

BBBBBBBBBBBB		AAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPP		
BBBBBBBBBBBB		AAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPP		
BBBBBBBBBBBB		AAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPP		
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
BBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKK	KKK	UUU	UUU	PPPPPPPPPPP	
BBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKK	KKK	UUU	UUU	PPPPPPPPPPP	
BBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKK	KKK	UUU	UUU	PPPPPPPPPPP	
BBB	BBB	AAAAAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPP		
BBB	BBB	AAAAAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPP		
BBB	BBB	AAAAAAAAAAAAA	CCCCCCCCCCCC	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	
BBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUU	UUU	PPP	
BBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUU	UUU	PPP	
BBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUU	UUU	PPP	

```
MM      MM      AAAAAA      IIIIII      NN      NN
MM      MM      AAAAAA      IIIIII      NN      NN
MMMM    MMMM    AA          AA      II      NN      NN
MMMM    MMMM    AA          AA      II      NN      NN
MM      MM      AA          AA      II      NNNN     NN
MM      MM      AA          AA      II      NNNN     NN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      II      NN      NN
MM      MM      AAAAAAAAAA     II      NN      NNNN
MM      MM      AAAAAAAAAA     II      NN      NNNN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      IIIIII     NN      NN
MM      MM      AA          AA      IIIIII     NN      NN
.....
.....
.....
.....
```

```
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
LLLLLLLLLLLL IIIIII      SSSSSSSS
LLLLLLLLLLLL IIIIII      SSSSSSSS
```

```

1 0001 0 MODULE MAIN      (%TITLE 'Main module'
2 0002 0                  MAIN = MAIN,
3 0003 0                  IDENT = 'V04-000'
4 0004 0                  ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 *  ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 *  TRANSFERRED.
20 0020 1 *
21 0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 *  CORPORATION.
24 0024 1 *
25 0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 **
33 0033 1 FACILITY:
34 0034 1     Backup/Restore
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1     This is the main module.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1     VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: M. Jack, CREATION DATE: 25-Aug-1980
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1     V03-005 JEP0003      J. Eric Pollack,      23-Apr-1983  10:53
48 0048 1     Add support for encrypted savesets.
49 0049 1
50 0050 1     V03-004 ACG0313      Andrew C. Goldstein,  12-Feb-1983  16:24
51 0051 1     Add routine subtitles
52 0052 1
53 0053 1     V03-003 LMP0044      L. Mark Pilant,      21-Oct-1982  15:15
54 0054 1     Add support for saving and restoring ACL's.
55 0055 1
56 0056 1     V03-002 LMP0032      L. Mark Pilant,      22-Jun-1982  10:36
57 0057 1     Add support for wildcard save set names on a LIST or RESTORE

```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

0058 1
0059 1
0060 1
0061 1
0062 1
0063 1
0064 1
0065 1
0066 1
0067 1
0068 1
0069 1
0070 1
0071 1
0072 1
0073 1
0074 1
0075 1
0076 1
0077 1
0078 1
0079 1
0080 1
0081 1
0082 1
0083 1
0084 1

operation.
V03-001 MLJ0088 Martin L. Jack, 13-Apr-1982 15:27
Avoid access violation for /LIST/VERIFY.
V02-006 MLJ0054 Martin L. Jack, 15-Oct-1981 17:06
Exit with proper \$STATUS value. Implement /VOLUME. Move
STAACP globals to common.
V02-005 MLJ0037 Martin L. Jack, 29-Aug-1981 16:06
Avoid STA_MOUNT of output for /IMAGE/COMPARE.
V02-004 ACG0211 Andrew C. Goldstein, 16-Jul-1981 10:29
Implement sequential disk support
V02-003 MLJ0020 Martin L. Jack, 20-Apr-1981 22:07
Implement /JOURNAL qualifier.
V02-002 MLJ0012 Martin L. Jack, 27-Mar-1981 16:06
Force /VERIFY off if /COMPARE specified
V02-001 MLJ0010 Martin L. Jack, 25-Mar-1981 16:12
Reorganize global storage. Add capability for listing
concurrent with another operation. Add image restore.
Add standalone operation.

```

: 86 0085 1 REQUIRE 'SRCS:COMMON';
: 87 1191 1 LIBRARY 'SYSSLIBRARY:STARLET';
: 88 1192 1 REQUIRE 'LIBS:BACKDEF';
: 89 1642 1
: 90 1643 1
: 91 1644 1 FORWARD ROUTINE
: 92 1645 1     MAIN_HANDLER,           ! Condition handler for main routine
: 93 1646 1     MAIN;                ! Main routine
: 94 1647 1
: 95 1648 1
: 96 1649 1 EXTERNAL ROUTINE
: 97 1650 1     LIB$CRC_TABLE: ADDRESSING_MODE(GENERAL),
: 98 1651 1     ! Compute CRC vectors
: 99 1652 1     STA_INIT: NOVALUE WEAK, ! Initialize standalone version
100 1653 1     STA_RESTART: NOVALUE WEAK, ! Restart standalone version
101 1654 1     STA_MOUNT: NOVALUE, ! Mount a volume
102 1655 1     COMMAND: NOVALUE, ! Parse command
103 1656 1     SAVE: NOVALUE, ! Driver for save
104 1657 1     RESTORE: NOVALUE, ! Driver for restore
105 1658 1     INIT_LIST: NOVALUE, ! Initialize listing
106 1659 1     FIN_LIST: NOVALUE, ! Finish listing
107 1660 1     OPEN_JOURNAL: NOVALUE WEAK, ! Open journal file
108 1661 1     CLOSE_JOURNAL: NOVALUE WEAK, ! Close journal file
109 1662 1     LIST: NOVALUE, ! Driver for list-only
110 1663 1     LIST_JOURNAL: NOVALUE WEAK, ! List journal file
111 1664 1     ANALYZE: NOVALUE, ! Driver for analysis
112 1665 1     FILE_ERROR: NOVALUE, ! Signal a file related error
113 1666 1     CRYPTO_INIT: NOVALUE WEAK, ! Initialize saveset encryption
114 1667 1     CRYPTO_FINI: NOVALUE WEAK, ! Finalize encryption streams
115 1668 1
116 1669 1 EXTERNAL LITERAL
117 1670 1     BACKUPS_BADSETCNT,
118 1671 1     BACKUPS_OPENIN,
119 1672 1     BACKUPS_NOFILES;
120 1673 1
121 1674 1
122 1675 1 OWN
123 1676 1     EXIT_STATUS: BBLOCK[4]; ! Image exit status
124 1677 1
125 1678 1
126 1679 1 G$DEFINE(GBL); ! Define global common area

