

CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL

```

SSSSSSSS EEEEEEEEE TTTTTTTTT PPPPPPP 000000 DDDDDDD I11111 SSSSSSS PPPPPPP
SSSSSSSS EEEEEEEEE TTTTTTTTT PPPPPPP 000000 DDDDDDD I11111 SSSSSSS PPPPPPP
SS          EE          TT          PP          PP 00          00 DD          DD I1          I1 SS          PP          PP
SS          EE          TT          PP          PP 00          00 DD          DD I1          I1 SS          PP          PP
SS          EE          TT          PP          PP 00          00 DD          DD I1          I1 SS          PP          PP
SSSSSS    EEEEEEEE    TT          PPPPPPP 00    00    00 DD          DD I1          I1 SSSSSS    PPPPPPP
SSSSSS    EEEEEEEE    TT          PPPPPPP 00    00    00 DD          DD I1          I1 SSSSSS    PPPPPPP
          SS          EE          TT          PP          PP 0000    00 DD          DD I1          I1          SS          PP
          SS          EE          TT          PP          PP 0000    00 DD          DD I1          I1          SS          PP
          SS          EE          TT          PP          PP 00          00 DD          DD I1          I1          SS          PP
          SS          EE          TT          PP          PP 00          00 DD          DD I1          I1          SS          PP
SSSSSSSS EEEEEEEEE TTT          PP          PP 000000 DDDDDDD I11111 SSSSSSS PPPPPPP
SSSSSSSS EEEEEEEEE TTT          PP          PP 000000 DDDDDDD I11111 SSSSSSS PPPPPPP

```

```

LL          I11111 SSSSSSS
LL          I11111 SSSSSSS
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
LLLLLLLLLL I11111 SSSSSSS
LLLLLLLLLL I11111 SSSSSSS

```

```
1 0001 0 MODULE setp0$disp ( IDENT = 'V04-000', MAIN = setp0$disp) =
2 0002 1 BEGIN
3 0003 1
4 0004 1
5 0005 1
6 0006 1
7 0007 1
8 0008 1
9 0009 1
10 0010 1
11 0011 1
12 0012 1
13 0013 1
14 0014 1
15 0015 1
16 0016 1
17 0017 1
18 0018 1
19 0019 1
20 0020 1
21 0021 1
22 0022 1
23 0023 1
24 0024 1
25 0025 1
26 0026 1
27 0027 1
28 0028 1
29 0029 1
30 0030 1
31 0031 1
32 0032 1
33 0033 1
34 0034 1
35 0035 1
36 0036 1
37 0037 1
38 0038 1
39 0039 1
40 0040 1
41 0041 1
42 0042 1
43 0043 1
44 0044 1
45 0045 1
46 0046 1
47 0047 1
48 0048 1
49 0049 1
50 0050 1
51 0051 1
52 0052 1
53 0053 1
54 0054 1
55 0055 1
56 0056 1
57 0057 1

*****
*
*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*   ALL RIGHTS RESERVED.
*
*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*   TRANSFERRED.
*
*   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*   CORPORATION.
*
*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

++
FACILITY: SET Command (SETPO.EXE)

ABSTRACT:

This is the main option dispatcher for all options
handled by the SETPO image which completely resides
is the P0 region, stack and all, in order to map
certain sections into P1 space.

ENVIRONMENT:

VAX/VMS operating system. unprivileged user mode.

AUTHOR: Tim Halvorsen, Dec 1979

Modified by:

V03-003 AEW0001 Anne E. Warner 20-Jul-1984
Turn on the capability to report messages which was
originally supressed.

V03-002 BLS0291 Benn Schreiber 24-MAR-1984
Move SET PASSWORD here from SET.

V03-001 GAS0112 Gerry Smith 29-Mar-1983
Use new CLI interface.

--
```

