

PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH

```
PPPPPPPP      AAAAAA      TTTTTTTTTT      BBBB8888      AAAAAA      SSSSSSSS
PPPPPPPP      AAAAAA      TTTTTTTTTT      88888888      AAAAAA      SSSSSSSS
PP      PP      AA      AA      TT      BB      BB      AA      AA      SS
PP      PP      AA      AA      TT      BB      BB      AA      AA      SS
PP      PP      AA      AA      TT      BB      BB      AA      AA      SS
PP      PP      AA      AA      TT      BB      BB      AA      AA      SS
PPPPPPPP      AA      AA      TT      88888888      AA      AA      SSSSSS
PPPPPPPP      AA      AA      TT      88888888      AA      AA      SSSSSS
PP      AAAAAAAAAA      TT      BB      BB      AAAAAAAAAA      SS
PP      AAAAAAAAAA      TT      BB      BB      AAAAAAAAAA      SS
PP      AA      AA      TT      BB      BB      AA      AA      SS
PP      AA      AA      TT      BB      BB      AA      AA      SS
PP      AA      AA      TT      88888888      AA      AA      SSSSSSSS
PP      AA      AA      TT      88888888      AA      AA      SSSSSSSS

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```



```
L 0001 0 MODULE PATBAS (%IF %VARIANT EQL 1
0002 0 %THEN
0003 0 ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
0004 0 NONEXTERNAL = LONG_RELATIVE),
0005 0 %FI
0006 0 IDENT = 'V04-000') =
0007 1 BEGIN
0008 1
0009 1 *****
0010 1 *
0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0013 1 * ALL RIGHTS RESERVED.
0014 1 *
0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0020 1 * TRANSFERRED.
0021 1 *
0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0024 1 * CORPORATION.
0025 1 *
0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0028 1 *
0029 1 *
0030 1 *****
0031 1
0032 1 FACILITY: PATCH
0033 1
0034 1 FUNCTIONAL DESCRIPTION:
0035 1 INITIALIZATION ROUTINES FOR PATCH.
0036 1
0037 1 History:
0038 1 Author:
0039 1 Carol Peters, 12 Aug 1976: Version 01
0040 1
0041 1 MODIFIED BY:
0042 1
0043 1 V03-003 MCN0142 Maria del C. Nasr 18-Jan-1982
0044 1 Change SIGNAL to SIGNAL_STOP. Also, define descriptors
0045 1 for logical device names, since their FABs do not
0046 1 have NAM blocks, and we cannot use the routine
0047 1 GETFILDSC to get it.
0048 1
0049 1 V03-002 MTR0007 Mike Rhodes 14-Jun-1982
0050 1 Use shared system messages. Affected modules include:
0051 1 DYNMEM.B32, PATBAS.B32, PATCMD.B32, PATIHD.B32, PATINT.B32,
0052 1 PATIO.B32, PATMAI.B32, PATMSG.MSG, PATWRT.B32, and PATSPA.B32.
0053 1
0054 1 The shared messages are defined by DYNMEM.B32's invocation of
0055 1 SHRMSG.REQ and we simply link against these symbols. They are
0056 1 declared as external literals below.
0057 1
```