```

```

128 1680 1 %SBTTL 'MAIN_HANDLER - main condition handler'
129 1681 1 ROUTINE MAIN_HANDLER(SIG,MECH)=
130 1682 1
131 1683 1 |**
132 1684 1 |
133 1685 1 | FUNCTIONAL DESCRIPTION:
134 1686 1 | This routine is established as the condition handler for routine MAIN.
135 1687 1 | It saves the most severe error for the image exit status.
136 1688 1 |
137 1689 1 | INPUT PARAMETERS:
138 1690 1 | SIG - Standard VMS condition handler parameters.
139 1691 1 | MECH
140 1692 1 |
141 1693 1 | IMPLICIT INPUTS:
142 1694 1 | NONE
143 1695 1 |
144 1696 1 | OUTPUT PARAMETERS:
145 1697 1 | NONE
146 1698 1 |
147 1699 1 | IMPLICIT OUTPUTS:
148 1700 1 | EXIT_STATUS - The most severe error status.
149 1701 1 |
150 1702 1 | ROUTINE VALUE:
151 1703 1 | $$$_RESIGNAL.
152 1704 1 |
153 1705 1 | SIDE EFFECTS:
154 1706 1 | NONE
155 1707 1 |
156 1708 1 | --
157 1709 1 |
158 1710 2 BEGIN
159 1711 2 MAP
160 1712 2 SIG: REF BBLOCK, ! Signal parameters
161 1713 2 MECH: REF BBLOCK; ! Mechanism parameters
162 1714 2 BIND
163 1715 2 COND= SIG[CHFSL_SIG_NAME]: BBLOCK; ! Condition
164 1716 2
165 1717 2
166 1718 2 IF
167 1719 2 NOT .COND AND
168 1720 2 (.COND[STSSV_SEVERITY] GTRU .EXIT_STATUS[STSSV_SEVERITY] OR
169 1721 2 .EXIT_STATUS[STSSV_SEVERITY])
170 1722 2 THEN
171 1723 2 EXIT_STATUS = .COND;
172 1724 2
173 1725 2
174 1726 2 $$$_RESIGNAL
175 1727 1 END;

