

[illegible]

[illegible]


```
1 0001 0 MODULE setp0$mess (IDENT = 'V04-000',
2 0002 0 ADDRESSING_MODE(EXTERNAL=GENERAL)) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: SET Command (SETP0.EXE)
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module processes the SET MESSAGE command
35 0035 1
36 0036 1 ENVIRONMENT:
37 0037 1
38 0038 1 VAX/VMS operating system. unprivileged user mode,
39 0039 1
40 0040 1 AUTHOR: Tim Halvorsen, Dec 1979
41 0041 1
42 0042 1 Modified by:
43 0043 1
44 0044 1 V03-006 AEW0001 Anne E. Warner 20-Jul-1984
45 0045 1 Enable the message cli$ confqual to be returned when
46 0046 1 a file-spec is specified with the /DELETE qualifier.
47 0047 1
48 0048 1 V03-005 BLS0295 Benn Schreiber 7-APR-1984
49 0049 1 Disable SYSPRV.
50 0050 1
51 0051 1 V03-004 BLS0291 Benn Schreiber 24-MAR-1984
52 0052 1 Make routine name SET$MESSAGE.
53 0053 1
54 0054 1 V03-003 GAS0112 Gerry Smith 29-Mar-1983
55 0055 1 Remove references to old CLI interface.
56 0056 1
57 0057 1 V03-002 GAS0091 Gerry Smith 19-Oct-1982
```

SETPOSMESS
V04-000

I 14
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 2
(1)

```
: 58      0058 1 | Change input request for new CLD syntax.
: 59      0059 1 |
: 60      0060 1 | V03-001 DWT0036      David Thiel      1-Apr-1982
: 61      0061 1 | Don't allow an indirect message section to be installed.
: 62      0062 1 |
: 63      0063 1 | V001      TMH0001      Tim Halvorsen  26-Dec-1981
: 64      0064 1 | Complete the code to delete a message section when
: 65      0065 1 | desired (/DELETE) by unmaping the section and deassigning
: 66      0066 1 | the channel to the file. Two new cells were added to P1
: 67      0067 1 | space to hold the virtual address range and the channel #.
: 68      0068 1 | --
: 69      0069 1 |
: 70      0070 1 |
: 71      0071 1 | Include files
: 72      0072 1 |
: 73      0073 1 |
: 74      0074 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32'; ! VAX/VMS common definitions
: 75      0075 1 |
```



```

77 0076 1  |
78 0077 1  | Table of contents
79 0078 1  |
80 0079 1  |
81 0080 1  | FORWARD ROUTINE
82 0081 1  |   set$message,
83 0082 1  |   map_section,
84 0083 1  |   delete_section;
85 0084 1  |
86 0085 1  |
87 0086 1  | Define literals
88 0087 1  |
89 0088 1  |
90 0089 1  | LITERAL
91 0090 1  |   true = 1,
92 0091 1  |   false = 0;
93 0092 1  |
94 0093 1  |
95 0094 1  | Macros
96 0095 1  |
97 0096 1  |
98 0097 1  | MACRO
99 M 0098 1  |   perform(command) =
100 M 0099 1  |   BEGIN
101 M 0100 1  |   LOCAL status;
102 M 0101 1  |   status = command;
103 M 0102 1  |   IF NOT .status
104 M 0103 1  |   THEN
105 M 0104 1  |     BEGIN
106 M 0105 1  |       setp0$l_status = .status;
107 M 0106 1  |       RETURN true;
108 M 0107 1  |     END;
109 M 0108 1  |   END%,
110 M 0109 1  |
111 M 0110 1  |   return_if_error(command) =
112 M 0111 1  |   BEGIN
113 M 0112 1  |   LOCAL status;
114 M 0113 1  |   status = command;
115 M 0114 1  |   IF NOT .status
116 M 0115 1  |   THEN
117 M 0116 1  |     RETURN .status;
118 M 0117 1  |   END%;
119 M 0118 1  |
120 M 0119 1  |
121 M 0120 1  |
122 M 0121 1  | External definitions
123 M 0122 1  |
124 M 0123 1  |
125 M 0124 1  | EXTERNAL
126 M 0125 1  |   setp0$l_status,
127 M 0126 1  |   ctl$gb_msgmask:  BYTE,
128 M 0127 1  |   ctl$gl_ppmsg:    VECTOR [2],
129 M 0128 1  |   ctl$gw_ppmsgchn: WORD,
130 M 0129 1  |   ctl$gl_ctlbasva;
131 M 0130 1  |
132 M 0131 1  | EXTERNAL LITERAL
133 M 0132 1  |   cli$_negated,

```

```

! Process SET MESSAGE
! Map message section into P1 region
! Delete message section from P1 region

```

```

! Status returned from option
! Default message flags
! Process permanent message section
! Channel to message section
! Base of fixed P1 region

! Qualifier explicitly negated

