FA0 to format data

L 13
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 -32 V4.0-742 Page 9
DISK\$ ER:[BACKUP.SRC]LIST.B32;1 (3)

00290	INPUT_PROC_LIST:	.BLKB	0
00294	INPUT_PLACEMENT:	.BLKB	4
0029C	INPUT_VBN_LIST:	.BLKB	8
002A4	INPUT_PLACE_LEN:	.BLKB	8
002A6	INPUT_PADDING_2:	.BLKB	2
002A8	OUTPUT_BEG:	.BLKB	2
002A8	OUTPUT_CHAN:	.BLKB	0
002AC	OUTPUT_FLAGS:	.BLKB	4
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	4
002D4	OUTPUT_ATTBUF:	.BLKB	8
00364	OUTPUT_END:	.BLKB	144
00364	LIST_TOTFILES:	.BLKB	0
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	4
		.BLKB	1

LIST
V04-000

List a save set
LIST_FAO - call FAO to format data

M 13
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 10
(3)

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:		

003E0 DIR_SEL_NTV: .BLKB 8
003E8 DIR_STRUCLEV: .BLKB 8
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

OUTPUT_FILENAME=
OUTPUT_RECATTR=

OUTPUT_ATTBUF
OUTPUT_ATTBUF+16

```

OUTPUT_FILECHAR= OUTPUT_ATTBUF+100
.EXTRN DEBLOCK, DECODE_DEVTYPE
.EXTRN FILE_ERROR, FIN-IN SAVE
.EXTRN GET_SUMMARY_ATTRIBUTES
.EXTRN GET_VOLUME_ATTRIBUTES
.EXTRN GET_FILE_ATTRIBUTES
.EXTRN GET_PHYSVOL_ATTRIBUTES
.EXTRN INIT_IN_SAVE, READ_BUFFER
.EXTRN RESTORE_HANDLER
.EXTRN CRYPTO_CHKSAV, BACKUP$_OPENOUT
.EXTRN BACKUP$_WRITEERR
.EXTRN BACKUP$_BACNOTENC
.EXTRN BACKUP$_CLOSEOUT
.EXTRN BACKUP$_ENCSAVSET
.EXTRN SYSSFAOL
.WEAK CRYPTO_INIDEC, CRYPTO_DECR_BLOCK

.PSECT CODE, NOWRT, 2

```

```

0000 00000
04 0C C2 00002
08 AE 04 AE AC 04 BC 9A 00005
08 01 C1 0000A
08 AC 9F 00010
08 5B DD 00013
10 AE 9F 00015
00000000G 00 04 FB 0001B
50 6E 3C 00022
6B 50 C2 00025
50 6E 3C 00028
04 AB 50 C0 0002B
04 0002F

```

```

.ENTRY LIST_FAO, Save nothing : 1740
SUBL2 #12, SP :
MOVZBL @CTRL, DESC : 1779
ADDL3 #1, CTRL, DESC+4 : 1780
PUSHAB PARAM : 1785
PUSHL P$
PUSHAB OUTLEN
PUSHAB DESC
CALLS #4, SYSSFAOL : 1791
MOVZWL OUTLEN, R0 :
SUBL2 R0, (P$) : 1792
MOVZWL OUTLEN, R0 :
ADDL2 R0, 4(P$) :
RET : 1793

```

; Routine Size: 48 bytes, Routine Base: CODE + 0000

```
243 1794 1 %SBTTL 'LIST_EOL - write buffer to listing file'
244 1795 1 GLOBAL ROUTINE LIST_EOL: L_PS NOVALUE=
245 1796 1
246 1797 1 !++
247 1798 1
248 1799 1 FUNCTIONAL DESCRIPTION:
249 1800 1 This routine writes the listing buffer to the listing file.
250 1801 1
251 1802 1 INPUT PARAMETERS:
252 1803 1 NONE
253 1804 1
254 1805 1 IMPLICIT INPUTS:
255 1806 1 NONE
256 1807 1
257 1808 1 OUTPUT PARAMETERS:
258 1809 1 NONE
259 1810 1
260 1811 1 IMPLICIT OUTPUTS:
261 1812 1 NONE
262 1813 1
263 1814 1 ROUTINE VALUE:
264 1815 1 NONE
265 1816 1
266 1817 1 SIDE EFFECTS:
267 1818 1 The listing is produced.
268 1819 1
269 1820 1 !--
270 1821 1
271 1822 2 BEGIN
272 1823 2 LOCAL
273 1824 2 LIST_FAB: REF BBLOCK; ! FAB for listing file
274 1825 2 L_DECL;
275 1826 2
276 1827 2
277 1828 2 ! Compute the length of the line.
278 1829 2
279 1830 2 LIST_FAB = .QUAL[QUAL LIST FC];
280 1831 2 BBLOCK[LIST_FAB[FC_RAB], RAB$W_RSZ] = LIST_SIZE - .LIST_DESC[0];
281 1832 2 BBLOCK[LIST_FAB[FC_RAB], RAB$L_RBF] = _LIST_BUFFER;
282 1833 2
283 1834 2
284 1835 2 ! Write the line.
285 1836 2
286 1837 3 IF NOT $PUT(RAB=LIST_FAB[FC_RAB])
287 1838 2 THEN
288 1839 2 FILE_ERROR(
289 1840 2 BACKUP$ WRITEERR + STS$K_SEVERE,
290 1841 2 .LIST_FAB,
291 1842 2 .BBLOCK[LIST_FAB[FC_RAB], RAB$L_STS],
292 1843 2 .BBLOCK[LIST_FAB[FC_RAB], RAB$L_STV]);
293 1844 2
294 1845 2
295 1846 2 ! Reinitialize the line buffer descriptor to describe the entire line.
296 1847 2
297 1848 2 LIST_DESC[0] = LIST_SIZE;
298 1849 2 LIST_DESC[1] = _LIST_BUFFER;
299 1850 1 END;
```

```

000C 00000
72 A2 0100 52 00000000' EF DO 00002
8F 6B A3 00009
53 08 AB 9E 00010
78 A2 53 DO 00014
50 A2 9F 00018
00000000G 00 01 FB 0001B
13 50 EB 00022
7E 58 A2 7D 00025
52 DD 00029
00000000G 00 8F DD 0002B
6B 0100 04 FB 00031
04 AB 8F 3C 00038 1$:
53 DO 0003D
04 00041

```

```

.EXTRN SYSSPUT
.ENTRY LIST_EOL, Save R2,R3 : 1795
MOVL QUAL$56, LIST_FAB : 1830
SUBW3 (P$), #256, 1T4(LIST_FAB) : 1831
MOVAB 8(R11), R3 : 1832
MOVL R3, 120(LIST_FAB) :
PUSHAB 80(R2) : 1837
CALLS #1, SYSSPUT :
BLBS R0, 1$ :
MOVQ 88(LIST_FAB), -(SP) : 1842
PUSHL LIST_FAB : 1841
PUSHL #BACKUP$ WRITEERR+4 : 1840
CALLS #4, FILE_ERROR :
MOVZWL #256, (P$) : 1848
MOVL R3, 4(P$) : 1849
RET : 1850

```

; Routine Size: 66 bytes, Routine Base: CODE + 0030


```

301 1851 1 %SBTTL 'LIST PROTECTION - format protection value'
302 1852 1 GLOBAL ROUTINE LIST_PROTECTION(PROT,CODES): L_PS NOVALUE=
303 1853 1
304 1854 1 !++
305 1855 1
306 1856 1 FUNCTIONAL DESCRIPTION:
307 1857 1 This routine formats a protection value.
308 1858 1
309 1859 1 INPUT PARAMETERS:
310 1860 1 PROT - A protection value.
311 1861 1 CODES - The four letter codes, e.g. 'RWED'.
312 1862 1
313 1863 1 IMPLICIT INPUTS:
314 1864 1 NONE
315 1865 1
316 1866 1 OUTPUT PARAMETERS:
317 1867 1 NONE
318 1868 1
319 1869 1 IMPLICIT OUTPUTS:
320 1870 1 NONE
321 1871 1
322 1872 1 ROUTINE VALUE:
323 1873 1 NONE
324 1874 1
325 1875 1 SIDE EFFECTS:
326 1876 1 The listing is produced.
327 1877 1
328 1878 1 !--
329 1879 1
330 1880 2 BEGIN
331 1881 2 MAP
332 1882 2 PROT: BITVECTOR[16], ! Protection value
333 1883 2 CODES: VECTOR[,BYTE]; ! Letter codes
334 1884 2 BIND
335 1885 2 WHO = UPLIT (
336 1886 2 UPLIT BYTE (%ASCIC 'System:'),
337 1887 2 UPLIT BYTE (%ASCIC ', Owner:'),
338 1888 2 UPLIT BYTE (%ASCIC ', Group:'),
339 1889 2 UPLIT BYTE (%ASCIC ', World:'))
340 1890 2 : VECTOR;
341 1891 2 L_DECL;
342 1892 2
343 1893 2
344 1894 2 INCR I FROM 0 TO 3 DO
345 1895 3 BEGIN
346 1896 3 FAO ('!AC', .WHO[I]);
347 1897 3 INCR J FROM 0 TO 3 DO
348 1898 4 BEGIN
349 1899 4 IF NOT .PROT[I*4+.J] THEN FAO('!AD', 1, CODES[J]);
350 1900 3 END;
351 1901 2 END;
352 1902 1 END;

```

```

3A 3A 6D 65 74 73 79 53 07 00072 P.AAB: .ASCII <7>\System:\
3A 72 65 6E 77 4F 20 2C 08 0007A P.AAC: .ASCII <8>\, Owner:\

```

3A 70 75 6F 72 47 20 2C 08	00083	P.AAD:	.ASCII <8>\, Group:\	:	
3A 64 6C 72 6F 57 20 2C 08	0008C	P.AAE:	.ASCII <8>\, World:\	:	
	00095		.BLKB 3	:	
00000000' 0000000C' 00000000' 00000000'	00098	P.AAA:	.ADDRESS P.AAB, P.AAC, P.AAD, P.AAE	:	
	43 41 21 03	000A8	P.AAF:	.ASCII <3>\!AC\	:
	44 41 21 03	000AC	P.AAG:	.ASCII <3>\!AD\	:
		WHO=	P.AAA		
	001C 00000		.ENTRY LIST_PROTECTION, Save R2,R3,R4	: 1852	
54 FF4A CF 9E 00002			MOVAB LIST_FA0, R4	: 1894	
	53 D4 00007		CLRL I	: 1896	
	DB AF43 DD 00009 1\$:		PUSHL WHO[I]	: 1897	
	E8 AF 9F 0000D		PUSHAB P.AAF	: 1899	
64	02 FB 00010		CALLS #2, LIST_FA0	:	
	52 D4 00013		CLRL J	: 1897	
	6243 DE 00015 2\$:		MOVAL (J)[I], R0	: 1899	
OC 04 AC	50 E0 00019		BBS R0, PROT, 3\$:	
	08 AC42 9F 0001E		PUSHAB CODES[J]	:	
	01 DD 00022		PUSHL #1	:	
	D5 AF 9F 00024		PUSHAB P.AAG	:	
	03 FB 00027		CALLS #3, LIST_FA0	:	
E7 64	03 F3 0002A 3\$:		AOBLEQ #3, J, 2\$: 1897	
D7 52	03 F3 0002E		AOBLEQ #3, I, 1\$: 1894	
	53 03 F3 00032		RET	: 1902	

; Routine Size: 51 bytes, Routine Base: CODE + 00B0

```
354 1903 1 %SBTTL 'FULL_LISTING - format full listing'
355 1904 1 ROUTINE FULL_LISTING: L_P$ NOVALUE=
356 1905 1
357 1906 1 |++
358 1907 1 |
359 1908 1 | FUNCTIONAL DESCRIPTION:
360 1909 1 | This routine formats the /FULL listing. The code in this routine is
361 1910 1 | intended to track the DIRECTORY/FULL listing.
362 1911 1 |
363 1912 1 | INPUT PARAMETERS:
364 1913 1 | NONE
365 1914 1 |
366 1915 1 | IMPLICIT INPUTS:
367 1916 1 | NONE
368 1917 1 |
369 1918 1 | OUTPUT PARAMETERS:
370 1919 1 | NONE
371 1920 1 |
372 1921 1 | IMPLICIT OUTPUTS:
373 1922 1 | NONE
374 1923 1 |
375 1924 1 | ROUTINE VALUE:
376 1925 1 | NONE
377 1926 1 |
378 1927 1 | SIDE EFFECTS:
379 1928 1 | The listing is produced.
380 1929 1 |
381 1930 1 | --
382 1931 1 |
383 1932 2 BEGIN
384 1933 2 LOCAL
385 1934 2     BLOCKS,           ! Used blocks
386 1935 2     DIGITS;       ! Number of digits needed for file size
387 1936 2 L_DECL:
388 1937 2
389 1938 2
390 1939 2 ! Compute file size.
391 1940 2 !
392 1941 2 BLOCKS = ROT(.OUTPUT_RECATTR[FAT$L_EFBLK], 16);
393 1942 2 IF .OUTPUT_RECATTR[FAT$W_FFBYTE] EQL 0 AND .BLOCKS NEQ 0
394 1943 2     THEN BLOCKS = .BLOCKS - 1;
395 1944 2
396 1945 2
397 1946 2 ! Compute digits to display file size.
398 1947 2 !
399 1948 2 IF
400 1949 2     .BLOCKS GTR 99999 OR           ! If larger than 5 digits
401 1950 2     .OUTPUT_ATTBUF[FAR_FILESIZE] GTR 99999
402 1951 2 THEN
403 1952 2     DIGITS = 10                 ! then make output totally variable
404 1953 2 ELSE
405 1954 2     DIGITS = 5;                 ! else use fixed field size
406 1955 2
407 1956 2
408 1957 2 ! Format first line.
409 1958 2 !
410 1959 2 IF .OUTPUT_FILENAME[DSC$W_LENGTH] GTRU 21     ! If long file name.
```

```
411 1960 2 THEN
412 1961 2 BEGIN
413 1962 2     FAO_(!AS', OUTPUT_FILENAME);           ! Display name on first line
414 1963 2     EOL_();
415 1964 2     FAO_(!22* ');
416 1965 2     END
417 1966 2 ELSE
418 1967 2     FAO_(!21AS ', OUTPUT_FILENAME);
419 1968 2
420 1969 2
421 P 1970 2     FAO_(!Size:  !#UL/!#<!UL!>    Created: !17%D',
422 1971 2         .DIGITS, .BLOCKS,
423 P 1972 2         .DIGITS, .OUTPUT_ATTBUF[FAR_FILESIZE],
424 1973 2         OUTPUT_ATTBUF[FAR_CREDATE]));
425 1974 2     EOL_();
426 1975 2
427 1976 2
428 1977 2 ! Format second line.
429 1978 2 !
430 P 1979 2     FAO_(!22* Owner:  [!30W,!30W]    Revised: ',
431 P 1980 2         .OUTPUT_ATTBUF[FAR_UICGROUP],
432 1981 2         .OUTPUT_ATTBUF[FAR_UICMEMBER]));
433 1982 2
434 1983 2 IF .(OUTPUT_ATTBUF[FAR_REVDATE]) EQL 0
435 1984 2 THEN
436 1985 2     FAO_('<None specified>')
437 1986 2 ELSE
438 P 1987 2     FAO_(!17%D (!UL)',
439 P 1988 2         .OUTPUT_ATTBUF[FAR_REVDATE],
440 1989 2         .OUTPUT_ATTBUF[FAR_REVISION]));
441 1990 2     EOL_();
442 1991 2
443 1992 2
444 1993 2 ! Format third line.
445 1994 2 !
446 P 1995 2     FAO_(!22* File ID: !14<(!UL,!UL,!UL)!> Expires: ',
447 P 1996 2         .OUTPUT_ATTBUF[FAR_FID_NUM] + .OUTPUT_ATTBUF[FAR_FID_NMX] ^ 16,
448 P 1997 2         .OUTPUT_ATTBUF[FAR_FID_SEQ],
449 1998 2         .OUTPUT_ATTBUF[FAR_FID_RVN]));
450 1999 2
451 2000 2 IF .(OUTPUT_ATTBUF[FAR_EXPDATE]) EQL 0
452 2001 2 THEN
453 2002 2     FAO_('<None specified>')
454 2003 2 ELSE
455 P 2004 2     FAO_(!17%D',
456 2005 2         .OUTPUT_ATTBUF[FAR_EXPDATE]);
457 2006 2     EOL_();
458 2007 2
459 2008 2
460 2009 2 ! Format fourth line.
461 2010 2 !
462 P 2011 2     FAO_(!46* Backup:  ',
463 P 2012 2         .OUTPUT_ATTBUF[FAR_FID_NUM] + .OUTPUT_ATTBUF[FAR_FID_NMX] ^ 16,
464 P 2013 2         .OUTPUT_ATTBUF[FAR_FID_SEQ],
465 2014 2         .OUTPUT_ATTBUF[FAR_FID_RVN]));
466 2015 2
467 2016 2 IF .(OUTPUT_ATTBUF[FAR_BAKDATE]) EQL 0
```

```
: 468      2017 2 THEN
: 469      2018 22    FAO_('<No backup done>')
: 470      2019 22    ELSE
: 471      2020 22    FAO_('!17%D'
: 472      2021 22    .OUTPUT_ATTBUF[FAR_BAKDATE]);
: 473      2022 22    EOL_();
: 474      2023 22
: 475      2024 22
: 476      2025 22    ! Format fifth line.
: 477      2026 22
: 478      2027 22    FAO_(' File protection: ');
: 479      2028 22    LIST_PROTECTION(.OUTPUT_ATTBUF[FAR_FPRO], 'RWED');
: 480      2029 22    EOL_();
: 481      2030 22
: 482      2031 22
: 483      2032 22    ! Format sixth line.
: 484      2033 22
: 485      2034 22    FAO_(' File organization: ');
: 486      2035 22    SELECTONEU .OUTPUT_RECATTR[FAT$V_FILEORG] OF
: 487      2036 22    SET
: 488      2037 22    [FAT$C_SEQUENTIAL]: FAO_('Sequential');
: 489      2038 22    [FAT$C_RELATIVE]: FAO_('Relative, maximum record number = !UL',
: 490      2039 22    .OUTPUT_RECATTR[FAT$W_MAXREC]);
: 491      2040 22    [FAT$C_INDEXED]: FAO_('Indexed');
: 492      2041 22    [OTHERWISE]: FAO_('Unknown type = !UL',
: 493      2042 22    .OUTPUT_RECATTR[FAT$V_FILEORG]);
: 494      2043 22
: 495      2044 22    TES;
: 496      2045 22    EOL_();
: 497      2046 22
: 498      2047 22    ! Format seventh line.
: 499      2048 22
: 500      2049 22    FAO_(' File attributes: Allocation = !UL, Extend = !UL',
: 501      2050 22    .OUTPUT_ATTBUF[FAR_FILESIZE],
: 502      2051 22    .OUTPUT_RECATTR[FAT$W_DEEXT]);
: 503      2052 22    IF .OUTPUT_RECATTR[FAT$B_BKTSIZE] NEQ 0 ! If non-zero bucket size,
: 504      2053 22    THEN
: 505      2054 22    BEGIN
: 506      2055 22    FAO_(' ');
: 507      2056 22    IF .OUTPUT_RECATTR[FAT$V_FILEORG] EQL FAT$C_INDEXED THEN FAO_('maximum ');
: 508      2057 22    FAO_('bucket size = !UL', .OUTPUT_RECATTR[FAT$B_BKTSIZE]);
: 509      2058 22    END;
: 510      2059 22    EOL_();
: 511      2060 22
: 512      2061 22
: 513      2062 22    ! Format eighth line.
: 514      2063 22
: 515      2064 22    FAO_('!22* Global Buffer Count = !UL',
: 516      2065 22    .OUTPUT_RECATTR[FAT$W_GBC]);
: 517      2066 22    IF .OUTPUT_FILECHAR[FCH$V_CONTIGB] ! If contiguous best try,
: 518      2067 22    THEN
: 519      2068 22    FAO_(' Contiguous-best-try')
: 520      2069 22    ELSE IF .OUTPUT_FILECHAR[FCH$V_CONTIG] ! If file is contiguous,
: 521      2070 22    THEN
: 522      2071 22    FAO_(' Contiguous');
: 523      2072 22    IF .OUTPUT_FILECHAR[FCH$V_LOCKED] ! If file is access locked,
: 524      2073 22    THEN
```

```

525 2074 2    FAO_(', Locked');
526 2075 2    EOL_();
527 2076 2
528 2077 2
529 2078 2    ! Format ninth line.
530 2079 2
531 2080 2    FAO_(' Record format: ');
532 2081 2    SELECT ONEU .OUTPUT_RECATTR[FAT$V_RTYPE] OF
533 2082 2    SET
534 P 2083 2    [FAT$C_FIXED]:    FAO_('Fixed length !UL byte records',
535 2084 2    .OUTPUT_RECATTR[FAT$W_RSIZE]);
536 2085 2    [FAT$C_VARIABLE]:  FAO_('Variable length');
537 P 2086 2    [FAT$C_VFC]:    FAO_('VFC, !UL byte header',
538 2087 2    .OUTPUT_RECATTR[FAT$B_VFCSIZE]);
539 2088 2    [FAT$C_STREAM]:    FAO_('Stream');
540 2089 2    [FAT$C_STREAMLF]:  FAO_('Stream_LF');
541 2090 2    [FAT$C_STREAMCR]:  FAO_('Stream_CR');
542 2091 2    [0]:    FAO_('Undefined');
543 P 2092 2    [OTHERWISE]:    FAO_('Unknown type = !UL',
544 2093 2    .OUTPUT_RECATTR[FAT$V_RTYPE]);
545 2094 2    TES;
546 2095 2    IF
547 2096 2    .OUTPUT_RECATTR[FAT$V_RTYPE] NEQ FAT$C_FIXED AND    ! If not fixed length,
548 2097 2    .OUTPUT_RECATTR[FAT$W_MAXREC] NEQ 0
549 2098 2    THEN
550 2099 2    FAO_(', maximum !UL byte!%S', .OUTPUT_RECATTR[FAT$W_MAXREC]);
551 2100 2    EOL_();
552 2101 2
553 2102 2
554 2103 2    ! Format tenth line.
555 2104 2
556 2105 2    FAO_(' Record attributes: ');
557 2106 2    IF .OUTPUT_RECATTR[FAT$V_IMPLIEDCC]
558 2107 2    THEN
559 2108 2    FAO_('Carriage return')
560 2109 2    ELSE IF .OUTPUT_RECATTR[FAT$V_FORTRANCC]
561 2110 2    THEN
562 2111 2    FAO_('Fortran carriage control')
563 2112 2    ELSE IF .OUTPUT_RECATTR[FAT$V_PRINTCC]
564 2113 2    THEN
565 2114 2    FAO_('Print file format');
566 2115 2
567 2116 2
568 2117 2    IF .OUTPUT_RECATTR[FAT$B_RATTRIB] EQL FAT$M_NOSPAN
569 2118 2    THEN
570 2119 2    FAO_('Non-spanned')
571 2120 2    ELSE
572 2121 2    IF .OUTPUT_RECATTR[FAT$V_NOSPAN]
573 2122 2    THEN
574 2123 2    FAO_(', Non-spanned');
575 2124 2
576 2125 2
577 2126 2    IF .OUTPUT_RECATTR[FAT$B_RATTRIB] EQL 0
578 2127 2    THEN
579 2128 2    FAO_('None');
580 2129 2
581 2130 2

