

[illegible]

```

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             SSSSSS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
1 0001 0 MODULE setshobro ( IDENT = 'V04-000',
2 0002 0 ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL=LONG_RELATIVE)
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: SET and SHOW Commands
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module implements the DCL commands SET BROADCAST
36 0036 1 and SHOW BROADCAST
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 VAX/VMS operating system, user mode
41 0041 1
42 0042 1 AUTHOR: Gerry Smith 25-Aug-1983
43 0043 1
44 0044 1 Modified by:
45 0045 1
46 0046 1 V03-003 JLV0391 Jake VanNoy 25-JUL-1984
47 0047 1 Enhance SHOW logic to also look at NOBRDCST bit.
48 0048 1
49 0049 1 V03-002 DAS0001 David Solomon 10-Jul-1984
50 0050 1 Fix truncation errors; make nonexternal refs LONG_RELATIVE.
51 0051 1
52 0052 1 V03-001 JLV0345 Jake VanNoy 8-APR-1984
53 0053 1 Fix USER1-USER16 so that appropriate bits are set.
54 0054 1
55 0055 1 --
```

```

: 57      0056 1 |
: 58      0057 1 | Include files
: 59      0058 1 |
: 60      0059 1 | LIBRARY 'SYSSLIBRARY:LIB';
: 61      0060 1 |
: 62      0061 1 |
: 63      0062 1 | Declare shared messages
: 64      0063 1 |
: 65      P 0064 1 | $SHR_MSGDEF (SET, 119, LOCAL,
: 66      P 0065 1 | (notterm, warning),
: 67      0066 1 | (readerr, warning));
: 68      0067 1 |
: 69      P 0068 1 | $SHR_MSGDEF (SHOW, 120, LOCAL,
: 70      P 0069 1 | (notterm, warning),
: 71      0070 1 | (readerr, warning));

```

! VAX/VMS common definitions


```

: 73      0071 1  !
: 74      0072 1  ! Declare some storage for tables
: 75      0073 1  !
: 76      0074 1  ! OWN
: 77      0075 1  !   class : VECTOR[8]
: 78      0076 1  !       INITIAL(%ASCID 'OPTION.BROADCAST.GENERAL',
: 79      0077 1  !           %ASCID 'OPTION.BROADCAST.PHONE',
: 80      0078 1  !           %ASCID 'OPTION.BROADCAST.MAIL',
: 81      0079 1  !           %ASCID 'OPTION.BROADCAST.QUEUE',
: 82      0080 1  !           %ASCID 'OPTION.BROADCAST.SHUTDOWN',
: 83      0081 1  !           %ASCID 'OPTION.BROADCAST.URGENT',
: 84      0082 1  !           %ASCID 'OPTION.BROADCAST.DCL',
: 85      0083 1  !           %ASCID 'OPTION.BROADCAST.OPCOM'),
: 86      0084 1  !   user : VECTOR[16]
: 87      0085 1  !       INITIAL(%ASCID 'OPTION.BROADCAST.USER1',
: 88      0086 1  !           %ASCID 'OPTION.BROADCAST.USER2',
: 89      0087 1  !           %ASCID 'OPTION.BROADCAST.USER3',
: 90      0088 1  !           %ASCID 'OPTION.BROADCAST.USER4',
: 91      0089 1  !           %ASCID 'OPTION.BROADCAST.USER5',
: 92      0090 1  !           %ASCID 'OPTION.BROADCAST.USER6',
: 93      0091 1  !           %ASCID 'OPTION.BROADCAST.USER7',
: 94      0092 1  !           %ASCID 'OPTION.BROADCAST.USER8',
: 95      0093 1  !           %ASCID 'OPTION.BROADCAST.USER9',
: 96      0094 1  !           %ASCID 'OPTION.BROADCAST.USER10',
: 97      0095 1  !           %ASCID 'OPTION.BROADCAST.USER11',
: 98      0096 1  !           %ASCID 'OPTION.BROADCAST.USER12',
: 99      0097 1  !           %ASCID 'OPTION.BROADCAST.USER13',
: 100     0098 1  !           %ASCID 'OPTION.BROADCAST.USER14',
: 101     0099 1  !           %ASCID 'OPTION.BROADCAST.USER15',
: 102     0100 1  !           %ASCID 'OPTION.BROADCAST.USER16');
: 103     0101 1  !