```

SETPOMESS
V04-000

K 14
16-Sep-1984 00:38:02 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:13 [CLIUTL.SRC]SETPOMESS.B32;1

Page 4
(2)

```
: 134      0133 1      cli$_confqual;  
: 135      0134 1  
: 136      0135 1 EXTERNAL ROUTINE  
: 137      0136 1      cli$_get_value,  
: 138      0137 1      cli$_present;
```

! Conflicting qualifiers

! Get a value from CLI
! See if qualifier present

SETPOMESS
V04-000

L 14
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 5
(3)

```

: 140      0138 1
: 141      0139 1
: 142      0140 1  ! Define flags to indicate which qualifiers are present
: 143      0141 1  !
: 144      0142 1  !
: 145      0143 1 LITERAL
: 146      P 0144 1      SEQUALST(qual,,0,1,
: 147      P 0145 1      (delete,),
: 148      0146 1      (file,));
: 149      0147 1
: 150      0148 1  !
: 151      0149 1  ! Work areas
: 152      0150 1  !
: 153      0151 1  !
: 154      0152 1 OWN
: 155      0153 1      cli_flags: BITVECTOR[32],      ! CLI qualifier flags
: 156      0154 1      set_mask:  BYTE INITIAL(0),      ! Mask of bits to explicitly set
: 157      0155 1      clear_mask: BYTE INITIAL(0);      ! Mask of bits to explicitly clear
```

SETPOSMESS
V04-000

M 14
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 6
(4)

```
: 159      0156 1 GLOBAL ROUTINE set$message =      ! SET MESSAGE command
: 160      0157 1
: 161      0158 1 ---
: 162      0159 1
: 163      0160 1      This routine processes the SET MESSAGE command
: 164      0161 1
: 165      0162 1      Inputs:
: 166      0163 1          None.
: 167      0164 1
: 168      0165 1      Outputs:
: 169      0166 1
: 170      0167 1          status code
: 171      0168 1      ---
: 172      0169 1
: 173      0170 2 BEGIN
: 174      0171 2
: 175      0172 2 ROUTINE
: 176      0173 2     set_flags =
: 177      0174 2     (BUILTIN ap; ctl$gb_msgmask = .ap <0,4>; true); ! Set message flags
```

```
.TITLE SETPOSMESS
.IDENT \V04-000\
.PSECT $OWNS$,NOEXE,2
```

```
0000 CLI_FLAGS:
      .BLKB 4
00 00004 SET_MASK:
      .BYTE 0
00 00005 CLEAR_MASK:
      .BYTE 0
```

```
.EXTRN SETPOS$ STATUS, CTL$GB MSGMASK
.EXTRN CTL$GL_PPMSG, CTL$GW_PPMSGCHN
.EXTRN CTL$GL_CTLBASVA
.EXTRN CLIS$ NEGATED, CLIS$ CONFQUAL
.EXTRN CLIS$GET_VALUE, CLIS$PRESENT
```

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

```
0000 00000 SET_FLAGS:
```

```
50          5C          04          00 EF 00002      .WORD      Save nothing
          00000000G      00          50 90 00007      EXTZV      #0, #4, AP, R0
          50          01 D0 0000E      MOV      R0, CTL$GB_MSGMASK
          04 00011      RET      #1, R0
```

```
: 0173
: 0174
:
```

