

[illegible]

.....

```
EEEEEEEEEE XX XX CCCCCCCC SSSSSSSS HH HH 000000 WW WW
EEEEEEEEEE XX XX CCCCCCCC SSSSSSSS HH HH 000000 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EEEEEEEE XX XX CC SSSSSS HHHHHHHHHH 00 00 WW WW
EEEEEEEE XX XX CC SSSSSS HHHHHHHHHH 00 00 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EE XX XX CC SS HH HH 00 00 WW WW
EEEEEEEEEE XX XX CCCCCCCC SSSSSSSS HH HH 000000 WW WW
EEEEEEEEEE XX XX CCCCCCCC SSSSSSSS HH HH 000000 WW WW
```

....
....
....
....

```
LL I I I I I SSSSSSSS
LL I I I I I SSSSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SSSSSS
LL I I SSSSSS
LL I I SS
LL I I SS
LL I I SS
LL I I SS
LLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLL I I I I I SSSSSSSS
```


[illegible]

EXCH\$SHOW
V04-000

SHOW verb dispatch routine
Module table of contents

D 11
16-Sep-1984 01:24:42
14-Sep-1984 12:29:08

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXC\$SHOW.B32;1

Page 2
(2)

```
: 51      0146 1 %SBTTL 'Module table of contents'
: 52      0147 1
: 53      0148 1 ! Module table of contents:
: 54      0149 1 !
: 55      0150 1 FORWARD ROUTINE
: 56      0151 1     exch$show_show
: 57      0152 1     ;
: 58      0153 1
: 59      0154 1 ! EXCHANGE facility routines
: 60      0155 1 !
: 61      0156 1 ! EXTERNAL ROUTINE
: 62      0157 1 !
: 63      0158 1
: 64      0159 1 ! Equated symbols:
: 65      0160 1 !
: 66      0161 1 ! LITERAL
: 67      0162 1 !
: 68      0163 1
: 69      0164 1 ! Bound declarations:
: 70      0165 1 !
: 71      0166 1 BIND
: 72      0167 1     null_string = %ASCID ''
: 73      0168 1     ;
```



```

: 75      0169 1 GLOBAL ROUTINE exch$show_show = %SBTTL 'exch$show_show'
: 76      0170 2 BEGIN
: 77      0171 2 ++
: 78      0172 2
: 79      0173 2 FUNCTIONAL DESCRIPTION:
: 80      0174 2
: 81      0175 2         This routine shows the mounted volumes.
: 82      0176 2
: 83      0177 2 INPUTS:
: 84      0178 2
: 85      0179 2         none
: 86      0180 2
: 87      0181 2 IMPLICIT INPUTS:
: 88      0182 2
: 89      0183 2         exch$a_gbl - pointer to all global data
: 90      0184 2
: 91      0185 2 OUTPUTS:
: 92      0186 2
: 93      0187 2         none
: 94      0188 2
: 95      0189 2 IMPLICIT OUTPUTS:
: 96      0190 2
: 97      0191 2         none
: 98      0192 2
: 99      0193 2 ROUTINE VALUE:
: 100     0194 2
: 101     0195 2         success
: 102     0196 2
: 103     0197 2 SIDE EFFECTS:
: 104     0198 2
: 105     0199 2         information is displayed on sys$output
: 106     0200 2 --
: 107     0201 2 $dbgtrc_prefix ('show_show> ');
: 108     0202 2
: 109     0203 2 LOCAL
: 110     0204 2     volb : $ref_bblock
: 111     0205 2     ;
: 112     0206 2
: 113     0207 2 ! Show VOLBs, volume control blocks, for mounted volumes
: 114     0208 2
: 115     0209 2 volb = .exch$a_gbl [excg$a_volb_use_flink]; ! Get pointer to first volume
: 116     0210 2 IF .volb EQL exch$a_gbl [excg$a_volb_use_flink] ! If pointer is same as header address, no mounted volumes
: 117     0211 2 THEN
: 118     0212 2     $exch_signal_return (exch$_novolumes);
: 119     0213 2
: 120     0214 2 ! Print the header lines
: 121     0215 2
: 122     0216 2 $print_desc (null_string);
: 123     0217 2 $print_lit ('Mounted volumes:');
```

```
125 0218 2 ! Print each mounted volume
126 0219 2
127 0220 2 WHILE .volb NEQ exch$a_gbl [excg$a_volb_use_flink]
128 0221 2 DO
129 0222 2 BEGIN
130 0223 2 BIND
131 0224 2 nam = volb [volb$a_nam] : $ref_bblock;
132 0225 2 LOCAL
133 0226 2 ptr1, ptr2;
134 0227 2
135 0228 2 $block_check (2, .volb, volb, 477);
136 0229 2 $print_desc (null_string);
137 0230 2
138 0231 2 $print_fao (' !AF', .volb [volb$l_vol_ident_len], volb [volb$t_vol_ident]);
139 0232 2
140 0233 2 $print_fao ('!8* volume format:!10* !AF', .volb [volb$l_vol_type_len], volb [volb$t_vol_type]);
141 0234 2
142 0235 2 IF .volb [volb$l_devclass] EQL dc$_disk
143 0236 2 THEN
144 0237 2 ptr1 = %ASCII 'disk'
145 0238 2 ELSE IF .volb [volb$l_devclass] EQL dc$_tape
146 0239 2 THEN
147 0240 2 ptr1 = %ASCII 'magtape'
148 0241 2 ELSE
149 0242 2 ptr1 = %ASCII 'unknown';
150 0243 2 IF .volb [volb$v_virtual]
151 0244 2 THEN
152 0245 2 ptr2 = %ASCII ' (virtual volume)'
153 0246 2 ELSE
154 0247 2 ptr2 = null_string;
155 0248 2 $print_fao ('!8* volume class:!11* !AS!AS', .ptr1, .ptr2);
156 0249 2
157 0250 2 IF .volb [volb$v_virtual]
158 0251 2 THEN
159 0252 2 $print_fao ('!8* virtual file name:!6* !AF', .nam [nam$b_rsl], .nam [nam$l_rsa])
160 0253 2 ELSE
161 0254 2 $print_fao ('!8* physical device name: !AF', .volb [volb$l_devnamlen], volb [volb$t_devnam]);
162 0255 2
163 0256 2 IF .volb [volb$b_vol_format] EQL volb$k_vfmt_rt11
164 0257 2 THEN
165 0258 2 BEGIN
166 0259 2 $print_fao ('!8* volume size:!12* !UL block!%S', .volb [volb$l_volmaxblock]);
167 0260 2 END;
168 0261 2
169 0262 2 IF NOT .volb [volb$v_write]
170 0263 2 THEN
171 0264 2 $print_fao ('!32* volume is write-locked');
172 0265 2
173 0266 2 IF .volb [volb$v_read_check]
174 0267 2 THEN
175 0268 2 $print_fao ('!32* /DATA_CHECK=READ is enabled');
176 0269 2
177 0270 2 IF .volb [volb$v_write_check]
178 0271 2 THEN
179 0272 2 $print_fao ('!32* /DATA_CHECK=WRITE is enabled');
180 0273 2
181 0274 2 volb = .volb [volb$a_flink]; ! Move to the next volume
```