```

```

: 105      0102 1  :
: 106      0103 1  : Table of contents
: 107      0104 1  :
: 108      0105 1  :
: 109      0106 1  FORWARD ROUTINE
: 110      0107 1      set$broadcast : NOVALUE,
: 111      0108 1      show$broadcast : NOVALUE,
: 112      0109 1      getset;
: 113      0110 1
: 114      0111 1  :
: 115      0112 1  : External routines
: 116      0113 1  :
: 117      0114 1  EXTERNAL ROUTINE
: 118      0115 1      cli$present,
: 119      0116 1      show$write_line : NOVALUE;
: 120      0117 1
: 121      0118 1  :
: 122      0119 1  : Declare literals defined elsewhere
: 123      0120 1  :
: 124      0121 1  EXTERNAL LITERAL
: 125      0122 1      cli$_absent,
: 126      0123 1      set$_writeerr;
: 127      0124 1
: 128      0125 1  :
: 129      0126 1  : Declare the shared messages
: 130      0127 1
: 131      P 0128 1  $$HR_MSGDEF (SET, 119, LOCAL,
: 132      0129 1      (syntax, error));
: 133      0130 1
```

```

: Set broadcast classes
: Display broadcast classes
: Get/set broadcast classes
```

```

! Write a line to SYSS$OUTPUT
```

```

! Qualifier not present
! Error modifying broadcast classes
```



```
135 0131 1 GLOBAL ROUTINE set$broadcast : NOVALUE =
136 0132 BEGIN
137 0133
138 0134 ++
139 0135 Functional description
140 0136
141 0137 This is the routine for the SET BROADCAST command. It is called
142 0138 from the SET command processor, and enables/disables broadcast
143 0139 messages for the process's terminal.
144 0140
145 0141 Inputs
146 0142 None
147 0143
148 0144 Outputs
149 0145 None
150 0146
151 0147 ----
152 0148
153 0149 LOCAL
154 0150 status,
155 0151 char : VECTOR [3], ! terminal characteristics
156 0152 flags : BITVECTOR[64]; ! Quadword for broadcast classes
157 0153
158 0154
159 0155 First, get the current settings.
160 0156
161 0157 IF NOT (status = getset(0, ! 0 ==> get broadcast classes
162 0158 flags,char)) ! Put broadcast flags here
163 0159 THEN
164 0160 BEGIN
165 0161 IF .status EQL 0 ! Status of 0 is not-a-terminal
166 0162 THEN status = set$_notterm; ! so translate it here
167 0163 SIGNAL(set$_readerr, ! Error reading dev chars,
168 0164 1,
169 0165 %ASCII 'broadcast classes',
170 0166 .status); ! and this is why
171 0167 RETURN;
172 0168 END;
173 0169
174 0170
175 0171 See if ALL or NONE was specified. If so, do the appropriate thing.
176 0172
177 0173 IF cli$present(%ASCII 'OPTION.BROADCAST.NONE')
178 0174 THEN
179 0175 BEGIN
180 0176 MAP
181 0177 flags : VECTOR[2];
182 0178 flags[0] = flags[1] = -1;
183 0179 END;
184 0180
185 0181 IF cli$present(%ASCII 'OPTION.BROADCAST.ALL')
186 0182 THEN
187 0183 BEGIN
188 0184 MAP
189 0185 flags : VECTOR[2];
190 0186 flags[0] = flags[1] = 0;
191 0187 END;
```

```
192 0188 2
193 0189 2
194 0190 2
195 0191 2
196 0192 2
197 0193 2
198 0194 2
199 0195 2
200 0196 2
201 0197 4
202 0198 4
203 0199 4
204 0200 4
205 0201 4
206 0202 4
207 0203 4
208 0204 4
209 0205 4
210 0206 4
211 0207 4
212 0208 4
213 0209 4
214 0210 4
215 0211 4
216 0212 4
217 0213 4
218 0214 4
219 0215 4
220 0216 4
221 0217 4
222 0218 4
223 0219 4
224 0220 4
225 0221 4
226 0222 4
227 0223 4
228 0224 4
229 0225 4
230 0226 4
231 0227 4
232 0228 4
233 0229 4
234 0230 1

Get the system broadcast classes.
INCR i FROM 0 TO 7 DO
    BEGIN
        status = cli$present(.class[i]);
        IF .status NEQ cli$_absent
        THEN
            BEGIN
                IF .status
                THEN flags[i] = 0;
                ELSE flags[i] = 1;
            END;
        END;
    END;

Now for the user stuff.
INCR i FROM 0 TO 15 DO
    BEGIN
        status = cli$present(.user[i]);
        IF .status NEQ cli$_absent
        THEN
            BEGIN
                IF .status
                THEN flags[32 + i] = 0;
                ELSE flags[32 + i] = 1;
            END;
        END;
    END;

Put in the new broadcast classes.
IF NOT (status = getset(1, flags))
THEN SIGNAL(set$_writeerr,
    1,
    %ASCII 'broadcast classes',
    .status);
RETURN;
END;
```

.TITLE SETSHOBRO
.IDENT \V04-000\

.PSECT \$PLITS,NOWRT,NOEXE,2

```
53 41 43 44 41 4F 52 42 2E 4E 4F 49 54 50 4F 00000 P.AAB: .ASCII \OPTION.BROADCAST.GENERAL\
4C 41 52 45 4E 45 47 2E 54 0000F
010E0018 00018 P.AAA: .LONG 17694744
00000000 0001C .ADDRESS P.AAB
53 41 43 44 41 4F 52 42 2E 4E 4F 49 54 50 4F 00020 P.AAD: .ASCII \OPTION.BROADCAST.PHONE\<0><0>
00 00 45 4E 4F 48 50 2E 54 0002F
010E0016 00038 P.AAC: .LONG 17694742
```