```

: 582
: 583
: 584

2131 2 EOL_(:);
2132 2 EOL_(:);
2133 1 END;

21	2F	4C	55	23	21	20	20	20	20	20	2A	32	32	21	03	000E3	P.AAH:	.ASCII	<3>\!AS\
65	72	43	20	20	20	20	20	20	3E	21	4C	55	21	3C	23	000E7	P.AAI:	.ASCII	<5>\!22* \
20	20	20	3A	72	65	6E	77	4F	20	2A	32	32	21	28	000ED	P.AAJ:	.ASCII	<6>\!21AS \	
20	20	20	5D	57	4F	33	21	2C	57	4F	33	21	5B	20	000F4	P.AAK:	.ASCII	\(Size: !#UL/!#<!UL!> Created: !17%\	
20	20	20	5D	57	4F	33	21	2C	57	4F	33	21	5B	20	00103				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00112				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	0011C	P.AAL:	.ASCII	\D\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	0011D	P.AAL:	.ASCII	\(!22* Owner: [!30W,!30W] Revised:\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	0012C				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	0013B				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00145	P.AAM:	.ASCII	\ \	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00146	P.AAM:	.ASCII	<16>\<None specified>\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	00155				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00157	P.AAN:	.ASCII	<11>\!17%D (!UL)\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00163	P.AAO:	.ASCII	\+!22* File ID: !14<(!UL,!UL,!UL)!> Expir\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	00172				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00181				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	0018B	P.AAP:	.ASCII	\es: \	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	0018F	P.AAP:	.ASCII	<16>\<None specified>\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	0019E				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	001A0	P.AAQ:	.ASCII	<5>\!17%D\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	001A6	P.AAR:	.ASCII	<14>\!46* Backup: \	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	001B5	P.AAS:	.ASCII	<16>\<No backup done>\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	001C4				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	001C6	P.AAT:	.ASCII	<5>\!17%D\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	001CC	P.AAU:	.ASCII	<22>\ File protection: \	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	001DB				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	001E3	P.AAV:	.ASCII	<22>\ File organization: \	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	001F2				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	001FA	P.AAW:	.ASCII	<10>\Sequential\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	00205	P.AAX:	.ASCII	\%Relative, maximum record number = !UL\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00214				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00223				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	0022B	P.AAY:	.ASCII	<7>\Indexed\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00233	P.AAZ:	.ASCII	<18>\Unknown type = !UL\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00242				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	00246	P.ABA:	.ASCII	\4 File attributes: Allocation = !UL,\	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00255				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00264				
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	0026E				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	0027B	P.ABB:	.ASCII	\ Extend = !UL\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	0027E	P.ABC:	.ASCII	<2>\, \	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	00287	P.ABD:	.ASCII	<8>\maximum \	
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	00296				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	00299	P.ABE:	.ASCII	<17>\bucket size = !UL\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	002A8				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	002B7				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	002B8	P.ABF:	.ASCII	<30>\!22* Global Buffer Count = !UL\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	002C7				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	002CE	P.ABG:	.ASCII	<21>\, Contiguous-best-try\	
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	002DB	P.ABH:	.ASCII	<12>\, Contiguous\	
20	3A	44	29	4C	55	21	28	20	44	25	37	31	21	0B	002DB				
55	21	2C	4C	55	21	2C	4C	55	21	28	3C	34	31	21	002DB				
65	69	66	69	63	65	70	73	20	65	6E	6F	4E	3C	10	002DB				

LIST V04-000

List a save set FULL_LISTING - format full Listing

61	6D	72	6F	66	20	64	72	6F	63	65	52	20	20	16	002E4	P.ABI:	.ASCII	<22>\	Record format:	\	:	
21	20	68	74	67	6E	65	6C	20	20	64	65	78	69	46	10	002F3					:	
73	64	72	6F	63	65	72	20	65	74	79	62	20	4C	55	0030A	P.ABJ:	.ASCII	<29>\	Fixed length !UL byte records\		:	
74	67	6E	65	6C	20	65	6C	62	61	69	72	61	56	0F	00319	P.ABK:	.ASCII	<15>\	Variable length\		:	
20	65	74	79	62	20	4C	55	21	20	2C	43	46	56	14	00329	P.ABL:	.ASCII	<20>\	VFC, !UL byte header\		:	
															68	00328					:	
															06	0033E	P.ABM:	.ASCII	<6>\	Stream\	:	
															09	00345	P.ABN:	.ASCII	<9>\	Stream_LF\	:	
															09	0034F	P.ABO:	.ASCII	<9>\	Stream_CR\	:	
															09	00359	P.ABP:	.ASCII	<9>\	Undefined\	:	
3D	20	65	70	79	74	20	6E	77	6F	6E	6B	6E	55	12	00363	P.ABQ:	.ASCII	<18>\	Unknown type = !UL\		:	
															21	20	00372				:	
20	4C	55	21	20	6D	75	6D	69	78	61	6D	20	2C	15	00376	P.ABR:	.ASCII	<21>\	, maximum !UL byte!%S\		:	
															62	00385					:	
69	72	74	74	61	20	64	72	6F	63	65	52	20	20	16	0038C	P.ABS:	.ASCII	<22>\	Record attributes:	\	:	
															62	0039B					:	
72	75	74	65	72	20	65	67	61	69	72	72	61	43	0F	003A3	P.ABT:	.ASCII	<15>\	Carriage return\		:	
															6E	003B2					:	
61	69	72	72	61	63	20	6E	61	72	74	72	6F	46	18	003B3	P.ABU:	.ASCII	<24>\	Fortran carriage control\		:	
															67	003C2					:	
72	6F	66	20	65	6C	69	66	20	74	6E	69	72	50	11	003CC	P.ABV:	.ASCII	<17>\	Print file format\		:	
															6D	003DB					:	
															0B	003DE	P.ABW:	.ASCII	<11>\	Non-spanned\		:
															0D	003FA	P.ABX:	.ASCII	<13>\	, Non-spanned\		:
64	65	6E	6E	61	70	73	2D	6E	6F	4E	20	2C	0D	003FA	P.ABX:	.ASCII	<13>\	, Non-spanned\		:		
															04	003F8	P.ABY:	.ASCII	<4>\	None\		:

003C 00000 FULL_LISTING:

55	FBFD	CF	9E	00002	.WORD	Save R2,R3,R4,R5	1904									
54	00000000	EF	9E	00007	MOVAB	LIST FAO, R5	:									
53	07	A4	10	9C	0000E	MOVAB	OUTPUT_RECATTR+1, R4	1941								
			A4	B5	00013	ROTL	#16, OUTPUT_RECATTR+8, BLOCKS	1942								
			06	12	00016	TSTW	OUTPUT_RECATTR+12	:								
			53	D5	00018	BNEQ	1\$:								
			02	13	0001A	TSTL	BLOCKS	:								
			53	D7	0001C	BEQL	1\$	1943								
0001869F	8F	53	D1	0001E	1\$:	DECL	BLOCKS	1949								
			0A	14	00025	CMPL	BLOCKS, #99999	:								
0001869F	8F	4B	A4	D1	00027	BGTR	2\$	1950								
			05	15	0002F	CMPL	OUTPUT_ATTBUF+92, #99999	:								
			52	0A	D0	00031	2\$:	1952								
			03	11	00034	BLEQ	3\$:								
			52	05	D0	00036	3\$:	1954								
			15	EF	A4	B1	00039	4\$:	1959							
					17	1B	0003D	:								
					EF	A4	9F	0003F	PUSHAB	OUTPUT_FILENAME	1962					
					FCA0	CF	9F	00042	PUSHAB	P.AAH	:					
						02	FB	00046	CALLS	#2, LIST_FAO	1963					
						30	A5	00	FB	00049	CALLS	#0, LIST_EOL	1964			
								CF	9F	0004D	PUSHAB	P.AAI	:			
								65	01	FB	00051	CALLS	#1, LIST_FAO	1959		
									0A	11	00054	BRB	6\$:		
									EF	A4	9F	00056	5\$:	PUSHAB	OUTPUT_FILENAME	1967

		FC93	CF	9F	00059	PUSHAB	P.AAJ		
	65		02	FB	0005D	CALLS	#2, LIST_FAO		
		1F	A4	9F	00060	6\$: PUSHAB	OUTPUT_ATTBUF+48		1973
		4B	A4	DD	00063	PUSHL	OUTPUT_ATTBUF+92		
			52	DD	00066	PUSHL	DIGITS		
			0C	BB	00068	PUSHR	#*M<R2,R3>		
		FC89	CF	9F	0006A	PUSHAB	P.AAK		
	65		06	FB	0006E	CALLS	#6, LIST_FAO		
30	A5		00	FB	00071	CALLS	#0, LIST_EOL		1974
	7E	4F	A4	3C	00075	MOVZWL	OUTPUT_ATTBUF+96, -(SP)		1981
	7E	51	A4	3C	00079	MOVZWL	OUTPUT_ATTBUF+98, -(SP)		
		FC9F	CF	9F	0007D	PUSHAB	P.AAL		
	65		03	FB	00081	CALLS	#3, LIST_FAO		
		27	A4	D5	00084	TSTL	OUTPUT_ATTBUF+56		1983
			09	12	00087	BNEQ	7\$		
		FCBC	CF	9F	00089	PUSHAB	P.AAM		1985
	65		01	FB	0008D	CALLS	#1, LIST_FAO		
			0E	11	00090	BRB	8\$		
	7E	61	A4	3C	00092	7\$: MOVZWL	OUTPUT_ATTBUF+114, -(SP)		1989
		27	A4	9F	00096	PUSHAB	OUTPUT_ATTBUF+56		
		FCBD	CF	9F	00099	PUSHAB	P.AAN		
	65		03	FB	0009D	CALLS	#3, LIST_FAO		
30	A5		00	FB	000A0	8\$: CALLS	#0, LIST_EOL		1990
	7E	43	A4	9A	000A4	MOVZBL	OUTPUT_ATTBUF+84, -(SP)		1998
	7E	41	A4	3C	000A8	MOVZWL	OUTPUT_ATTBUF+82, -(SP)		
	50	3F	A4	3C	000AC	MOVZWL	OUTPUT_ATTBUF+80, R0		
	51	44	A4	9A	000B0	MOVZBL	OUTPUT_ATTBUF+85, R1		
51	51		10	78	000B4	ASHL	#16, RT, R1		
			6140	9F	000B8	PUSHAB	(R1)[R0]		
		FCA7	CF	9F	000BB	PUSHAB	P.AAO		
	65		04	FB	000BF	CALLS	#4, LIST_FAO		
		2F	A4	D5	000C2	TSTL	OUTPUT_ATTBUF+64		2000
			09	12	000C5	BNEQ	9\$		
		FCC7	CF	9F	000C7	PUSHAB	P.AAP		2002
	65		01	FB	000CB	CALLS	#1, LIST_FAO		
			0A	11	000CE	BRB	10\$		
		2F	A4	9F	000D0	9\$: PUSHAB	OUTPUT_ATTBUF+64		2005
		FCCC	CF	9F	000D3	PUSHAB	P.AAQ		
	65		02	FB	000D7	CALLS	#2, LIST_FAO		
30	A5		00	FB	000DA	10\$: CALLS	#0, LIST_EOL		2006
	7E	43	A4	9A	000DE	MOVZBL	OUTPUT_ATTBUF+84, -(SP)		2014
	7E	41	A4	3C	000E2	MOVZWL	OUTPUT_ATTBUF+82, -(SP)		
	50	3F	A4	3C	000E6	MOVZWL	OUTPUT_ATTBUF+80, R0		
	51	44	A4	9A	000EA	MOVZBL	OUTPUT_ATTBUF+85, R1		
51	51		10	78	000EE	ASHL	#16, RT, R1		
			6140	9F	000F0	PUSHAB	(R1)[R0]		
		FCB0	CF	9F	000F5	PUSHAB	P.AAR		
	65		04	FB	000F9	CALLS	#4, LIST_FAO		
		37	A4	D5	000F7	TSTL	OUTPUT_ATTBUF+72		2016
			09	12	000FF	BNEQ	11\$		
		FCB3	CF	9F	00101	PUSHAB	P.AAS		2018
	65		01	FB	00105	CALLS	#1, LIST_FAO		
			0A	11	00108	BRB	12\$		
		37	A4	9F	0010A	11\$: PUSHAB	OUTPUT_ATTBUF+72		2021
		FCB8	CF	9F	0010D	PUSHAB	P.AAT		
	65		02	FB	00111	CALLS	#2, LIST_FAO		
30	A5		00	FB	00114	12\$: CALLS	#0, LIST_EOL		2022