```

```

.TITLE MAIN Main module
.IDENT \V04-000\

.PSECT COMMON,NOEXE, OVR,2

0000 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:

0011C	RWSV_IN_ERRORS:	.BLKB	4
		.BLKB	2
0011E	RWSV_IN_XORUSE:	.BLKB	2
		.BLKB	2
00120	RWSV_IN_ORGERR:	.BLKB	8
		.BLKB	4
00128	RWSV_IN_VBN:	.BLKB	4
		.BLKB	4
0012C	RWSV_IN_VBN 0:	.BLKB	4
		.BLKB	4
00130	RWSV_ALLOC:	.BLKB	4
		.BLKB	4
00134	RWSV_EOF:	.BLKB	4
		.BLKB	4
00138	RWSV_OUT_SEQ:	.BLKB	4
		.BLKB	4
0013C	RWSV_OUT_VBN:	.BLKB	4
		.BLKB	4
00140	RWSV_OUT_BLOCK_COUNT:	.BLKB	4
		.BLKB	4
00144	RWSV_OUT_ERRORS:	.BLKB	2
		.BLKB	2
00146	RWSV_SEQ_ERRORS:	.BLKB	2
		.BLKB	2
00148	RWSV_OUT_GROUP_COUNT:	.BLKB	1
		.BLKB	1
00149	RWSV_PADDING:	.BLKB	3
		.BLKB	112
0014C	QUAL:	.BLKB	112
001BC	COM_SSNAME:	.BLKB	8
		.BLKB	8
001C4	COM_VALID_TYPES:	.BLKB	2
		.BLKB	2
001C6	COM_FLAGS:	.BLKB	2
		.BLKB	2
001C8	COM_PADDING:	.BLKB	1
		.BLKB	1
001C9	COM_BUFF_COUNT:	.BLKB	1
		.BLKB	1
001CA	COM_I_SETCOUNT:	.BLKB	1
		.BLKB	1
001CB	COM_O_SETCOUNT:	.BLKB	1
		.BLKB	1
001CC	COM_I_STRUCNAME:	.BLKB	12
		.BLKB	12
001DB	COM_O_STRUCNAME:	.BLKB	12
		.BLKB	12
001E4	COM_O_BSRDATE:	.BLKB	8
		.BLKB	8
001EC	ALT_SSNAME:	.BLKB	32
		.BLKB	32
0020C	INPUT_FUNC:	.BLKB	1
		.BLKB	1
0020D	INPUT_RTYPE:	.BLKB	1
		.BLKB	1
0020E	OUTPUT_FUNC:	.BLKB	1

MAIN
V04-000

Main module
MAIN_HANDLER - main condition handler

L 3
16-Sep-1984 00:11:39
14-Sep-1984 11:53:55

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[BACKUP.SRC]MAIN.B32;1 Page 7
(3)

0020F	FAST_STRUCLEV:	.BLKB	1
00210	INPUT_BEG:	.BLKB	1
00210	INPUT_CHAN:	.BLKB	0
00214	INPUT_FLAGS:	.BLKB	4
00216	INPUT_PADDING:	.BLKB	2
00218	INPUT_FAB:	.BLKB	2
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	4
00240	INPUT_STATBLK:	.BLKB	8
00248	INPUT_HDR_BEG:	.BLKB	8
00248	INPUT_CREDATE:	.BLKB	0
00250	INPUT_REVDATE:	.BLKB	8
00258	INPUT_EXPDATE:	.BLKB	8
00260	INPUT_BAKDATE:	.BLKB	8
00268	INPUT_FILEOWNER:	.BLKB	8
0026C	INPUT_FILECHAR:	.BLKB	4
00270	INPUT_RECATTR:	.BLKB	4
00290	INPUT_HDR_END:	.BLKB	32
00290	INPUT_END:	.BLKB	0
00290	INPUT_PROC_LIST:	.BLKB	0
00294	INPUT_PLACEMENT:	.BLKB	4
0029C	INPUT_VBN_LIST:	.BLKB	8
		.BLKB	8

002A4 INPUT_PLACE_LEN:
 .BLRB 2
002A6 INPUT_PADDING_2:
 .BLKB 2
002A8 OUTPUT_BEG:
 .BLKB 0
002A8 OUTPUT_CHAN:
 .BLKB 4
002AC OUTPUT_FLAGS:
 .BLKB 2
002AE OUTPUT_PADDING:
 .BLKB 2
002B0 OUTPUT_FAB:
 .BLKB 4
002B4 OUTPUT_NAM:
 .BLKB 4
002B8 OUTPUT_BCB:
 .BLKB 4
002BC OUTPUT_QUAL:
 .BLKB 4
002C0 OUTPUT_BAD:
 .BLKB 4
002C4 OUTPUT_BLOCK:
 .BLKB 4
002C8 OUTPUT_MAXBLOCK:
 .BLKB 4
002CC OUTPUT_DEVGEO:
 .BLKB 8
002D4 OUTPUT_ATTBUF:
 .BLKB 144
00364 OUTPUT_END:
 .BLKB 0
00364 LIST_TOTFILES:
 .BLKB 4
00368 LIST_TOTSIZE:
 .BLKB 4
0036C VERIFY_FAB:
 .BLKB 4
00370 VERIFY_USE_COUNT:
 .BLKB 4
00374 VERIFY_QUAL:
 .BLKB 4
00378 COMPARE_BCB:
 .BLKB 4
0037C FAST_BUFFER:
 .BLKB 4
00380 FAST_BUFFER_SIZE:
 .BLRB 4
00384 FAST_RVN:
 .BLKB 1
00385 FAST_PADDING:
 .BLKB 1
00386 DIR_VERLIMIT:
 .BLKB 2
00388 FAST_VOL_BEG:
 .BLKB 0
00388 FAST_IMAP_SIZE:

MAIN
V04-000

Main module
MAIN_HANDLER - main condition handler

N 3
16-Sep-1984 00:11:39
14-Sep-1984 11:53:55

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

0038C	FAST_IMAP:	.BLKB	4
00390	FAST_HDR_OFFSET:	.BLKB	4
00394	FAST_BOOT_LBN:	.BLKB	4
00398	FAST_VOL_END:	.BLKB	4
00398	JOUR_BUFFER:	.BLKB	0
0039C	JOUR_DIR:	.BLKB	4
003A0	JOUR_HIBLK:	.BLKB	4
003A4	JOUR_EFBLK:	.BLKB	4
003A8	JOUR_INBLK:	.BLKB	4
003AC	JOUR_FFBYTE:	.BLKB	2
003AE	JOUR_INBYTE:	.BLKB	2
003B0	JOUR_STRUCT_LEV:	.BLKB	2
003B2	JOUR_COUNT:	.BLKB	1
003B3	JOUR_REVERSE:	.BLKB	1
003B4	JOUR_EXSZ:	.BLKB	2
003B6	JOUR_PADDING:	.BLKB	2
003B8	CHKPT_HIGH_SP:	.BLKB	4
003BC	CHKPT_LOW_SP:	.BLKB	4
003C0	CHKPT_STACK:	.BLKB	4
003C4	CHKPT_VARS:	.BLKB	4
003C8	CHKPT_STATUS:	.BLKB	4
003CC	DIR_BEG:	.BLKB	0
003CC	DIR_CHAN:	.BLKB	4
003D0	DIR_NAM:	.BLKB	4
003D4	DIR_DEV_DESC:	.BLKB	4
003D8	DIR_SEL_DIR:	.BLKB	8
003E0	DIR_SEL_NTV:	.BLKB	8
003E8	DIR_STRUCLEV:	.BLKB	1
003E9	DIR_LEVELS:	.BLKB	1