EXCH\$SHOW
V04-000

SHOW verb dispatch routine
exch\$show_show

G 11
16-Sep-1984 01:24:42
14-Sep-1984 12:29:08

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCH\$SHOW.B32;1

Page 5
(4)

```
: 182      0275 2      END;
: 183      0276 2
: 184      0277 2 $print_desc (null_string);
: 185      0278 2
: 186      0279 2 RETURN true;
: 187      0280 1 END;
```

```
.TITLE EXCH$SHOW SHOW verb dispatch routine
.IDENT \V04-000\

.PSECT EXCH$SHOW_PLIT,NOWRT,2

00000 P.AAB: .BLKB 0
010E0000 00000 P.AAA: .LONG 17694720
00000000' 00004 .ADDRESS P.AAB
73 65 6D 75 6C 6F 76 20 64 65 74 6E 75 6F 4D 00008 P.AAD: .ASCII \Mounted volumes:\
3A 00017
010E0010 00018 P.AAC: .LONG 17694736
00000000' 0001C .ADDRESS P.AAD
46 41 21 20 00020 P.AAF: .ASCII \!AF\
010E0004 00024 P.AAE: .LONG 17694724
00000000' 00028 .ADDRESS P.AAF
6D 72 6F 66 20 65 6D 75 6C 6F 76 20 2A 38 21 0002C P.AAH: .ASCII \!8* volume format:!10* !AF\<0><0>
00 00 46 41 21 20 2A 30 31 21 3A 74 61 0003B
010E001A 00048 P.AAG: .LONG 17694746
00000000' 0004C .ADDRESS P.AAH
6B 73 69 64 00050 P.AAJ: .ASCII \disk\
010E0004 00054 P.AAI: .LONG 17694724
00000000' 00058 .ADDRESS P.AAJ
00 65 70 61 74 67 61 6D 0005C P.AAL: .ASCII \magtape\<0>
010E0007 00064 P.AAK: .LONG 17694727
00000000' 00068 .ADDRESS P.AAL
00 6E 77 6F 6E 6B 6E 75 0006C P.AAN: .ASCII \unknown\<0>
010E0007 00074 P.AAM: .LONG 17694727
00000000' 00078 .ADDRESS P.AAN
75 6C 6F 76 20 6C 61 75 74 72 69 76 28 20 20 0007C P.AAP: .ASCII \ (virtual volume)\<0><0>
00 00 29 65 6D 0008B
010E0012 00090 P.AAO: .LONG 17694738
00000000' 00094 .ADDRESS P.AAP
73 61 6C 63 20 65 6D 75 6C 6F 76 20 2A 38 21 00098 P.AAR: .ASCII \!8* volume class:!11* !AS!AS\
53 41 21 53 41 21 20 2A 31 31 21 3A 73 000A7
010E001C 000B4 P.AAQ: .LONG 17694748
00000000' 000B8 .ADDRESS P.AAR
6C 69 66 20 6C 61 75 74 72 69 76 20 2A 38 21 000BC P.AAT: .ASCII \!8* virtual file name:!6* !AF\<0><0>
00 46 41 21 20 2A 36 21 3A 65 6D 61 6E 20 65 000CB
00 000DA
00 000DB
010E001D 000DC P.AAS: .ASCII <0>
00000000' 000E0 .LONG 17694749
65 64 20 6C 61 63 69 73 79 68 70 20 2A 38 21 000E4 P.AAT: .ADDRESS P.AAT
41 21 20 20 20 3A 65 6D 61 6E 20 65 63 69 76 000F3 P.AAV: .ASCII \!8* physical device name: !AF\<0>
00 46 00102
010E001F 00104 P.AAU: .LONG 17694751
00000000' 00108 .ADDRESS P.AAV
65 7A 69 73 20 65 6D 75 6C 6F 76 20 2A 38 21 0010C P.AAX: .ASCII \!8* volume size:!12* !UL block!XS\<0><0>
6B 63 6F 6C 62 20 4C 55 21 20 2A 32 31 21 3A 0011B
```

```
00 00 53 25 21 0012A
                                00 0012F