53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00000000'	0003C	.ADDRESS P.AAD
						00	00	00	4C	49	41	54	50	4F	00040	P.AAF:	.ASCII \OPTION.BROADCAST.MAIL\<0><0><0>
												4D	2E	54	0004F		
															010E0015	00058	P.AAE: .LONG 17694741
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00000000'	0005C	.ADDRESS P.AAF
						00	00	45	55	45	55	51	2E	54	00060	P.AAH:	.ASCII \OPTION.BROADCAST.QUEUE\<0><0>
															0006F		
															010E0016	00078	P.AAG: .LONG 17694742
															0007C		
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00080	P.AAJ:	.ADDRESS P.AAH
		00	00	00	4E	57	4F	44	54	55	48	53	2E	54	0008F		.ASCII \OPTION.BROADCAST.SHUTDOWN\<0><0><0>
															0009C	P.AAI:	.LONG 17694745
															000A0		.ADDRESS P.AAJ
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	000A4	P.AAL:	.ASCII \OPTION.BROADCAST.URGENT\<0>
						00	54	4E	45	47	52	55	2E	54	000B3		
															010E0017	000BC	P.AAK: .LONG 17694743
															000C0		.ADDRESS P.AAL
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	000C4	P.AAN:	.ASCII \OPTION.BROADCAST.DCL\
												44	2E	54	000D3		
															010E0014	000D8	P.AAM: .LONG 17694740
															000DC		.ADDRESS P.AAN
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	000E0	P.AAP:	.ASCII \OPTION.BROADCAST.OPCOM\<0><0>
						00	00	4D	4F	43	50	4F	2E	54	000EF		
															010E0016	000F8	P.AAO: .LONG 17694742
															000FC		.ADDRESS P.AAP
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00100	P.AAR:	.ASCII \OPTION.BROADCAST.USER1\<0><0>
						00	00	31	52	45	53	55	2E	54	0010F		
															010E0016	00118	P.AAQ: .LONG 17694742
															0011C		.ADDRESS P.AAR
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00120	P.AAT:	.ASCII \OPTION.BROADCAST.USER2\<0><0>
						00	00	32	52	45	53	55	2E	54	0012F		
															010E0016	00138	P.AAS: .LONG 17694742
															0013C		.ADDRESS P.AAT
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00140	P.AAV:	.ASCII \OPTION.BROADCAST.USER3\<0><0>
						00	00	33	52	45	53	55	2E	54	0014F		
															010E0016	00158	P.AAU: .LONG 17694742
															0015C		.ADDRESS P.AAV
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00160	P.AAX:	.ASCII \OPTION.BROADCAST.USER4\<0><0>
						00	00	34	52	45	53	55	2E	54	0016F		
															010E0016	00178	P.AAW: .LONG 17694742
															0017C		.ADDRESS P.AAX
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00180	P.AAZ:	.ASCII \OPTION.BROADCAST.USER5\<0><0>
						00	00	35	52	45	53	55	2E	54	0018F		
															010E0016	00198	P.AAY: .LONG 17694742
															0019C		.ADDRESS P.AAZ
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	001A0	P.ABB:	.ASCII \OPTION.BROADCAST.USER6\<0><0>
						00	00	36	52	45	53	55	2E	54	001AF		
															010E0016	001B8	P.ABA: .LONG 17694742
															001BC		.ADDRESS P.ABB
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	001C0	P.ABD:	.ASCII \OPTION.BROADCAST.USER7\<0><0>
						00	00	37	52	45	53	55	2E	54	001CF		
															010E0016	001D8	P.ABC: .LONG 17694742
															001DC		.ADDRESS P.ABD
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	001E0	P.ABF:	.ASCII \OPTION.BROADCAST.USER8\<0><0>
						00	00	38	52	45	53	55	2E	54	001EF		
															010E0016	001F8	P.ABE: .LONG 17694742
															00000000'	001FC	.ADDRESS P.ABF

