


```

PPPPPPPP      AAAAAA      TTTTTTTTTT      MM      MM      000000      DDDDDDDD
PPPPPPPP      AAAAAA      TTTTTTTTTT      MM      MM      000000      DDDDDDDD
PP      PP      AA      AA      TT      MMMM      MMMM      00      00      DD      DD
PP      PP      AA      AA      TT      MMMM      MMMM      00      00      DD      DD
PP      PP      AA      AA      TT      MM      MM      00      00      DD      DD
PP      PP      AA      AA      TT      MM      MM      00      00      DD      DD
PPPPPPPP      AA      AA      TT      MM      MM      00      00      DD      DD
PPPPPPPP      AA      AA      TT      MM      MM      00      00      DD      DD
PP      AAAAAAAAAA      TT      MM      MM      00      00      DD      DD
PP      AAAAAAAAAA      TT      MM      MM      00      00      DD      DD
PP      AA      AA      TT      MM      MM      00      00      DD      DD
PP      AA      AA      TT      MM      MM      00      00      DD      DD
PP      AA      AA      TT      MM      MM      000000      DDDDDDDD
PP      AA      AA      TT      MM      MM      000000      DDDDDDDD

```

```

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
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE PATMOD (
2 L 0002 0 %IF %VARIANT EQL 1
3 0003 0 %THEN
4 0004 0 ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
5 0005 0 %FI
6 0006 0 IDENT = 'V04-000') =
7 0007 1 BEGIN
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1 FACILITY: PATCH
33 0033 1
34 0034 1 **
35 0035 1 FUNCTIONAL DESCRIPTION:
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1
39 0039 1 Version: X01.10
40 0040 1
41 0041 1 History:
42 0042 1 Author:
43 0043 1 Carol Peters, 03 Jul 1976: Version 01
44 0044 1
45 0045 1 Modified by:
46 0046 1 Kathleen Morse, 19 Oct 1977: Version X01.00
47 0047 1
48 0048 1 Revision history:
49 0049 1
50 0050 1 NO DATE PROGRAMMER PURPOSE
51 0051 1 -- ---- -
52 0052 1
53 0053 1 00 19-OCT-77 K.D. MORSE ADAPT VERSION 24 FOR PATCH
54 0054 1 01 4-JAN-78 K.D. MORSE NO CHANGES FOR VERS 25.
55 0055 1 DISPLAY SETTING OF MODE_SYMBOLIC
56 0056 1 IN SHOW MODES. (26)
57 0057 1 NO CHANGES FOR VERS 27-28.

```

PATMOD
V04-000

N 13
16-Sep-1984 00:34:55 VAX-11 Bliss-32 V4.0-742 Page 2
14-Sep-1984 12:52:40 DISK\$VMSMASTER:[PATCH.SRC]PATMOD.B32;1 (1)

:	58	0058	1	:	02	24-JAN-78	K.D. MORSE
:	59	0059	1	:	03	13-FEB-78	K.D. MORSE
:	60	0060	1	:			
:	61	0061	1	:	04	25-APR-78	K.D. MORSE
:	62	0062	1	:	05	18-MAY-78	K.D. MORSE
:	63	0063	1	:	06	18-MAY-78	K.D. MORSE
:	64	0064	1	:	07	13-JUN-78	K.D. MORSE
:	65	0065	1	:	08	15-JUN-78	K.D. MORSE
:	66	0066	1	:			
:	67	0067	1	:	09	21-JUN-78	K.D. MORSE
:	68	0068	1	:	10	28-JUN-78	K.D. MORSE
:	69	0069	1	:			
:	70	0070	1	:			
:	71	0071	1	:	--		

NO CHANGES FOR 29.
ADD OUTPUT TO INDIRECT COMMAND
FILE FOR "SET MODE" MODES.
CONVERT TO NATIVE MODE.
NO CHANGES FOR VERS 30-31.
NO CHANGES FOR VERS 32.
ADD FAO COUNTS TO SIGNALS.
ALWAYS CALL PAT\$WRITEFILE TO
OUTPUT TO APPENDED TEXT BUFFERS.
NO CHANGES FOR VERS 33.
CHANGED PAT\$SHOW_DEFAL TO WOFK
FROM THE CURRENT MODE LEVEL (34).

```
.. 73      0072 1 FORWARD ROUTINE
.. 74      0073 1
.. 75      0074 1
.. 76      0075 1
.. 77      0076 1
.. 78      0077 1
.. 79      0078 1
.. 80      0079 1
.. 81      0080 1 REQUIRE 'SRCS:VXSMAC.REQ':
.. 82      0145 1 REQUIRE 'SRCS:PATPCT.REQ':
.. 83      0185 1 REQUIRE 'SRCS:PATGEN.REQ':
.. 84      0407 1 REQUIRE 'SRCS:PATTER.REQ':
.. 85      0614 1 REQUIRE 'SRCS:BSTRUC.REQ':
.. 86      0690 1 REQUIRE 'SRCS:LISTEL.REQ':
.. 87      0732 1 REQUIRE 'SRCS:SYSSER.REQ':
```

```
! Sets modes to a specified level
! Sets default modes to initialization value
! Sets a list of modes
! Sets level of modes
! Sets a new mode
! Action routine to show default
```

PATMOD
V04-000

C 14
16-Sep-1984 00:34:55
15-Sep-1984 22:50:49

VAX-11 Bliss-32 V4.0-742
_\$255\$DUA28:[PATCH.SRC]SYSSER.REQ;1

Page 4
(1)

: R0764 1
: R0765 1
: R0766 1
: R0767 1
: R0768 1

SWITCHES LIST (SOURCE);

EXTERNAL ROUTINE

PAT\$fao_out;

! formats a line and outputs to the terminal

PA
VO

```

: 88      0814 1 REQUIRE 'SRCS:PREFIX.REQ';
: 89      1002 1 REQUIRE 'SRCS:PATPRE.REQ';
: 90      1165 1
: 91      1166 1 EXTERNAL ROUTINE
: 92      1167 1     PAT$WRITEFILE;
: 93      1168 1 EXTERNAL
: 94      1169 1     PAT$GL_COMRAB,
: 95      1170 1     PAT$GL_FLAGS,
: 96      1171 1     PAT$GL_CONTEXT : BITVECTOR,
: 97      1172 1     PAT$GB_MOD_PTR: REF VECTOR [, BYTE],
: 98      1173 1     PAT$GB_DEF_MOD: VECTOR [, BYTE],
: 99      1174 1     PAT$GL_HEAD_LST;