                                010E0021 00130 P.AAW: .ASCII <0>
                                00000000' 00134 P.AAZ: .LONG 17694753
20 73 69 20 65 6D 75 6C 6F 76 20 2A 32 33 21 00138 P.AAZ: .ADDRESS P.AAX
00 64 65 6B 63 6F 6C 2D 65 74 69 72 77 00147 P.AAY: .ASCII \!32* volume is write-locked\<0>
                                010E001B 00154 P.AAY: .LONG 17694747
                                00000000' 00158 P.AAZ: .ADDRESS P.AAZ
43 45 48 43 5F 41 54 41 44 2F 20 2A 32 33 21 0015C P.ABB: .ASCII \!32* /DATA_CHECK=READ is enabled\
6C 62 61 6E 65 20 73 69 20 44 41 45 52 3D 4B 0016B P.ABB: .ADDRESS P.AAB
                                64 65 0017A P.ABA: .LONG 17694752
                                010E0020 0017C P.ABA: .ADDRESS P.ABB
                                00000000' 00180 P.ABD: .ASCII \!32* /DATA_CHECK=WRITE is enabled\<0><0>
43 45 48 43 5F 41 54 41 44 2F 20 2A 32 33 21 00184 P.ABD: .LONG 17694753
62 61 6E 65 20 73 69 20 45 54 49 52 57 3D 4B 00193 P.ABD: .ADDRESS P.ABD
00 00 64 65 6C 001A2 P.ABC: .ASCII <0>
                                00 001A7 P.ABC: .LONG 17694753
                                010E0021 001A8 P.ABC: .ADDRESS P.ABD
                                00000000' 001AC
```

```
NULL_STRING= P.AAA
.EXTRN EXCH$A_GBL, EXCH$ NOVOLUMES
.EXTRN LIB$PUT_OUTPUT, EXCH$UTIL_BLOCK_CHECK
.EXTRN EXCH$UTIL_FAO_BUFFER
```

```
.PSECT EXCH$SHOW_CODE, NOWRT, 2
```

```
00FC 00000
57 00000000G EF 9E 00002 .ENTRY EXCH$SHOW SHOW, Save R2,R3,R4,R5,R6,R7
56 00000000G EF 9E 00009 MOVAB EXCH$A_GBL, R7
55 0000' CF 9E 00010 MOVAB EXCH$UTIL_FAO_BUFFER, R6
54 00000000G 00 9E 00015 MOVAB NULL_STRING, R5
50 67 D0 0001C MOVAB LIB$PUT_OUTPUT, R4
53 00C0 C0 D0 0001F MOVL EXCH$A_GBL, R0
50 00C0 C0 9E 00024 MOVL 192(R0), VOLB
50 53 D1 00029 MOVAB 192(R0), R0
14 12 0002C CMPL VOLB, R0
52 00000000G 8F D0 0002E BNEQ 1$
52 DD 00035 MOVL #EXCH$_NOVOLUMES, TEMP
00 01 FB 00037 PUSHL TEMP
50 52 D0 0003E CALLS #1, LIB$SIGNAL
04 00041 MOVL TEMP, R0
RET
55 DD 00042 1$: PUSHL R5
64 01 FB 00044 CALLS #1, LIB$PUT_OUTPUT
18 A5 9F 00047 PUSHAB P.AAC
64 01 FB 0004A CALLS #1, LIB$PUT_OUTPUT
50 000000C0 8F C1 0004D 2$: ADDL3 #192, EXCH$A_GBL, R0
50 53 D1 00055 CMPL VOLB, R0
03 12 00058 BNEQ 3$
00EE 31 0005A BRW 15$
52 041B00F3 8F D0 0005D 3$: MOVL #68878579, R2
51 01DD 8F 3C 00064 MOVZWL #477, R1
50 53 D0 00069 MOVL VOLB, R0
00000000G EF 16 0006C JSB EXCH$UTIL_BLOCK_CHECK
55 DD 00072 PUSHL R5
64 01 FB 00074 CALLS #1, LIB$PUT_OUTPUT
69 A3 9F 00077 PUSHAB 105(VOLB)
```