53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00200	P.ABH:	.ASCII	\OPTION.BROADCAST.USER9\<0><0>
						00	00	39	52	45	53	55	2E	54	0020F			
													010E0016		00218	P.ABG:	.LONG	17694742
													00000000		0021C		.ADDRESS	P.ABH
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00220	P.ABJ:	.ASCII	\OPTION.BROADCAST.USER10\<0>
						00	30	31	52	45	53	55	2E	54	0022F			
													010E0017		00238	P.ABI:	.LONG	17694743
													00000000		0023C		.ADDRESS	P.ABJ
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00240	P.ABL:	.ASCII	\OPTION.BROADCAST.USER11\<0>
						00	31	31	52	45	53	55	2E	54	0024F			
													010E0017		00258	P.ABK:	.LONG	17694743
													00000000		0025C		.ADDRESS	P.ABL
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00260	P.ABN:	.ASCII	\OPTION.BROADCAST.USER12\<0>
						00	32	31	52	45	53	55	2E	54	0026F			
													010E0017		00278	P.ABM:	.LONG	17694743
													00000000		0027C		.ADDRESS	P.ABN
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	00280	P.ABP:	.ASCII	\OPTION.BROADCAST.USER13\<0>
						00	33	31	52	45	53	55	2E	54	0028F			
													010E0017		00298	P.ABO:	.LONG	17694743
													00000000		0029C		.ADDRESS	P.ABP
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	002A0	P.ABR:	.ASCII	\OPTION.BROADCAST.USER14\<0>
						00	34	31	52	45	53	55	2E	54	002AF			
													010E0017		002B8	P.ABQ:	.LONG	17694743
													00000000		002BC		.ADDRESS	P.ABR
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	002C0	P.ABT:	.ASCII	\OPTION.BROADCAST.USER15\<0>
						00	35	31	52	45	53	55	2E	54	002CF			
													010E0017		002D8	P.ABS:	.LONG	17694743
													00000000		002DC		.ADDRESS	P.ABT
53	41	43	44	41	4F	52	42	2E	4E	4F	49	54	50	4F	002E0	P.ABV:	.ASCII	\OPTION.BROADCAST.USER16\<0>
						00	36	31	52	45	53	55	2E	54	002EF			
													010E0017		002F8	P.ABU:	.LONG	17694743
													00000000		002FC		.ADDRESS	P.ABV
73	73	61	6C	63	20	74	73	61	63	64	61	6F	72	62	00300	P.ABX:	.ASCII	\broadcast classes\<0><0><0>
										00	00	00	73	65	0030F			

				.EXTRN	CLISPRESNT, SHOW\$WRITE LINE	
				.EXTRN	CLIS_ABSENT, SETS_WRITEERR	
				.PSECT	\$CODE\$,NOWRT,2	
				.ENTRY	SETSBROADCAST, Save R2,R3,R4,R5,R6,R7	0131
57	00000000V	EF	9E	MOVAB	GETSET, R7	
56	00000000G	8F	D0	MOVL	#CLIS_ABSENT, R6	
55	00000000G	00	9E	MOVAB	CLISPRESNT, R5	
54	00000000'	EF	9E	MOVAB	P.ABW, R4	
5E		14	C2	SUBL2	#20, SP	
	08	AE	9F	PUSHAB	CHAR	0157
	04	AE	9F	PUSHAB	FLAGS	
		7E	D4	CLRL	-(SP)	
67		03	FB	CALLS	#3, GETSET	
53		50	D0	MOVL	R0, STATUS	
18		53	EB	BLBS	STATUS, 2\$	
		07	12	BNEQ	1\$	0161
53	007712D8	8F	D0	MOVL	#7803608, STATUS	0162
		53	DD	PUSHL	STATUS	0166
		54	DD	PUSHL	R4	0164
		01	DD	PUSHL	#1	0163
	007710B0	8F	DD	PUSHL	#7803056	
		0083	31	BRW	11\$	
	20	A4	9F	PUSHAB	P.ABY	0173
65		01	FB	CALLS	#1, CLISPRESNT	
07		50	E9	BLBC	R0, 3\$	
AE		01	CE	MNEGL	#1, FLAGS+4	0178
6E		01	CE	MNEGL	#1, FLAGS	
	3C	A4	9F	PUSHAB	P.ACA	0181
65		01	FB	CALLS	#1, CLISPRESNT	
02		50	E9	BLBC	R0, 4\$	
		6E	7C	CLRQ	FLAGS	0186
		52	D4	CLRL	I	0192
	00000000'EF	42	DD	PUSHL	CLASSE[I]	0194
65		01	FB	CALLS	#1, CLISPRESNT	
53		50	D0	MOVL	R0, STATUS	
56		53	D1	CMPL	STATUS, R6	0195
		0D	13	BEQL	7\$	
06		53	E9	BLBC	STATUS, 6\$	0198
6E		52	E5	BBCC	I, FLAGS, 7\$	0199
		04	11	BRB	7\$	
00		52	E2	BBSS	I, FLAGS, 7\$	0200
DD		07	F3	AOBLEQ	#7, I, 5\$	0192
		52	D4	CLRL	I	0207
	00000000'EF	42	DD	PUSHL	USER[I]	0209
65		01	FB	CALLS	#1, CLISPRESNT	
53		50	D0	MOVL	R0, STATUS	
56		53	D1	CMPL	STATUS, R6	0210
		11	13	BEQL	10\$	
50		A2	9E	MOVAB	32(R2), R0	0214
06		53	E9	BLBC	STATUS, 9\$	
6E		50	E5	BBCC	R0, FLAGS, 10\$	
		04	11	BRB	10\$	
00		50	E2	BBSS	R0, FLAGS, 10\$	0215
D9		0F	F3	AOBLEQ	#15, I, 8\$	0207
		5E	DD	PUSHL	SP	0222