: Routine Size: 18 bytes, Routine Base: \$CODE\$ + 0000

```
: 178      0175 2
: 179      0176 2 LOCAL
: 180      0177 2     status,      ! status code
: 181      0178 2     priv_mask : $BBLOCK[8],      ! Mask to clear SYSPRV
: 182      0179 2     file_desc : $BBLOCK[dsc$c_s_bln],      ! File descriptor
: 183      0180 2     new_mask:  BYTE;      ! New message bit mask
```



```

184 0181 2
185 0182 2
186 0183 2 Drop SYSPRV
187 0184 2
188 0185 2 ch$fill(0,8,priv_mask);
189 0186 2 priv_mask[priv$sysprv] = 1;
190 0187 2 $SETPRV(enbflg=0,privadr=priv_mask);
191 0188 2
192 0189 2 Get message file name
193 0190 2
194 0191 2 $init_dyndesc(file_desc);
195 0192 2 cli$get_value(%ASCID 'FILE', file_desc);
196 0193 2 cli_flags[qual_delete] = cli$present(%ASCID 'DELETE');
197 0194 2
198 0195 2
199 0196 2 Get qualifiers on command line
200 0197 2
201 0198 2 status = cli$present(%ASCID 'TEXT'); ! /[NO]TEXT
202 0199 2 IF .status
203 0200 2 THEN set_mask = .set_mask OR 1
204 0201 2 ELSE IF .status EQL cli$_negated
205 0202 2 THEN clear_mask = .clear_mask OR 1;
206 0203 2
207 0204 2 status = cli$present(%ASCID 'IDENT'); ! /[NO]IDENT
208 0205 2 IF .status
209 0206 2 THEN set_mask = .set_mask OR 2
210 0207 2 ELSE IF .status EQL cli$_negated
211 0208 2 THEN clear_mask = .clear_mask OR 2;
212 0209 2
213 0210 2 status = cli$present(%ASCID 'SEVERITY'); ! /[NO]SEVERITY
214 0211 2 IF .status
215 0212 2 THEN set_mask = .set_mask OR 4
216 0213 2 ELSE IF .status EQL cli$_negated
217 0214 2 THEN clear_mask = .clear_mask OR 4;
218 0215 2
219 0216 2 status = cli$present(%ASCID 'FACILITY'); ! /[NO]FACILITY
220 0217 2 IF .status
221 0218 2 THEN set_mask = .set_mask OR 8
222 0219 2 ELSE IF .status EQL cli$_negated
223 0220 2 THEN clear_mask = .clear_mask OR 8;
224 0221 2
225 0222 2
226 0223 2 Set the new message display flag settings first
227 0224 2
228 0225 2
229 0226 2 IF .set_mask NEQ 0 OR .clear_mask NEQ 0 ! If we gotta change something,
230 0227 2 THEN
231 0228 2 BEGIN
232 0229 2 new_mask = (.ctl$gb_msgmask OR .set_mask) AND NOT .clear_mask;
233 0230 2
234 0231 2 perform($CMKRNL(ROUTIN = set_flags,
235 0232 2 ARGVST = .new_mask)); ! Set new message flags
236 0233 2 END;
237 0234 2
238 0235 2
239 0236 2 If /DELETE was specified, then delete the current section
240 0237 2

```



```

241      0238 2 IF .cli_flags [qual_delete]      ! If /DELETE specified,
242      0239 THEN
243      0240 BEGIN
244      0241 IF .file_desc[dsc$w_length] NEQ 0      ! If file spec specified,
245      0242 THEN
246      0243 BEGIN
247      0244 setp0$l_status = cli$_confqual;
248      0245 RETURN      ! return conflicting qualifiers
249      0246 END;
250      0247
251      P 0248 perform($CMKRNL(ROUTIN=delete_section, ! then delete the section
252      0249 ARGST=file_desc));
253      0250 END;
254      0251
255      0252 !
256      0253 If message file name specified, then map the file into
257      0254 the P1 region and set the process permanent message pointer.
258      0255 !
259      0256
260      0257 IF .file_desc[dsc$w_length] NEQ 0      ! If parameter specified,
261      0258 AND NOT .cli_flags [qual_delete]      ! and not deleting the section
262      0259 THEN
263      0260 BEGIN
264      0261 !
265      0262 Only allow 1 section to be mapped at a time for now
266      0263 This can be extended at a later date.
267      0264 !
268      0265 IF .ctl$gl_ppmsg NEQ 0      ! If section already mapped,
269      0266 THEN
270      0267 perform($CMKRNL(ROUTIN=delete_section,ARGST=file_desc));! unmap it
271      0268 !
272      0269 Map the specified file as the current section
273      0270 !
274      P 0271 perform($CMKRNL(ROUTIN=map_section, ! map the section
275      0272 ARGST=file_desc));
276      0273 END;
277      0274
278      0275 RETURN true;
279      0276
280      0277 1 END;

```

INFO#212 L1:0244
: Null expression appears in value-required context

.PSECT \$PLITS,NOWRT,NOEXE,2

```

          45 4C 49 46 00000 P.AAB: .ASCII \FILE\
          010E0004 00004 P.AAA: .LONG 17694724
          00000000' 00008 .ADDRESS P.AAB
00 00 45 54 45 4C 45 44 0000C P.AAD: .ASCII \DELETE\<0><0>
          010E0006 00014 P.AAC: .LONG 17694726
          00000000' 00018 .ADDRESS P.AAD
          54 58 45 54 0001C P.AAF: .ASCII \TEXT\
          010E0004 00020 P.AAE: .LONG 17694724
          00000000' 00024 .ADDRESS P.AAF
00 00 00 54 4E 45 44 49 00028 P.AAH: .ASCII \IDENT\<0><0><0>