			FCB3	CF 9F 00118	PUSHAB P.AAU	2027
		65		01 FB 0011C	CALLS #1, LIST_FAO	
			44455752	8F DD 0011F	PUSHL #145395026	2028
		7E	5D	A4 3C 00125	MOVZWL OUTPUT_ATTBUF+110, -(SP)	
			00B0	02 FB 00129	CALLS #2, LIST_PROTECTION	
		30		00 FB 0012E	CALLS #0, LIST_EOL	2029
			FCB0	CF 9F 00132	PUSHAB P.AAV	2034
		65		01 FB 00136	CALLS #1, LIST_FAO	
52	FF		A4	04 EF 00139	EXTZV #4, #4, OUTPUT_RECATTR, R2	2035
		04		06 12 0013F	BNEQ 13\$	2037
			FCB8	CF 9F 00141	PUSHAB P.AAW	
				18 11 00145	BRB 15\$	
		01		52 D1 00147 13\$:	CMPL R2, #1	2038
				0A 12 0014A	BNEQ 14\$	
		7E	OF	A4 3C 0014C	MOVZWL OUTPUT_RECATTR+16, -(SP)	2039
			FCB4	CF 9F 00150	PUSHAB P.AAX	
				14 11 00154	BRB 17\$	
		02		52 D1 00156 14\$:	CMPL R2, #2	2040
				09 12 00159	BNEQ 16\$	
			FCCF	CF 9F 0015B	PUSHAB P.AAY	
		65		01 FB 0015F 15\$:	CALLS #1, LIST_FAO	
				09 11 00162	BRB 18\$	
				52 DD 00164 16\$:	PUSHL R2	2042
			FCCC	CF 9F 00166	PUSHAB P.AAZ	
		65		02 FB 0016A 17\$:	CALLS #2, LIST_FAO	
			30	A5 00 FB 0016D 18\$:	CALLS #0, LIST_EOL	2044
		7E	11	A4 3C 00171	MOVZWL OUTPUT_RECATTR+18, -(SP)	2051
			4B	A4 DD 00175	PUSHL OUTPUT_ATTBUF+92	
			FCCD	CF 9F 00178	PUSHAB P.ABA	
		65		03 FB 0017C	CALLS #3, LIST_FAO	
			0D	A4 95 0017F	TSTB OUTPUT_RECATTR+14	2052
				21 13 00182	BEQL 20\$	
			FCF6	CF 9F 00184	PUSHAB P.ABB	2055
		65		01 FB 00188	CALLS #1, LIST_FAO	
02	FF		A4	04 ED 0018B	CMPZV #4, #4, OUTPUT_RECATTR, #2	2056
		04		07 12 00191	BNEQ 19\$	
			FCEA	CF 9F 00193	PUSHAB P.ABC	
		65		01 FB 00197	CALLS #1, LIST_FAO	
		7E	0D	A4 9A 0019A 19\$:	MOVZBL OUTPUT_RECATTR+14, -(SP)	2057
			FCEB	CF 9F 0019E	PUSHAB P.ABD	
		65		02 FB 001A2	CALLS #2, LIST_FAO	
			30	A5 00 FB 001A5 20\$:	CALLS #0, LIST_EOL	2059
		7E	13	A4 3C 001A9	MOVZWL OUTPUT_RECATTR+20, -(SP)	2065
			FCEB	CF 9F 001AD	PUSHAB P.ABE	
		65		02 FB 001B1	CALLS #2, LIST_FAO	
	06	53	A4	05 E1 001B4	BBC #5, OUTPUT_FILECHAR, 21\$	2066
			FCFE	CF 9F 001B9	PUSHAB P.ABF	2068
				09 11 001BD	BRB 22\$	
			53	A4 95 001BF 21\$:	TSTB OUTPUT_FILECHAR	2069
				07 18 001C2	BGEQ 23\$	
			FD09	CF 9F 001C4	PUSHAB P.ABG	2071
		65		01 FB 001C8 22\$:	CALLS #1, LIST_FAO	
			07	53 A4 06 E1 001CB 23\$:	BBC #6, OUTPUT_FILECHAR, 24\$	2072
			FD0A	CF 9F 001D0	PUSHAB P.ABH	2074
		65		01 FB 001D4	CALLS #1, LIST_FAO	
			30	A5 00 FB 001D7 24\$:	CALLS #0, LIST_EOL	2075
			FD08	CF 9F 001DB	PUSHAB P.ABI	2080

52	FF	A4	65	01	FB	001DF	CALLS	#1, LIST FAO	2081	
			04	00	EF	001E2	EXTZV	#0, #4, OUTPUT_RECATTR, R2	2083	
			01	52	D1	001E8	CMPL	R2, #1	2084	
			7E	0A	12	001EB	BNEQ	25\$	2085	
				01	A4	3C	MOV7WL	OUTPUT_RECATTR+2, -(SP)	2086	
				FD09	CF	9F	PUSHAB	P.ABJ	2087	
					4E	11	BRB	33\$	2088	
			02	52	D1	001F7	CMPL	R2, #2	2089	
				06	12	001FA	BNEQ	26\$	2090	
				FD1C	CF	9F	PUSHAB	P.ABK	2091	
					38	11	BRB	31\$	2092	
			03	52	D1	00202	CMPL	R2, #3	2093	
				0A	12	00205	BNEQ	27\$	2094	
			7E	0E	A4	9A	MOVZSL	OUTPUT_RECATTR+15, -(SP)	2095	
				FD1D	CF	9F	PUSHAB	P.ABL	2096	
					34	11	BRB	33\$	2097	
			04	52	D1	00211	CMPL	R2, #4	2098	
				06	12	00214	BNEQ	28\$	2099	
				FD27	CF	9F	PUSHAB	P.ABM	2100	
					1E	11	BRB	31\$	2101	
			05	52	D1	0021C	CMPL	R2, #5	2102	
				06	12	0021F	BNEQ	29\$	2103	
				FD23	CF	9F	PUSHAB	P.ABN	2104	
					13	11	BRB	31\$	2105	
			06	52	D1	00227	CMPL	R2, #6	2106	
				06	12	0022A	BNEQ	30\$	2107	
				FD22	CF	9F	PUSHAB	P.ABO	2108	
					08	11	BRB	31\$	2109	
				52	D5	00232	TSTL	R2	2110	
				09	12	00234	BNEQ	32\$	2111	
				FD22	CF	9F	PUSHAB	P.ABP	2112	
			65	01	FB	0023A	CALLS	#1, LIST_FAO	2113	
				09	11	0023D	BRB	34\$	2114	
				52	DD	0023F	PUSHL	R2	2115	
				FD21	CF	9F	PUSHAB	P.ABQ	2116	
			65	02	FB	00245	CALLS	#2, LIST FAO	2117	
			04	00	ED	00248	CMPL	#0, #4, OUTPUT_RECATTR, #1	2118	
				10	13	0024E	BEQL	35\$	2119	
				0F	A4	B5	TSTW	OUTPUT_RECATTR+16	2120	
				0B	13	00253	BEQL	35\$	2121	
			7E	0F	A4	3C	MOVZWL	OUTPUT_RECATTR+16, -(SP)	2122	
				FD1C	CF	9F	PUSHAB	P.ABR	2123	
					02	FB	0025D	CALLS	#2, LIST_FAO	2124
			30	65	00	FB	00260	CALLS	#0, LIST_EOL	2125
			A5	FD27	CF	9F	PUSHAB	P.ABS	2126	
					01	FB	00268	CALLS	#1, LIST FAO	2127
			65	01	E1	0026B	BBC	#1, OUTPUT_RECATTR+1, 36\$	2128	
			64	FD33	CF	9F	PUSHAB	P.ABT	2129	
					11	11	BRB	38\$	2130	
			06	64	E9	00275	BLBC	OUTPUT_RECATTR+1, 37\$	2131	
				FD3A	CF	9F	PUSHAB	P.ABU	2132	
					08	11	BRB	38\$	2133	
				07	02	E1	BBC	#2, OUTPUT_RECATTR+1, 39\$	2134	
			64	FD49	CF	9F	PUSHAB	P.ABV	2135	
					01	FB	00286	CALLS	#1, LIST FAO	2136
			65	01	FB	00286	CALLS	#1, LIST FAO	2137	
			08	64	91	00289	CMPL	OUTPUT_RECATTR+1, #8	2138	
				06	12	0028C	BNEQ	40\$	2139	

LIST
V04-000

List a save set
FULL_LISTING - format full listing

		FD4F	CF 9F 0028E	PUSHAB P.ABW	: 2119
			08 11 00292	BRB 41\$: 2119
07	64		03 E1 00294 40\$:	BBC #3, OUTPUT_RECATTR+1, 42\$: 2121
		FD51	CF 9F 00298	PUSHAB P.ABX	: 2123
	65		01 FB 0029C 41\$:	CALLS #1, LIST_FAO	: 2123
			64 95 0029F 42\$:	TSTB OUTPUT_RECATTR+1	: 2126
			07 12 002A1	BNEQ 43\$: 2126
		FD54	CF 9F 002A3	PUSHAB P.ABY	: 2128
	65		01 FB 002A7	CALLS #1, LIST_FAO	: 2128
	30 A5		00 FB 002AA 43\$:	CALLS #0, LIST_EOL	: 2131
	30 A5		00 FB 002AE	CALLS #0, LIST_EOL	: 2132
			04 002B2	RET	: 2133

: Routine Size: 691 bytes, Routine Base: CODE + 03FD

```

: 586 2134 1 %SBTTL 'LIST_ONE_RECORD - list save set record'
: 587 2135 1 ROUTINE LIST_ONE_RECORD(REC,BCB): L_PS NOVALUE=
: 588 2136 1
: 589 2137 1 ++
: 590 2138 1
: 591 2139 1 FUNCTIONAL DESCRIPTION:
: 592 2140 1 This routine lists the contents of one save set record.
: 593 2141 1
: 594 2142 1 INPUT PARAMETERS:
: 595 2143 1 REC - Pointer to save set record.
: 596 2144 1 BCB - Pointer to BCB for buffer.
: 597 2145 1
: 598 2146 1 IMPLICIT INPUTS:
: 599 2147 1 NONE
: 600 2148 1
: 601 2149 1 OUTPUT PARAMETERS:
: 602 2150 1 NONE
: 603 2151 1
: 604 2152 1 IMPLICIT OUTPUTS:
: 605 2153 1 NONE
: 606 2154 1
: 607 2155 1 ROUTINE VALUE:
: 608 2156 1 NONE
: 609 2157 1
: 610 2158 1 SIDE EFFECTS:
: 611 2159 1 The listing is produced.
: 612 2160 1
: 613 2161 1 --
: 614 2162 1
: 615 2163 2 BEGIN
: 616 2164 2
: 617 2165 2 LITERAL
: 618 2166 2 DEVTYP_BUF_LEN = 6 ;
: 619 2167 2
: 620 2168 2 MAP
: 621 2169 2 REC: REF BBLOCK, ! Pointer to save set record
: 622 2170 2 BCB: REF BBLOCK; ! Pointer to buffer control block
: 623 2171 2
: 624 2172 2 L_DECL;
: 625 2173 2
: 626 2174 2 CASE .REC[BRH$W_RTYPE] FROM BRH$K_SUMMARY TO BRH$K_FID OF
: 627 2175 2 SET
: 628 2176 2
: 629 2177 2 [BRH$K_VBN, BRH$K_LBN, BRH$K_FID, OUTRANGE]:
: 630 2178 2 0;
: 631 2179 2
: 632 2180 2
: 633 2181 2 [BRH$K_SUMMARY]:
: 634 2182 2 BEGIN
: 635 2183 2 LOCAL
: 636 2184 2 VOLNUM;
: 637 2185 2
: 638 2186 2 GET_SUMMARY_ATTRIBUTES(.REC);
: 639 2187 2
: 640 2188 2 VOLNUM = .BBLOCK[BCB[BCB_BUFFER], BBH$W_VOLNUM];
: 641 2189 2 IF .QUAL[QUAL_OSAV] THEN VOLNUM = .RWSV_VOL_NUMBER;
: 642 2190 3

```

```

: 643      2191  3      IF .VOLNUM GTRU 1 THEN EOL_();
: 644      2192  3
: 645      2193  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_SSNAME], DSC$W_LENGTH] NEQ 0
: 646      2194  3      THEN
: 647      2195  4          BEGIN
: 648      2196  4          FAO_ ('Save set:           !AD',
: 649      2197  4          .BBLOCK[OUTPUT_ATTBUF[BSR_SSNAME], DSC$W_LENGTH],
: 650      2198  4          .BBLOCK[OUTPUT_ATTBUF[BSR_SSNAME], DSC$A_POINTER]);
: 651      2199  4          IF .VOLNUM GTRU 1 THEN FAO_ (' volume !UL', .VOLNUM);
: 652      2200  4          EOL_();
: 653      2201  3          END;
: 654      2202  3
: 655      2203  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_USERNAME], DSC$W_LENGTH] NEQ 0
: 656      2204  3      THEN
: 657      2205  4          BEGIN
: 658      2206  4          FAO_ ('Written by:           !AD',
: 659      2207  4          .BBLOCK[OUTPUT_ATTBUF[BSR_USERNAME], DSC$W_LENGTH],
: 660      2208  4          .BBLOCK[OUTPUT_ATTBUF[BSR_USERNAME], DSC$A_POINTER]);
: 661      2209  4          EOL_();
: 662      2210  3          END;
: 663      2211  3
: 664      2212  3      IF .OUTPUT_ATTBUF[BSR_USERUIC] NEQ 0
: 665      2213  3      THEN
: 666      2214  4          BEGIN
: 667      2215  4          FAO_ ('UIC:           [!OB,!OB]',
: 668      2216  4          .(OUTPUT_ATTBUF[BSR_USERUIC])<16,16>,
: 669      2217  4          .(OUTPUT_ATTBUF[BSR_USERUIC])<0,16>);
: 670      2218  4          EOL_();
: 671      2219  3          END;
: 672      2220  3
: 673      2221  3      IF .(OUTPUT_ATTBUF[BSR_DATE]) NEQ 0
: 674      2222  3      THEN
: 675      2223  4          BEGIN
: 676      2224  4          FAO_ ('Date:           !XD',
: 677      2225  4          .OUTPUT_ATTBUF[BSR_DATE]);
: 678      2226  4          EOL_();
: 679      2227  3          END;
: 680      2228  3
: 681      2229  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_COMMAND], DSC$W_LENGTH] NEQ 0
: 682      2230  3      THEN
: 683      2231  4          BEGIN
: 684      2232  4          LOCAL
: 685      2233  4          LENGTH,
: 686      2234  4          ADDRESS;
: 687      2235  4          LITERAL
: 688      2236  4          CHUNK_SIZE = LIST_SIZE - 19;
: 689      2237  4
: 690      2238  4          LENGTH = .BBLOCK[OUTPUT_ATTBUF[BSR_COMMAND], DSC$W_LENGTH];
: 691      2239  4          ADDRESS = .BBLOCK[OUTPUT_ATTBUF[BSR_COMMAND], DSC$A_POINTER];
: 692      2240  4          WHILE .LENGTH GTR 0 DO
: 693      2241  5              BEGIN
: 694      2242  5                  IF .LENGTH EQL .BBLOCK[OUTPUT_ATTBUF[BSR_COMMAND], DSC$W_LENGTH]
: 695      2243  5                      THEN FAO_ ('Command:
: 696      2244  5                      ELSE FAO_ ('!19* ');
: 697      2245  5                  FAO_ ('!AD',
: 698      2246  5                  .LENGTH,
: 699      2247  5                  .ADDRESS);