D 9
16-Sep-1984 01:15:24
14-Sep-1984 12:09:19

Page 10
(5)

			01	DD	000B5		PUSHL	#1
	67		02	FB	000B7		CALLS	#2, GETSET
	53		50	D0	000BA		MOVL	R0, STATUS
	14		53	E8	000BD		BLBS	STATUS, 12\$
			53	DD	000C0		PUSHL	STATUS
		58	A4	9F	000C2		PUSHAB	P.ACC
			01	DD	000C5		PUSHL	#1
		00000000G	8F	DD	000C7		PUSHL	#SET\$ WRITEERR
00000000G	00		04	FB	000CD	11\$:	CALLS	#4, LIB\$SIGNAL
			04	000D4	12\$:		RET	

0227
0225
0224

0230

; Routine Size: 213 bytes, Routine Base: \$CODES + 0000


```
236 0231 1 GLOBAL ROUTINE show$broadcast : NOVALUE =
237 0232 2 BEGIN
238 0233 3
239 0234 4 ++
240 0235 5 Functional description
241 0236 6
242 0237 7 This is the routine for the SHOW BROADCAST command. It is called
243 0238 8 from the SHOW command processor, and displays classes of broadcast
244 0239 9 for the process's terminal.
245 0240 10
246 0241 11 Inputs
247 0242 12 None
248 0243 13
249 0244 14 Outputs
250 0245 15 None
251 0246 16
252 0247 17 ----
253 0248 18
254 0249 19 LOCAL
255 0250 20 status,
256 0251 21 char : VECTOR [3], ! terminal characteristics
257 0252 22 flags : BITVECTOR[64]; ! Broadcast flags quadword
258 0253 23
259 0254 24
260 0255 25 Get the current settings.
261 0256 26
262 0257 27 IF NOT (status = getset(0, ! 0 ==> get broadcast classes
263 0258 28 flags,char)) ! Put them here
264 0259 29 THEN ! If an error,
265 0260 30 BEGIN ! tell the user
266 0261 31 IF .status EQL 0 ! If 0, then really not-a-terminal
267 0262 32 THEN status = show$_notterm; ! so tell user that.
268 0263 33 SIGNAL(show$_readerr,
269 0264 34 1,
270 0265 35 %ASCII 'broadcast classes',
271 0266 36 .status);
272 0267 37 RETURN;
273 0268 38 END;
274 0269 39
275 0270 40
276 0271 41 If the entire quadword is zero, then all broadcasts are enabled.
277 0272 42
278 0273 43 BEGIN
279 0274 44 MAP flags : VECTOR[2];
280 0275 45 BIND devdepend = CHAR [1]: $BBLOCK;
281 0276 46
282 0277 47 IF (.flags[0] EQL 0 AND .flags[1] EQL 0)
283 0278 48 AND NOT (.devdepend [IT$V_NOBRDCST])
284 0279 49 THEN
285 0280 50 BEGIN
286 0281 51 show$write_line(%ASCII 'Broadcasts are enabled for all classes', %REF(0));
287 0282 52 RETURN;
288 0283 53 END;
289 0284 54 END;
290 0285 55
291 0286 56 BEGIN
292 0287 57 MAP flags : VECTOR[2];
```

```
0288 3 BIND devdepend = CHAR [1]: $BBLOCK;
0289 3
0290 4 IF (.flags[0] EQL %X'ffffffff' AND .flags[1] EQL %X'ffffffff')
0291 4 OR (.devdepend [TT$V_NOBRDCST])
0292 4 THEN
0293 4 BEGIN
0294 4 show$write_line(%ASCII 'Broadcasts are disabled for all classes', %REF(0));
0295 4 RETURN;
0296 4 END;
0297 3 END;
0298 3
0299 3 If we have a more usual case, start processing here, first with a header.
0300 3
0301 3 show$write_line(%ASCII 'Broadcasts are currently disabled for:',
0302 3 %REF(0));
0303 3
0304 3
0305 3 Display disabled system broadcast classes.
0306 3
0307 3 INCR i FROM 0 TO 7 DO
0308 3 BEGIN
0309 3 IF .flags[i]
0310 3 THEN
0311 3 BEGIN
0312 3 LOCAL
0313 3 desc : VECTOR[2];
0314 3 BIND string = .class[i] : $BBLOCK[dsc$c_s_bln];
0315 3 desc[0] = .string[dsc$w_length] - 17;
0316 3 desc[1] = .string[dsc$a_pointer] + 17;
0317 3 show$write_line(%ASCII 'AS', %REF(desc));
0318 3 END;
0319 3 END;
0320 3
0321 3
0322 3 Display enabled user broadcast classes.
0323 3
0324 3 INCR i FROM 0 TO 15 DO
0325 3 BEGIN
0326 3 IF .flags[32 + .i]
0327 3 THEN
0328 3 BEGIN
0329 3 LOCAL
0330 3 desc : VECTOR[2];
0331 3 BIND string = .user[i] : $BBLOCK[dsc$c_s_bln];
0332 3 desc[0] = .string[dsc$w_length] - 17;
0333 3 desc[1] = .string[dsc$a_pointer] + 17;
0334 3 show$write_line(%ASCII 'AS', %REF(desc));
0335 3 END;
0336 3 END;
0337 3
0338 3 RETURN;
0339 1 END;
```