```

! Writes a line to a file
! RAB for output command file
! CLI flags
! Context longword
! Pointer to modes
! Block for modes
! Head of argument list

```

101 1175 1 !++
102 1176 1 | Counted strings for FAO.
103 1177 1 |--
104 1178 1
105 1179 1 BIND
106 1180 1     CS_OCTAL      = UPLIT ( %ASCIC 'octal' ),
107 1181 1     CS_DECIMAL   = UPLIT ( %ASCIC 'decimal' ),
108 1182 1     CS_HEXADecimal = UPLIT ( %ASCIC 'hexadecimal' ),
109 1183 1     CS_BYTE      = UPLIT ( %ASCIC 'byte' ),
110 1184 1     CS_WORD      = UPLIT ( %ASCIC 'word' ),
111 1185 1     CS_LONGWORD  = UPLIT ( %ASCIC 'long' ),
112 1186 1     CS_BOOLEAN_ON = UPLIT ( %ASCIC ' ' ),
113 1187 1     CS_BOOLEAN_OFF = UPLIT ( %ASCIC ' no' );
114 1188 1
115 1189 1 !++
116 1190 1 | The following literals define the byte offset into the MODE_NAME_TBL for
117 1191 1 | the keywords for the 'SET MODE' command. The order of these offsets must be
118 1192 1 | the same as the order of the names in the MODE_NAME_TBL.
119 1193 1 |--
120 1194 1 LITERAL
121 1195 1     BYTE_NAME = 0,
122 1196 1     WORD_NAME = 0 + %CHARCOUNT(%ASCIC 'B'),
123 1197 1     LONG_NAME = WORD_NAME + %CHARCOUNT(%ASCIC 'W'),
124 1198 1     DECIMAL_NAME = LONG_NAME + %CHARCOUNT(%ASCIC 'LO'),
125 1199 1     HEX_NAME = DECIMAL_NAME + %CHARCOUNT(%ASCIC 'DEC'),
126 1200 1     OCTAL_NAME = HEX_NAME + %CHARCOUNT(%ASCIC 'H'),
127 1201 1     INSTRUC_NAME = OCTAL_NAME + %CHARCOUNT(%ASCIC 'OC'),
128 1202 1     NOINSTRUC_NAME = INSTRUC_NAME + %CHARCOUNT(%ASCIC 'I'),
129 1203 1     ASCII_NAME = NOINSTRUC_NAME + %CHARCOUNT(%ASCIC 'NOI'),
130 1204 1     NOASCII_NAME = ASCII_NAME + %CHARCOUNT(%ASCIC 'AS'),
131 1205 1     SYMBOL_NAME = NOASCII_NAME + %CHARCOUNT(%ASCIC 'NOAS'),
132 1206 1     NOSYMBOL_NAME = SYMBOL_NAME + %CHARCOUNT(%ASCIC 'SY'),
133 1207 1     SCOPE_NAME = NOSYMBOL_NAME + %CHARCOUNT(%ASCIC 'NOSY'),
134 1208 1     NOSCOPE_NAME = SCOPE_NAME + %CHARCOUNT(%ASCIC 'SC'),
135 1209 1     GLOBAL_NAME = NOSCOPE_NAME + %CHARCOUNT(%ASCIC 'NOSC'),
136 1210 1     NOGLOBAL_NAME = GLOBAL_NAME + %CHARCOUNT(%ASCIC 'GL');
137 1211 1
138 1212 1 !++
139 1213 1 | This table holds the names of the modes acceptable to a 'SET MODE' command.
140 1214 1 | It is used to write the output command file and the appended patch commands.
141 1215 1 |--
142 1216 1 BIND
143 1217 1     MODE_NAME_TBL = UPLIT BYTE (
144 1218 1         %ASCIC 'B',
145 1219 1         %ASCIC 'W',
146 1220 1         %ASCIC 'LO',
147 1221 1         %ASCIC 'DEC',
148 1222 1         %ASCIC 'H',
149 1223 1         %ASCIC 'OC',
150 1224 1         %ASCIC 'I',
151 1225 1         %ASCIC 'NOI',
152 1226 1         %ASCIC 'AS',
153 1227 1         %ASCIC 'NOAS',
154 1228 1         %ASCIC 'SY',
155 1229 1         %ASCIC 'NOSY',
156 1230 1         %ASCIC 'SC',
157 1231 1         %ASCIC 'NOSC',