```

```

: 700      2248  5      EOL ();
: 701      2249  5      LENGTH = .LENGTH - CHUNK_SIZE;
: 702      2250  5      ADDRESS = .ADDRESS + CHUNK_SIZE;
: 703      2251  4      END;
: 704      2252  3      END;
: 705      2253  3
: 706      2254  3      IF .OUTPUT_ATTBUF[BSR_OPSYS] EQL BACKUP$K_OPSYS
: 707      2255  3      THEN
: 708      2256  4      BEGIN
: 709      2257  4      FAO ('Operating system: VAX/VMS');
: 710      2258  4      IF .BBLOCK[OUTPUT_ATTBUF[BSR_SYSVER], DSC$W_LENGTH] NEQ 0
: 711      2259  4      THEN
: 712      P 2260  4      FAO (' version !AD',
: 713      P 2261  4      .BBLOCK[OUTPUT_ATTBUF[BSR_SYSVER], DSC$W_LENGTH],
: 714      2262  4      .BBLOCK[OUTPUT_ATTBUF[BSR_SYSVER], DSC$A_POINTER]);
: 715      2263  4      EOL ();
: 716      2264  3      END;
: 717      2265  3
: 718      2266  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_BACKVER], DSC$W_LENGTH] NEQ 0
: 719      2267  3      THEN
: 720      2268  4      BEGIN
: 721      P 2269  4      FAO ('BACKUP version: !AD',
: 722      P 2270  4      .BBLOCK[OUTPUT_ATTBUF[BSR_BACKVER], DSC$W_LENGTH],
: 723      2271  4      .BBLOCK[OUTPUT_ATTBUF[BSR_BACKVER], DSC$A_POINTER]);
: 724      2272  4      EOL ();
: 725      2273  3      END;
: 726      2274  3
: 727      2275  3      IF .OUTPUT_ATTBUF[BSR_SIR] NEQ 0
: 728      2276  3      THEN
: 729      2277  4      BEGIN
: 730      P 2278  4      FAO ('CPU ID register: !XL',
: 731      2279  4      .OUTPUT_ATTBUF[BSR_SIR]);
: 732      2280  4      EOL ();
: 733      2281  3      END;
: 734      2282  3
: 735      2283  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_NODENAME], DSC$W_LENGTH] NEQ 0
: 736      2284  3      THEN
: 737      2285  4      BEGIN
: 738      P 2286  4      FAO ('Node name: !AD',
: 739      P 2287  4      .BBLOCK[OUTPUT_ATTBUF[BSR_NODENAME], DSC$W_LENGTH],
: 740      2288  4      .BBLOCK[OUTPUT_ATTBUF[BSR_NODENAME], DSC$A_POINTER]);
: 741      2289  4      EOL ();
: 742      2290  3      END;
: 743      2291  3
: 744      2292  3      IF .BBLOCK[OUTPUT_ATTBUF[BSR_DRIVEID], DSC$W_LENGTH] NEQ 0
: 745      2293  3      THEN
: 746      2294  4      BEGIN
: 747      P 2295  4      FAO ('Written on: !AD',
: 748      P 2296  4      .BBLOCK[OUTPUT_ATTBUF[BSR_DRIVEID], DSC$W_LENGTH],
: 749      2297  4      .BBLOCK[OUTPUT_ATTBUF[BSR_DRIVEID], DSC$A_POINTER]);
: 750      2298  4      EOL ();
: 751      2299  3      END;
: 752      2300  3
: 753      2301  3      IF .OUTPUT_ATTBUF[BSR_BLOCKSIZE] NEQ 0
: 754      2302  3      THEN
: 755      2303  4      BEGIN
: 756      P 2304  4      FAO ('Block size: !UL',

```

```
757 2305 4 .OUTPUT_ATTBUF[BSR_BLOCKSIZ]);
758 2306 4 EOL_();
759 2307 3 END;
760 2308 3
761 2309 3 IF .OUTPUT_ATTBUF[BSR_XORSIZE] NEQ 0
762 2310 3 THEN
763 2311 4 BEGIN
764 P 2312 4 FAO_ ('Group size: !UL',
765 2313 4 .OUTPUT_ATTBUF[BSR_XORSIZE]);
766 2314 4 EOL_();
767 2315 3 END;
768 2316 3
769 2317 3 IF .OUTPUT_ATTBUF[BSR_BUFFERS] NEQ 0
770 2318 3 THEN
771 2319 4 BEGIN
772 P 2320 4 FAO_ ('Buffer count: !UL',
773 2321 4 .OUTPUT_ATTBUF[BSR_BUFFERS]);
774 2322 4 EOL_();
775 2323 3 END;
776 2324 3
777 2325 3 IF .OUTPUT_ATTBUF[BSR_NVOLS] NEQ 0
778 2326 3 THEN
779 2327 4 BEGIN
780 2328 4 EOL_();
781 2329 4 FAO_ ('Image save of volume set');
782 2330 4 EOL_();
783 2331 4
784 2332 4 IF .BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$W_LENGTH] NEQ 0
785 2333 4 THEN IF (.BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$A_POINTER]) < 0,8 > GTRU %C' '
786 2334 4 THEN
787 2335 5 BEGIN
788 P 2336 5 FAO_ ('Volume set name: !AD',
789 P 2337 5 .BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$W_LENGTH],
790 2338 5 .BBLOCK[OUTPUT_ATTBUF[BSR_VOLSETNAM], DSC$A_POINTER]);
791 2339 5 EOL_();
792 2340 4 END;
793 2341 4
794 P 2342 4 FAO_ ('Number of volumes: !UL',
795 2343 4 .OUTPUT_ATTBUF[BSR_NVOLS]);
796 2344 4 EOL_();
797 2345 3 END;
798 2346 3
799 2347 3 IF .BBLOCK[OUTPUT_ATTBUF[BSR_CRYPTDATA], BSASB_CRYPTYP] NEQU 0
800 2348 3 THEN
801 2349 4 BEGIN
802 2350 4 FAO_ ('Saveset Encrypted');
803 2351 4 EOL_();
804 2352 3 END;
805 2353 3
806 2354 3 EOL_();
807 2355 2 END;
808 2356 2
809 2357 2
810 2358 2 [BRH$K VOLUME]:
811 2359 2 BEGIN
812 2360 2 GET_VOLUME_ATTRIBUTES(.REC);
813 2361 2
```



```

814 2362 3
815 2363 3
816 2364 3
817 2365 3
818 2366 3
819 2367 3
820 2368 3
821 2369 3
822 2370 3
823 2371 4
824 P 2372 4
825 2373 4
826 2374 4
827 2375 4
828 2376 3
829 2377 3
830 2378 3
831 2379 4
832 P 2380 4
833 P 2381 4
834 2382 4
835 2383 4
836 2384 3
837 2385 3
838 2386 3
839 2387 3
840 2388 4
841 P 2389 4
842 P 2390 4
843 2391 4
844 2392 4
845 2393 3
846 2394 3
847 2395 3
848 2396 3
849 2397 4
850 P 2398 4
851 P 2399 4
852 2400 4
853 2401 4
854 2402 3
855 2403 3
856 2404 3
857 2405 3
858 2406 4
859 P 2407 4
860 2408 4
861 2409 4
862 2410 3
863 2411 3
864 2412 3
865 2413 3
866 2414 4
867 P 2415 4
868 2416 4
869 2417 4
870 2418 3

FAO_('Volume attributes');
IF .OUTPUT_ATTBUF[VSR_RVN] NEQ 0
THEN
  FAO_(' for relative volume !UL', .OUTPUT_ATTBUF[VSR_RVN]);
EOL_();

IF .OUTPUT_ATTBUF[VSR_VOLSTRUCT] NEQ 0
THEN
  BEGIN
    FAO_('Structure level: !UL',
      .(OUTPUT_ATTBUF[VSR_VOLSTRUCT])<8,8>);
    EOL_();
  END;

IF .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH] NEQ 0
THEN
  BEGIN
    FAO_('Label: !AD',
      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER]);
    EOL_();
  END;

IF .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH] NEQ 0
THEN
  BEGIN
    FAO_('Owner: !AD',
      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH],
      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$A_POINTER]);
    EOL_();
  END;

IF .OUTPUT_ATTBUF[VSR_VOLOWNER] NEQ 0
THEN
  BEGIN
    FAO_('Owner UIC: [!OB,!OB]',
      .(OUTPUT_ATTBUF[VSR_VOLOWNER])<16,16>,
      .(OUTPUT_ATTBUF[VSR_VOLOWNER])<0,16>);
    EOL_();
  END;

IF .(OUTPUT_ATTBUF[VSR_VOLDATE]) NEQ 0
THEN
  BEGIN
    FAO_('Creation date: !%D',
      OUTPUT_ATTBUF[VSR_VOLDATE]);
    EOL_();
  END;

IF .OUTPUT_ATTBUF[VSR_SERIALNUM] NEQ 0
THEN
  BEGIN
    FAO_('Serial number: !OL',
      OUTPUT_ATTBUF[VSR_SERIALNUM]);
    EOL_();
  END;

```

```

: 871      2419      3
: 872      2420      3
: 873      2421      3
: 874      2422      4
: 875      2423      4
: 876      2424      4
: 877      2425      4
: 878      2426      3
: 879      2427      3
: 880      2428      3
: 881      2429      3
: 882      2430      3
: 883      2431      3
: 884      2432      3
: 885      2433      3
: 886      2434      3
: 887      2435      3
: 888      2436      3
: 889      2437      3
: 890      2438      3
: 891      2439      3
: 892      2440      3
: 893      2441      3
: 894      2442      3
: 895      2443      3
: 896      2444      3
: 897      2445      3
: 898      2446      3
: 899      2447      3
: 900      2448      3
: 901      2449      3
: 902      2450      3
: 903      2451      3
: 904      2452      3
: 905      2453      3
: 906      2454      3
: 907      2455      3
: 908      2456      3
: 909      2457      3
: 910      2458      3
: 911      2459      3
: 912      2460      3
: 913      2461      3
: 914      2462      3
: 915      2463      3
: 916      2464      3
: 917      2465      3
: 918      2466      3
: 919      2467      4
: 920      2468      4
: 921      2469      4
: 922      2470      4
: 923      2471      3
: 924      2472      3
: 925      2473      3
: 926      2474      3
: 927      2475      4

IF .OUTPUT_ATTBUF[VSR_VOLSIZE] NEQ 0
THEN
  BEGIN
    FAO_ ('Total blocks:      !UL',
      .OUTPUT_ATTBUF[VSR_VOLSIZE]);
    EOL_ ();
  END;

P 2428      3
FAO_ ('Access count:      !UL',
  .OUTPUT_ATTBUF[VSR_LRU_LIM]);
EOL_ ();

P 2432      3
FAO_ ('Cluster size:      !UL',
  .OUTPUT_ATTBUF[VSR_CLUSTER]);
EOL_ ();

888      2436      3
FAO_ ('Data check:      ');
889      2437      3
IF (.OUTPUT_ATTBUF[VSR_VOLCHAR] AND HM2$M_READCHECK) EQL 0
890      2438      3
  THEN FAO_ ('No ');
891      2439      3
FAO_ ('Read, ');
892      2440      3
IF (.OUTPUT_ATTBUF[VSR_VOLCHAR] AND HM2$M_WRITECHECK) EQL 0
893      2441      3
  THEN FAO_ ('No ');
894      2442      3
FAO_ ('Write');
895      2443      3
EOL_ ();

P 2445      3
FAO_ ('Extension size:      !UL',
  .OUTPUT_ATTBUF[VSR_EXTEND]);
EOL_ ();

901      2449      3
FAO_ ('File protection:      ');
902      2450      3
LIST_PROTECTION(.OUTPUT_ATTBUF[VSR_FILEPROT], 'RWED');
903      2451      3
EOL_ ();

P 2453      3
FAO_ ('Maximum files:      !UL',
  .OUTPUT_ATTBUF[VSR_MAXFILES]);
EOL_ ();

909      2457      3
FAO_ ('Volume protection:      ');
910      2458      3
LIST_PROTECTION(.OUTPUT_ATTBUF[VSR_PROTECT], 'RWCD');
911      2459      3
EOL_ ();

P 2461      3
FAO_ ('Windows:      !UL',
  .OUTPUT_ATTBUF[VSR_WINDOW]);
EOL_ ();

917      2465      3
IF (.OUTPUT_ATTBUF[VSR_RETAINMIN]) NEQ 0
918      2466      3
THEN
919      2467      4
  BEGIN
920      2468      4
    FAO_ ('Minimum retention: !XD',
921      2469      4
      .OUTPUT_ATTBUF[VSR_RETAINMIN]);
922      2470      4
    EOL_ ();
923      2471      3
  END;

925      2473      3
IF (.OUTPUT_ATTBUF[VSR_RETAINMAX]) NEQ 0
926      2474      3
THEN
927      2475      4
  BEGIN
```

```

: 928 P 2476 4          FAO_('Maximum retention: !%D',
: 929   2477 4          _OUTPUT_ATTBUF[VSR_RETAINMAX]);
: 930   2478 4          EOL_();
: 931   2479 3          END;
: 932   2480 3
: 933   2481 3          EOL_();
: 934   2482 2          END;
: 935   2483 2
: 936   2484 2
: 937   2485 2          [BRH$K_PHYSVOL]:
: 938   2486 2          BEGIN
: 939   2487 2          GET_PHYSVOL_ATTRIBUTES(.REC);
: 940   2488 3
: 941   2489 3
: 942   2490 3          FAO_('Physical save of volume');
: 943   2491 3          EOL_();
: 944   2492 3
: 945   2493 3          IF .BBLOCK[OUTPUT_ATTBUF[PVA_DEVNAM], DSC$W_LENGTH] NEQ 0
: 946   2494 3          THEN
: 947   2495 4          BEGIN
: 948   2496 4          FAO_('Device:          !AD',
: 949 P 2497 4          _BBLOCK[OUTPUT_ATTBUF[PVA_DEVNAM], DSC$W_LENGTH],
: 950   2498 4          _BBLOCK[OUTPUT_ATTBUF[PVA_DEVNAM], DSC$A_POINTER]);
: 951   2499 4          EOL_();
: 952   2500 3          END;
: 953   2501 3
: 954   2502 3          IF .OUTPUT_ATTBUF[PVA_DEVTYP] LEQU DTS_RZF01
: 955   2503 3          THEN
: 956   2504 3          | If the devtype is this small we must be dealing with
: 957   2505 3          | the old form of the attribute which was simply the 1 byte
: 958   2506 3          | devtyp from the UCB
: 959   2507 3          |
: 960   2508 4          BEGIN
: 961   2509 4          BIND
: 962   2510 4          DEVTYPES = UPLIT (
: 963   2511 4          UPLIT BYTE (%ASCIC ''),          | 0
: 964   2512 4          UPLIT BYTE (%ASCIC 'RK06'),      | 1
: 965   2513 4          UPLIT BYTE (%ASCIC 'RK07'),      | 2
: 966   2514 4          UPLIT BYTE (%ASCIC 'RP04'),      | 3
: 967   2515 4          UPLIT BYTE (%ASCIC 'RP05'),      | 4
: 968   2516 4          UPLIT BYTE (%ASCIC 'RP06'),      | 5
: 969   2517 4          UPLIT BYTE (%ASCIC 'RM03'),      | 6
: 970   2518 4          UPLIT BYTE (%ASCIC 'RP07'),      | 7
: 971   2519 4          UPLIT BYTE (%ASCIC 'RP07HT'),    | 8
: 972   2520 4          UPLIT BYTE (%ASCIC 'RL01'),      | 9
: 973   2521 4          UPLIT BYTE (%ASCIC 'RL02'),      | 10
: 974   2522 4          UPLIT BYTE (%ASCIC 'RX02'),      | 11
: 975   2523 4          UPLIT BYTE (%ASCIC 'RX04'),      | 12
: 976   2524 4          UPLIT BYTE (%ASCIC 'RM80'),      | 13
: 977   2525 4          UPLIT BYTE (%ASCIC 'TU58'),      | 14
: 978   2526 4          UPLIT BYTE (%ASCIC 'RM05'),      | 15
: 979   2527 4          UPLIT BYTE (%ASCIC 'RX01'),      | 16
: 980   2528 4          UPLIT BYTE (%ASCIC 'ML11'),      | 17
: 981   2529 4          UPLIT BYTE (%ASCIC 'RB02'),      | 18
: 982   2530 4          UPLIT BYTE (%ASCIC 'RB80'),      | 19
: 983   2531 4          UPLIT BYTE (%ASCIC 'RA80'),      | 20
: 984   2532 4          UPLIT BYTE (%ASCIC 'RA81'),      | 21

```

```

: 985      2533  4          UPLIT BYTE (%ASCIC 'RA60'),      ! 22
: 986      2534  4          UPLIT BYTE (%ASCIC 'RZ01'),      ! 23
: 987      2535  4          UPLIT BYTE (%ASCIC 'RZF01'),      ! 24
: 988      2536  4          : VECTOR;
: 989      2537  4
: 990      P 2538  4          FAO_('Device type:      !AC',
: 991      2539  4          .DEVTYPE$[.OUTPUT_ATTBUF[PVA_DEVTYP]]);
: 992      2540  4          EOL_();
: 993      2541  4          END;
: 994      2542  3          ELSE IF .OUTPUT_ATTBUF[PVA_DEVTYP] GTRU %X'FF'
: 995      2543  3          THEN
: 996      2544  4          BEGIN
: 997      2545  4          ! If the devtyp attribute is greater than 1 byte we must be
: 998      2546  4          ! dealing with the new form which is the longword MEDIA_ID
: 999      2547  4          ! from the UCB. This longword must now be decoded.
: 1000     2548  4
: 1001     2549  4          LOCAL
: 1002     2550  4          LENGTH      : LONG      INITIAL (DEVTYPE_BUF_LEN),
: 1003     2551  4          BUFFER      : VECTOR[DEVTYPE_BUF_LEN, BYTE];
: 1004     2552  4
: 1005     2553  4          DECODE_DEVTYP ( .OUTPUT_ATTBUF[PVA_DEVTYP], LENGTH, BUFFER,
: 1006     2554  4          0, 0 );
: 1007     2555  4
: 1008     P 2556  4          FAO_('Device type:      !AF',
: 1009     2557  4          .LENGTH, BUFFER );
: 1010     2558  4          EOL_();
: 1011     2559  4
: 1012     2560  3          END .
: 1013     2561  3
: 1014     2562  3          IF .OUTPUT_ATTBUF[PVA_SERIAL] NEQ 0
: 1015     2563  3          THEN
: 1016     2564  4          BEGIN
: 1017     P 2565  4          FAO_('Serial number:      !OL',
: 1018     2566  4          .OUTPUT_ATTBUF[PVA_SERIAL]);
: 1019     2567  4          EOL_();
: 1020     2568  3          END;
: 1021     2569  3
: 1022     2570  3          IF .BBLOCK[OUTPUT_ATTBUF[PVA_LABEL], DSC$W_LENGTH] NEQ 0
: 1023     2571  3          THEN
: 1024     2572  4          BEGIN
: 1025     P 2573  4          FAO_('Label:      !AD',
: 1026     P 2574  4          .BBLOCK[OUTPUT_ATTBUF[PVA_LABEL], DSC$W_LENGTH],
: 1027     2575  4          .BBLOCK[OUTPUT_ATTBUF[PVA_LABEL], DSC$A_POINTER]);
: 1028     2576  4          EOL_();
: 1029     2577  3          END;
: 1030     2578  3
: 1031     2579  3          IF .OUTPUT_ATTBUF[PVA_MAXBLOCK] NEQ 0
: 1032     2580  3          THEN
: 1033     2581  4          BEGIN
: 1034     P 2582  4          FAO_('Total blocks:      !UL',
: 1035     2583  4          .OUTPUT_ATTBUF[PVA_MAXBLOCK]);
: 1036     2584  4          EOL_();
: 1037     2585  3          END;
: 1038     2586  3
: 1039     2587  3          IF .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSC$W_LENGTH] NEQ 0
: 1040     2588  3          THEN
: 1041     2589  4          BEGIN