```
003EA DIR_FLAGS:
      .BLKB 1
003EB DIR_STATUS:
      .BLKB 1
003EC DIR_STRING:
      .BLKB 320
0052C DIR_STACK:
      .BLKB 612
00790 DIR_SP: .BLKB 4
00794 DIR_SEL_LATEST:
      .BLKB 4
00798 DIR_END: .BLKB 0
00798 DIR_SCANLIMIT:
      .BLKB 36
007BC INPUT_MTL:
      .BLKB 4
007C0 OUTPUT_MTL:
      .BLKB 4
007C4 CURRENT_MTL:
      .BLKB 4
007C8 CURRENT_VCB:
      .BLKB 4
007CC CURRENT_WCB:
      .BLKB 4
007D0 ACL_FIB_DESCR:
      .BLKB 8
007D8 ACL_FIB: .BLKB 64
00818 ACL_LENGTH:
      .BLKB 4
0081C ACL_BUFFER:
      .BLKB 4
00820 CRYP_IN_CONTEXT:
      .BLKB 4
00824 CRYP_OU_CONTEXT:
      .BLKB 4
00828 CRYP_DA_CONTEXT:
      .BLKB 4
0082C CRYP_DATA_ENCIV:
      .BLKB 8
00834 CRYP_DATA_CODE:
      .BLKB 4
00838 CRYP_DATA_KEY:
      .BLKB 8
00840 CRYP_DATA_IV:
      .BLKB 8
00848 CRYP_DATA_CKSM:
      .BLKB 4
      .PSFCT DATA,NOEXE,2
00000 EXIT_STATUS:
      .BLKB 4
      .EXTRN LIB$CRC_TABLE, STA_MOUNT
      .EXTRN COMMAND, SAVE, RESTORE
      .EXTRN INIT_LIST, FIN_LIST
      .EXTRN LIST, ANALYZE, FILE_ERROR
```

MAIN
V04-000

Main module
MAIN_HANDLER - main condition handler

C 4
16-Sep-1984 00:11:39
14-Sep-1984 11:53:55

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

.EXTRN BACKUP\$BADSETCNT
.EXTRN BACKUP\$OPENIN, BACKUP\$NOFILES
.WEAK STA_INIT, STA_RESTART
.WEAK OPEN_JOURNAL, CLOSE_JOURNAL
.WEAK LIST_JOURNAL, CRYPTO_INIT
.WEAK CRYPTO_FINI

.PSECT CODE, NOWRT, 2

0004 00000 MAIN_HANDLER:

		52	00000000'	EF	9E	00002	.WORD	Save R2	:	1681	
	50	04	AC	04	C1	00009	MOVAB	EXIT_STATUS, R2	:		
			12	60	E8	0000E	ADDL3	#4, SIG, R0	:	1715	
51	62		03	00	EF	00011	BLBS	(R0), 2\$:	1719	
51	60		03	00	ED	00016	EXTZV	#0, #3, EXIT_STATUS, R1	:	1720	
				03	1A	0001B	CMPZV	#0, #3, (R0), R1	:		
			03	62	E9	0001D	BGTRU	1\$:		
			62	60	D0	00020	BLBC	EXIT_STATUS, 2\$:	1721	
			50	0918	8F	3C	1\$:	MOVL	(R0), EXIT_STATUS	:	1723
						04	2\$:	MOVZWL	#2328, R0	:	1727
								RET	:		

; Routine Size: 41 bytes, Routine Base: CODE + 0000