```



```

158 1232 1          %ASCIC 'GL',
159 1233 1          %ASCIC 'NOGL'
160 1234 1
161 1235 1          ) : VECTOR [, BYTE];
162 1236 1 ! This macro is probably temporary - see routine PUT_DEFAL
163 1237 1
164 1238 1 MACRO
165 1239 1     YES_NO ( VALUE ) =
166 1240 1         (IF VALUE
167 1241 1             THEN CS_BOOLEAN_ON
168 1242 1             ELSE CS_BOOLEAN_OFF
169 1243 1         ) %;
170 1244 1
171 1245 1
172 1246 1 !+
173 1247 1 ! The following table holds the valid mode settings and values
174 1248 1 ! relevant to them. Each entry in the table is four bytes long.
175 1249 1 ! The entry has four fields, each of them one byte long.
176 1250 1
177 1251 1 ! -----
178 1252 1 ! value ! offset ! keyword ! name_offset !
179 1253 1 ! -----
180 1254 1 ! The keyword field holds the token value that represents the particular
181 1255 1 ! mode, e.g., BYTE_TOKEN indicates that the mode is BYTE.
182 1256 1 ! The offset field holds the location of the mode setting in any of
183 1257 1 ! the mode levels.
184 1258 1 ! The value field holds the value that represents a particular mode
185 1259 1 ! to patch in an internal sense.
186 1260 1 ! The name_offset field holds the byte offset into the mode name table,
187 1261 1 ! MODE_NAME_TBL, to the ascic string for this mode. This is used to reconstruct
188 1262 1 ! the command line for the output command file and appended text.
189 1263 1
190 1264 1 LITERAL
191 1265 1     KEYWORD_FIELD = 0,
192 1266 1     OFFSET_FIELD  = 1,
193 1267 1     VALUE_FIELD   = 2,
194 1268 1     NAME_OFF_FIELD = 3,
195 1269 1     MODE_ENTRY_LEN = 4;
196 1270 1
197 1271 1 BIND
198 1272 1     MODE_TABLE = UPLIT BYTE (
199 1273 1
200 1274 1     BYTE_TOKEN, MODE_LENGTH, BYTE_LENGTH, BYTE_NAME,
201 1275 1     WORD_TOKEN, MODE_LENGTH, WORD_LENGTH, WORD_NAME,
202 1276 1     LONG_TOKEN, MODE_LENGTH, LONG_LENGTH, LONG_NAME,
203 1277 1     DECIMAL_TOKEN, MODE_RADIX, DECIMAL_RADIX, DECIMAL_NAME,
204 1278 1     HEXADECTM_TOKEN, MODE_RADIX, HEX_RADIX, HEX_NAME,
205 1279 1     OCTAL_TOKEN, MODE_RADIX, OCTAL_RADIX, OCTAL_NAME,
206 1280 1     INSTRUC_TOKEN, MODE_INSTRUC, TRUE, INSTRUC_NAME,
207 1281 1     NOINSTRUC_TOKEN, MODE_INSTRUC, FALSE, NOINSTRUC_NAME,
208 1282 1     ASCII_TOKEN, MODE_ASCII, TRUE, ASCII_NAME,
209 1283 1     NOASCII_TOKEN, MODE_ASCII, FALSE, NOASCII_NAME,
210 1284 1     SYMBOLS_TOKEN, MODE_SYMBOLS, TRUE, SYMBOL_NAME,
211 1285 1     NOSYMBOLS_TOKEN, MODE_SYMBOLS, FALSE, NOSYMBOL_NAME,
212 1286 1     SCOPE_TOKEN, MODE_SCOPE, TRUE, SCOPE_NAME,
213 1287 1     NOSCOPE_TOKEN, MODE_SCOPE, FALSE, NOSCOPE_NAME,
214 1288 1     GLOBALS_TOKEN, MODE_GLOBALS, TRUE, GLOBAL_NAME,

```



```

220 1293 1 GLOBAL ROUTINE PAT$INIT_MODES (GOAL_LEVEL, SOURCE_LEVEL) : NOVALUE =
221 1294 1
222 1295 1 +-
223 1296 1 FUNCTIONAL DESCRIPTION:
224 1297 1
225 1298 1     Sets all levels from local level to the goal level specified
226 1299 1     with the mode settings of the source level.
227 1300 1
228 1301 1 CALLING SEQUENCE:
229 1302 1
230 1303 1     PAT$INIT_MODES ( )
231 1304 1
232 1305 1 INPUTS:
233 1306 1
234 1307 1     GOAL_LEVEL     - Highest level to set.
235 1308 1     SOURCE_LEVEL  - Level from which to obtain mode settings.
236 1309 1
237 1310 1 IMPLICIT INPUTS:
238 1311 1
239 1312 1     none
240 1313 1
241 1314 1 OUTPUTS:
242 1315 1
243 1316 1     none
244 1317 1
245 1318 1 IMPLICIT OUTPUTS:
246 1319 1
247 1320 1     The mode settings of the local level to the goal level
248 1321 1     are reset to the mode settings of the source level.
249 1322 1
250 1323 1 ROUTINE VALUE:
251 1324 1
252 1325 1     novalue
253 1326 1
254 1327 1 SIDE EFFECTS:
255 1328 1
256 1329 1     none
257 1330 1
258 1331 1 --
259 1332 1
260 1333 2 BEGIN
261 1334 2
262 1335 2 LOCAL
263 1336 2     TEMP_LEVEL: REF VECTOR [, BYTE];
264 1337 2
265 1338 2 TEMP_LEVEL = PAT$GB DEF MOD [.SOURCE_LEVEL * MODE_LVL_SIZE];
266 1339 2 DECR I FROM LOCAL_MODE TO .GOAL_LEVEL DO
267 1340 3     BEGIN
268 1341 3     PAT$GB_MOD_PTR = PAT$GB DEF MOD [.I * MODE_LVL_SIZE];
269 1342 3     PAT$GB_MOD_PTR [MODE_RADIX] = .TEMP_LEVEL [MODE_RADIX];
270 1343 3     PAT$GB_MOD_PTR [MODE_LENGTH] = .TEMP_LEVEL [MODE_LENGTH];
271 1344 3     PAT$GB_MOD_PTR [MODE_SYMBOLS] = .TEMP_LEVEL [MODE_SYMBOLS];
272 1345 3     PAT$GB_MOD_PTR [MODE_INSTRUC] = .TEMP_LEVEL [MODE_INSTRUC];
273 1346 3     PAT$GB_MOD_PTR [MODE_ASCII] = .TEMP_LEVEL [MODE_ASCII];
274 1347 3     PAT$GB_MOD_PTR [MODE_SCOPE] = .TEMP_LEVEL [MODE_SCOPE];
275 1348 3     PAT$GB_MOD_PTR [MODE_GLOBALS] = .TEMP_LEVEL [MODE_GLOBALS];
276 1349 2     END;

```


					.EXTRN	PAT\$GB_DEF_MOD, PAT\$GL_HEAD_LST	
					.WEAK	ACCESS_CHECK	
					.PSECT	_PAT\$CODE,NOWRT,2	
			001C 00000		.ENTRY	PAT\$INIT MODES, Save R2,R3,R4	: 1293
		54	00000000G	EF 9E 00002	MOVAB	PAT\$GB_DEF_MOD, R4	: 1338
50	08	53	00000000G	EF 9E 00009	MOVAB	PAT\$GB_MOD_PTR, R3	: 1348
52		AC		07 C5 00010	MULL3	#7, SOURCE_LEVEL, R0	: 1341
		50		54 C1 00015	ADDL3	R4, R0, TEMP_LEVEL	: 1342
		50		03 D0 00019	MOVL	#3, I	: 1346
				1A 11 0001C	BRB	2\$: 1348
51		50		07 C5 0001E 1\$:	MULL3	#7, I, R1	: 1341
63		51		54 C1 00022	ADDL3	R4, R1, PAT\$GB_MOD_PTR	: 1342
		51		63 D0 00026	MOVL	PAT\$GB_MOD_PTR, R1	: 1346
		61		62 D0 00029	MOVL	(TEMP_LEVEL), (R1)	: 1348
	04	A1	04	A2 B0 0002C	MOVW	4(TEMP_LEVEL), 4(R1)	: 1339
	06	A1	06	A2 90 00031	MOVB	6(TEMP_LEVEL), 6(R1)	: 1350
				50 D7 00036	DECL	I	
	04	AC		50 D1 00038 2\$:	CPL	I, GOAL_LEVEL	
				E0 18 0003C	BGEQ	1\$	
				04 0003E	RET		