		65	A3	DD	0007A	PUSHL	101(VOLB)	:	
		24	A5	9F	0007D	PUSHAB	P.AAE	:	
	66		03	FB	00080	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	00083	PUSHL	R0	:	
	64		01	FB	00085	CALLS	#1, LIB\$PUT_OUTPUT	:	
		5D	A3	9F	00088	PUSHAB	93(VOLB)	:	0233
		59	A3	DD	0008B	PUSHL	89(VOLB)	:	
		48	A5	9F	0008E	PUSHAB	P.AAG	:	
	66		03	FB	00091	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	00094	PUSHL	R0	:	
	64		01	FB	00096	CALLS	#1, LIB\$PUT_OUTPUT	:	
	01	30	A3	D1	00099	CMPL	48(VOLB), #1	:	0235
			06	12	0009D	BNEQ	4\$	:	
	51	54	A5	9E	0009F	MOVAB	P.AAI, PTR1	:	0237
			10	11	000A3	BRB	6\$	:	
	02	30	A3	D1	000A5	CMPL	48(VOLB), #2	:	0238
			06	12	000A9	BNEQ	5\$	:	
	51	64	A5	9E	000AB	MOVAB	P.AAK, PTR1	:	0240
			04	11	000AF	BRB	6\$	:	
07		51	A5	9E	000B1	MOVAB	P.AAM, PTR1	:	0242
	48	A3	04	E1	000B5	BBC	#4, 72(VOLB), 7\$	:	0243
		50	C5	9E	000BA	MOVAB	P.AAO, PTR2	:	0245
			03	11	000BF	BRB	8\$	:	
		50	65	9E	000C1	MOVAB	NULL_STRING, PTR2	:	0247
			50	DD	000C4	PUSHL	PTR2	:	0248
			51	DD	000C6	PUSHL	PTR1	:	
			00B4	C5	9F	000C8	PUSHAB	P.AAQ	
	66		03	FB	000CC	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	000CF	PUSHL	R0	:	
	64		01	FB	000D1	CALLS	#1, LIB\$PUT_OUTPUT	:	
11		A3	04	E1	000D4	BBC	#4, 72(VOLB), 9\$	:	0250
	48	50	A3	D0	000D9	MOVL	24(VOLB), R0	:	0252
			04	A0	DD	000DD	PUSHL	4(R0)	
		7E	03	A0	9A	000E0	MOVZBL	3(R0), -(SP)	
			00DC	C5	9F	000E4	PUSHAB	P.AAS	
			0B	11	000E8	BRB	10\$	:	
		00E9	C3	9F	000EA	PUSHAB	233(VOLB)	:	0254
			A3	DD	000EE	PUSHL	56(VOLB)	:	
		38	C5	9F	000F1	PUSHAB	P.AAU	:	
		0104	03	FB	000F5	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:	
	66		50	DD	000F8	PUSHL	R0	:	
			01	FB	000FA	CALLS	#1, LIB\$PUT_OUTPUT	:	
	64		A3	91	000FD	CMPB	88(VOLB), #3	:	0256
	03		0F	12	00101	BNEQ	11\$	:	
		44	A3	DD	00103	PUSHL	68(VOLB)	:	0259
			C5	9F	00106	PUSHAB	P.AAW	:	
	66		02	FB	0010A	CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	0010D	PUSHL	R0	:	
	64		01	FB	0010F	CALLS	#1, LIB\$PUT_OUTPUT	:	
0C		A3	05	E0	00112	BBS	#5, 72(VOLB), 12\$	:	0262
			C5	9F	00117	PUSHAB	P.AAY	:	0264
	66		01	FB	0011B	CALLS	#1, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	0011E	PUSHL	R0	:	
	64		01	FB	00120	CALLS	#1, LIB\$PUT_OUTPUT	:	
0C		A3	01	E1	00123	BBC	#1, 72(VOLB), 13\$	:	0266
			C5	9F	00128	PUSHAB	P.ABA	:	0268
		017C	01	FB	0012C	CALLS	#1, EXCH\$UTIL_FAO_BUFFER	:	