```

```

: 1042 2590 4 LOCAL P: REF VECTOR;
: 1043 2591 4 P = .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSCSA_POINTER];
: 1044 2592 4 WHILE .P LSSA
: 1045 2593 4 .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSCSA_POINTER] +
: 1046 2594 4 .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSCSW_LENGTH]
: 1047 2595 4 DO
: 1048 2596 4 BEGIN
: 1049 2597 5 IF .P EQL .BBLOCK[OUTPUT_ATTBUF[PVA_BADBLOCK], DSCSA_POINTER]
: 1050 2598 5 THEN FAO ('Bad block numbers: ')
: 1051 2599 5 ELSE FAO ('!19* ');
: 1052 2600 5 FAO ('!UL', .P[0]);
: 1053 2601 5 IF .P[1] NEQ 1 THEN FAO (' to !UL', .P[0] + .P[1] - 1);
: 1054 2602 5 EOL ();
: 1055 2603 5 P = .P + 8;
: 1056 2604 4 END;
: 1057 2605 4 END;
: 1058 2606 4
: 1059 2607 4 EOL ();
: 1060 2608 4 END;
: 1061 2609 4
: 1062 2610 4 [BRH$K FILE]:
: 1063 2611 4 BEGIN
: 1064 2612 4 LOCAL
: 1065 2613 4 BLOCKS; ! File size
: 1066 2614 4
: 1067 2615 4
: 1068 2616 4
: 1069 2617 4 GET_FILE_ATTRIBUTES(.REC);
: 1070 2618 4
: 1071 2619 4
: 1072 2620 4 BLOCKS = ROT(.OUTPUT_RECATTR[FAT$E_FBLK], 16);
: 1073 2621 4 IF .OUTPUT_RECATTR[FAT$W_FFBYTE] EQL 0 AND .BLOCKS NEQ 0
: 1074 2622 4 THEN BLOCKS = .BLOCKS - 1;
: 1075 2623 4
: 1076 2624 4
: 1077 2625 4 LIST_TOTFILES = .LIST_TOTFILES + 1;
: 1078 2626 4 LIST_TOTSIZE = .LIST_TOTSIZE + .BLOCKS;
: 1079 2627 4
: 1080 2628 4
: 1081 2629 4 IF .QUAL[QUAL_FULL]
: 1082 2630 4 THEN
: 1083 2631 4 FULL_LISTING()
: 1084 2632 4 ELSE
: 1085 2633 4 BEGIN
: 1086 2634 4 IF .OUTPUT_FILENAME[DSCSW_LENGTH] GTR 52
: 1087 2635 4 THEN
: 1088 2636 4 BEGIN
: 1089 2637 4 FAO ('!AS', OUTPUT_FILENAME);
: 1090 2638 4 EOL ();
: 1091 2639 4 FAO ('!52* ');
: 1092 2640 4 END
: 1093 2641 4 ELSE
: 1094 2642 4 FAO ('!52AS', OUTPUT_FILENAME);
: 1095 2643 4
: 1096 2644 4 FAO ('!9SL !17%D', .BLOCKS, OUTPUT_ATTBUF[FAR_CREDATE]);
: 1097 2645 4 EOL ();
: 1098 2646 4 END;

```

LIST
V04-000

List a save set
LIST_ONE_RECORD - list save set record

M 15
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 36
(7)

: 1099 2647 2 END;
: 1100 2648 2
: 1101 2649 2
: 1102 2650 2 TES;
: 1103 2651 1 END;

20	20	20	20	20	3A	74	65	73	20	65	76	61	53	16	006B0	P.ABZ:	.ASCII	<22>\Save set:	!AD\	
							44	41	21	20	20	20	20	20	006BF					
20	20	4C	55	21	20	65	6D	75	6C	6F	76	20	2C	0C	006C7	P.ACA:	.ASCII	<12>\, volume !UL\		
		20	3A	79	62	20	6E	65	74	74	69	72	57	16	006D4	P.ACB:	.ASCII	<22>\Written by:	!AD\	
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	006E3					
		5D	42	4F	21	2C	42	4F	21	5B	20	20	20	20	006EB	P.ACC:	.ASCII	<28>\UIC:	[!OB,!OB]\	
20	20	20	20	20	20	20	20	20	20	3A	65	74	61	44	16	00708	P.ACD:	.ASCII	<22>\Date:	!XD\
20	20	20	20	20	20	3A	64	6E	61	6D	6D	6F	43	13	00717					
										20	20	20	20	20	0071F	P.ACE:	.ASCII	<19>\Command:	\	
										20	20	20	20	20	0072E					
										2A	39	31	21	05	00733	P.ACF:	.ASCII	<5>\!19* \		
74	73	79	73	20	67	6E	69	74	61	72	65	70	4F	1A	00739	P.ACG:	.ASCII	<3>\!AD\		
			53	4D	56	2F	58	41	56	20	20	3A	6D	65	0073D	P.ACH:	.ASCII	<26>\Operating system:	VAX/VMS\	
		44	41	21	20	6E	6F	69	73	72	65	76	20	0C	0074C					
6E	6F	69	73	72	65	76	20	50	55	4B	43	41	42	16	00758	P.ACI:	.ASCII	<12>\ version !AD\		
							44	41	21	20	20	20	20	3A	00765	P.ACJ:	.ASCII	<22>\BACKUP version:	!AD\	
65	74	73	69	67	65	72	20	44	49	20	55	50	43	16	00774					
							4C	58	21	20	20	20	3A	72	0077C	P.ACK:	.ASCII	<22>\CPU ID register:	!XL\	
20	20	20	20	3A	65	6D	61	6E	20	65	64	6F	4E	16	0078B					
							44	41	21	20	20	20	20	20	00793	P.ACL:	.ASCII	<22>\Node name:	!AD\	
20	20	20	3A	6E	6F	20	6E	65	74	74	69	72	57	16	007A2					
							44	41	21	20	20	20	20	20	007AA	P.ACM:	.ASCII	<22>\Written on:	!AD\	
20	20	20	3A	65	7A	69	73	20	6B	63	6F	6C	42	16	007B9					
							4C	55	21	20	20	20	20	20	007C1	P.ACN:	.ASCII	<22>\Block size:	!UL\	
20	20	20	3A	5	7A	69	73	20	70	75	6F	72	47	16	007D0					
							4C	55	21	20	20	20	20	20	007DB	P.ACO:	.ASCII	<22>\Group size:	!UL\	
20	3A	74	6E	75	6F	63	20	72	65	66	66	75	42	16	007E7					
							4C	55	21	20	20	20	20	20	007EF	P.ACP:	.ASCII	<22>\Buffer count:	!UL\	
20	66	6F	20	65	76	61	73	20	65	67	61	6D	49	18	007FE					
							4C	55	21	20	20	20	20	20	00806	P.ACQ:	.ASCII	<24>\Image save of volume set\		
6D	61	6E	20	74	65	73	20	65	6D	75	6C	6F	56	16	00815					
							44	41	21	20	20	20	3A	65	0081F	P.ACR:	.ASCII	<22>\Volume set name:	!AD\	
75	6C	6F	76	20	66	6F	20	72	65	62	6D	75	4E	16	0082E					
							4C	55	21	20	3A	73	65	6D	00836	P.ACS:	.ASCII	<22>\Number of volumes:	!UL\	
70	79	72	63	6E	45	20	74	65	73	65	76	61	53	11	00845					
												64	65	74	0084D	P.ACT:	.ASCII	<17>\Saveset Encrypted\		
75	62	69	72	74	74	61	20	65	6D	75	6C	6F	56	11	0085C					
												73	65	74	0085F	P.ACU:	.ASCII	<17>\Volume attributes\		
												73	65	74	0086E					
20	65	76	69	74	61	6C	65	72	20	72	6F	66	20	18	00871	P.ACV:	.ASCII	<24>\ for relative volume !UL\		
							4C	55	21	20	65	6D	75	6C	6F	76	00880			
65	76	65	6C	20	65	72	75	74	63	75	72	74	53	16	0088A	P.ACW:	.ASCII	<22>\Structure level:	!UL\	
							4C	55	21	20	20	20	3A	6C	00899					
20	20	20	20	20	20	20	20	3A	6C	65	62	61	4C	16	008A1	P.ACX:	.ASCII	<22>\Label:	!AD\	
							44	41	21	20	20	20	20	20	008B0					
20	20	20	20	20	20	20	20	3A	72	65	6E	77	4F	16	008B8	P.ACY:	.ASCII	<22>\Owner:	!AD\	
							44	41	21	20	20	20	20	20	008C7					
20	20	20	20	3A	43	49	55	20	72	65	6E	77	4F	1C	008CF	P.ACZ:	.ASCII	<28>\Owner UIC:	[!OB,!OB]\	

LIST
V04-000

List a save set
LIST_ONE_RECORD - List save set record

LIST	5D	42	4F	21	2C	42	4F	21	5B	20	20	20	20	20	008DE
3A	65	74	61	64	20	6E	6F	69	74	61	65	72	43	16	008EC P.ADA: .ASCII <22>\Creation date: !%D\ 008FB
3A	72	65	62	6D	75	6E	20	6C	61	69	72	65	53	16	00903 P.ADB: .ASCII <22>\Serial number: !OL\ 00912
20	3A	73	6B	63	6F	6C	62	20	6C	61	74	6F	54	16	0091A P.ADC: .ASCII <22>\Total blocks: !UL\ 00929
20	3A	74	6E	75	6E	63	20	73	73	65	63	63	41	16	00931 P.ADD: .ASCII <22>\Access count: !UL\ 00940
20	3A	65	7A	69	73	20	72	65	74	73	75	6C	43	16	00948 P.ADE: .ASCII <22>\Cluster size: !UL\ 00957
20	20	20	3A	6B	63	65	68	63	20	61	74	61	44	13	0095F P.ADF: .ASCII <19>\Data check: \ 0096E
															00973 P.ADG: .ASCII <3>\No \ 00977 P.ADH: .ASCII <6>\Read, \ 0097E P.ADI: .ASCII <3>\No \ 00982 P.ADJ: .ASCII <5>\Write\ 00988 P.ADK: .ASCII <22>\Extension size: !UL\ 00997
65	7A	69	73	20	6E	6F	69	73	6E	65	74	78	45	16	0099F P.ADL: .ASCII <19>\File protection: \ 009AE
6F	69	74	63	65	74	6F	72	70	20	65	6C	69	46	13	009B3 P.ADM: .ASCII <22>\Maximum files: !UL\ 009C2
3A	73	65	6C	69	66	20	6D	75	6D	69	78	61	4D	16	009CA P.ADN: .ASCII <19>\Volume protection: \ 009D9
74	63	65	74	6F	72	70	20	65	6D	75	6C	6F	56	13	009DE P.ADO: .ASCII <22>\Windows: !UL\ 009ED
20	20	20	20	20	20	3A	73	77	6F	64	6E	69	57	16	009F5 P.ADP: .ASCII <22>\Minimum retention: !%D\ 00A04
74	6E	65	74	65	72	20	6D	75	6D	69	78	61	4D	16	00A0C P.ADQ: .ASCII <22>\Maximum retention: !%D\ 00A1B
74	6E	65	74	65	72	20	6D	75	6D	69	78	61	4D	16	00A23 P.ADR: .ASCII <23>\Physical save of volume\ 00A32
20	65	76	61	73	20	6C	61	63	69	73	79	68	50	17	00A3B P.ADS: .ASCII <22>\Device: !AD\ 00A4A
20	20	20	20	20	20	20	3A	65	6C	6F	76	20	66	6F	00A52 P.ADU: .ASCII <0> 00A53 P.ADV: .ASCII <4>\RK06\ 00A58 P.ADW: .ASCII <4>\RK07\ 00A5D P.ADX: .ASCII <4>\RP04\ 00A62 P.ADY: .ASCII <4>\RP05\ 00A67 P.ADZ: .ASCII <4>\RP06\ 00A6C P.AEA: .ASCII <4>\RM03\ 00A71 P.AEB: .ASCII <4>\RP07\ 54 48 00A76 P.AEC: .ASCII <6>\RP07HT\ 00A7D P.AED: .ASCII <4>\RL01\ 00A82 P.AEE: .ASCII <4>\RL02\ 00A87 P.AEF: .ASCII <4>\RX02\ 00A8C P.AEG: .ASCII <4>\RX04\ 00A91 P.AEH: .ASCII <4>\RM80\ 00A96 P.AEI: .ASCII <4>\TU58\ 00A9B P.AEJ: .ASCII <4>\RM05\ 00AA0 P.AEK: .ASCII <4>\RX01\ 00AA5 P.AEL: .ASCII <4>\ML11\ 00AAA P.AEM: .ASCII <4>\RB02\ 00AAF P.AEN: .ASCII <4>\RB80\ 00AB4 P.AEO: .ASCII <4>\RA80\ 00AB9 P.AEP: .ASCII <4>\RA81\ 00

			06	1B	0004E		BLEQU	4\$		
			53	D6	00050		INCL	R3		
30	A4		00	FB	00052		CALLS	#0, LIST_EOL		
	50		65	3C	00056	4\$:	MOVZWL	OUTPUT_ATTBUF, R0		2193
			1C	13	00059		BEQL	6\$		
		04	A5	DD	0005B		PUSHL	OUTPUT_ATTBUF+4		2198
			50	DD	0005E		PUSHL	R0		
		FA63	CF	9F	00060		PUSHAB	P.ABZ		
	64		03	FB	00064		CALLS	#3, LIST_FAO		
	09		53	E9	00067		BLBC	R3, 5\$		2199
			52	DD	0006A		PUSHL	VOLNUM		
		FA6E	CF	9F	0006C		PUSHAB	P.ACA		
	64		02	FB	00070		CALLS	#2, LIST_FAO		
30	A4		00	FB	00073	5\$:	CALLS	#0, LIST_EOL		2200
	50		18	A5	00077	6\$:	MOVZWL	OUTPUT_ATTBUF+24, R0		2203
			10	13	0007B		BEQL	7\$		
			1C	A5	0007D		PUSHL	OUTPUT_ATTBUF+28		2208
			50	DD	00080		PUSHL	R0		
		FA65	CF	9F	00082		PUSHAB	P.ACB		
	64		03	FB	00086		CALLS	#3, LIST_FAO		
30	A4		00	FB	00089		CALLS	#0, LIST_EOL		2209
			5C	A5	0008D	7\$:	TSTL	OUTPUT_ATTBUF+92		2212
			13	13	00090		BEQL	8\$		
	7E		5C	A5	00092		MOVZWL	OUTPUT_ATTBUF+92, -(SP)		2217
	7E		5E	A5	00096		MOVZWL	OUTPUT_ATTBUF+94, -(SP)		
		FA64	CF	9F	0009A		PUSHAB	P.ACC		
	64		03	FB	0009E		CALLS	#3, LIST_FAO		
30	A4		00	FB	000A1		CALLS	#0, LIST_EOL		2218
			24	A5	000A5	8\$:	TSTL	OUTPUT_ATTBUF+36		2221
			0E	13	000A8		BEQL	9\$		
			24	A5	000AA		PUSHAB	OUTPUT_ATTBUF+36		2225
		FA6E	CF	9F	000AD		PUSHAB	P.ACD		
	64		02	FB	000B1		CALLS	#2, LIST_FAO		
30	A4		00	FB	000B4		CALLS	#0, LIST_EOL		2226
	50		08	A5	000B8	9\$:	MOVZWL	OUTPUT_ATTBUF+8, R0		2229
			39	13	000BC		BEQL	13\$		
			50	D0	000BE		MOVL	R0, LENGTH		2238
			53	A5	000C1		MOVL	OUTPUT_ATTBUF+12, ADDRESS		2239
			52	D5	000C5	10\$:	TSTL	LENGTH		2240
			2E	15	000C7		BLEQ	13\$		
52		08	A5				CMPZV	#0, #16, OUTPUT_ATTBUF+8, LENGTH		2242
			10	00	ED	000C9	BNEQ	11\$		
			06	12	000CF		PUSHAB	P.ACE		2243
		A61	CF	9F	000D1		BRB	12\$		
			04	11	000D5		PUSHAB	P.ACF		2244
		FA6F	CF	9F	000D7	11\$:	PUSHAB	P.ACF		
	64		01	FB	000DB	12\$:	CALLS	#1, LIST_FAO		
			0C	BB	000DE		PUSHR	#*M<R2,R3>		2247
		FA6C	CF	9F	000E0		PUSHAB	P.ACG		
	64		03	FB	000E4		CALLS	#3, LIST_FAO		
30	A4		00	FB	000E7		CALLS	#0, LIST_EOL		2248
	52	FF13	C2	9E	000EB		MOVAB	-237(R2), LENGTH		2249
	53	00ED	C3	9E	000F0		MOVAB	237(R3), ADDRESS		2250
			CE	11	000F5		BRB	10\$		2240
0400	8F		6C	A5	000F7	13\$:	CMPW	OUTPUT_ATTBUF+108, #1024		2254
			1D	12	000FD		BNEQ	15\$		
		FA51	CF	9F	000FF		PUSHAB	P.ACH		2257
	64		01	FB	00103		CALLS	#1, LIST_FAO		