; Routine Size: 63 bytes, Routine Base: _PAT\$CODE + 0000

```

279 1351 1 GLOBAL ROUTINE PAT$RESET_DEF : NOVALUE =
280 1352 1
281 1353 1 +-+
282 1354 1 FUNCTIONAL DESCRIPTION:
283 1355 1
284 1356 1     Cancels user set defaults for mode settings and resets all mode
285 1357 1     levels to PATCH defaults.
286 1358 1
287 1359 1 CALLING SEQUENCE:
288 1360 1
289 1361 1     PAT$RESET_DEF ( )
290 1362 1
291 1363 1 INPUTS:
292 1364 1
293 1365 1     none
294 1366 1
295 1367 1 IMPLICIT INPUTS:
296 1368 1
297 1369 1     The default modes.
298 1370 1
299 1371 1 OUTPUTS:
300 1372 1
301 1373 1     none
302 1374 1
303 1375 1 IMPLICIT OUTPUTS:
304 1376 1
305 1377 1     none
306 1378 1
307 1379 1 ROUTINE VALUE:
308 1380 1
309 1381 1     novalue
310 1382 1
311 1383 1 SIDE EFFECTS:
312 1384 1
313 1385 1     Resets default values for modes.
314 1386 1
315 1387 1 --
316 1388 1
317 1389 2 BEGIN
318 1390 2
319 1391 2 PAT$GB_MOD_PTR = PAT$GB_DEF_MOD [DEFAULT_MODE * MODE_LVL_SIZE];
320 1392 2 PAT$GB_MOD_PTR [MODE_RADIX] = DEF_MODE_RADIX;
321 1393 2 PAT$GB_MOD_PTR [MODE_LENGTH] = DEF_MODE_LENGTH;
322 1394 2 PAT$GB_MOD_PTR [MODE_SYMBOLS] = TRUE;
323 1395 2 PAT$GB_MOD_PTR [MODE_INSTRUC] = FALSE;
324 1396 2 PAT$GB_MOD_PTR [MODE_ASCII] = FALSE;
325 1397 2 PAT$GB_MOD_PTR [MODE_SCOPE] = TRUE;
326 1398 2 pat$GB_MOD_PTR [MODE_GLOBALS] = FALSE;
327 1399 2 PAT$INIT_MODES (USER_DEF_MODE, DEFAULT_MODE);
328 1400 1 END;

```

52 00000000G EF 0004 0000
9E 00002

.ENTRY PAT\$RESET_DEF, Save R2
MOVAB PAT\$GB_MOD_PTR, R2

: 1351
:

PATMOD
V04-000

L 14
16-Sep-1984 00:34:55
14-Sep-1984 12:52:40

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATMOD.B32;1 (5) Page 13

	62	00000000G	EF	9E	00009	MOVAB	PAT\$GB_DEF_MOD, PAT\$GB_MOD_PTR	:	1391
	50		62	D0	00010	MOVL	PAT\$GB_MOD_PTR, R0	:	1392
	60	00010410	8F	D0	00013	MOVL	#66576, (R0)	:	
04	A0	0100	8F	B0	0001A	MOVW	#256, 4(R0)	:	1396
		06	A0	94	00020	CLRB	6(R0)	:	1398
	7E		01	7D	00023	MOVQ	#1, -(SP)	:	1399
97	AF		02	FB	00026	CALLS	#2, PAT\$INIT_MODES	:	
			04	00	002A	RET		:	1400

; Routine Size: 43 bytes, Routine Base: _PAT\$CODE + 003F

```

330 1401 1 GLOBAL ROUTINE PAT$SET_MOD_LST (LEVEL) : NOVALUE =
331 1402 1
332 1403 1 ++
333 1404 1 FUNCTIONAL DESCRIPTION:
334 1405 1
335 1406 1     Sets a list of modes at the specified level.
336 1407 1
337 1408 1 CALLING SEQUENCE:
338 1409 1
339 1410 1     PAT$SET_MOD_LST ( )
340 1411 1
341 1412 1 INPUTS:
342 1413 1
343 1414 1     LEVEL - Level at which to set modes.
344 1415 1
345 1416 1 IMPLICIT INPUTS:
346 1417 1
347 1418 1     -The address of the linked list which holds the switches.
348 1419 1     -The global pointers to the current mode levels.
349 1420 1
350 1421 1 OUTPUTS:
351 1422 1
352 1423 1     none
353 1424 1
354 1425 1 IMPLICIT OUTPUTS:
355 1426 1
356 1427 1     none
357 1428 1
358 1429 1 ROUTINE VALUE:
359 1430 1
360 1431 1     novalue
361 1432 1
362 1433 1 SIDE EFFECTS:
363 1434 1
364 1435 1     The appropriate modes are set.
365 1436 1
366 1437 1 --
367 1438 1
368 1439 2 BEGIN
369 1440 2
370 1441 2 LOCAL
371 1442 2     POINTER;
372 1443 2 PAT$SET_MOD_LVL (.LEVEL);
373 1444 2 POINTER = .PAT$GL_HEAD_LST;
374 1445 2 DO
375 1446 2     PAT$SET_NEW_MOD ( .LIST_ELEM_EXP1 (.POINTER) )
376 1447 2     UNTIL (POINTER = .LIST_ELEM_FLINK (.POINTER)) EQL 0;
377 1448 1 END;

```

			0004 0000	.ENTRY	PAT\$SET_MOD_LST, Save R2	: 1401
		04	AC DD 0002	PUSHL	LEVEL	: 1443
0000000V	EF		01 FB 0005	CALLS	#1, PAT\$SET_MOD_LVL	:
	52 0000000G	EF	D0 000C	MOVL	PAT\$GL_HEAD_LST, POINTER	: 1444

PATMOD
V04-000

N 14
16-Sep-1984 00:34:55
14-Sep-1984 12:52:40

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATMOD.B32;1 Page 15 (6)