```


59	54	49	52	45	56	45	53	010E0005	00030	P.AAG:	.LONG	17694725		
								00000000	00034		.ADDRESS	P.AAH		
								010E0008	00038	P.AAJ:	.ASCII	\SEVERITY\		
								00000000	00040	P.AAI:	.LONG	17694728		
								00000000	00044		.ADDRESS	P.AAJ		
59	54	49	4C	49	43	41	46	010E0008	00048	P.AAL:	.ASCII	\FACILITY\		
								00000000	00050	P.AAK:	.LONG	17694728		
								00000000	00054		.ADDRESS	P.AAL		
											.EXTRN	SYSS\$SETPRV, SYSS\$CMKRNL		
											.PSECT	\$CODE\$,NOWRT,2		
											.ENTRY	SET\$MESSAGE, Save R2,R3,R4,R5,R6,R7,R8,R9,-	0156	
												R10,R11		
											MOVAB	SETPOS\$ STATUS, R11		
											MOVAB	P.AAA, R10		
											MOVAB	SYSS\$CMKRNL, R9		
											MOVL	#CLIS\$ NEGATED, R8		
											MOVAB	CLIS\$PRESENT, R7		
											MOVAB	CLEAR_MASK, R6		
											SUBL2	#16, SP		
											MOVC5	#0, (SP), #0, #8, PRIV_MASK	0185	
											BISB2	#16, PRIV_MASK+3	0186	
											CLRQ	-(SP)	0187	
											PUSHAB	PRIV_MASK		
											CLRL	-(SP)		
											CALLS	#4, SYSS\$SETPRV		
											MOVL	#34471936, FILE_DESC	0191	
											CLRL	FILE_DESC+4		
											PUSHR	#*M<R10, SP>	0192	
											CALLS	#2, CLIS\$GET_VALUE		
											PUSHAB	P.AAC	0193	
											CALLS	#1, CLIS\$PRESENT		
											INSV	R0, #0, #1, CLI_FLAGS		
											PUSHAB	P.AAE	0198	
											CALLS	#1, CLIS\$PRESENT		
											BLBC	STATUS, 1\$	0199	
											BISB2	#1, SET_MASK	0200	
											BRB	2\$		
											CMPL	STATUS, R8	0201	
											BNEQ	2\$		
											BISB2	#1, CLEAR_MASK	0202	
											PUSHAB	P.AAG	0204	
											CALLS	#1, CLIS\$PRESENT		
											BLBC	STATUS, 3\$	0205	
											BISB2	#2, SET_MASK	0206	
											BRB	4\$		
											CMPL	STATUS, R8	0207	
											BNEQ	4\$		
											BISB2	#2, CLEAR_MASK	0208	
											PUSHAB	P.AAI	0210	
											CALLS	#1, CLIS\$PRESENT		
											BLBC	STATUS, 5\$	0211	
											BISB2	#4, SET_MASK	0212	
											BRB	6\$		

58		50	D1	000A2	5\$:	CMPL	STATUS, R8	: 0213
		03	12	000A5		BNEQ	6\$	: 0214
66		04	88	000A7		BISB2	#4, CLEAR_MASK	: 0216
	4C	AA	9F	000AA	6\$:	PUSHAB	P.AAK	: 0217
67		01	FB	000AD		CALLS	#1, CLISPRESNT	: 0218
06		50	E9	000B0		BLBC	STATUS, 7\$	: 0219
FF		08	88	000B3		BISB2	#8, SET_MASK	: 0220
A6		08	11	000B7		BRB	8\$	: 0226
58		50	D1	000B9	7\$:	CMPL	STATUS, R8	: 0229
		03	12	000BC		BNEQ	8\$	: 0232
66		08	88	000BE		BISB2	#8, CLEAR_MASK	: 0238
50		A6	9A	000C1	8\$:	MOVZBL	SET_MASK, -R0	: 0241
	FF	04	12	000C5		BNEQ	9\$	: 0243
		66	95	000C7		TSTB	CLEAR_MASK	: 0249
		1B	13	000C9		BEQL	10\$	: 0257
52	00000000G	00	9A	000CB	9\$:	MOVZBL	CTL\$GB_MSGMASK, R2	: 0258
50		52	C8	000D2		BISL2	R2, R0	: 0265
50		66	8B	000D5		BICB3	CLEAR_MASK, R0, NEW_MASK	: 0267
7E		51	9A	000D9		MOVZBL	NEW_MASK, -(SP)	: 0272
	FF0E	CF	9F	000DC		PUSHAB	SET_FLAGS	: 0275
69		02	FB	000E0		CALLS	#2, SYSSCMKRNL	: 0277
45		50	E9	000E3		BLBC	STATUS, 14\$	: 0275
19		A6	E9	000E6	10\$:	BLBC	CLI_FLAGS, 12\$	: 0277
	FB	6E	B5	000EA		TSTW	FILE_DESC	: 0275
		09	13	000EC		BEQL	11\$	: 0277
6B	00000000G	8F	D0	000EE		MOVL	#CLIS_CONFQUAL, SETPOS_L_STATUS	: 0275
		3B	11	000F5		BRB	16\$	: 0275
		5E	DD	000F7	11\$:	PUSHL	SP	: 0275
	0000V	CF	9F	000F9		PUSHAB	DELETE_SECTION	: 0275
69		02	FB	000FD		CALLS	#2, SYSSCMKRNL	: 0275
28		50	E9	00100		BLBC	STATUS, 14\$	: 0275
		6E	B5	00103	12\$:	TSTW	FILE_DESC	: 0275
		27	13	00105		BEQL	15\$	: 0275
23		A6	E8	00107		BLBS	CLI_FLAGS, 15\$	: 0275
	FB	00	D5	0010B		TSTL	CTL\$GL_PPMSG	: 0275
	00000000G	0C	13	00111		BEQL	13\$	: 0275
		5E	DD	00113		PUSHL	SP	: 0275
	0000V	CF	9F	00115		PUSHAB	DELETE_SECTION	: 0275
69		02	FB	00119		CALLS	#2, SYSSCMKRNL	: 0275
0C		50	E9	0011C		BLBC	STATUS, 14\$	: 0275
		5E	DD	0011F	13\$:	PUSHL	SP	: 0275
	0000V	CF	9F	00121		PUSHAB	MAP_SECTION	: 0275
69		02	FB	00125		CALLS	#2, SYSSCMKRNL	: 0275
03		50	E8	00128		BLBS	STATUS, 15\$	: 0275
6B		50	D0	0012B	14\$:	MOVL	STATUS, SETPOS_L_STATUS	: 0275
50		01	D0	0012E	15\$:	MOVL	#1, R0	: 0275
			04	00131		RET		: 0277
		50	D4	00132	16\$:	CLRL	R0	: 0277
			04	00134		RET		: 0277