	50	2C	A5	3C	00106	MOVZWL	OUTPUT_ATTBUF+44, R0	2258
			0C	13	0010A	BEQL	14\$	
		30	A5	DD	0010C	PUSHL	OUTPUT_ATTBUF+48	2262
			50	DD	0010F	PUSHL	R0	
			FA5A	CF	9F	00111	PUSHAB	P.ACJ
	64		03	FB	00115	CALLS	#3, LIST_FAO	
30	A4		00	FB	00118	CALLS	#0, LIST_EOL	2263
	50	44	A5	3C	0011C	MOVZWL	OUTPUT_ATTBUF+68, R0	2266
			10	13	00120	BEQL	16\$	
		48	A5	DD	00122	PUSHL	OUTPUT_ATTBUF+72	2271
			50	DD	00125	PUSHL	R0	
			FA51	CF	9F	00127	PUSHAB	P.ACJ
	64		03	FB	0012B	CALLS	#3, LIST_FAO	
30	A4		00	FB	0012E	CALLS	#0, LIST_EOL	2272
	50	60	A5	DD	00132	MOVL	OUTPUT_ATTBUF+96, R0	2275
			0D	13	00136	BEQL	17\$	
			50	DD	00138	PUSHL	R0	2279
			FA55	CF	9F	0013A	PUSHAB	P.ACK
	64		02	FB	0013E	CALLS	#2, LIST_FAO	
30	A4		00	FB	00141	CALLS	#0, LIST_EOL	2280
	50	34	A5	3C	00145	MOVZWL	OUTPUT_ATTBUF+52, R0	2283
			10	13	00149	BEQL	18\$	
		38	A5	DD	0014B	PUSHL	OUTPUT_ATTBUF+56	2288
			50	DD	0014E	PUSHL	R0	
			FA56	CF	9F	00150	PUSHAB	P.ACL
	64		03	FB	00154	CALLS	#3, LIST_FAO	
30	A4		00	FB	00157	CALLS	#0, LIST_EOL	2289
	50	3C	A5	3C	0015B	MOVZWL	OUTPUT_ATTBUF+60, R0	2292
			10	13	0015F	BEQL	19\$	
		40	A5	DD	00161	PUSHL	OUTPUT_ATTBUF+64	2297
			50	DD	00164	PUSHL	R0	
			FA57	CF	9F	00166	PUSHAB	P.ACM
	64		03	FB	0016A	CALLS	#3, LIST_FAO	
30	A4		00	FB	0016D	CALLS	#0, LIST_EOL	2298
	50	64	A5	DD	00171	MOVL	OUTPUT_ATTBUF+100, R0	2301
			0D	13	00175	BEQL	20\$	
			50	DD	00177	PUSHL	R0	2305
			FA5B	CF	9F	00179	PUSHAB	P.ACN
	64		02	FB	0017D	CALLS	#2, LIST_FAO	
30	A4		00	FB	00180	CALLS	#0, LIST_EOL	2306
	50	6E	A5	3C	00184	MOVZWL	OUTPUT_ATTBUF+110, R0	2309
			0D	13	00188	BEQL	21\$	
			50	DD	0018A	PUSHL	R0	2313
			FA5F	CF	9F	0018C	PUSHAB	P.ACO
	64		02	FB	00190	CALLS	#2, LIST_FAO	
30	A4		00	FB	00193	CALLS	#0, LIST_EOL	2314
	50	70	A5	3C	00197	MOVZWL	OUTPUT_ATTBUF+112, R0	2317
			0D	13	0019B	BEQL	22\$	
			50	DD	0019D	PUSHL	R0	2321
			FA63	CF	9F	0019F	PUSHAB	P.ACP
	64		02	FB	001A3	CALLS	#2, LIST_FAO	
30	A4		00	FB	001A6	CALLS	#0, LIST_EOL	2322
		72	A5	B5	001AA	TSTW	OUTPUT_ATTBUF+114	2325
			3A	13	001AD	BEQL	24\$	
30	A4		00	FB	001AF	CALLS	#0, LIST_EOL	2328
			FA66	CF	9F	001B3	PUSHAB	P.ACQ
	64		01	FB	001B7	CALLS	#1, LIST_FAO	2329

30	A4		00	FB	001BA		CALLS	#0, LIST_EOL	2330
	50	4C	A5	3C	001BE		MOVZWL	OUTPUT_ATTBUF+76, R0	2332
			16	13	001C2		BEQL	23\$	
	20	50	B5	91	001C4		CMPB	@OUTPUT_ATTBUF+80, #32	2333
			10	1B	001C8		BLEQU	23\$	
		50	A5	DD	001CA		PUSHL	OUTPUT_ATTBUF+80	2338
			50	DD	001CD		PUSHL	R0	
		FA63	CF	9F	001CF		PUSHAB	P.ACR	
	64		03	FB	001D3		CALLS	#3, LIST_FAO	
30	A4		00	FB	001D6		CALLS	#0, LIST_EOL	2339
	7E	72	A5	3C	001DA	23\$:	MOVZWL	OUTPUT_ATTBUF+114, -(SP)	2343
		FA6B	CF	9F	001DE		PUSHAB	P.ACS	
	64		02	FB	001E2		CALLS	#2, LIST_FAO	
30	A4		00	FB	001E5		CALLS	#0, LIST_EOL	2344
		74	A5	95	001E9	24\$:	TSTB	OUTPUT_ATTBUF+116	2347
			03	12	001EC		BNEQ	25\$	
			02EF	31	001EE		BRW	59\$	
		FA6F	CF	9F	001F1	25\$:	PUSHAB	P.ACT	2350
	64		01	FB	001F5		CALLS	#1, LIST_FAO	
			0183	31	001F8		BRW	38\$	2351
			52	DD	001FB	26\$:	PUSHL	R2	2360
00000000G	0J		01	FB	001FD		CALLS	#1, GET_VOLUME_ATTRIBUTES	
		FA6E	CF	9F	00204		PUSHAB	P.ACU	2363
	64		01	FB	00208		CALLS	#1, LIST_FAO	
	50	42	A5	3C	0020B		MOVZWL	OUTPUT_ATTBUF+66, R0	2364
			09	13	0020F		BEQL	27\$	
			50	DD	00211		PUSHL	R0	2366
		FA71	CF	9F	00213		PUSHAB	P.ACV	
	64		02	FB	00217		CALLS	#2, LIST_FAO	
30	A4		00	FB	0021A	27\$:	CALLS	#0, LIST_EOL	2367
		40	A5	B5	0021E		TSTW	OUTPUT_ATTBUF+64	2369
			0F	13	00221		BEQL	28\$	
	7E	41	A5	9A	00223		MOVZBL	OUTPUT_ATTBUF+65, -(SP)	2373
		FA76	CF	9F	00227		PUSHAB	P.ACW	
	64		02	FB	0022B		CALLS	#2, LIST_FAO	
30	A4		00	FB	0022E		CALLS	#0, LIST_EOL	2374
	50		65	3C	00232	28\$:	MOVZWL	OUTPUT_ATTBUF, R0	2377
			10	13	00235		BEQL	29\$	
		04	A5	DD	00237		PUSHL	OUTPUT_ATTBUF+4	2382
			50	DD	0023A		PUSHL	R0	
		FA78	CF	9F	0023C		PUSHAB	P.ACX	
	64		03	FB	00240		CALLS	#3, LIST_FAO	
30	A4		00	FB	00243		CALLS	#0, LIST_EOL	2383
	50	08	A5	3C	00247	29\$:	MOVZWL	OUTPUT_ATTBUF+8, R0	2386
			10	13	0024B		BEQL	30\$	
		0C	A5	DD	0024D		PUSHL	OUTPUT_ATTBUF+12	2391
			50	DD	00250		PUSHL	R0	
		FA79	CF	9F	00252		PUSHAB	P.ACY	
	64		03	FB	00256		CALLS	#3, LIST_FAO	
30	A4		00	FB	00259		CALLS	#0, LIST_EOL	2392
		28	A5	D5	0025D	30\$:	TSTL	OUTPUT_ATTBUF+40	2395
			13	13	00260		BEQL	31\$	
	7E	28	A5	3C	00262		MOVZWL	OUTPUT_ATTBUF+40, -(SP)	2400
	7E	2A	A5	3C	00266		MOVZWL	OUTPUT_ATTBUF+42, -(SP)	
		FA78	CF	9F	0026A		PUSHAB	P.ACZ	
	64		03	FB	0026E		CALLS	#3, LIST_FAO	
30	A4		00	FB	00271		CALLS	#0, LIST_EOL	2401

		18	A5	D5	00275	31\$:	TSTL	OUTPUT_ATTBUF+24	2404
			0E	13	00278		BEQL	32\$	2405
		18	A5	9F	0027A		PUSHAB	OUTPUT_ATTBUF+24	2408
			CF	9F	0027D		PUSHAB	P.ADA	2409
	64		02	FB	00281		CALLS	#2, LIST_FAO	2410
30	A4		00	FB	00284		CALLS	#0, LIST_EOL	2411
	50	3C	A5	D0	00288	32\$:	MOVL	OUTPUT_ATTBUF+60, R0	2412
			0D	13	0028C		BEQL	33\$	2413
			50	DD	0028E		PUSHL	R0	2416
		FA86	CF	9F	00290		PUSHAB	P.ADB	2417
	64		02	FB	00294		CALLS	#2, LIST_FAO	2418
30	A4		00	FB	00297		CALLS	#0, LIST_EOL	2419
	50	2C	A5	D0	0029B	33\$:	MOVL	OUTPUT_ATTBUF+44, R0	2420
			0D	13	0029F		BEQL	34\$	2421
			50	DD	002A1		PUSHL	R0	2424
		FA8A	CF	9F	002A3		PUSHAB	P.ADC	2425
	64		02	FB	002A7		CALLS	#2, LIST_FAO	2426
30	A4		00	FB	002AA		CALLS	#0, LIST_EOL	2427
	7E	53	A5	9A	002AE	34\$:	MOVZBL	OUTPUT_ATTBUF+83, -(SP)	2429
			CF	9F	002B2		PUSHAB	P.ADD	2430
	64		02	FB	002B6		CALLS	#2, LIST_FAO	2431
30	A4		00	FB	002B9		CALLS	#0, LIST_EOL	2432
	7E	4E	A5	3C	002BD		MOVZWL	OUTPUT_ATTBUF+78, -(SP)	2433
			CF	9F	002C1		PUSHAB	P.ADE	2434
	64		02	FB	002C5		CALLS	#2, LIST_FAO	2435
30	A4		00	FB	002C8		CALLS	#0, LIST_EOL	2436
		FAA6	CF	9F	002CC		PUSHAB	P.ADF	2437
	64		01	FB	002D0		CALLS	#1, LIST_FAO	2438
	07	4A	A5	E8	002D3		BLBS	OUTPUT_ATTBUF+74, 35\$	2439
			CF	9F	002D7		PUSHAB	P.ADG	2440
	64		01	FB	002DB		CALLS	#1, LIST_FAO	2441
		FAAC	CF	9F	002DE	35\$:	PUSHAB	P.ADH	2442
	64		01	FB	002E2		CALLS	#1, LIST_FAO	2443
07	4A	A5	01	E0	002E5		BBS	#1, OUTPUT_ATTBUF+74, 36\$	2444
			CF	9F	002EA		PUSHAB	P.ADI	2445
	64		01	FB	002EE		CALLS	#1, LIST_FAO	2446
		FAA4	CF	9F	002F1	36\$:	PUSHAB	P.ADJ	2447
	64		01	FB	002F5		CALLS	#1, LIST_FAO	2448
30	A4		00	FB	002F8		CALLS	#0, LIST_EOL	2449
	7E	4C	A5	3C	002FC		MOVZWL	OUTPUT_ATTBUF+76, -(SP)	2450
			CF	9F	00300		PUSHAB	P.ADY	2451
	64		02	FB	00304		CALLS	#2, LIST_FAO	2452
30	A4		00	FB	00307		CALLS	#0, LIST_EOL	2453
		FAA7	CF	9F	0030B		PUSHAB	P.ADL	2454
	64		01	FB	0030F		CALLS	#1, LIST_FAO	2455
		44455752	8F	DD	00312		PUSHL	#1145395026	2456
	7E	46	A5	3C	00318		MOVZWL	OUTPUT_ATTBUF+70, -(SP)	2457
00B0	C4		02	FB	0031C		CALLS	#2, LIST_PROTECTION	2458
30	A4		00	FB	00321		CALLS	#0, LIST_EOL	2459
		34	A5	DD	00325		PUSHL	OUTPUT_ATTBUF+52	2460
			CF	9F	00328		PUSHAB	P.ADM	2461
	64		02	FB	0032C		CALLS	#2, LIST_FAO	2462
30	A4		00	FB	0032F		CALLS	#0, LIST_EOL	2463
		FAAA	CF	9F	00333		PUSHAB	P.ADN	2464
	64		01	FB	00337		CALLS	#1, LIST_FAO	2465
		44435752	8F	DD	0033A		PUSHL	#1145263954	2466
	7E	44	A5	3C	00340		MOVZWL	OUTPUT_ATTBUF+68, -(SP)	2467

0080	C4	02	FB	00344	CALLS	#2, LIST_PROTECTION	:	
30	A4	00	FB	00349	CALLS	#0, LIST_EOL	:	2459
	7E	52	A5	9A 0034D	MOVZBL	OUTPUT_ATTBUF+82, -(SP)	:	2462
		FAA0	CF	9F 00351	PUSHAB	P.ADO	:	
	64	02	FB	00355	CALLS	#2, LIST_FAO	:	
30	A4	00	FB	00358	CALLS	#0, LIST_EOL	:	2463
		60	A5	D5 0035C	TSTL	OUTPUT_ATTBUF+96	:	2465
		0E	13	0035F	BEQL	37\$:	
		60	A5	9F 00361	PUSHAB	OUTPUT_ATTBUF+96	:	2469
		FAA4	CF	9F 00364	PUSHAB	P.ADP	:	
	64	02	FB	00368	CALLS	#2, LIST_FAO	:	
30	A4	00	FB	0036B	CALLS	#0, LIST_EOL	:	2470
		68	A5	D5 0036F	TSTL	OUTPUT_ATTBUF+104	:	2473
		0E	13	00372	BEQL	39\$:	
		68	A5	9F 00374	PUSHAB	OUTPUT_ATTBUF+104	:	2477
		FAA8	CF	9F 00377	PUSHAB	P.ADQ	:	
	64	02	FB	0037B	CALLS	#2, LIST_FAO	:	
30	A4	00	FB	0037E	CALLS	#0, LIST_EOL	:	2478
		015B	31	00382	BRW	59\$:	2481
		52	DD	00385	PUSHL	R2	:	2487
00000000G	00	01	FB	00387	CALLS	#1, GET_PHYSVOL_ATTRIBUTES	:	
		FAA8	CF	9F 0038E	PUSHAB	P.ADR	:	2490
	64	01	FB	00392	CALLS	#1, LIST_FAO	:	
30	A4	00	FB	00395	CALLS	#0, LIST_EOL	:	2491
	50	65	3C	00399	MOVZWL	OUTPUT_ATTBUF, R0	:	2493
		10	13	0039C	BEQL	41\$:	
		04	A5	DD 0039E	PUSHL	OUTPUT_ATTBUF+4	:	2498
		50	DD	003A1	PUSHL	R0	:	
		FAAB	CF	9F 003A3	PUSHAB	P.ADS	:	
	64	03	FB	003A7	CALLS	#3, LIST_FAO	:	
30	A4	00	FB	003AA	CALLS	#0, LIST_EOL	:	2499
	52	24	A5	D0 003AE	MOVL	OUTPUT_ATTBUF+36, R2	:	2502
	18	52	D1	003B2	CMPL	R2, #24	:	
		0E	1A	003B5	BGTRU	42\$:	
		FB2B	CF	42 DD 003B7	PUSHL	DEVTYPE\$[R2]	:	2539
		FB8B	CF	9F 003BC	PUSHAB	P.AET	:	
	64	02	FB	003C0	CALLS	#2, LIST_FAO	:	
000000FF	8F	2A	11	003C3	BRB	43\$:	2540
		52	D1	003C5	CMPL	R2, #255	:	2542
		25	1B	003CC	BLEQU	44\$:	
	6E	06	D0	003CE	MOVL	#6, LENGTH	:	2544
		7E	7C	003D1	CLRQ	-(SP)	:	2553
		0C	AE	9F 003D3	PUSHAB	BUFFER	:	
		0C	AE	9F 003D6	PUSHAB	LENGTH	:	
		52	DD	003D9	PUSHL	R2	:	
00000000G	00	05	FB	003DB	CALLS	#5, DECODE_DEV TYP	:	
		04	AE	9F 003E2	PUSHAB	BUFFER	:	2557
		04	AE	DD 003E5	PUSHL	LENGTH	:	
		FB76	CF	9F 003E8	PUSHAB	P.AEU	:	
	64	03	FB	003EC	CALLS	#3, LIST_FAO	:	
30	A4	00	FB	003EF	CALLS	#0, LIST_EOL	:	2558
	50	1C	A5	D0 003F3	MOVL	OUTPUT_ATTBUF+28, R0	:	2562
		0D	13	003F7	BEQL	45\$:	
		50	DD	003F9	PUSHL	R0	:	2566
		FB7A	CF	9F 003FB	PUSHAB	P.AEV	:	
	64	02	FB	003FF	CALLS	#2, LIST_FAO	:	
30	A4	00	FB	00402	CALLS	#0, LIST_EOL	:	2567