00000000V	EF	04	A2	DD	00013	1\$:	PUSHL	4(POINTER)	:	1446
	52		01	FB	00016		CALLS	#1, PAT\$SET_NEW MOD	:	
			62	D0	0001D		MOVL	(POINTER), POINTER	:	1447
			F1	12	00020		BNEQ	1\$:	
			04	00022			RET		:	1448

; Routine Size: 35 bytes, Routine Base: _PAT\$CODE + 006A

```

: 379 1449 1 |++
: 380 1450 1 | The following routines control interpretation and typeout modes.
: 381 1451 1 |
: 382 1452 1 |     Address interpretation and data typeout are controlled by four mode
: 383 1453 1 |     settings. They are 'DEFAULT_MODE', which is the mode set by
: 384 1454 1 |     PATCH initialization; 'USER_DEF_MODE', which is the mode set
: 385 1455 1 |     by user 'SET MODE' commands; 'OVERRIDE_MODE', which is
: 386 1456 1 |     the mode set by a single line override mode command; and
: 387 1457 1 |     'LOCAL_MODE', which is the mode set for a single variable.
: 388 1458 1 |     The mode used depends on the pointer into the block PAT$GB_DEF_MOD.
: 389 1459 1 |
: 390 1460 1 |     At initialization time, the four mode settings are initialized to
: 391 1461 1 |     the same values. When these values are changed by user commands, some
: 392 1462 1 |     propagation of mode values must move up and down through the different
: 393 1463 1 |     blocks so that the items displayed or interpreted will have valid
: 394 1464 1 |     and timely mode settings for both radix and length.
: 395 1465 1 |
: 396 1466 1 |     The routines that follow set the mode block pointers to their new
: 397 1467 1 |     value and propagate values as necessary.
: 398 1468 1 |
: 399 1469 1 |     The general strategy is as follows:
: 400 1470 1 |     DEFAULT_MODE settings are never changed.
: 401 1471 1 |     USER_DEF_MODE settings are never changed as a result of pointer jiggling.
: 402 1472 1 |     They are only changed explicitly in action routines.
: 403 1473 1 |     OVERRIDE_MODE settings are changed to reflect user-set defaults
: 404 1474 1 |     when they are the object of the pointer.
: 405 1475 1 |     LOCAL_MODE settings are copied from OVERRIDE_MODE settings when they
: 406 1476 1 |     are the object of the pointer.
: 407 1477 1 |     When the pointer is moving up the block, from LOCAL_MODE toward
: 408 1478 1 |     DEFAULT_MODE, no values are propagated.
: 409 1479 1 |     Setting of LOCAL_MODE and OVERRIDE_MODE to default settings at the end
: 410 1480 1 |     of a command line is the explicit responsibility of the
: 411 1481 1 |     END-OF-COMMAND action routine.
: 412 1482 1 | --

```

```

414 1483 1 GLOBAL ROUTINE PAT$SET_MOD_LVL (LEVEL) : NOVALUE =
415 1484 1
416 1485 1 |++
417 1486 1 | FUNCTIONAL DESCRIPTION
418 1487 1 |
419 1488 1 |     See description on page 2.
420 1489 1 |
421 1490 1 | CALLING SEQUENCE:
422 1491 1 |
423 1492 1 |     PAT$SET_MOD_LVL ( )
424 1493 1 |
425 1494 1 | INPUTS:
426 1495 1 |
427 1496 1 |     novalue
428 1497 1 |
429 1498 1 | IMPLICIT INPUTS:
430 1499 1 |
431 1500 1 |     none
432 1501 1 |
433 1502 1 | OUTPUTS:
434 1503 1 |
435 1504 1 |     none
436 1505 1 |
437 1506 1 | IMPLICIT OUTPUTS:
438 1507 1 |
439 1508 1 |     none
440 1509 1 |
441 1510 1 | ROUTINE VALUE:
442 1511 1 |
443 1512 1 |     novalue
444 1513 1 |
445 1514 1 | SIDE EFFECTS:
446 1515 1 |
447 1516 1 |     none
448 1517 1 |
449 1518 1 | --
450 1519 1 |
451 1520 2 BEGIN
452 1521 2
453 1522 2 LOCAL
454 1523 2     TEMP_LEVEL: REF VECTOR [, BYTE];
455 1524 2
456 1525 2 IF (PAT$GB_DEF_MOD [.LEVEL * MODE_LVL_SIZE] LEQA .PAT$GB_MOD_PTR) OR
457 1526 3     (.LEVEL EQL USER_DEF_MODE)
458 1527 2 THEN
459 1528 3     BEGIN
460 1529 3     PAT$GB_MOD_PTR = PAT$GB_DEF_MOD [.LEVEL * MODE_LVL_SIZE];
461 1530 3     RETURN
462 1531 2     END;
463 1532 2 PAT$GL_CONTEXT [OVERRIDE] = TRUE;
464 1533 2 SELECT .LEVEL OF
465 1534 2
466 1535 2     SET
467 1536 2
468 1537 2     [OVERRIDE MODE]:
469 1538 3     BEGIN
470 1539 3     TEMP_LEVEL = PAT$GB_DEF_MOD [USER_DEF_MODE * MODE_LVL_SIZE];

```

```

: 471 1540 3 PAT$GB_MOD_PTR = PAT$GB_DEF_MOD [OVERRIDE_MODE * MODE_LVL_SIZE];
: 472 1541 2 END;
: 473 1542 2
: 474 1543 2
: 475 1544 2 [LOCAL_MODE]:
: 476 1545 3 BEGIN
: 477 1546 4 IF (.PAT$GB_MOD_PTR EQLA PAT$GB_DEF_MOD [OVERRIDE_MODE * MODE_LVL_SIZE])
: 478 1547 3 THEN
: 479 1548 4 BEGIN
: 480 1549 4 TEMP_LEVEL = .PAT$GB_MOD_PTR;
: 481 1550 4 END
: 482 1551 3 ELSE
: 483 1552 4 BEGIN
: 484 1553 4 TEMP_LEVEL = PAT$GB_DEF_MOD [USER_DEF_MODE * MODE_LVL_SIZE];
: 485 1554 3 END;
: 486 1555 3 PAT$GB_MOD_PTR = PAT$GB_DEF_MOD [LOCAL_MODE * MODE_LVL_SIZE];
: 487 1556 2 END;
: 488 1557 2
: 489 1558 2 TES;
: 490 1559 2 PAT$GB_MOD_PTR [MODE_RADIX] = .TEMP_LEVEL [MODE_RADIX];
: 491 1560 2 PAT$GB_MOD_PTR [MODE_LENGTH] = .TEMP_LEVEL [MODE_LENGTH];
: 492 1561 2 PAT$GB_MOD_PTR [MODE_SYMBOLS] = .TEMP_LEVEL [MODE_SYMBOLS];
: 493 1562 2 PAT$GB_MOD_PTR [MODE_INSTRUC] = .TEMP_LEVEL [MODE_INSTRUC];
: 494 1563 2 PAT$GB_MOD_PTR [MODE_ASCII] = .TEMP_LEVEL [MODE_ASCII];
: 495 1564 2 PAT$GB_MOD_PTR [MODE_SCOPE] = .TEMP_LEVEL [MODE_SCOPE];
: 496 1565 2 PAT$GB_MOD_PTR [MODE_GLOBALS] = .TEMP_LEVEL [MODE_GLOBALS];
: 497 1566 1 END;