.PSECT SPLITS,NOWRT,NOEXE,2

SETSHOBRO
V04-000

6 9
16-Sep-1984 01:15:24
14-Sep-1984 12:09:19

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETSHOBRO.B32;1

Page 13
(6)

```
73 73 61 6C 63 20 74 73 61 63 64 61 6F 72 62 00374 P.ACF: .ASCII \broadcast classes\<0><0><0>
      00 00 00 73 65 00383
      010E0011 00388 P.ACE: .LONG 17694737
      00000000' 0038C .ADDRESS P.ACF
20 65 72 61 20 73 74 73 61 63 64 61 6F 72 42 00390 P.ACH: .ASCII \Broadcasts are enabled for all classes-
6C 6C 61 20 72 6F 66 20 64 65 6C 62 61 6E 65 0039F \<0>
      00 73 65 73 73 61 6C 63 20 003AE
      00 00 003B7
      010E0026 003B8 P.ACG: .ASCII <0>
      00000000' 003BC .LONG 17694758
      003C0 P.ACJ: .ADDRESS P.ACH
      003CF .ASCII \Broadcasts are disabled for all classes-
      003DE \<0>
      010E0027 003E8 P.ACI: .LONG 17694759
      00000000' 003EC .ADDRESS P.ACJ
20 65 72 61 20 73 74 73 61 63 64 61 6F 72 42 003F0 P.ACL: .ASCII \Broadcasts are currently disabled for:-
62 61 73 69 64 20 79 6C 74 6E 65 72 72 75 63 003FF \<0>
      00 3A 72 6F 66 20 64 65 6C 0040E
      00 00 00417
      010E0026 00418 P.ACK: .ASCII <0>
      00000000' 0041C .LONG 17694758
      00420 P.ACN: .ADDRESS P.ACL
      010E0007 00428 P.ACM: .ASCII \ !AS\<0>
      00000000' 0042C .LONG 17694727
      00430 P.ACP: .ADDRESS P.ACN
      010E0007 00438 P.ACO: .ASCII \ !AS\<0>
      00000000' 0043C .LONG 17694727
      .ADDRESS P.ACP
```

```
.PSECT $CODE$,NOWRT,2
.ENTRY SHOW$BROADCAST, Save R2,R3,R4
MOVAB SHOW$WRITE_LINE, R4
MOVAB P_ACE, R3
SUBL2 #32, SP
PUSHAB CHAR
PUSHAB FLAGS
CLRL -(SP)
CALLS #3, GETSET
BLBS STATUS, 2$
TSTL STATUS
BNEQ 1$
MOVL #7869144, STATUS
PUSHL STATUS
PUSHL R3
PUSHL #1
PUSHL #7868592
CALLS #4, LIB$SIGNAL
RET
TSTL FLAGS
BNEQ 3$
TSTL FLAGS+4
BNEQ 3$
BBS #1, DEVDEPEND+2, 3$
CLRL (SP)
PUSHL SP
```