```
58 0058 1 |
59 0059 1 | V03-001 MTR0006 Mike Rhodes 07-JUN-1982
60 0060 1 | Add statement to clear the I_HAT_SEEN bit for end
61 0061 1 | of command processing in routine PAT$SET_CONTEXT.
62 0062 1 |
63 0063 1 | V02-001 PCG0001 Peter George 02-FEB-1981
64 0064 1 | Add require statement for LIB$:PATDEF.REQ
65 0065 1 |
66 0066 1 | MODIFICATIONS:
67 0067 1 |
68 0068 1 | NO DATE PROGRAMMER PURPOSE
69 0069 1 | -- ----
70 0070 1 |
71 0071 1 | 00 17-OCT-77 K.D. MORSE ADAPT VERSION 54 FOR PATCH.
72 0072 1 | 01 4-JAN-78 K.D. MORSE NO CHANGES FOR 55-82.
73 0073 1 | 02 24-JAN-78 K.D. MORSE NO CHANGES FOR 83-84.
74 0074 1 | 03 28-FEB-78 K.D. MORSE NO CHANGES FOR 85-86.
75 0075 1 | 04 01-MAR-78 K.D. MORSE ADD CONTEXT BIT RESET FOR /ALL.
76 0076 1 | 05 01-MAR-78 K.D. MORSE NO CHANGES FOR 87.
77 0077 1 | 06 24-MAR-78 K.D. MORSE NO CHANGES FOR 88-91.
78 0078 1 | 07 04-APR-78 K.D. MORSE NO CHANGES FOR 92-93.
79 0079 1 | 08 14-APR-78 K.D. MORSE NO CHANGES FOR 94.
80 0080 1 | 09 18-APR-78 K.D. MORSE ADD LITERAL BIT AND OPN COM_FILE.
81 0081 1 | 10 25-APR-78 K.D. MORSE CONVERT TO NATIVE COMPILER.
82 0082 1 | 11 18-MAY-78 K.D. MORSE NO CHANGES FOR VERS 95.
83 0083 1 | 12 18-MAY-78 K.D. MORSE NO CHANGES FOR VERS 96-97.
84 0084 1 | 13 06-JUN-78 K.D. MORSE INITIALIZE LISTHEADS FOR
85 0085 1 | OLD AND NEW LABELS, PAT$GL_OLDLABLS
86 0086 1 | AND PAT$GL_NEWLABLS.
87 0087 1 | 14 13-JUN-78 K.D. MORSE ADD FAO COUNTS TO SIGNALS.
88 0088 1 | 15 16-JUN-78 K.D. MORSE INITIALIZE APPENDED COMMAND
89 0089 1 | TEXT BUFFERS.
90 0090 1 | 16 19-JUN-78 K.D. MORSE NO CHANGES FOR VERS 98-99.
91 0091 1 | 17 28-JUN-78 K.D. MORSE NO CHANGES FOR VERS 100-102.
92 0092 1 |
93 0093 1 | --
```



```

95      0094 1 FORWARD ROUTINE
96      0095 1          PAT$SET_CONTEXT: NOVALUE,
97      0096 1          PAT$INIT: NOVALUE;
98      0097 1
99      0098 1 LIBRARY 'SYSS$LIBRARY:STARLET.L32';
100     0099 1 REQUIRE 'SRC$:VXSMAC.REQ';
101     0164 1 REQUIRE 'SRC$:PATPCT.REQ';
102     0204 1 REQUIRE 'SRC$:PATGEN.REQ';
103     0426 1 REQUIRE 'SRC$:BSTRUC.REQ';
104     0502 1 REQUIRE 'SRC$:DLLNAM.REQ';
105     0560 1 REQUIRE 'LIB$:PATDEF.REQ';
106     0614 1 REQUIRE 'LIB$:PATMSG.REQ';
107     0788 1 REQUIRE 'SRC$:PREFIX.REQ';
108     0976 1 REQUIRE 'SRC$:PATPRE.REQ';
109     1139 1
110     1140 1 EXTERNAL ROUTINE
111     1141 1          PAT$FREEZ,
112     1142 1          PAT$FREEINIT,
113     1143 1          PAT$RESET_DEF;
114     1144 1
115     1145 1 EXTERNAL
116     1146 1          PAT$GB_ECOLVL: BYTE,
117     1147 1          PAT$GB_ERRNAME,
118     1148 1          PAT$GB_INPNAME,
119     1149 1          PAT$GB_OUTNAME,
120     1150 1          PAT$GL_CONTEXT: BITVECTOR,
121     1151 1          PAT$GL_INPFAB: BLOCK [, BYTE],
122     1152 1          PAT$GL_INPRAB: BLOCK [, BYTE],
123     1153 1          PAT$GL_OUTFAB: BLOCK [, BYTE],
124     1154 1          PAT$GL_OUTRAB: BLOCK [, BYTE],
125     1155 1          PAT$GL_ERRFAB: BLOCK [, BYTE],
126     1156 1          PAT$GL_ERRRAB: BLOCK [, BYTE],
127     1157 1          PAT$GL_SYMHEAD,
128     1158 1          PAT$GL_SYMTBPTR,
129     1159 1          PAT$GL_OLDLABLS,
130     1160 1          PAT$GL_NEWLABLS,
131     1161 1          PAT$GL_RLCLABLS,
132     1162 1          PAT$GL_ERRCODE,
133     1163 1          PAT$GL_TXTFREE,
134     1164 1          PAT$GL_TXTLHD : REF BLOCK[,BYTE],
135     1165 1          PAT$GL_TXTTAIL;
136     1166 1
137     1167 1 EXTERNAL LITERAL
138     1168 1          PAT$K_ERRNAMLNG,
139     1169 1          PAT$K_OUTNAMLNG,
140     1170 1          PAT$K_INPNAMLNG,
141     1171 1
142     1172 1          Define shared message references. (resolved @ link time)
143     1173 1
144     1174 1          PAT$_CLOSEIN,
145     1175 1          PAT$_CLOSEOUT,
146     1176 1          PAT$_OPENIN,
147     1177 1          PAT$_OPENOUT,
148     1178 1          PAT$_READERR,
149     1179 1          PAT$_SYSERROR,
150     1180 1          PAT$_WRITEERR;