```

			000C 00000	.ENTRY	PAT\$SET MOD_LVL, Save R2,R3	: 1483
	53	00000000G	EF 9E 00002	MOVAB	PAT\$GB_DEF_MOD+7, R3	
	52	00000000G	EF 9E 00009	MOVAB	PAT\$GB_MOD_PTR, R2	
	51	04	AC D0 00010	MOVL	LEVEL, R1	: 1525
50	51		07 C5 00014	MULL3	#7, R1, R0	
	50	F9 A340	9E 00018	MOVAB	PAT\$GB_DEF_MOD[R0], R0	
	62		50 D1 0001D	CMPL	R0, PAT\$GB_MOD_PTR	
			05 18 00020	BLEQU	1\$	
	01		51 D1 00022	CMPL	R1, #1	: 1526
			04 12 00025	BNEQ	2\$	
	62		50 D0 00027 1\$:	MOVL	R0, PAT\$GB_MOD_PTR	: 1529
			04 0002A	RET		: 1528
	00000000G	EF	04 88 0002B 2\$:	BISB2	#4, PAT\$GL_CONTEXT+1	: 1532
	02		51 D1 00032	CMPL	R1, #2	: 1537
			07 12 00035	BNEQ	3\$	
	50		63 9E 00037	MOVAB	PAT\$GB_DEF_MOD+7, TEMP_LEVEL	: 1539
	62	07	A3 9E 0003A	MOVAB	PAT\$GB_DEF_MOD+14, PAT\$GB_MOD_PTR	: 1540
	03		51 D1 0003E 3\$:	CMPL	R1, #3	: 1544
			15 12 00041	BNEQ	6\$	
	51	07	A3 9E 00043	MOVAB	PAT\$GB_DEF_MOD+14, R1	: 1546
	51		62 D1 00047	CMPL	PAT\$GB_MOD_PTR, R1	
			05 12 0004A	BNEQ	4\$	
	50		62 D0 0004C	MOVL	PAT\$GB_MOD_PTR, TEMP_LEVEL	: 1549
			03 11 0004F	BRB	5\$: 1546
	50		63 9E 00051 4\$:	MOVAB	PAT\$GB_DEF_MOD+7, TEMP_LEVEL	: 1553

PATMOD
V04-000

	62	0E	A3	9E	00054	5\$:	MOVAB	PAT\$GB_DEF_MOD+21, PAT\$GB_MOD_PTR	:	1555
	51		62	D0	00058	6\$:	MOVL	PAT\$GB_MOD_PTR, R1	:	1559
	61		60	D0	0005B		MOVL	(TEMP_LEVEL), (R1)	:	
04	A1	04	A0	B0	0005E		MOVW	4(TEMP_LEVEL), 4(R1)	:	1563
06	A1	06	A0	90	00063		MOVB	6(TEMP_LEVEL), 6(R1)	:	1565
			04	00068			RET		:	1566

; Routine Size: 105 bytes, Routine Base: _PAT\$CODE + 008D

```

499 1567 1 GLOBAL ROUTINE PAT$SET_NEW_MOD ( SWITCH_VALUE ): NOVALUE =
500 1568 1
501 1569 1 |++
502 1570 1 | FUNCTIONAL DESCRIPTION:
503 1571 1 |
504 1572 1 |     This routine sets one mode according to the value
505 1573 1 |     specified.
506 1574 1 |
507 1575 1 | CALLING SEQUENCE:
508 1576 1 |
509 1577 1 |     PAT$SET_NEW_MOD ( )
510 1578 1 |
511 1579 1 | INPUTS:
512 1580 1 |
513 1581 1 |     SWITCH_VALUE     - Value to set
514 1582 1 |
515 1583 1 | IMPLICIT INPUTS:
516 1584 1 |
517 1585 1 |     The current level of modes being used; held in PAT$GB_MOD_PTR.
518 1586 1 |
519 1587 1 | OUTPUTS:
520 1588 1 |
521 1589 1 |     No value returned.  Modifies the value of the entry in whatever level
522 1590 1 |     we are in of the mode data structure.
523 1591 1 |
524 1592 1 | IMPLICIT OUTPUTS:
525 1593 1 |
526 1594 1 |     none
527 1595 1 |
528 1596 1 | ROUTINE VALUE:
529 1597 1 |
530 1598 1 |     novalue
531 1599 1 |
532 1600 1 | SIDE EFFECTS:
533 1601 1 |
534 1602 1 |     none
535 1603 1 |
536 1604 1 | --
537 1605 1 |
538 1606 2 BEGIN
539 1607 2
540 1608 2 LOCAL
541 1609 2     TABLE_PTR : REF VECTOR [, BYTE];
542 1610 2
543 1611 2 TABLE_PTR = MODE_TABLE;
544 1612 2 REPEAT
545 1613 3     BEGIN
546 1614 4     IF (.TABLE_PTR [KEYWORD_FIELD] EQL 0)
547 1615 3     THEN EXITLOOP;
548 1616 4     IF (.SWITCH_VALUE EQL .TABLE_PTR [KEYWORD_FIELD])
549 1617 3     THEN
550 1618 4         BEGIN
551 1619 4         PAT$GB_MOD_PTR [.TABLE_PTR [OFFSET_FIELD]] = .TABLE_PTR [VALUE_FIELD];
552 1620 5         IF (.PAT$G[_CONTEXT[MODE_BIT]])
553 1621 4         THEN
554 1622 4             PAT$WRITEFILE(.MODE_NAME_TBL[.TABLE_PTR[NAME_OFF_FIELD]],
555 1623 4                 CH$PTR(MODE_NAME_TBL[1], .TABLE_PTR[NAME_OFF_FIELD]),