0231
0257
0261
0262
0266
0264
0263
0260
0277
0278
0281

			30	A3	9F	00057	PUSHAB	P.ACG		
				20	11	0005A	BRB	6\$		
	FFFFFFF	8F	0C	AE	D1	0005C	3\$:	CMP	FLAGS, #-1	0290
				0A	12	00064	BNEQ	4\$		
	FFFFFFF	8F	10	AE	D1	00066	CMP	FLAGS+4, #-1		
				05	13	0006E	BEQ	5\$		
0B		1A	AE	01	E1	00070	4\$:	BBC	#1, DEVDEPEND+2, 7\$	0291
				6E	D4	00075	5\$:	CLRL	(SP)	0294
				5E	DD	00077	PUSHL	SP		
			60	A3	9F	00079	PUSHAB	P.ACI		
				02	FB	0007C	6\$:	CALLS	#2, SHOW\$WRITE_LINE	
					04	0007F	RET			0293
				6E	D4	00080	7\$:	CLRL	(SP)	0302
				5E	DD	00082	PUSHL	SP		
			0090	C3	9F	00084	PUSHAB	P.ACK		0301
				02	FB	00088	CALLS	#2, SHOW\$WRITE_LINE		
				52	D4	0008B	CLRL	I		0307
23		0C	AE	50	E1	0008D	8\$:	BBC	I, FLAGS, 9\$	0309
				42	D0	00092	MOVL	CLASS[I], R0		0314
				60	3C	0009A	MOVZWL	(R0), DESC		0315
				11	C2	0009E	SUBL2	#17, DESC		
0B	AE			11	C1	000A2	ADDL3	#17, 4(R0), DESC+4		0316
				AE	9E	000A8	MOVAB	DESC, (SP)		0317
				5E	DD	000AC	PUSHL	SP		
			00A0	C3	9F	000AE	PUSHAB	P.ACM		
				02	FB	000B2	CALLS	#2, SHOW\$WRITE_LINE		
D4				07	F3	000B5	9\$:	AOBLEQ	#7, I, 8\$	0307
				52	D4	000B9	CLRL	I		0324
			20	A2	9E	000BB	10\$:	MOVAB	32(R2), R0	0326
23		0C	AE	50	E1	000BF	BBC	R0, FLAGS, 11\$		
				42	D0	000C4	MOVL	USER[I], R0		0331
				60	3C	000CC	MOVZWL	(R0), DESC		0332
				11	C2	000D0	SUBL2	#17, DESC		
0B	AE			11	C1	000D4	ADDL3	#17, 4(R0), DESC+4		0333
				AE	9E	000DA	MOVAB	DESC, (SP)		0334
				5E	DD	000DE	PUSHL	SP		
			00B0	C3	9F	000E0	PUSHAB	P.ACO		
				02	FB	000E4	CALLS	#2, SHOW\$WRITE_LINE		
D0				0F	F3	000E7	11\$:	AOBLEQ	#15, I, 10\$	0324
				04	000EB	RET				0339

; Routine Size: 236 bytes, Routine Base: \$CODE\$ + 00D5


```

346 0340 1 ROUTINE getset (key, flags, char: REF $BBLOCK) =
347 0341 BEGIN
348 0342
349 0343 ++
350 0344
351 0345 Get or set the broadcast classes.
352 0346
353 0347 Inputs:
354 0348     key - if 0, get the broadcast classes
355 0349         if 1, set the broadcast classes
356 0350
357 0351 Outputs:
358 0352     flags - if key=0, then flags will be filled in with the broadcast
359 0353             class quadword.
360 0354             if key=1, then flags will be used as input to the setmode
361 0355             to set the broadcast classes.
362 0356
363 0357     char - terminal characteristics buffer
364 0358
365 0359 Status return:
366 0360     ss$_normal
367 0361
368 0362     Any valid error from $QIOW, or $ASSIGN
369 0363     0 ==> SYS$COMMAND is not a terminal
370 0364
371 0365 ---
372 0366
373 0367 MAP
374 0368     flags : REF $BBLOCK;
375 0369
376 0370 LOCAL
377 0371     status,
378 0372     chan : WORD,
379 0373     iosb : VECTOR[4,WORD],
380 0374     func;
381 0375
382 0376
383 0377 First, open a channel. Always use SYS$COMMAND, since that doesn't
384 0378 ever get reassigned.
385 0379
386 P 0380 status = $ASSIGN(CHAN = chan,
387 0381                  DEVNAM = %ASCII 'SYS$COMMAND:');
388 0382
389 0383 IF NOT .status
390 0384 THEN RETURN .status;
391 0385
392 0386 Now to do what is requested.
393 0387
394 0388 IF .key
395 0389 THEN func = io$_setmode OR io$m_brdcst
396 0390 ELSE
397 0391 BEGIN
398 P 0392     status = $QIOW(CHAN = .chan,
399 P 0393                  FUNC = io$_sensemode,
400 P 0394                  IOSB = iosb,
401 P 0395                  P1 = .char,
402 0396                  P2 = 12);
```

```

! Key=1 means
! set the broadcast stuff
! But if key=0, then
! first check to see if
! this is a terminal...
```

```
403 0397 IF .status
404 0398 THEN status = .iosb[0];
405 0399 IF NOT .status
406 0400 THEN
407 0401 BEGIN
408 0402 $DASSGN(CHAN = .chan);
409 0403 RETURN .status;
410 0404 END;
411 0405 IF .char[0,0,8,0] NEQ dc$_term
412 0406 THEN
413 0407 BEGIN
414 0408 $DASSGN(CHAN = .chan);
415 0409 RETURN 0;
416 0410 END;
417 0411 func = io$_sensemode OR io$m_brdcst;
418 0412 END;
419 0413
420 P 0414 status = $QIOW(CHAN = .chan,
421 P 0415 FUNC = .func,
422 P 0416 IOSB = iosb,
423 0417 P1 = .flags);
424 0418 $DASSGN(CHAN = .chan);
425 0419
426 0420
427 0421 ! Check to see if all is well.
428 0422
429 0423 IF .status
430 0424 THEN status = .iosb[0];
431 0425
432 0426 RETURN .status;
433 0427 1 END;
```

```
! If not a terminal,
! refuse, and tell why.
```

```
! Otherwise get ready to
! get the broadcast classes
```

```
! Do it...
```

```
! Get rid of channel
```

```
! See if set/sense
! worked.
```

```
3A 44 4E 41 4D 4D 4F 43 24 53 59 53 00440 P.ACR: .PSECT $SPLIT$,NOWRT,NOEXE,2
010E000C 0044C P.ACQ: .ASCII \SYSS$COMMAND:\
00000000' 00450 .LONG 17694732
.ADDRESS P.ACR
```

```
.EXTRN SYSS$ASSIGN, SYSS$QIOW
.EXTRN SYSS$DASSGN
```

```
.PSECT $CODE$,NOWRT,2
```

```
003C 00000 GETSET: .WORD Save R2,R3,R4,R5
55 00000000G 00 9E 00002 MOVAB SYSS$QIOW, R5
54 00000000G 00 9E 00009 MOVAB SYSS$DASSGN, R4
5E 0C C2 00010 SUBL2 #12, SP
7E 7C 00013 CLRQ -(SP)
08 AE 9F 00015 PUSHAB CHAN
00000000' EF 9F 00018 PUSHAB P.ACQ
00000000G 00 04 FB 0001E CALLS #4, SYSS$ASSIGN
52 50 D0 00025 MOVL R0, STATUS
76 52 E9 00028 BLBC STATUS, 6$
07 04 AC E9 0002B BLBC KEY, 1$
53 4023 8F 3C 0002F MOVZWL #16419, FUNC
```

```
0340
```

```
0381
```

```
0382
```

```
0388
```

```
0389
```