```

177 1728 1 %SBTTL 'MAIN - main routine for BACKUP'
178 1729 1 ROUTINE MAIN=
179 1730 1
180 1731 1 !++
181 1732 1
182 1733 1 ! FUNCTIONAL DESCRIPTION:
183 1734 1 ! This is the main routine. It provides the top-level control.
184 1735 1
185 1736 1 ! INPUT PARAMETERS:
186 1737 1 ! Standard VMS activation parameters (not used).
187 1738 1
188 1739 1 ! IMPLICIT INPUTS:
189 1740 1 ! NONE
190 1741 1
191 1742 1 ! OUTPUT PARAMETERS:
192 1743 1 ! NONE
193 1744 1
194 1745 1 ! IMPLICIT OUTPUTS:
195 1746 1 ! NONE
196 1747 1
197 1748 1 ! ROUTINE VALUE:
198 1749 1 ! Exit status.
199 1750 1
200 1751 1 ! SIDE EFFECTS:
201 1752 1 ! NONE
202 1753 1
203 1754 1 !--
204 1755 1
205 1756 2 BEGIN
206 1757 2 BUILTIN
207 1758 2 FP;
208 1759 2
209 1760 2 LOCAL
210 1761 2 FAB : REF BBLOCK; ! Pointer to the input FAB
211 1762 2
212 1763 2 ! Establish the handler.
213 1764 2
214 1765 2 EXIT STATUS = $$$ NORMAL;
215 1766 2 .FP = MAIN_HANDLER;
216 1767 2
217 1768 2
218 1769 2 ! Initialize the standalone version if STA_INIT is present. (This will
219 1770 2 ! reestablish the handler in STA_INIT for this routine.)
220 1771 2
221 1772 2 IF STA_INIT NEQ 0 THEN STA_INIT();
222 1773 2
223 1774 2
224 1775 2 ! Get the command.
225 1776 2
226 1777 2 COMMAND();
227 1778 2
228 1779 2
229 1780 2 ! Initialize various globals.
230 1781 2
231 1782 2 LIB$CRC_TABLE(UPLIT(%'120001'), RWSV_CRC16);
232 1783 2 LIB$CRC_TABLE(UPLIT(%'EDB88320'), RWSV_AUTODIN);
233 1784 2 INPUT_FUNC = IOS_READVBLK;

```

```

: 234      1785 2 INPUT RTYPE = BRH$K_VBN;
: 235      1786 2 OUTPUT FUNC = IOS_WRITEVBLK;
: 236      1787 2 COM_VA[ID TYPES =
: 237      1788 2     1^BRH$K_NULL +
: 238      1789 2     1^BRH$K_SUMMARY +
: 239      1790 2     1^BRH$K_VOLUME +
: 240      1791 2     1^BRH$K_FILE +
: 241      1792 2     1^BRH$K_VBN +
: 242      1793 2     1^BRH$K_FID +
: 243      1794 2     1^BRH$K_FILE_EXT;
: 244      1795 2 IF .QUAL[QUAL_PHYS]
: 245      1796 2 THEN
: 246      1797 2 BEGIN
: 247      1798 2     INPUT_FUNC = IOS_READLBLK;
: 248      1799 2     INPUT_RTYPE = BRH$K_LBN;
: 249      1800 2     OUTPUT_FUNC = IOS_WRITELBLK;
: 250      1801 2     COM_VA[ID TYPES =
: 251      1802 2         1^BRH$K_NULL +
: 252      1803 2         1^BRH$K_SUMMARY +
: 253      1804 2         1^BRH$K_PHYSVOL +
: 254      1805 2         1^BRH$K_LBN;
: 255      1806 2     END;
: 256      1807 2
: 257      1808 2 ! Initialize the saveset encryption context if required
: 258      1809 2
: 259      1810 2 IF .QUAL[QUAL_SS_ENCRYP] AND (CRYPTO_INIT NEQ 0 THEN CRYPTO_INIT();
: 260      1811 2
: 261      1812 2 ! Call the standalone ACP to mount the input and output volumes if required.
: 262      1813 2
: 263      1814 2 IF NOT .QUAL[QUAL_PHYS]
: 264      1815 2 THEN
: 265      1816 2 BEGIN
: 266      1817 2     IF .QUAL[QUAL_IF11] AND .COM_FLAGS[COM_STANDALONE]
: 267      1818 2     THEN
: 268      1819 2         STA_MOUNT(0);
: 269      1820 2
: 270      1821 2
: 271      1822 2     IF .QUAL[QUAL_OF11] AND (.COM_FLAGS[COM_STANDALONE] OR (.QUAL[QUAL_IMAG] AND NOT .QUAL[QUAL_COMP]))
: 272      1823 2     THEN
: 273      1824 2         BEGIN
: 274      1825 2             IF .QUAL[QUAL_VOLU]
: 275      1826 2             THEN
: 276      1827 2                 BEGIN
: 277      1828 2                     LOCAL
: 278      1829 2                         MTL: REF BBLOCK; ! Pointer to MTL entry
: 279      1830 2
: 280      1831 2                         STA_MOUNT(1, 0);
: 281      1832 2                         MTL = .OUTPUT_MTL;
: 282      1833 2                         IF .MTL[MTL_SETCOUNT] NEQ 1 THEN SIGNAL(BACKUP$BADSETCNT);
: 283      1834 2                         MTL[MTL_RVN_BASE] = .QUAL[QUAL_VOLU_VALUE];
: 284      1835 2                         STA_MOUNT(1, .QUAL[QUAL_VOLU_VALUE]);
: 285      1836 2                     END
: 286      1837 2                 ELSE
: 287      1838 2                     STA_MOUNT(1);
: 288      1839 2                 END;
: 289      1840 2             END;
: 290      1841 2         END;

```