; Routine Size: 309 bytes, Routine Base: \$CODE\$ + 0012


```

282 0278 1 ROUTINE map_section =
283 0279 1
284 0280 1 ---
285 0281 1
286 0282 1 This routine maps the message file into the P1 region
287 0283 1 as a process permanent message section. The file name
288 0284 1 is described by the parameter descriptor block.
289 0285 1
290 0286 1 Inputs:
291 0287 1
292 0288 1 ap = Address of parameter descriptor block
293 0289 1
294 0290 1 Outputs:
295 0291 1
296 0292 1 The file is mapped; else status returned in setp0$l_status.
297 0293 1 ---
298 0294 1
299 0295 2 BEGIN
300 0296 2
301 0297 2 BUILTIN
302 0298 2 AP, ! Register AP
303 0299 2 PROBER; ! Built-in function PROBER
304 0300 2
305 0301 2 MAP
306 0302 2 ap: REF BLOCK [,BYTE]; ! Address of parm desc. block
307 0303 2
308 0304 2 LOCAL
309 0305 2 fab: BLOCK [fab$c_bln,BYTE], ! FAB to access file with
310 0306 2 status, ! temporary status
311 0307 2 range: VECTOR [2], ! CRMPSC address range
312 0308 2 section: REF BLOCK[,BYTE]; ! Use to access actual section
313 0309 2
314 0310 2
315 0311 2 Open the message file and obtain a channel to it
316 0312 2
317 0313 2
318 P 0314 2 $FAB_INIT(FAB = fab, ! Initialize FAB
319 P 0315 2 FNA = .ap [dsc$a_pointer], ! Address of file name string
320 P 0316 2 FNS = .ap [dsc$w_length], ! Length of file name string
321 P 0317 2 DNA = UPLIT(' .EXE'), ! Default name string
322 P 0318 2 DNS = 4,
323 0319 2 FOP = UFO); ! User file open
324 0320 2
325 0321 2 return_if_error($OPEN(FAB = fab)); ! Open the file
326 0322 2
327 0323 2
328 0324 2 Map the section into P1 space
329 0325 2
330 0326 2
331 0327 2 range [0] = %X'40000000'; ! Signal to EXPREG in P1 region
332 0328 2 range [1] = .range [0];
333 0329 2
334 P 0330 2 return_if_error($CRMPSC(INADR = range, ! Map the section into P1 space
335 P 0331 2 RETADR = range, ! Get resulting address range
336 P 0332 2 FLAGS = sec$m_expreg, ! Expand region to map section
337 P 0333 2 CHAN = .fab [fab$l_stv], ! Channel number of open file
338 P 0334 2 ACMODE = psl$c_exec, ! Owned by exec mode, read only