		42	11	00034	BRB	5\$	
		7E	7C	00036	1\$: CLRQ	-(SP)	0396
		7E	7C	00038	CLRQ	-(SP)	
		0C	DD	0003A	PUSHL	#12	
		0C	AC	DD	0003C	PUSHL	CHAR
		7E	7C	0003F	CLRQ	-(SP)	
		24	AE	9F	00041	PUSHAB	IOSB
		27	DD	00044	PUSHL	#39	
	7E	28	AE	3C	00046	MOVZWL	CHAN, -(SP)
		7E	D4	0004A	CLRL	-(SP)	
	65		0C	FB	0004C	CALLS	#12, SYSSQIOW
	52		50	DD	0004F	MOVL	R0, STATUS
	07		52	E9	00052	BLBC	STATUS, 2\$
	52	04	AE	3C	00055	MOVZWL	IOSB, STATUS
	08		52	E8	00059	BLBS	STATUS, 3\$
	7E		6E	3C	0005C	2\$: MOVZWL	CHAN, -(SP)
	64		01	FB	0005F	CALLS	#1, SYSSDASSGN
			3D	11	00062	BRB	6\$
42	8F	0C	BC	91	00064	3\$: CMPB	@CHAR, #66
			08	13	00069	BEQL	4\$
	7E		6E	3C	0006B	MOVZWL	CHAN, -(SP)
	64		01	FB	0006E	CALLS	#1, SYSSDASSGN
			32	11	00071	BRB	7\$
	53	4027	8F	3C	00073	4\$: MOVZWL	#16423, FUNC
			7E	7C	00078	5\$: CLRQ	-(SP)
			7E	7C	0007A	CLRQ	-(SP)
			7E	D4	0007C	CLRL	-(SP)
		08	AC	DD	0007E	PUSHL	FLAGS
		24	7E	7C	00081	CLRQ	-(SP)
			AE	9F	00083	PUSHAB	IOSB
		53	DD	00086	PUSHL	FUNC	
	7E	28	AE	3C	00088	MOVZWL	CHAN, -(SP)
			7E	D4	0008C	CLRL	-(SP)
	65		0C	FB	0008E	CALLS	#12, SYSSQIOW
	52		50	DD	00091	MOVL	R0, STATUS
	7E		6E	3C	00094	MOVZWL	CHAN, -(SP)
	64		01	FB	00097	CALLS	#1, SYSSDASSGN
	04		52	E9	0009A	BLBC	STATUS, 6\$
	52	04	AE	3C	0009D	MOVZWL	IOSB, STATUS
	50		52	DD	000A1	6\$: MOVL	STATUS, R0
			04	000A4	RET		
		50	D4	000A5	7\$: CLRL	R0	0427
			04	000A7	RET		

; Routine Size: 168 bytes, Routine Base: \$CODE\$ + 01C1

: 434 0428 1
: 435 0429 1 END
: 436 0430 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	1108	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$OWNS	96	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$CODES	617	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	22	0	1000	00:01.8

COMMAND QUALIFIERS

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

Size: 617 code + 1204 data bytes
Run Time: 00:14.8
Elapsed Time: 00:52.2
Lines/CPU Min: 1744
Lexemes/CPU-Min: 19983
Memory Used: 120 pages
Compilation Complete

0054 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SETPROCES
LIS

SETSHOBRO
LIS

SETVOLUME
LIS

SETPWD
LIS

SETTERM
LIS

SETQUEUE
LIS

SETTIME
LIS