	50	08	A5	3C	00406	45\$:	MOVZWL	OUTPUT_ATTBUF+8, R0	2570
			10	13	0040A		BEQL	46\$	
		0C	A5	DD	0040C		PUSHL	OUTPUT_ATTBUF+12	2575
			50	DD	0040F		PUSHL	R0	
		FB7B	CF	9F	00411		PUSHAB	P.AEW	
	64		03	FB	00415		CALLS	#3, LIST_FAO	
30	A4		00	FB	00418		CALLS	#0, LIST_EOL	2576
	50	18	A5	D0	0041C	46\$:	MOVL	OUTPUT_ATTBUF+24, R0	2579
			0D	13	00420		BEQL	47\$	
			50	DD	00422		PUSHL	R0	2583
		FB7F	CF	9F	00424		PUSHAB	P.AEX	
	64		02	FB	00428		CALLS	#2, LIST_FAO	
30	A4		00	FB	0042B		CALLS	#0, LIST_EOL	2584
		10	A5	B5	0042F	47\$:	TSTW	OUTPUT_ATTBUF+16	2587
			03	12	00432		BNEQ	49\$	
			00A9	31	00434	48\$:	BRW	59\$	
	52	14	A5	D0	00437	49\$:	MOVL	OUTPUT_ATTBUF+20, P	2591
	50	10	A5	3C	0043B	50\$:	MOVZWL	OUTPUT_ATTBUF+16, R0	2594
	50	14	A5	C0	0043F		ADDL2	OUTPUT_ATTBUF+20, R0	
	50		52	D1	00443		CMPL	P, R0	2593
			EC	1E	00446		BGEQU	48\$	
	14	A5	52	D1	00448		CMPL	P, OUTPUT_ATTBUF+20	2597
			06	12	0044C		BNEQ	51\$	
		FB6C	CF	9F	0044E		PUSHAB	P.AEY	2598
			04	11	00452		BRB	52\$	
		FB7A	CF	9F	00454	51\$:	PUSHAB	P.AEZ	2599
	64		01	FB	00458	52\$:	CALLS	#1, LIST_FAO	
			62	DD	0045B		PUSHL	(P)	2600
		FB77	CF	9F	0045D		PUSHAB	P.AFA	
	64		02	FB	00461		CALLS	#2, LIST_FAO	
	01	04	A2	D1	00464		CMPL	4(P), #1	2601
			0F	13	00468		BEQL	53\$	
50	62	04	A2	C1	0046A		ADDL3	4(P), (P), R0	
		FF	A0	9F	0046F		PUSHAB	-1(R0)	
		FB66	CF	9F	00472		PUSHAB	P.AFB	
	64		02	FB	00476		CALLS	#2, LIST_FAO	
	30	A4	00	FB	00479	53\$:	CALLS	#0, LIST_EOL	2602
	52		08	C0	0047D		ADDL2	#8, P	2603
			B9	11	00480		BRB	50\$	2592
			52	DD	00482	54\$:	PUSHL	R2	2617
		00000000G	00	01	FB	00484	CALLS	#1, GET FILE ATTRIBUTES	
52	18	A5	10	9C	0048B		ROTL	#16, OUTPUT_RECATTR+8, BLOCKS	2620
		1C	A5	B5	00490		TSTW	OUTPUT_RECATTR+12	2621
			06	12	00493		BNEQ	55\$	
			52	D5	00495		TSTL	BLOCKS	
			02	13	00497		BEQL	55\$	
			52	D7	00499		DECL	BLOCKS	2622
		0090	C5	D6	0049B	55\$:	INCL	LIST_TOTFILES	2625
	0094	C5	52	C0	0049F		ADDL2	BLOCKS, LIST_TOTSIZE	2626
		FE81	C5	95	004A4		TSTB	QUAL+9	2629
			06	18	004A8		BGEQ	56\$	
	03FD	C4	00	FB	004AA		CALLS	#0, FULL_LISTING	2631
			04	004AF			RET		
	34		65	B1	004B0	56\$:	CMPW	OUTPUT_FILENAME, #52	2634
			16	1B	004B3		BLEQU	57\$	
			55	DD	004B5		PUSHL	R5	2637
		FB29	CF	9F	004B7		PUSHAB	P.AFC	

LIST
V04-000

List a save set
LIST_ONE_RECORD - list save set record

I 16
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 45
(7)

30	64		02	FB	0048B	CALLS	#2, LIST_FAO	:	
	A4		00	FB	0048E	CALLS	#0, LIST_EOL	:	2638
		FB22	CF	9F	004C2	PUSHAB	P.AFD	:	2639
	64		01	FB	004C6	CALLS	#1, LIST_FAO	:	
			09	11	004C9	BRB	58\$:	2634
			55	DD	004CB	PUSHL	R5	:	2642
		FB1D	CF	9F	004CD	PUSHAB	P.AFE	:	
	64		02	FB	004D1	CALLS	#2, LIST_FAO	:	
		30	A5	9F	004D4	PUSHAB	OUTPUT_ATTBUF+48	:	2644
			52	DD	004D7	PUSHL	BLOCKS	:	
		FB17	CF	9F	004D9	PUSHAB	P.AFF	:	
	64		03	FB	004DD	CALLS	#3, LIST_FAO	:	
30	A4		00	FB	004E0	CALLS	#0, LIST_EOL	:	2645
			04	004E4	60\$:	RET		:	2651

; Routine Size: 1253 bytes, Routine Base: CODE + 0BE9

```

: 1105 2652 1 %SBTTL 'INIT LIST - initialize listing'
: 1106 2653 1 GLOBAL ROUTINE INIT_LIST: NOVALUE=
: 1107 2654 1
: 1108 2655 1 |++
: 1109 2656 1 |
: 1110 2657 1 | FUNCTIONAL DESCRIPTION:
: 1111 2658 1 |     This routine initializes a listing.
: 1112 2659 1 |
: 1113 2660 1 | INPUT PARAMETERS:
: 1114 2661 1 |     NONE
: 1115 2662 1 |
: 1116 2663 1 | IMPLICIT INPUTS:
: 1117 2664 1 |     NONE
: 1118 2665 1 |
: 1119 2666 1 | OUTPUT PARAMETERS:
: 1120 2667 1 |     NONE
: 1121 2668 1 |
: 1122 2669 1 | IMPLICIT OUTPUTS:
: 1123 2670 1 |     NONE
: 1124 2671 1 |
: 1125 2672 1 | ROUTINE VALUE:
: 1126 2673 1 |     NONE
: 1127 2674 1 |
: 1128 2675 1 | SIDE EFFECTS:
: 1129 2676 1 |     NONE
: 1130 2677 1 |
: 1131 2678 1 | --
: 1132 2679 1 |
: 1133 2680 2 BEGIN
: 1134 2681 2 LOCAL
: 1135 2682 2     LIST_FAB:      REF BBLOCK,      ! FAB for listing file
: 1136 2683 2     PSAREA:      VECTOR[P$SIZE]; ! Impure area
: 1137 2684 2 GLOBAL REGISTER
: 1138 2685 2     P$ = 11:      REF VECTOR;      ! Impure area base register
: 1139 2686 2
: 1140 2687 2
: 1141 2688 2 ! Initialize impure area.
: 1142 2689 2 |
: 1143 2690 2 P$ = PSAREA;
: 1144 2691 2 LIST_DESC[0] = LIST_SIZE;
: 1145 2692 2 LIST_DESC[1] = _LIST_BUFFER;
: 1146 2693 2
: 1147 2694 2
: 1148 2695 2 ! Open the listing file.
: 1149 2696 2 |
: 1150 2697 2 LIST_FAB = .QUAL[QUAL_LIST_FC];
: 1151 2698 2 LIST_FAB[FAB$B_DNS] = %CHARCOUNT('BACKUP.LIS');
: 1152 2699 2 LIST_FAB[FAB$L_DNA] = UPLIT BYTE('BACKUP.LIS');
: 1153 2700 2 LIST_FAB[FAB$V_PUT] = TRUE;
: 1154 2701 2 IF .[LIST_FAB[FAB$B_FNS] EQL 0
: 1155 2702 2 THEN
: 1156 2703 2     BEGIN
: 1157 2704 2     LIST_FAB[FAB$B_FNS] = %CHARCOUNT('SYSS$OUTPUT:');
: 1158 2705 2     LIST_FAB[FAB$L_FNA] = UPLIT BYTE('SYSS$OUTPUT:');
: 1159 2706 2     END;
: 1160 2707 2 LIST_FAB[FAB$V_SQO] = TRUE;
: 1161 2708 2 LIST_FAB[FAB$V_CR] = TRUE;
```



```

: 1162 2709 2 LIST FAB[FAB$B_RFM] = FAB$C VAR;
: 1163 2710 2 BBLOCK[LIST_FAB[FC_RAB], RAB$V_WBH] = TRUE;
: 1164 2711 2
: 1165 2712 2
: 1166 2713 2 IF NOT $CREATE(FAB=.LIST_FAB)
: 1167 2714 2 THEN
: 1168 2715 2 FILE_ERROR(
: 1169 2716 2   BACKUP$ OPENOUT + STS$K_SEVERE,
: 1170 2717 2   .LIST_FAB,
: 1171 2718 2   .LIST_FAB[FAB$L_STS],
: 1172 2719 2   .LIST_FAB[FAB$L_STV]);
: 1173 2720 2
: 1174 2721 2
: 1175 2722 2 IF NOT $CONNECT(RAB=LIST_FAB[FC_RAB])
: 1176 2723 2 THEN
: 1177 2724 2 FILE_ERROR(
: 1178 2725 2   BACKUP$ OPENOUT + STS$K_SEVERE,
: 1179 2726 2   .LIST_FAB,
: 1180 2727 2   .BBLOCK[LIST_FAB[FC_RAB], RAB$L_STS],
: 1181 2728 2   .BBLOCK[LIST_FAB[FC_RAB], RAB$L_STV]);
: 1182 2729 2
: 1183 2730 2
: 1184 2731 2 ! Put a header on the listing.
: 1185 2732 2 !
: 1186 2733 2 FAO_('Listing of ');
: 1187 2734 2 IF .QUAL[QUAL_INPU_LIST] NEQ 0
: 1188 2735 2 THEN
: 1189 2736 2 BEGIN
: 1190 2737 2   FAO_('save set');
: 1191 2738 2   EOL_();
: 1192 2739 2 END
: 1193 2740 2 ELSE
: 1194 2741 2   FAO_('BACKUP journal');
: 1195 2742 2   EOL_();
: 1196 2743 2 END;
```

			53	49	4C	2E	50	55	4B	43	41	42	010CE	P.AFG:	.ASCII	\BACKUP.LIS\	:	
			3A	54	55	50	54	55	4F	24	53	59	53	010D8	P.AFH:	.ASCII	\SYS\$OUTPUT:\	:
	20	66	6F	20	67	6E	69	74	73	69	4C	0B	010E3	P.AFI:	.ASCII	<11>\Listing of \	:	
				74	65	73	20	65	76	61	73	0B	010EF	P.AFJ:	.ASCII	<8>\save set\	:	
6C	61	6E	72	75	6F	6A	20	50	55	4B	43	41	42	0E	010F8	P.AFK:	.ASCII	<14>\BACKUP journal\
																.EXTRN	SYS\$CREATE, SYS\$CONNECT	:
																.ENTRY	INIT_LIST, Save R2,R3,R4,R5,R11	: 2653
								55	00000000G	00	9E	00002				MOVAB	FILE_ERROR, R5	:
								54	00000000G	8F	D0	00009				MOVL	#BACKUP\$ OPENOUT+4, R4	:
								53	EEEE	CF	9E	00010				MOVAB	LIST FAO, R3	:
								5E	FEF8	CE	9E	00015				MOVAB	-264(SP), SP	:
								5B		6E	9E	0001A				MOVAB	P\$AREA, P\$: 2690
								6B	0100	8F	3C	0001D				MOVZWL	#256, (P\$)	: 2691
						04	AB	08	AB	9E	00022					MOVAB	8(R11), 4(P\$)	: 2692
							52	00000000'	EF	D0	00027					MOVL	QUAL+56, LIST_FAB	: 2697
						35	A2		0A	90	0002E					MOVAB	#10, 53(LIST_FAB)	: 2698
						30	A2	92	AF	9E	00032					MOVAB	P.AFG, 48(LIST_FAB)	: 2699

LIST
V04-000

List a save set
INIT_LIST - initialize listing

L 16
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 48
(8)

16	A2		01	88	00037	BISB2	#1, 22(LIST_FAB)	:	2700
		34	A2	95	0003B	TSTB	52(LIST_FAB)	:	2701
			09	12	0003E	BNEQ	1\$:	
34	A2		0B	90	00040	MOVB	#11, 52(LIST_FAB)	:	2704
2C	A2	8A	AF	9E	00044	MOVAB	P.AFH, 44(LIST_FAB)	:	2705
04	A2	40	8F	88	00049	BISB2	#64, 4(LIST_FAB)	:	2707
1E	A2		02	88	0004E	BISB2	#2, 30(LIST_FAB)	:	2708
1F	A2		02	90	00052	MOVB	#2, 31(LIST_FAB)	:	2709
55	A2		04	88	00056	BISB2	#4, 85(LIST_FAB)	:	2710
			52	DD	0005A	PUSHL	LIST_FAB	:	2713
00000000G	00		01	FB	0005C	CALLS	#1, SYSS\$CREATE	:	
	0B		50	E8	00063	BLBS	R0, 2\$:	
	7E	08	A2	7D	00066	MOVQ	8(LIST_FAB), -(SP)	:	2718
			52	DD	0006A	PUSHL	LIST_FAB	:	2717
			54	DD	0006C	PUSHL	R4	:	2716
	65		04	FB	0006E	CALLS	#4, FILE_ERROR	:	
00000000G	00	50	A2	9F	00071	PUSHAB	80(R2)	:	2722
	0B		01	FB	00074	CALLS	#1, SYSS\$CONNECT	:	
	7E	58	50	E8	0007B	BLBS	R0, 3\$:	
			A2	7D	0007E	MOVQ	88(LIST_FAB), -(SP)	:	2727
			52	DD	00082	PUSHL	LIST_FAB	:	2726
			54	DD	00084	PUSHL	R4	:	2725
	65		04	FB	00086	CALLS	#4, FILE_ERROR	:	
		FF4F	CF	9F	00089	PUSHAB	P.AFI	:	2733
	63		01	FB	0008D	CALLS	#1, LIST_FAO	:	
		00000000'	EF	D5	00090	TSTL	QUAL	:	2734
			0D	13	00096	BEQL	4\$:	
		FF4C	CF	9F	00098	PUSHAB	P.AFJ	:	2737
	63		01	FB	0009C	CALLS	#1, LIST_FAO	:	
30	A3		00	FB	0009F	CALLS	#0, LIST_EOL	:	2738
			07	11	000A3	BRB	5\$:	2734
		FF48	CF	9F	000A5	PUSHAB	P.AFK	:	2741
	63		01	FB	000A9	CALLS	#1, LIST_FAO	:	
30	A3		00	FB	000AC	CALLS	#0, LIST_EOL	:	2742
			04	000B0	RET			:	2743

; Routine Size: 177 bytes. Routine Base: CODE + 1107

```
.. 1198 2744 1 %SBTTL 'FIN LIST - finish listing'
.. 1199 2745 1 GLOBAL ROUTINE FIN_LIST(CLOSE_LIST): NOVALUE=
.. 1200 2746 1
.. 1201 2747 1 :++
.. 1202 2748 1
.. 1203 2749 1 : FUNCTIONAL DESCRIPTION:
.. 1204 2750 1 : This routine terminates a listing.
.. 1205 2751 1
.. 1206 2752 1 : INPUT PARAMETERS:
.. 1207 2753 1 : CLOSE_LIST - TRUE TO CLOSE LISTING FILE;
.. 1208 2754 1 : FALSE TO LEAVE LISTING FILE OPEN
.. 1209 2755 1
.. 1210 2756 1 : IMPLICIT INPUTS:
.. 1211 2757 1 : NONE
.. 1212 2758 1
.. 1213 2759 1 : OUTPUT PARAMETERS:
.. 1214 2760 1 : NONE
.. 1215 2761 1
.. 1216 2762 1 : IMPLICIT OUTPUTS:
.. 1217 2763 1 : NONE
.. 1218 2764 1
.. 1219 2765 1 : ROUTINE VALUE:
.. 1220 2766 1 : NONE
.. 1221 2767 1
.. 1222 2768 1 : SIDE EFFECTS:
.. 1223 2769 1 : NONE
.. 1224 2770 1
.. 1225 2771 1 :--
.. 1226 2772 1
.. 1227 2773 2 BEGIN
.. 1228 2774 2 LOCAL
.. 1229 2775 2 LIST_FAB: REF BBLOCK, ! FAB for listing file
.. 1230 2776 2 PSAREA: VECTOR[PSSIZE]; ! Impure area
.. 1231 2777 2 GLOBAL REGISTER
.. 1232 2778 2 PS = 11: REF VECTOR; ! Impure area base register
.. 1233 2779 2
.. 1234 2780 2
.. 1235 2781 2 ! Initialize impure area.
.. 1236 2782 2 !
.. 1237 2783 2 PS = PSAREA;
.. 1238 2784 2 LIST_DESC[0] = LIST_SIZE;
.. 1239 2785 2 LIST_DESC[1] = _LIST_BUFFER;
.. 1240 2786 2
.. 1241 2787 2
.. 1242 2788 2 ! Close the listing file (if desired).
.. 1243 2789 2 !
.. 1244 2790 2 IF .CLOSE_LIST
.. 1245 2791 2 THEN
.. 1246 2792 3 BEGIN
.. 1247 2793 3 LIST_FAB = .QUAL[QUAL LIST FC];
.. 1248 2794 4 IF NOT $CLOSE(FAB=.LIST_FAB)
.. 1249 2795 3 THEN
.. 1250 2796 3 FILE ERROR(
.. 1251 2797 3 BACKUP$ CLOSEOUT + STS$K_SEVERE,
.. 1252 2798 3 .LIST_FAB,
.. 1253 2799 3 .LIST_FAB[FAB$L_STS],
.. 1254 2800 3 .LIST_FAB[FAB$L_STV]);
```