```

291 1842 2
292 1843 2
293 1844 2
294 1845 2
295 1846 2
296 1847 2
297 1848 2
298 1849 2
299 1850 2
300 1851 2
301 1852 2
302 1853 2
303 1854 2
304 1855 2
305 1856 2
306 1857 4
307 1858 4
308 1859 4
309 1860 5
310 1861 5
311 1862 5
312 1863 5
313 1864 5
314 1865 5
315 1866 4
316 1867 5
317 1868 5
318 1869 5
319 1870 5
320 1871 5
321 1872 5
322 1873 5
323 1874 5
324 1875 5
325 1876 5
326 1877 5
327 1878 5
328 1879 4
329 1880 4
330 1881 4
331 1882 3
332 1883 4
333 1884 4
334 1885 4
335 1886 4
336 1887 4
337 1888 4
338 1889 4
339 1890 5
340 1891 5
341 1892 5
342 1893 5
343 1894 5
344 1895 5
345 1896 4
346 1897 4
347 1898 3

: Initialize the listing if required.
: IF .QUAL[QUAL_LIST] THEN INIT_LIST();
: Execute the command.
:
INPUT_FLAGS[INPUT_SSFIND] = FALSE;          ! No save set found as yet
DO
BEGIN
IF .QUAL[QUAL_LIST] AND .QUAL[QUAL_OUTP_LIST] EQL 0
THEN
BEGIN
IF .QUAL[QUAL_INPU_LIST] EQL 0
THEN
BEGIN
: List journal.
IF .QUAL[QUAL_JOUR] THEN LIST_JOURNAL();
END
ELSE
BEGIN
: List save set.
: Set all record types valid and call LIST or ANALYZE as appropriate.
QUAL[QUAL_VERI] = FALSE;
COM_VALID_TYPES = -1;
IF .QUAL[QUAL_ANAL]
THEN
ANALYZE()
ELSE
LIST()
END;
IF NOT .COM_FLAGS[COM_EOV] THEN FIN_LIST(FALSE);
END
ELSE
BEGIN
: If disk-to-disk or disk-to-tape operation, call SAVE. If tape-to-disk
: operation, call RESTORE.
IF .QUAL[QUAL_IF11]
THEN
BEGIN
IF NOT .QUAL[QUAL_OSAV] THEN QUAL[QUAL_JOUR] = FALSE;
IF .QUAL[QUAL_JOUR] THEN OPEN_JOURNAL(TRUE);
SAVE();
IF .QUAL[QUAL_JOUR] THEN CLOSE_JOURNAL(TRUE);
END
ELSE
RESTORE();
END;

```



```

1899      END
1900      FILE NOT .COM_FLAGS[COM_EOV] AND .INPUT_FLAGS[INPUT_WILDSAVE];
1901
1902      IF NOT .COM_FLAGS[COM_VERIFYING]
1903      THEN
1904          BEGIN
1905              IF .INPUT_FLAGS[INPUT_WILDSAVE]
1906              THEN IF NOT .INPUT_FLAGS[INPUT_SSFIND]
1907              THEN
1908                  BEGIN
1909                      FAB = .RWSV_SAVE FAB;
1910                      FILE_ERROR (BACKUPS_OPENIN+STSSK_SEVERE, .FAB, SS$_NOSUCHFILE);
1911                  END
1912              ELSE IF NOT .QUAL[QUAL_PHYS] AND NOT .COM_FLAGS[COM_FILESEEN] AND NOT .QUAL[QUAL_LIST]
1913              THEN SIGNAL (BACKUPS_NOFILES, 1,
1914                          BBLOCK[.QUAL[QUAL_INPU_LIST], QUAL_EXP_DESC]);
1915          END;
1916
1917      ! Finish the listing if required.
1918
1919      IF .QUAL[QUAL_LIST] THEN FIN_LIST(TRUE);
1920
1921
1922      ! If this is the standalone version, restart the image.
1923
1924      IF .COM_FLAGS[COM_STANDALONE] THEN STA_RESTART();
1925
1926      ! Close any remaining encryption contexts
1927
1928      IF .QUAL[QUAL_SS_ENCRYP] AND crypto_fini NEQA 0 THEN crypto_fini();
1929
1930      ! Return to operating system.
1931
1932      .EXIT_STATUS OR STSSM_INHIB_MSG
1933      1 END;

```