SETPO\$DISP
V04-000

L 13
16-Sep-1984 00:37:26 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:12 [CLIUTL.SRC]SETPCDISP.B32;1

```

: 58      0058 1 !
: 59      0059 1 ! Include files
: 60      0060 1 !
: 61      0061 1 !
: 62      0062 1 LIBRARY 'SYSS$LIBRARY:STARLET.L32'; ! VAX/VMS common definitions
: 63      0063 1
```

```

65 0064 1 | Table of contents
66 0065 1 |
67 0066 1 |
68 0067 1 |
69 0068 1 | FORWARD ROUTINE
70 0069 1 | setp0$disp; ! Main option dispatcher
71 0070 1 |
72 0071 1 |
73 0072 1 | External routines
74 0073 1 |
75 0074 1 |
76 0075 1 | EXTERNAL ROUTINE
77 0076 1 | cli$get_value; ! Get value from CLI
78 0077 1 |
79 0078 1 |
80 0079 1 | Global definitions
81 0080 1 |
82 0081 1 |
83 0082 1 | GLOBAL
84 0083 1 | setp0$l_status: INITIAL(ss$_normal); ! Status returned from option
85 0084 1 |
86 0085 1 |
87 0086 1 | Macro to set up two associated tables. The first table is a list of
88 0087 1 | descriptor addresses. These descriptors contain the option names.
89 0088 1 | The second table is a corresponding list of addresses of option routines.
90 0089 1 |
91 0090 1 | If a new option is added to SETPO, all that is required in this
92 0091 1 | module is to add one line of code, the option name, e.g. WORKING SET.
93 0092 1 | Then, the name of the global routine that is dispatched to from this
94 0093 1 | routine will be named SET$WORKING_SET.
95 0094 1 |
96 0095 1 | MACRO
97 0096 1 |
98 0097 1 | option_name [option] = %EXACTSTRING(4, 0, option)%,
99 0098 1 |
100 0099 1 | option_address [option] = %NAME(%STRING('set$', %STRING(option)))%,
101 0100 1 |
102 0101 1 | option_declare [option] = %NAME(%STRING('set$', %STRING(option))) : NOVALUE%,
103 0102 1 |
104 M 0103 1 | make_table (name) =
105 M 0104 1 | [ITERAL %NAME(%STRING(name, 'table length')) = %LENGTH - 1;
106 M 0105 1 | EXTERNAL ROUTINE option_declare(%REMAINING);
107 M 0106 1 | OWN
108 M 0107 1 | %NAME(%STRING(name, 'option')) : VECTOR[%LENGTH - 1]
109 M 0108 1 | INITIAL (option_name(%REMAINING)),
110 M 0109 1 |
111 M 0110 1 | %NAME(%STRING(name, 'routine')) : VECTOR[%LENGTH - 1]
112 M 0111 1 | INITIAL (option_address(%REMAINING));%;
113 0112 1 |
114 0113 1 |
115 0114 1 |
116 0115 1 | Set up a table of all options, and another table pointing to the address
117 0116 1 | of the routine for each option.
118 0117 1 |
119 0118 1 |
120 P 0119 1 | make_table (set,
121 P 0120 1 | message,

```

SETPODISP
V04-000

: 122

0121 1

password);

N 13
16-Sep-1984 00:37:26
14-Sep-1984 12:09:12

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

Page 4
(2)

```

124 0122 1 ROUTINE handler (sigargs, mechargs) =
125 0123 2 BEGIN
126 0124 2
127 0125 2 |---
128 0126 2 |
129 0127 2 | This routine is a condition handler established by the main
130 0128 2 | routine. It saves the most severe condition for the exit status.
131 0129 2 |---
132 0130 2
133 0131 2
134 0132 2 MAP
135 0133 2     sigargs : REF $BBLOCK,
136 0134 2     mechargs : REF $BBLOCK;
137 0135 2 BIND
138 0136 2     signame = sigargs[chf$l_sig_name] : $BBLOCK;           ! Name of signal
139 0137 2
140 0138 2
141 0139 2 IF .setp0$l_status EQL 1                                     ! If no errors yet, use
142 0140 2 THEN setp0$t_status = .signame;                             ! this one.
143 0141 2
144 0142 2 IF NOT .signame                                             ! If an error signal
145 0143 2 AND .signame[sts$v_severity]                                  ! and severity is worse
146 0144 2 GTRU . $BBLOCK[setp0$l_status, sts$v_severity]         ! than current saved severity
147 0145 2 THEN setp0$l_status = .signame;                             ! then save it for exit
148 0146 2
149 0147 2 RETURN ss$_resignal;                                         ! Resignal to get message
150 0148 1 END;