```

```

: 556      1624  4
: 557      1625  4
: 558      1626  4
: 559      1627  3
: 560      1628  3
: 561      1629  2
: 562      1630  1 END;

```

```

EXITLOOP
END
ELSE
TABLE_PTR = TABLE_PTR [0] + MODE_ENTRY_LEN;
END;

```

PAT\$GL_COMRAB);

04	AC	62	53	00000000'	EF	9E	00002	.ENTRY	PAT\$SET NEW_MOD, Save R2,R3	: 1567
			52		63	9E	00009	MOVAB	MODE_TABLE, R3	: 1611
					62	95	0000C	MOVAB	MODE_TABLE, TABLE_PTR	: 1614
					3E	13	0000E	TSTB	(TABLE_PTR)	
			08		00	ED	00010	BEQL	3\$	
					31	12	00016	CMPZV	#0, #8, (TABLE_PTR), SWITCH_VALUE	: 1616
			50	01	A2	9A	00018	BNEQ	2\$	
			50	00000000G	EF	C0	0001C	MOVZBL	1(TABLE_PTR), R0	: 1619
			60	02	A2	90	00023	ADDL2	PAT\$GB_MOD_PTR, R0	
			20	00000000G	EF	E9	00027	MOVAB	2(TABLE_PTR), (R0)	
				00000000G	EF	9F	0002E	BLBC	PAT\$GL_CONTEXT, 3\$: 1620
			50	03	A2	9A	00034	PUSHAB	PAT\$GL_COMRAB	: 1623
				CB A340	9F	00038		MOVZBL	3(TABLE_PTR), R0	
			7E	CA A340	9A	0003C		PUSHAB	MODE_NAME_TBL+1[R0]	
		00000000G	EF		03	FB	00041	MOVZBL	MODE_NAME_TBL[R0], -(SP)	
						04	00048	CALLS	#3, PAT\$WRITEFILE	
			52		04	C0	00049	RET		: 1618
					BE	11	0004C	ADDL2	#4, TABLE_PTR	: 1628
						04	0004E	BRB	1\$: 1611
								RET		: 1630

; Routine Size: 79 bytes, Routine Base: _PAT\$CODE + 00F6

```

564 1631 1 GLOBAL ROUTINE PAT$SHOW_DEFAL : NOVALUE =
565 1632 1
566 1633 1 :++
567 1634 1 : FUNCTIONAL DESCRIPTION:
568 1635 1
569 1636 1 :     Action routine to show default modes.
570 1637 1
571 1638 1 : CALLING SEQUENCE:
572 1639 1
573 1640 1 :     PAT$SHOW_DEFAL ( )
574 1641 1
575 1642 1 : INPUTS:
576 1643 1
577 1644 1 :     none
578 1645 1
579 1646 1 : IMPLICIT INPUTS:
580 1647 1
581 1648 1 :     The default modes.
582 1649 1
583 1650 1 : OUTPUTS:
584 1651 1
585 1652 1 :     none
586 1653 1
587 1654 1 : IMPLICIT OUTPUTS:
588 1655 1
589 1656 1 :     none
590 1657 1
591 1658 1 : ROUTINE VALUE:
592 1659 1
593 1660 1 :     novalue
594 1661 1
595 1662 1 : SIDE EFFECTS:
596 1663 1
597 1664 1 :     The modes are displayed on the output device.
598 1665 1
599 1666 1 :--
600 1667 1
601 1668 2 BEGIN
602 1669 2
603 1670 2 LOCAL
604 1671 2     LEVEL_PTR : REF VECTOR [, BYTE];           ! Current mode level
605 1672 2
606 1673 2 'EVEL_PTR = .PAT$GB_MOD_PTR;
607 1674 2
608 1675 2 :++
609 1676 2 : A temporary fix to hide the symbolic/non-symbolic mode setting is here so that
610 1677 2 : user is not confused by the reporting of a mode that has no effect yet.
611 1678 2 :--
612 P 1679 2 $FAO TT_OUT ( 'modes:!ACsymbols,!ACinstruction,!ACascii,!ACscope,!ACglobals, !AC !AC' ,
613 P 1680 2     YES_NO ( .LEVEL_PTR [MODE_SYMBOLS] ),
614 P 1681 2     YES_NO ( .LEVEL_PTR [MODE_INSTRUC] ),
615 P 1682 2     YES_NO ( .LEVEL_PTR [MODE_ASCII] ),
616 P 1683 2     YES_NO ( .LEVEL_PTR [MODE_SCOPE] ),
617 P 1684 2     YES_NO ( .LEVEL_PTR [MODE_GLOBALS] ),
618 P 1685 2
619 P 1686 2     (SELECT .LEVEL_PTR[ MODE_RADIX ] OF
620 P 1687 2

```

P
V
.....


```

: 621 P 1688 2
: 622 P P 1689 2
: 623 P P 1690 2
: 624 P P 1691 2
: 625 P P 1692 2
: 626 P P 1693 2
: 627 P P 1694 2
: 628 P P 1695 2
: 629 P P 1696 2
: 630 P P 1697 2
: 631 P P 1698 2
: 632 P P 1699 2
: 633 P P 1700 2
: 634 P P 1701 2
: 635 P P 1702 2
: 636 P P 1703 2
: 637 P P 1704 2
: 638 P P 1705 2
: 639 P P 1706 2
: 640 P P 1707 2
: 641 P P 1708 2
: 642 P P 1709 2
: 643 P P 1710 2
: 644 P P 1711 2
: 645 P P 1712 2
: 646 P P 1713 2
: 647 P 1714 2
: 648 1715 2
: 649 1716 2
: 650 1717 1 END;

```

```

SET
[OCTAL_RADIX]:
  CS_OCTAL;
[HEX_RADIX]:
  CS_HEXADecimal;
[DECIMAL_RADIX]:
  CS_DECIMAL;
TES),
(SELECT .LEVEL_PTR[ MODE_LENGTH ] OF
SET
[BYTE_LENGTH]:
  CS_BYTE;
[WORD_LENGTH]:
  CS_WORD;
[LONG_LENGTH]:
  CS_LONGWORD;
TES)
);