LIST
V04-000

List a save set
FIN_LIST - finish listing

B 1
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 50
(9)

```

: 1255      2801   3   RETURN;
: 1256      2802   2   END;
: 1257      2803   2
: 1258      2804   2
: 1259      2805   2   ! finish up the current save set listing.
: 1260      2806   2
: 1261      2807   2   EOL_();
: 1262      2808   2   !F .LIST_TOTFILES NEQ 0
: 1263      2809   2   THEN
: 1264      2810   3       BEGIN
: 1265      2811   3       FAO_ ('Total of !UL file!%S, !UL block!%S', .LIST_TOTFILES, .LIST_TOTSIZE);
: 1266      2812   3       EOL_();
: 1267      2813   3       END;
: 1268      2814   2   FAO_ ('End of ');
: 1269      2815   2   IF .QUAL[QUAL INPU_LIST] NEQ 0
: 1270      2816   2       THEN FAO_ ('save set!/')
: 1271      2817   2       ELSE FAO_ ('BACKUP journal!/')
: 1272      2818   2   EOL_();
: 1273      2819   1   END;
```

66	20	4C	55	21	20	66	6F	20	6C	61	74	6F	54	22	011B8	P.AFL:	.ASCII	\'Total of !UL file!%S, !UL block!%S\	:			
6F	6C	62	20	4C	55	21	20	2C	53	25	21	65	6C	69	011C7				:			
										53	25	21	6B	63	011D6				:			
										20	66	6F	20	64	6E	45	07	011DB	P.AFM:	.ASCII	<7>\End of \	:
				2F	21	74	65	73	20	65	76	61	73	0A	011E3	P.AFN:	.ASCII	<10>\save set!/\	:			
6C	61	6E	72	75	6F	6A	20	50	55	4B	43	41	42	10	011EE	P.AFO:	.ASCII	<16>\BACKUP journal!/\	:			
													2F	21	011FD				:			

.EXTRN SYSSCLOSE

									081C	00000		.ENTRY	FIN_LIST, Save R2,R3,R4,R11	:	2745
							54	00000000'	EF	9E	00002	MOVAB	QUAC+56, R4	:	
							53	EE24	CF	9E	00009	MOVAB	LIST_EOL, R3	:	
							5E	FEF8	CE	9E	0000E	MOVAB	-264(SP), SP	:	
							5B		6E	9E	00013	MOVAB	PSAREA, PS	:	2783
							6B	0100	8F	3C	00016	MOVZWL	#256, (PS)	:	2784
						04	AB	08	AB	9E	0001B	MOVAB	8(R11), 4(PS)	:	2785
							23	04	AC	E9	00020	BLBC	CLOSE_LIST, 1\$:	2790
							52		64	D0	00024	MOVL	QUAL+56, LIST_FAB	:	2793
									52	DD	00027	PUSHL	LIST_FAB	:	2794
						00000000G	00		01	FB	00029	CALLS	#1, SYSSCLOSE	:	
							4D		50	E8	00030	BLBS	R0, 5\$:	
							7E	08	A2	7D	00033	MOVQ	8(LIST_FAB), -(SP)	:	2799
									52	DD	00037	PUSHL	LIST_FAB	:	2798
						00000000G	00	00000000G	8F	DD	00039	PUSHL	#BACKUP\$_CLOSEOUT+4	:	2797
									04	FB	0003F	CALLS	#4, FILE_ERROR	:	
									04	00046	RET		:	2792	
							63		00	FB	00047	1\$: CALLS	#0, LIST_EOL	:	2807
							50	01E0	C4	D0	0004A	MOVL	LIST_TOTFILES, R0	:	2808
									11	13	0004F	BEQL	2\$:	
								01E4	C4	DD	00051	PUSHL	LIST_TOTSIZE	:	2811
									50	DD	00055	PUSHL	R0	:	
								FF5E	CF	9F	00057	PUSHAB	P.AFL	:	
						D0	A3		03	FB	0005B	CALLS	#3, LIST_FAO	:	
							63		00	FB	0005F	CALLS	#0, LIST_EOL	:	2812

LIST
V04-000

List @ save set
FIN_LIST - finish listing

C 1
16-Sep-1984 00:06:23 VAX-11 Bliss-32 V4.0-742 Page 51
14-Sep-1984 11:53:54 DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 (9)

		FF76	CF	9F	00062	2\$:	PUSHAB	P.AFM	:	2814
D0	A3		01	FB	00066		CALLS	#1, LIST_FAO	:	
		C8	A4	D5	0006A		TSTL	QUAL	:	2815
			06	13	0006D		BEQL	3\$:	
		FF71	CF	9F	0006F		PUSHAB	P.AFN	:	2816
			04	11	00073		BRB	4\$:	
		FF76	CF	9F	00075	3\$:	PUSHAB	P.AFO	:	2817
D0	A3		01	FB	00079	4\$:	CALLS	#1, LIST_FAO	:	
	63		00	FB	0007D		CALLS	#0, LIST_EOL	:	2818
			04	00080	5\$:		RET		:	2819

; Routine Size: 129 bytes, Routine Base: CODE + 11FF

```

1275 2820 1 %SBTTL 'CONCURRENT_LIST - generate concurrent listing'
1276 2821 1 GLOBAL ROUTINE CONCURRENT_LIST(BCB): NOVALUE=
1277 2822 1
1278 2823 1 !++
1279 2824 1
1280 2825 1 FUNCTIONAL DESCRIPTION:
1281 2826 1 This routine generates a concurrent listing; -- that is, a listing
1282 2827 1 during a save, restore, compare, or copy operation.
1283 2828 1
1284 2829 1 INPUT PARAMETERS:
1285 2830 1 BCB - Pointer to BCB for buffer to list.
1286 2831 1
1287 2832 1 IMPLICIT INPUTS:
1288 2833 1 NONE
1289 2834 1
1290 2835 1 OUTPUT PARAMETERS:
1291 2836 1 NONE
1292 2837 1
1293 2838 1 IMPLICIT OUTPUTS:
1294 2839 1 NONE
1295 2840 1
1296 2841 1 ROUTINE VALUE:
1297 2842 1 NONE
1298 2843 1
1299 2844 1 SIDE EFFECTS:
1300 2845 1 Listing is produced.
1301 2846 1
1302 2847 1 !--
1303 2848 1
1304 2849 2 BEGIN
1305 2850 2 LOCAL
1306 2851 2 SAVE_ATTBUF: BBLOCK[ATTBUF_LENGTH], ! Save for OUTPUT_ATTBUF
1307 2852 2 PSAREA: VECTOR[P$SIZE]; ! Impure area
1308 2853 2 GLOBAL REGISTER
1309 2854 2 P$ = 11: REF VECTOR; ! Impure area base register
1310 2855 2
1311 2856 2
1312 2857 2 ! Initialize impure area.
1313 2858 2
1314 2859 2 P$ = PSAREA;
1315 2860 2 LIST_DESC[0] = LIST_SIZE;
1316 2861 2 LIST_DESC[1] = _LIST_BUFFER;
1317 2862 2
1318 2863 2
1319 2864 2 ! Save attribute area so that attribute processing during the listing
1320 2865 2 will be transparent to the caller.
1321 2866 2
1322 2867 2 CH$MOVE(ATTBUF_LENGTH, OUTPUT_ATTBUF, SAVE_ATTBUF);
1323 2868 2
1324 2869 2
1325 2870 2 ! Do the listing. The third parameter to DEBLOCK prevents the buffer
1326 2871 2 from being released.
1327 2872 2
1328 2873 2 DEBLOCK(.BCB, LIST_ONE_RECORD, TRUE);
1329 2874 2
1330 2875 2
1331 2876 2 ! Restore saved area.

```

LIST
V04-000

List a save set
CONCURRENT_LIST - generate concurrent listing

E 1
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 53
(10)

: 1332
: 1333
: 1334

2877 2 !
2878 2 CH\$MOVE(ATTBUF_LENGTH, SAVE_ATTBUF, OUTPUT_ATTBUF);
2879 1 END;

				087C 00000	.ENTRY	CONCURRENT_LIST, Save R2,R3,R4,R5,R6,R11	: 2821
		56	00000000'	EF 9E 00002	MOVAB	OUTPUT_ATTBUF, R6	: :
		5E	FE68	CE 9E 00009	MOVAB	-408(SP), SP	: :
		5B		6E 9E 0000E	MOVAB	PSAREA, PS	: 2859
		6B	0100	8F 3C 00011	MOVZWL	#256, (PS)	: 2860
		AB	08	AB 9E 00016	MOVAB	8(R11), 4(PS)	: 2861
FF70	CD	66	0090	8F 28 0001B	MOV3	#144, OUTPUT_ATTBUF, SAVE_ATTBUF	: 2867
				01 DD 00023	PUSHL	#1	: 2873
			F940	CF 9F 00025	PUSHAB	LIST_ONE_RECORD	: :
			04	AC DD 00029	PUSHL	BCB	: :
		00000000G	00	03 FB 0002C	CALLS	#3, DEBLOCK	: :
		66	FF70 CD	8F 28 00033	MOV3	#144, SAVE_ATTBUF, OUTPUT_ATTBUF	: 2878
				04 0003B	RET		: 2879

: Routine Size: 60 bytes, Routine Base: CODE + 1280

```

: 1336 2880 1 %SBTTL 'LIST - main list routine'
: 1337 2881 1 GLOBAL ROUTINE LIST: NOVALUE=
: 1338 2882 1
: 1339 2883 1 !++
: 1340 2884 1
: 1341 2885 1 FUNCTIONAL DESCRIPTION:
: 1342 2886 1 This routine is the driver for listing-only generation.
: 1343 2887 1
: 1344 2888 1 INPUT PARAMETERS:
: 1345 2889 1 NONE
: 1346 2890 1
: 1347 2891 1 IMPLICIT INPUTS:
: 1348 2892 1 NONE
: 1349 2893 1
: 1350 2894 1 OUTPUT PARAMETERS:
: 1351 2895 1 NONE
: 1352 2896 1
: 1353 2897 1 IMPLICIT OUTPUTS:
: 1354 2898 1 NONE
: 1355 2899 1
: 1356 2900 1 ROUTINE VALUE:
: 1357 2901 1 NONE
: 1358 2902 1
: 1359 2903 1 SIDE EFFECTS:
: 1360 2904 1 NONE
: 1361 2905 1
: 1362 2906 1 !--
: 1363 2907 1
: 1364 2908 2 BEGIN
: 1365 2909 2 LOCAL
: 1366 2910 2 BCB, ! Pointer to buffer control block
: 1367 2911 2 CHK_SAVESET: BYTE, ! Check if save set encr on frst pass
: 1368 2912 2 PSAREA: VECTOR[P$SIZE]; ! Impure area
: 1369 2913 2
: 1370 2914 2 GLOBAL REGISTER
: 1371 2915 2 P$ = 11: REF VECTOR; ! Impure area base register
: 1372 2916 2
: 1373 2917 2 BUILTIN
: 1374 2918 2 FP;
: 1375 2919 2
: 1376 2920 2
: 1377 2921 2 ! Establish the handler.
: 1378 2922 2
: 1379 2923 2 .FP = RESTORE_HANDLER;
: 1380 2924 2
: 1381 2925 2
: 1382 2926 2 ! Initialize impure area.
: 1383 2927 2
: 1384 2928 2 P$ = PSAREA;
: 1385 2929 2 LIST_DESC[0] = LIST_SIZE;
: 1386 2930 2 LIST_DESC[1] = LIST_BUFFER;
: 1387 2931 2 CHK_SAVESET = 1;
: 1388 2932 2
: 1389 2933 2
: 1390 2934 2 ! Do the listing.
: 1391 2935 2
: 1392 2936 2 INIT_IN_SAVE(FALSE);

```



```

: 1393 2937 2 IF .INPUT_FLAGS[INPUT_WILDSAVE] AND .COM_FLAGS[COM_EOV] THEN RETURN;
: 1394 2938 2
: 1395 2939 2 WHILE (BCB = READ_BUFFER()) NEQ 0 DO
: 1396 2940 3 BEGIN
: 1397 2941 3 IF .QUAL[QUAL_SS_ENCRYP]
: 1398 2942 3 THEN
: 1399 2943 4 BEGIN
: 1400 2944 4     | If we are decrypting an encrypted saveset and we don't yet have
: 1401 2945 4     | the required decrypt context, do a special scan of the first block
: 1402 2946 4     | to locate the backup summary record, extract the datakey information
: 1403 2947 4     | and initialize the decryption context.
: 1404 2948 4     |
: 1405 2949 4     IF .CRYP DATA CODE EQLU 0
: 1406 2950 5     THEN IF NOT (QUAL[QUAL_SS_ENCRYP] = CRYPTO_INIDEC(.BCB))
: 1407 2951 4         THEN SIGNAL(BACKUP$_BACNOTENC);
: 1408 2952 4
: 1409 2953 4     | Decrypt the buffer
: 1410 2954 4     |
: 1411 2955 4     CRYPTO_DECR_BLOCK(.BCB);
: 1412 2956 4     END
: 1413 2957 4
: 1414 2958 3 ELSE
: 1415 2959 4 BEGIN
: 1416 2960 4     |
: 1417 2961 4     | Make sure that the save set is not encrypted.
: 1418 2962 4     |
: 1419 2963 4     IF .CHK_SAVESET
: 1420 2964 4     THEN IF CRYPTO_CHKSAV(.BCB)
: 1421 2965 4         THEN SIGNAL(BACKUP$_ENCSAVSET) ;
: 1422 2966 4     CHK_SAVESET = 0 ;
: 1423 2967 4     END ;
: 1424 2968 3
: 1425 2969 3
: 1426 2970 3 DEBLOCK(.BCB, LIST_ONE_RECORD);
: 1427 2971 3
: 1428 2972 2 END;
: 1429 2973 2 FIN_IN_SAVE(FALSE);
: 1430 2974 1 END;

```

```

: 2881 .ENTRY LIST, Save R2,R3,R4,R5,R11
: 2923 MOVAB LIB$SIGNAL, R5
: 2928 MOVAB QUAL+12, R4
: 2929 MOVAB -264(SP), SP
: 2930 MOVAB RESTORE_HANDLER, (FP)
: 2931 MOVAB PSAREA, PS
: 2936 MOVZWL #256, (PS)
: 2937 MOVAB 8(R11), 4(PS)
: 2938 MOVB #1, CHK_SAVESET
: 2939 CLRL -(SP)
: 2939 CALLS #1, INIT_IN_SAVE
: 2937 BBC #6, INPUT_FLAGS, 1$
: 2939 BLBS COM_FLAGS, 7$
: 2939 CALLS #0, READ_BUFFER

```

LIST
V04-000

List a save set
LIST - main list routine

H 1
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 Page 56
(11)

		52		50	DO	00046	MOVL	R0, BCB		
				5A	13	00049	BEQL	6\$		
	2C	02	A4	04	E1	0004B	BBC	#4, QUAL+14, 3\$		2941
			06DC	C4	D5	00050	TSTL	CRYP_DATA_CODE		2949
				1B	12	00054	BNEQ	2\$		
				52	DD	00056	PUSHL	BCB		2950
	02	A4	01	00000000G	00	01	FB	00058	CALLS	#1, CRYPTO_INIDEC
					04	50	FO	0005F	INSV	R0, #4, #1, QUAL+14
					09	50	E8	00065	BLBS	R0, 2\$
				00000000G	00	8F	DD	00068	PUSHL	#BACKUP\$ BACNOTENC
					65	01	FB	0006E	CALLS	#1, LIB\$SIGNAL
						52	DD	00071	2\$:	PUSHL BCB
				00000000G	00	01	FB	00073	CALLS	#1, CRYPTO_DECR_BLOCK
					15	1A	11	0007A	BRB	5\$
						53	E9	0007C	3\$:	BLBC CHK_SAVESET, 4\$
						52	DD	0007F	PUSHL	BCB
				00000000G	00	01	FB	00081	CALLS	#1, CRYPTO_CHKSAV
					09	50	E9	00088	BLBC	R0, 4\$
				00000000G	00	8F	DD	0008B	PUSHL	#BACKUP\$ ENCSAVSET
					65	01	FB	00091	CALLS	#1, LIB\$SIGNAL
						53	94	00094	4\$:	CLRB CHK_SAVESET
						CF	9F	00096	5\$:	PUSHAB LIST_ONE_RECORD
						52	DD	0009A	PUSHL	BCB
				00000000G	00	02	FB	0009C	CALLS	#2, DEBLOCK
						9A	11	000A3	BRB	1\$
						7E	D4	000A5	6\$:	CLRL -(SP)
				00000000G	00	01	FB	000A7	CALLS	#1, FIN_IN_SAVE
						04	000AE	7\$:	RET	2974

; Routine Size: 175 bytes, Routine Base: CODE + 12BC

LIST
V04-000

List a save set
LIST - main list routine

I 1
16-Sep-1984 00:06:23
14-Sep-1984 11:53:54

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]LIST.B32;1 (12) Page 57

: 1432
: 1433
2975 1 END
2976 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	4971	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]LIB.L32;1	18619	60	0	1000	00:02.2

COMMAND QUALIFIERS

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

: Size: 2650 code + 4445 data bytes
: Run Time: 01:04.7
: Elapsed Time: 03:28.5
: Lines/CPU Min: 2760
: Lexemes/CPU-Min: 32477
: Memory Used: 595 pages
: Compilation Complete

0011 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window shows a different screen from the VAX/VMS operating system. The screens contain various system messages, command prompts, and data listings. Some screens are clearly labeled with titles like 'FASTSCAN LIS', 'JOURNAL LIS', and 'LIST LIS'. The text is small and dense, typical of a terminal display. The overall appearance is that of a multi-terminal environment, possibly a console or a batch processing system.