```

```

.TITLE SETPO$DISP
.IDENT \V04-000\
.PSECT $OWNS,NOEXE,2

```

```

53 53 45 4D 0000 SET_OPTION:
53 53 41 50 0004 .ASCII \MESS\
00000000G 00000000G 00008 SET_ROUTINE: .ASCII \PASS\
.ADDRESS SET$MESSAGE, SET$PASSWORD
.PSECT $GLOBALS,NOEXE,2

```

```

00000001 0000 SETPO$L_STATUS::
.LONG 1
.EXTRN CLISGET VALUE, SET$MESSAGE
.EXTRN SET$PASSWORD
.PSECT $CODE$,NOWRT,2

```

```

0004 0000 HANDLER: .WORD Save R2
50 04 52 0000' CF 9E 00002 MOVAB SETPO$L STATUS, R2
AC 04 C1 00007 ADDL3 #4, SIGARGS, R0
01 62 D1 0000C CMPL SETPO$L STATUS, #1
03 12 0000F BNEQ 1$
62 60 D0 00011 MOVL (R0), SETPO$L STATUS
OF 60 E8 00014 1$: BLBS (R0), 2$
: 0122
: 0136
: 0139
: 0140
: 0142

```

SETPOS\$DISP
V04-000

C 14
16-Sep-1984 00:37:26 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:12 [CLIUTL.SRC]SETPODISP.B32;1

Page 6
(3)

51
51

62
60

03
03

62
50

0918

00 EF 00017
00 ED 0001C
03 1B 00021
60 D0 00023
8F 3C 00026 2\$:
04 0002B

EXTZV #0, #3 SETPOS\$ STATUS, R1
CMPZV #0, #3, (R0), RT
BLEQU 2\$
MOVL (R0), SETPOS\$ STATUS
MOVZWL #2328, R0
RET

: 0144
:
:
: 0145
: 0147
: 0148

; Routine Size: 44 bytes, Routine Base: \$CODE\$ + 0000


```

152 0149 1 ROUTINE setp0$disp = ! Main dispatching routine
153 0150 1
154 0151 1 |**
155 0152 1 | Functional description
156 0153 1 |
157 0154 1 | This routine decodes the set option and dispatches to
158 0155 1 | the option specific routine.
159 0156 1 |
160 0157 1 | Calling sequence
161 0158 1 |
162 0159 1 | Called from the Command Language Interpreter
163 0160 1 |
164 0161 1 | Input parameters
165 0162 1 |
166 0163 1 | None
167 0164 1 |
168 0165 1 | Output parameters
169 0166 1 |
170 0167 1 | None
171 0168 1 |
172 0169 1 | Routine value
173 0170 1 |
174 0171 1 | Error returned from option routine or SS$_NORMAL.
175 0172 1 |
176 0173 1 |----
177 0174 1
178 0175 2 BEGIN
179 0176 2
180 0177 2 LOCAL
181 0178 2 desc : BLOCK[dsc$_s_bln, BYTE], ! Option descriptor
182 0179 2 status; ! status code
183 0180 2
184 0181 2 ENABLE handler; ! Enable the condition handler
185 0182 2
186 0183 2 |
187 0184 2 | Get option, and dispatch to it
188 0185 2 |
189 0186 2 $init_dyndesc(desc); ! Make descriptor dynamic
190 0187 3 IF NOT (status = cli$get_value(%ASCII 'OPTION', desc))
191 0188 2 THEN RETURN .status;
192 0189 2
193 0190 2 desc[dsc$_w_length] = MINU (.desc[dsc$_w_length], 4);
194 0191 2
195 0192 2 INCR index FROM 0 TO set_table_length - 1 DO
196 0193 3 BEGIN
197 0194 3 IF CH$EQL(.desc[dsc$_w_length], .desc[dsc$_a_pointer],
198 0195 3 .desc[dsc$_w_length], set_option[.index])
199 0196 3 THEN
200 0197 4 BEGIN
201 0198 4 (.set_routine[.index])();
202 0199 4 EXITLOOP
203 0200 3 END;
204 0201 2 END;
205 0202 2
206 0203 2 RETURN (.setp0$_status); ! Exit with message
207 0204 2
208 0205 1 END;