```

```

339      VBN = 2));      ! Start after image header
340      ! NOTE: Assume 1 block image header
341
342
343      !
344      ! Change the page protection to user read only. This
345      ! is so that the probing in $GETMSG works for all callers.
346      !
347
348      status = $SETPRT(INADR=range,PROT=prt$c_ur);      ! Change to user read only
349
350      IF NOT .status      ! If error changing protection
351      THEN
352      BEGIN
353      $DELTVA(INADR=range);      ! Unmap section
354      $DASSGN(CHAN=.fab [fab$l_stv]);      ! and deassign channel
355      RETURN .status;      ! return with status
356      END;
357
358      !
359      ! Verify validity of message section
360      !
361
362      section = .range [0];      ! Address the contents of the section
363
364      IF (.section [plv$l_type] NEQ plv$c_typ_msg)      ! Verify validity of section
365      OR (.section [plv$l_version] NEQ 0)
366      OR (.section [plv$l_msgdsp] NEQ 6)
367      OR (.section [plv$l_exec] NEQ %X'65160101')
368      OR (
369      BEGIN
370      BIND
371      rel = section [plv$l_usrundwn];      ! self-relative pointer
372      msc = rel + .rel : BLOCK [, BYTE];      ! message section
373
374      IF PROBER (%REF (psl$c_user), %REF (1), msc [msc$b_type])
375      THEN
376      .msc [msc$b_type] NEQ msc$c_msg      ! only message text type allowed
377      ELSE
378      true
379      END
380      )
381      THEN
382      ! Message file illegal, unmap it
383      BEGIN
384      $DELTVA(INADR=range);      ! Unmap section
385      $DASSGN(CHAN=.fab [fab$l_stv]);      ! and deassign channel
386      RETURN ss$_ivsecidctl;      ! Signal invalid message file
387      END;
388
389      ctl$gl_ppmsg [0] = .range [0];      ! Set address of section vector
390      ctl$gl_ppmsg [1] = .range [1];
391      ctl$gw_ppmsgchn = .fab [fab$l_stv];      ! Save channel to message section
392
393      ctl$gl_ctlbasva = .range [0];      ! Set new base of fixed P1 region
394
395      RETURN true;      ! Return successful

```


: 396
: 397 0392 2
0393 1 END;.PSECT \$PLITS\$,NOWRT,NOEXE,2
45 58 45 2E 00058 P.AAM: .ASCII \.EXE\
: .EXTRN SYSS\$OPEN, SYSS\$CRMPSC
: .EXTRN SYSS\$SETPRT, SYSS\$DELTVA
: .EXTRN SYSS\$DASSGN

.PSECT \$CODE\$,NOWRT,2

00FC 00000 MAP_SECTION:

0050	8F	00	57	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	: 0278	
			56	00000000G	00	9E	00009	MOVAB	SYSS\$DASSGN, R7		
			5E	A8	AE	9E	00010	MOVAB	SYSS\$DELTVA, R6		
			6E		00	2C	00014	MOVAB	-88(SP), SP		
				08	AE		0001B	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	: 0319	
	08	AE		5003	8F	B0	0001D	MOVW	#20483, \$RMS_PTR		
	0C	AE		00020000	8F	D0	00023	MOVL	#131072, \$RMS_PTR+4		
	1E	AE			02	90	0002B	MOVB	#2, \$RMS_PTR+22		
	27	AE			02	90	0002F	MOVB	#2, \$RMS_PTR+31		
	34	AE		04	AC	D0	00033	MOVL	4(AP), \$RMS_PTR+44		
	38	AE		0000	CF	9E	00038	MOVAB	P.AAM, \$RMS_PTR+48		
	3C	AE			6C	90	0003E	MOVB	(AP), \$RMS_PTR+52		
	3D	AE			04	90	00042	MOVB	#4, \$RMS_PTR+53		
				08	AE	9F	00046	PUSHAB	FAB	: 0321	
		00000000G	00		01	FB	00049	CALLS	#1, SYSS\$OPEN		
			2D		50	E9	00050	BLBC	STATUS, 1\$		
			6E	40000000	8F	D0	00053	MOVL	#1073741824, RANGE	: 0327	
	04	AE			6E	D0	0005A	MOVL	RANGE, RANGE+4	: 0328	
					7E	7C	0005E	CLRQ	-(SP)	: 0335	
					02	DD	00060	PUSHL	#2		
					7E	D4	00062	CLRL	-(SP)		
				24	AE	DD	00064	PUSHL	FAB+12		
					7E	7C	00067	CLRQ	-(SP)		
					7E	D4	00069	CLRL	-(SP)		
				00020000	8F	DD	0006B	PUSHL	#131072		
					01	DD	00071	PUSHL	#1		
					28	AE	9F	00073	PUSHAB	RANGE	
					2C	AE	9F	00076	PUSHAB	RANGE	
		00000000G	00		0C	FB	00079	CALLS	#12, SYSS\$CRMPSC		
			01		50	E8	00080	BLBS	STATUS, 2\$		
						04	00083	RET			
			7E		0F	7D	00084	MOVQ	#15, -(SP)	: 0344	
					7E	7C	00087	CLRQ	-(SP)		
					AE	9F	00089	PUSHAB	RANGE		
				10	05	FB	0008C	CALLS	#5, SYSS\$SETPRT		
		00000000G	00		50	D0	00093	MOVL	R0, STATUS		
			52		52	E8	00096	BLBS	STATUS, 3\$	: 0346	
			12		7E	7C	00099	CLRQ	-(SP)	: 0349	
					AE	9F	0009B	PUSHAB	RANGE		
			66		03	FB	0009E	CALLS	#3, SYSS\$DELTVA		