EXCH\$SHOW
V04-000

SHOW verb dispatch routine
exch\$show_show

J 11
16-Sep-1984 01:24:42
14-Sep-1984 12:29:08

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCH\$SHOW.B32;1

Page 8
(4)

		64	50	DD	0012F	PUSHL	R0	:	
		A3	01	FB	00131	CALLS	#1, LIB\$PUT_OUTPUT	:	
OC	48		02	E1	00134	BBC	#2, 72(VOLB), 14\$	:	0270
			C5	9F	00139	PUSHAB	P.ABC	:	0272
		66	01	FB	0013D	CALLS	#1, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	00140	PUSHL	R0	:	
		64	01	FB	00142	CALLS	#1, LIB\$PUT_OUTPUT	:	
		53	63	D0	00145	MOVL	(VOLB), VOLB	:	0274
			FF02	31	00148	BRW	2\$	:	0220
			55	DD	0014B	PUSHL	R5	:	0277
		64	01	FB	0014D	CALLS	#1, LIB\$PUT_OUTPUT	:	
		50	01	D0	00150	MOVL	#1, R0	:	0279
			04	00153	RET			:	0280

; Routine Size: 340 bytes, Routine Base: EXCH\$SHOW_CODE + 0000

EXCH\$SHOW
V04-000

SHOW verb dispatch routine
exch\$show_show

K 11
16-Sep-1984 01:24:42
14-Sep-1984 12:29:08

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXC\$SHOW.B32;1

Page 9
(5)

: 189 0281 1 END
: 190 0282 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
EXCH\$SHOW_PLIT	432 NOVEC,NOWRT, RD	EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
EXCH\$SHOW_CODE	340 NOVEC,NOWRT, RD	EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	5	0	1000	00:01.8
_\$255\$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1	1151	31	2	79	00:01.4

COMMAND QUALIFIERS

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

: Size: 340 code + 432 data bytes
: Run Time: 00:10.1
: Elapsed Time: 00:39.3
: Lines/CPU Min: 1683
: Lexemes/CPU-Min: 12919
: Memory Used: 131 pages
: Compilation Complete

0163 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