```

```

6C 6F 62 6D 79 73 43 41 21 3A 73 65 64 6F 45 000B3 P.AAK:
6F 69 74 63 75 72 74 73 6E 69 43 41 21 2C 6D 000B4
6C 67 43 41 21 2C 65 70 6F 63 61 43 41 21 2C 6E 000C3
43 41 21 20 43 41 21 20 2C 73 6C 61 62 6F 000D2
43 41 21 20 43 41 21 20 2C 73 6C 61 62 6F 000DC
43 41 21 20 43 41 21 20 2C 73 6C 61 62 6F 000EB

```

```

.PSECT _PAT$PLIT,NOWRT,NOEXE,0
.BYTE 69
.ASCII \modes:!ACsymbols,!ACinstruction,!ACascii\
.ASCII \,!ACscope,!ACglobals,!AC !AC\

```

```

54 00000000' EF 9E 00002
50 00000000G EF D0 00009
53 01 A0 9A 00010
51 01 CE 00014
01 53 91 00017
07 12 0001A
52 E8 A4 9E 0001C
51 52 D0 00020
02 53 91 00023
07 12 00026

```

```

.PSECT _PAT$CODE,NOWRT,2
.ENTRY PAT$SHOW DEFAL, Save R2,R3,R4 : 1631
MOVAB CS_BOOLEAN_ON, R4
MOVL PAT$GB_MOD_PTR, LEVEL_PTR : 1673
MOVZBL 1(LEVEL_PTR), R3 : 1715
MNEGL #1, R1
CMPB R3, #1
BNEQ 1$
MOVAB CS_BYTE, R2
MOVL R2, R1
CMPB R3, #2
BNEQ 2$

```

```

52      F0      A4      9E      00028      MOVAB      CS_WORD, R2
51      52      D0      0002C      MOVL       R2, R1
04      53      91      0002F      2$:      CMPB      R3, #4
          07      12      00032      BNEQ      3$
52      F8      A4      9E      00034      MOVAB      CS_LONGWORD, R2
51      52      D0      00038      MOVL       R2, R1
          51      DD      0003B      3$:      PUSHL     R1
51      01      CE      0003D      MNEGL     #1, R1
08      60      91      00040      CMPB      (LEVEL_PTR), #8
          07      12      00043      BNEQ      4$
52      CC      A4      9E      00045      MOVAB      CS_OCTAL, R2
51      52      D0      00049      MOVL       R2, R1
10      60      91      0004C      4$:      CMPB      (LEVEL_PTR), #16
          07      12      0004F      BNEQ      5$
52      DC      A4      9E      00051      MOVAB      CS_HEXADECIMAL, R2
51      52      D0      00055      MOVL       R2, R1
0A      60      91      00058      5$:      CMPB      (LEVEL_PTR), #10
          07      12      0005B      BNEQ      6$
52      D4      A4      9E      0005D      MOVAB      CS_DECIMAL, R2
51      52      D0      00061      MOVL       R2, R1
          51      DD      00064      6$:      PUSHL     R1
05      06      A0      E9      00066      BLBC      6(LEVEL_PTR), 7$
51      64      9E      0006A      MOVAB     CS_BOOLEAN_ON, R1
          04      11      0006D      BRB      8$
51      04      A4      9E      0006F      7$:      MOVAB     CS_BOOLEAN_OFF, R1
          51      DD      00073      8$:      PUSHL     R1
05      05      A0      E9      00075      BLBC      5(LEVEL_PTR), 9$
51      64      9E      00079      MOVAB     CS_BOOLEAN_ON, R1
          04      11      0007C      BRB      10$
51      04      A4      9E      0007E      9$:      MOVAB     CS_BOOLEAN_OFF, R1
          51      DD      00082      10$:     PUSHL     R1
05      04      A0      E9      00084      BLBC      4(LEVEL_PTR), 11$
51      64      9E      00088      MOVAB     CS_BOOLEAN_ON, R1
          04      11      0008B      BRB      12$
51      04      A4      9E      0008D      11$:     MOVAB     CS_BOOLEAN_OFF, R1
          51      DD      00091      12$:     PUSHL     R1
05      03      A0      E9      00093      BLBC      3(LEVEL_PTR), 13$
51      64      9E      00097      MOVAB     CS_BOOLEAN_ON, R1
          04      11      0009A      BRB      14$
51      04      A4      9E      0009C      13$:     MOVAB     CS_BOOLEAN_OFF, R1
          51      DD      000A0      14$:     PUSHL     R1
05      02      A0      E9      000A2      BLBC      2(LEVEL_PTR), 15$
50      64      9E      000A6      MOVAB     CS_BOOLEAN_ON, R0
          04      11      000A9      BRB      16$
50      04      A4      9E      000AB      15$:     MOVAB     CS_BOOLEAN_OFF, R0
          50      DD      000AF      16$:     PUSHL     R0
          7F      A4      9F      000B1      PUSHAB   P.AAK
0000000G EF      08      FB      000B4      CALLS    #8, PAT$FAO_OUT
          04      000BB      RET

```

; Routine Size: 188 bytes, Routine Base: _PAT\$CODE + 0145

PATMOD
V04-000

K 15
16-Sep-1984 00:34:55
14-Sep-1984 12:52:40

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[PATCH.SRC]PATMOD.B32;1 (11) Page 25

: 652 1718 1 END
: 653 1719 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
PAT\$PLIT	249	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(0)
PAT\$CODE	513	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
ABS	0	NOVEC,NOWRT,NORD ,NOEXE,NOSHR, LCL, ABS, CON,NOPIC,ALIGN(0)

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS\$:PATMOD/OBJ=OBJ\$:PATMOD MSRCS\$:PATMOD/UPDATE=(ENH\$:PATMOD)

: Size: 513 code + 249 data bytes
: Run Time: 00:17.0
: Elapsed Time: 00:53.5
: Lines/CPU Min: 6070
: Lexemes/CPU-Min: 31974
: Memory Used: 122 pages
: Compilation Complete

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 pattern. Each window shows a different VAX/VMS command or system output. Several windows are clearly legible and contain text such as 'PATMAC LIS', 'PATMAT LIS', 'PATPAR LIS', 'PATLST LIS', 'PATIO LIS', 'PATLEX LIS', 'PATMOO LIS', and 'PATMSG LIS'. Other windows show various system prompts, error messages, and data listings. The overall appearance is that of a dense collection of system logs or command outputs.