SETPOSMESS
V04-000

H 15
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 14
(5)

		14	AE	DD	000A1	PUSHL	FAB+12	: 0350
	67		01	FB	000A4	CALLS	#1, SYSSDASSGN	: 0351
	50		52	D0	000A7	MOVL	STATUS, R0	: 0358
				04	000AA	RET		: 0360
	50		6E	D0	000AB	3\$: MOVL	RANGE, SECTION	: 0361
	02		60	D1	000AE	CMPL	(SECTION), #2	: 0362
			25	12	000B1	BNEQ	4\$	: 0363
		04	A0	D5	000B3	TSTL	4(SECTION)	: 0368
			20	12	000B6	BNEQ	4\$	: 0369
	06	08	A0	D1	000B8	CMPL	8(SECTION), #6	: 0371
			1A	12	000BC	BNEQ	4\$	: 0373
65160101	8F	0C	A0	D1	000BE	CMPL	12(SECTION), #1695940865	: 0380
			10	12	000C6	BNEQ	4\$	: 0381
	50		10	C0	000C8	ADDL2	#16, R0	: 0382
	50		60	C0	000CB	ADDL2	(R0), R0	: 0385
60	01		03	0C	000CE	PROBER	#3, #1, (R0)	: 0387
			04	13	000D2	BEQL	4\$	: 0389
			60	95	000D4	TSTB	(R0)	: 0391
			14	13	000D6	BEQL	5\$	: 0393
			7E	7C	000D8	4\$: CLRQ	-(SP)	: 0393
		08	AE	9F	000DA	PUSHAB	RANGE	: 0393
	66		03	FB	000DD	CALLS	#3, SYSSDELTV	: 0393
		14	AE	DD	000E0	PUSHL	FAB+12	: 0393
	67		01	FB	000E3	CALLS	#1, SYSSDASSGN	: 0393
	50	02E4	8F	3C	000E6	MOVZWL	#740, R0	: 0393
				04	000EB	RET		: 0393
00000000G	00		6E	7D	000EC	5\$: MOVQ	RANGE, CTL\$GL_PPMSG	: 0393
00000000G	00	14	AE	B0	000F3	MOVW	FAB+12, CTL\$GL_PPMSGCHN	: 0393
00000000G	00		6E	D0	000FB	MOVL	RANGE, CTL\$GL_CTLBASVA	: 0393
	50		01	D0	00102	MOVL	#1, R0	: 0393
			04	00105	RET			: 0393

; Routine Size: 262 bytes, Routine Base: \$CODE\$ + 0147

001C 00000 DELETE_SECTION:

PC	Op	OpC	OpD	OpI	OpJ	OpK	OpL	OpM	OpN	OpO	OpP	OpQ	OpR	OpS	OpT	OpU	OpV	OpW	OpX	OpY	OpZ	OpAA	OpAB	OpAC	OpAD	OpAE	OpAF	OpAG	OpAH	OpAI	OpAJ	OpAK	OpAL	OpAM	OpAN	OpAO	OpAP	OpAQ	OpAR	OpAS	OpAT	OpAU	OpAV	OpAW	OpAX	OpAY	OpAZ	OpBA	OpBB	OpBC	OpBD	OpBE	OpBF	OpBG	OpBH	OpBI	OpBJ	OpBK	OpBL	OpBM	OpBN	OpBO	OpBP	OpBQ	OpBR	OpBS	OpBT	OpBU	OpBV	OpBW	OpBX	OpBY	OpBZ	OpCA	OpCB	OpCC	OpCD	OpCE	OpCF	OpCG	OpCH	OpCI	OpCJ	OpCK	OpCL	OpCM	OpCN	OpCO	OpCP	OpCQ	OpCR	OpCS	OpCT	OpCU	OpCV	OpCW	OpCX	OpCY	OpCZ	OpDA	OpDB	OpDC	OpDD	OpDE	OpDF	OpDG	OpDH	OpDI	OpDJ	OpDK	OpDL	OpDM	OpDN	OpDO	OpDP	OpDQ	OpDR	OpDS	OpDT	OpDU	OpDV	OpDW	OpDX	OpDY	OpDZ	OpEA	OpEB	OpEC	OpED	OpEE	OpEF	OpEG	OpEH	OpEI	OpEJ	OpEK	OpEL	OpEM	OpEN	OpEO	OpEP	OpEQ	OpER	OpES	OpET	OpEU	OpEV	OpEW	OpEX	OpEY	OpEZ	OpFA	OpFB	OpFC	OpFD	OpFE	OpFF	OpFG	OpFH	OpFI	OpFJ	OpFK	OpFL	OpFM	OpFN	OpFO	OpFP	OpFQ	OpFR	OpFS	OpFT	OpFU	OpFV	OpFW	OpFX	OpFY	OpFZ	OpGA	OpGB	OpGC	OpGD	OpGE	OpGF	OpGG	OpGH	OpGI	OpGJ	OpGK	OpGL	OpGM	OpGN	OpGO	OpGP	OpGQ	OpGR	OpGS	OpGT	OpGU	OpGV	OpGW	OpGX	OpGY	OpGZ	OpHA	OpHB	OpHC	OpHD	OpHE	OpHF	OpHG	OpHH	OpHI	OpHJ	OpHK	OpHL	OpHM	OpHN	OpHO	OpHP	OpHQ	OpHR	OpHS	OpHT	OpHU	OpHV	OpHW	OpHX	OpHY	OpHZ	OpIA	OpIB	OpIC	OpID	OpIE	OpIF	OpIG	OpIH	OpII	OpIJ	OpIK	OpIL	OpIM	OpIN	OpIO	OpIP	OpIQ	OpIR	OpIS	OpIT	OpIU	OpIV	OpIW	OpIX	OpIY	OpIZ	OpJA	OpJB	OpJC	OpJD	OpJE	OpJF	OpJG	OpJH	OpJI	OpJJ	OpJK	OpJL	OpJM	OpJN	OpJO	OpJP	OpJQ	OpJR	OpJS	OpJT	OpJU	OpJV	OpJW	OpJX	OpJY	OpJZ	OpKA	OpKB	OpKC	OpKD	OpKE	OpKF	OpKG	OpKH	OpKI	OpKJ	OpKK	OpKL	OpKM	OpKN	OpKO	OpKP	OpKQ	OpKR	OpKS	OpKT	OpKU	OpKV	OpKW	OpKX	OpKY	OpKZ	OpLA	OpLB	OpLC	OpLD	OpLE	OpLF	OpLG	OpLH	OpLI	OpLJ	OpLK	OpLL	OpLM	OpLN	OpLO	OpLP	OpLQ	OpLR	OpLS	OpLT	OpLU	OpLV	OpLW	OpLX	OpLY	OpLZ	OpMA	OpMB	OpMC	OpMD	OpME	OpMF	OpMG	OpMH	OpMI	OpMJ	OpMK	OpML	OpMM	OpMN	OpMO	OpMP	OpMQ	OpMR	OpMS	OpMT	OpMU	OpMV	OpMW	OpMX	OpMY	OpMZ	OpNA	OpNB	OpNC	OpND	OpNE	OpNF	OpNG	OpNH	OpNI	OpNJ	OpNK	OpNL	OpNM	OpNN	OpNO	OpNP	OpNQ	OpNR	OpNS	OpNT	OpNU	OpNV	OpNW	OpNX	OpNY	OpNZ	OpOA	OpOB	OpOC	OpOD	OpOE	OpOF	OpOG	OpOH	OpOI	OpOJ	OpOK	OpOL	OpOM	OpON	OpOO	OpOP	OpOQ	OpOR	OpOS	OpOT	OpOU	OpOV	OpOW	OpOX	OpOY	OpOZ	OpPA	OpPB	OpPC	OpPD	OpPE	OpPF	OpPG	OpPH	OpPI	OpPJ	OpPK	OpPL	OpPM	OpPN	OpPO	OpPP	OpPQ	OpPR	OpPS	OpPT	OpPU	OpPV	OpPW	OpPX	OpPY	OpPZ	OpQA	OpQB	OpQC	OpQD	OpQE	OpQF	OpQG	OpQH	OpQI	OpQJ	OpQK	OpQL	OpQM	OpQN	OpQO	OpQP	OpQQ	OpQR	OpQS	OpQT	OpQU	OpQV	OpQW	OpQX	OpQY	OpQZ	OpRA	OpRB	OpRC	OpRD	OpRE	OpRF	OpRG	OpRH	OpRI	OpRJ	OpRK	OpRL	OpRM	OpRN	OpRO	OpRP	OpRQ	OpRR	OpRS	OpRT	OpRU	OpRV	OpRW	OpRX	OpRY	OpRZ	OpSA	OpSB	OpSC	OpSD	OpSE	OpSF	OpSG	OpSH	OpSI	OpSJ	OpSK	OpSL	OpSM	OpSN
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

SETPOMESS
V04-000

J 15
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 16
(6)

; Routine Size: 69 bytes, Routine Base: \$CODE\$ + 0240

SETPOMESS
V04-000

K 15
16-Sep-1984 00:38:02
14-Sep-1984 12:09:13

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

Page 17
(7)

: 433 0427 1 END
: 434 0428 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	6	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	658	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	92	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]LIB.L32;1	18619	76	0	1000	00:01.8

: Information: 1
: Warnings: 0
: Errors: 0

COMMAND QUALIFIERS

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

: Size: 658 code + 98 data bytes
: Run Time: 00:15.5
: Elapsed Time: 00:51.7
: Lines/CPU Min: 1658
: Lexemes/CPU-Min: 25612
: Memory Used: 160 pages
: Compilation Complete

0053 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