```

				.PSECT \$SPLITS\$,NOWRT,NOEXE,2								
00	00	4E	4F	49	54	50	4F	00000	P.AAB:	.ASCII	\OPTION\<0><0>	:
						010E0006		00008	P.AAA:	.LONG	17694726	:
						00000000'		0000C		.ADDRESS	P.AAB	:
				.PSECT \$CODES\$,NOWRT,2								
				001C 00000 SETPO\$DISP:								
									.WORD	Save R2,R3,R4		0149
		5E			04	C2		00002	SUBL2	#4, SP		:
		6D			004A	CF		00005	MOVAL	6\$, (FP)		0175
					020E0000	8F		0000A	PUSHL	#34471936		0186
					04	AE		00010	CLRL	DESC+4		:
						5E		00013	PUSHL	SP		0187
					0000'	CF		00015	PUSHAB	P.AAA		:
		0000G				02		00019	CALLS	#2, CLISGET_VALUE		:
						50		0001E	BLBC	STATUS, 5\$:
						6E		00021	MOVZWL	DESC, R0		0190
						50		00024	CMPW	R0, #4		:
						03		00027	BLEQU	1\$:
						04		00029	MOVL	#4, R0		:
						50		0002C	MOVW	R0, DESC		:
						54		0002F	CLRL	INDEX		0194
					0000'	CF44		00031	PUSHAL	SET_OPTION[INDEX]		0195
9E		08			04	AE		00036	CMPC3	DESC, @DESC+4, @(SP)+		:
						0B		0003C	BNEQ	3\$:
					0000'	CF44		0003E	MOVL	SET_ROUTINE[INDEX], R0		0198
						00		00044	CALLS	#0, -(R0)		:
						04		00047	BRB	4\$		0197
E4						01		00049	AOBLEQ	#1, INDEX, 2\$		0192
					0000'	CF		0004D	MOVL	SETPOS_L_STATUS, R0		0203
						04		00052	RET			0205
						0000		00053	.WORD	Save nothing		0175
						7E		00055	CLRL	-(SP)		:
						5E		00057	PUSHL	SP		:
						AC		00059	MOVQ	4(AP), -(SP)		:
		FF72				03		0005D	CALLS	#3, HANDLER		:
						04		00062	RET			:

; Routine Size: 99 bytes, Routine Base: \$CODE\$ + 002C

SETPODISP
V04-000

F 14
16-Sep-1984 00:37:26
14-Sep-1984 12:09:12

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

Page 9
(5)

: 210 0206 1 END
: 211 0207 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	16	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	143	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	16	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	13	0	581	00:01.0

COMMAND QUALIFIERS

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

: Size: 143 code + 36 data bytes
: Run Time: 00:04.9
: Elapsed Time: 00:20.8
: Lines/CPU Min: 2529
: Lexemes/CPU-Min: 14969
: Memory Used: 54 pages
: Compilation Complete

0053 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 terminal session with various system commands and their outputs. The text is small and dense, but several windows are clearly legible and contain the following commands:

- SETFILE LIS
- SETPOMESS LIS
- SETP001SP LIS
- SETMISC LIS
- SETPRO LIS
- SETMAIN LIS

Other windows show various system status reports, error messages, and command-line interactions. The overall appearance is that of a multi-user system terminal screen from the VAX/VMS era.