```

0000A001 00029 .BLKB 3
EDB88320 0002C P.AAA: .LONG 40961
00030 P.AAB: .LONG -306674912

```

```

OFFC 00000 MAIN: .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 : 1729
5B 00000000' EF 9E 00002 MOVAB EXIT_STATUS, R11
5A 00000000G 00 9E 00009 MOVAB CRYPTO_FINI, R10
59 00000000G 00 9E 00010 MOVAB FIN_LIST, R9
58 00000000G 00 9E 00017 MOVAB LIB$SIGNAL, R8
57 00000000G 00 9E 0001E MOVAB CRYPTO_INIT, R7
56 00000000G 00 9E 00025 MOVAB LIB$CRC_TABLE, R6
55 00000000G 00 9E 0002C MOVAB STA_INIT, R5
54 00000000G 00 9E 00033 MOVAB STA_MOUNT, R4
53 00000000' EF 9E 0003A MOVAB QUAL+12, R3
6B 01 D0 00041 MOVAB #1, EXIT_STATUS : 1765
6D 85 AF 9E 00044 MOVAB MAIN_HANDLER, (FP) : 1766
50 65 9E 00048 MOVAB STA_INIT, R0 : 1772

```

			03	13	0004B	BEQL	1\$				
			00	FB	0004D	CALLS	#0, STA_INIT				
	00000000G	00	00	FB	00050	1\$:	CALLS	#0, COMMAND	1777		
			FF00	C3	9F	00057	PUSHAB	RW\$V_CRC16	1782		
			9A	AF	9F	0005B	PUSHAB	P.AAX			
		66	02	FB	0005E	CALLS	#2, LIB\$CRC_TABLE				
			FF40	C3	9F	00061	PUSHAB	RW\$V_AUTODIR	1783		
			94	AF	9F	00065	PUSHAB	P.AAB			
		66	02	FB	00068	CALLS	#2, LIB\$CRC_TABLE				
	00B4	C3	0431	8F	80	0006B	MOVW	#1073, INPUT_FUNC	1784		
	00B6	C3		30	90	00072	MOVB	#48, OUTPUT_FUNC	1786		
	6C	A3	019F	8F	80	00077	MOVW	#415, COM_VALID_TYPES	1793		
11		63		05	E1	0007D	BBC	#5, QUAL+T2, 2\$	1795		
	00B4	C3	0621	8F	80	00081	MOVW	#1569, INPUT_FUNC	1798		
	00B6	C3		20	90	00088	MOVB	#32, OUTPUT_FUNC	1800		
	6C	A3	63	8F	9B	0008D	MOVZBW	#99, COM_VALID_TYPES	1804		
08		A3		04	E1	00092	2\$:	BBC	#4, QUAL+14, 3\$	1810	
	02	A3		67	9E	00097	MOVAB	CRYPTO_INIT, R0			
		50		03	13	0009A	BEQL	3\$			
		67		00	FB	0009C	CALLS	#0, CRYPTO_INIT			
56		63		05	E0	0009F	3\$:	BBS	#5, QUAL+12, 8\$	1814	
0A	03	A3		04	E1	000A3	BBC	#4, QUAL+15, 4\$	1817		
05	6E	A3		01	E1	000A8	BBC	#1, COM_FLAGS, 4\$			
				7E	D4	000AD	CLRL	-(SP)	1819		
		64		01	FB	000AF	CALLS	#1, STA_MOUNT			
42	03	A3		06	E1	000B2	4\$:	BBC	#6, QUAL+15, 8\$	1822	
0A	6E	A3		01	E0	000B7	BBS	#1, COM_FLAGS, 5\$			
38	FE	A3		03	E1	000BC	BBC	#3, QUAL+10, 8\$			
			FC	A3	95	000C1	TSTB	QUAL+8			
				33	19	000C4	BLSS	8\$			
		2A	02	A3	E9	000C6	5\$:	BLBC	QUAL+14, 7\$	1825	
		7E		01	7D	000CA	MOVQ	#1, -(SP)	1831		
		64		02	FB	000CD	CALLS	#2, STA_MOUNT			
		52	0668	C3	D0	000D0	MOVL	OUTPUT_MTL, MTL	1832		
		01	1F	A2	91	000D5	CMPB	31(MTL), #1	1833		
				09	13	000D9	BEQL	6\$			
		00000000G		8F	DD	000DB	PUSHL	#BACKUP\$ BADSETCNT			
		68		01	FB	000E1	CALLS	#1, LIB\$SIGNAL			
	30	A2	43	A3	90	000E4	6\$:	MOVB	QUAL+79, 48(MTL)	1834	
		7E	43	A3	9A	000E9	MOVZBL	QUAL+79, -(SP)	1835		
				01	DD	000ED	PUSHL	#1			
		64		02	FB	000EF	CALLS	#2, STA_MOUNT			
				05	11	000F2	BRB	8\$	1825		
				01	DD	000F4	7\$:	PUSHL	#1	1838	
		64		01	FB	000F6	CALLS	#1, STA_MOUNT			
		07	FF	A3	E9	000F9	8\$:	BLBC	QUAL+11, 9\$	1845	
	00000000G	00		00	FB	000FD	CALLS	#0, INIT_LIST			
	00BD	C3		01	8A	00104	9\$:	BICB2	#1, INPUT_FLAGS+1	1851	
		40	FF	A3	E9	00109	10\$:	BLBC	QUAL+11, T4\$	1855	
			F8	A3	D5	0010D	TSTL	QUAL+4			
				3B	12	00110	BNEQ	14\$			
			F4	A3	D5	00112	TSTL	QUAL	1858		
				0E	12	00115	BNEQ	11\$			
26	FE	A3		06	E1	00117	BBC	#6, QUAL+10, 13\$	1864		
	00000000G	00		00	FB	0011C	CALLS	#0, LIST_JOURNAL			
				1D	11	00123	BRB	13\$	1858		
		01	A3	80	8F	8A	00125	11\$:	BICB2	#128, QUAL+13	1872

6C	A3		01	AE	0012A	MNEGW	#1, COM_VALID_TYPES	1873
	09	FC	A3	E9	0012E	BLBC	QUAL+8, -12\$	1874
00000000G	00		00	FB	00132	CALLS	#0, ANALYZE	1876
			07	11	00139	BRB	13\$	
00000000G	00		00	FB	0013B	CALLS	#0, LIST	1878
	4F	6E	A3	E8	00142	BLBS	COM_FLAGS, 19\$	1880
			7E	D4	00146	CLRL	-(SP)	
			01	FB	00148	CALLS	#1, FIN_LIST	
	69		3B	11	0014B	BRB	18\$	1855
2F	03	A3	04	E1	0014D	BBC	#4, QUAL+15, 17\$	1888
			03	A3	95	TSTB	QUAL+15	1891
			05	19	00155	BLSS	15\$	
	FE	A3	40	8F	8A	BICB2	#64, QUAL+10	
09	FE	A3	06	E1	0015C	BBC	#6, QUAL+10, 16\$	1892
			C1	DD	00161	PUSHL	#1	
00000000G	00		01	FB	00163	CALLS	#1, OPEN_JOURNAL	
00000000G	00		00	FB	0016A	CALLS	#0, SAVE	1893
12	FE	A3	06	E1	00171	BBC	#6, QUAL+10, 18\$	1894
			01	DD	00176	PUSHL	#1	
00000000G	00		01	FB	00178	CALLS	#1, CLOSE_JOURNAL	
			07	11	0017F	BRB	18\$	1888
00000000G	00		00	FB	00181	CALLS	#0, RESTORE	1897
			09	E8	00188	BLBS	COM_FLAGS, 19\$	1900
03	00BC	C3	06	E1	0018C	BBC	#6, INPUT_FLAGS, 19\$	
			FF74	31	00192	BRW	10\$	
42	6E	A3	03	E0	00195	BBS	#3, COM_FLAGS, 21\$	1902
3C	00BC	C3	06	E1	0019A	BBC	#6, INPUT_FLAGS, 21\$	1905
		00BD	C3	E8	001A0	BLBS	INPUT_FLAGS+1, 20\$	1906
		A0	A3	D0	001A5	MOVL	RWSV SAVE FAB, FAB	1909
		0910	8F	3C	001A9	MOVZWL	#2320, -(SP)	1910
			50	DD	001AE	PUSHL	FAB	
		00000000G	8F	DD	001B0	PUSHL	#BACKUP\$ OPENIN+4	
00000000G	00		03	FB	001B6	CALLS	#3, FILE_ERROR	
			1D	11	001BD	BRB	21\$	1906
19	63		05	E0	001BF	BBS	#5, QUAL+12, 21\$	1912
14	6E	A3	02	E0	001C3	BBS	#2, COM_FLAGS, 21\$	
			A3	E8	001C8	BLBS	QUAL+11, 22\$	
7E	F4	A3	08	C1	001CC	ADDL3	#8, QUAL, -(SP)	1914
			01	DD	001D1	PUSHL	#1	
		00000000G	8F	DD	001D3	PUSHL	#BACKUP\$ NOFILES	
	68		03	FB	001D9	CALLS	#3, LIB\$SIGNAL	
	05	FF	A3	E9	001DC	BLBC	QUAL+11, 23\$	1919
			01	DD	001E0	PUSHL	#1	
	69		01	FB	001E2	CALLS	#1, FIN_LIST	
07	6E	A3	01	E1	001E5	BBC	#1, COM_FLAGS, 24\$	1924
00000000G	00		00	FB	001EA	CALLS	#0, STA_RESTART	
08	02	A3	04	E1	001F1	BBC	#4, QUAL+14, 25\$	1928
		50	6A	9E	001F6	MOVAB	CRYPTO_FINI, R0	
			03	13	001F9	BEQL	25\$	
	6A		00	FB	001FB	CALLS	#0, CRYPTO_FINI	
50	6B	10000000	8F	C9	001FE	BISL3	#268435456, EXIT_STATUS, R0	1932
			04	00	206	RET		1933

; Routine Size: 519 bytes, Routine Base: CODE + 0034

MAIN
V04-000

Main module
MAIN - main routine for BACKUP

J 4
16-Sep-1984 00:11:39
14-Sep-1984 11:53:55

VAX-11 BLISS-32 V4.0-742
DISK&VMSMASTER:[BACKUP.SRC]MAIN.B32;1 Page 18
(5)

: 384
: 385
: 1934 1 END
: 1935 0 ELUDOM

.EXTRN LIBSSIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
DATA	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
COMMON	2124	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	571	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_S255SDUA28:[SYSLIB]STARLET.L32;1	9776	12	0	581	00:01.2

COMMAND QUALIFIERS

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

: Size: 560 code + 2139 data bytes
: Run Time: 00:21.7
: Elapsed Time: 01:11.3
: Lines/CPU Min: 5347
: Lexemes/CPU-Min: 44813
: Memory Used: 324 pages
: Compilation Complete