! Routine to initialize context bits
! Routine to initialize PATCH

! Defines PSECTS

! Defines literals

! Routine to allocate a block of free storage
! Routine to initialize free storage
! Sets up default mode settings

! ECO level for current patch
! Logical error channel name
! Logical input channel name
! Logical output channel name
! PATCH command context bits
! FAB for command input stream
! RAB for command input stream
! FAB for command output stream
! RAB for command output stream
! FAB for error messages
! RAB for error messages
! Listhead for user-defined symbol table
! Pointer to default symbol table
! Listhead for old contents instruction labe
! Listhead for new contents un-relocated ins
! Listhead for newcontents relocated instruc
! Error code
! Pointer to next free byte in last text buf
! Pointer to first text buffer
! Pointer to last text buffer

! Length of error channel name
! Length of output channel name
! Length of input channel name

! Error closing input file.
! Error closing output file.
! Error opening input file.
! Error opening output file.
! Error reading from file.
! System Service error.
! Error writing to file.
```

```
1181 1 GLOBAL ROUTINE PAT$SET_CONTEXT : NOVALUE =
1182 1
1183 1 ++
1184 1 FUNCTIONAL DESCRIPTION:
1185 1
1186 1     Initializes context bits that are necessary for command
1187 1     processing. These bits are valid only during the processing
1188 1     of a single command. They are all reset after each command.
1189 1
1190 1 CALLING SEQUENCE:
1191 1
1192 1     PAT$SET_CONTEXT ( )
1193 1
1194 1 INPUTS:
1195 1
1196 1     none
1197 1
1198 1 IMPLICIT INPUTS:
1199 1
1200 1     The names of the context bits that are to be turned off.
1201 1
1202 1 OUTPUTS:
1203 1
1204 1     none
1205 1
1206 1 IMPLICIT OUTPUTS:
1207 1
1208 1     none
1209 1
1210 1 ROUTINE VALUE:
1211 1
1212 1     NOVALUE
1213 1
1214 1 SIDE EFFECTS:
1215 1
1216 1     The context bits are set to FALSE.
1217 1 --
1218 1
1219 2 BEGIN
1220 2
1221 2 PAT$GL_CONTEXT [OVERRIDE] = FALSE;
1222 2 PAT$GL_CONTEXT [MODE_BIT] = FALSE;
1223 2 PAT$GL_CONTEXT [SET_NOT_ECO] = FALSE;
1224 2 PAT$GL_CONTEXT [EXAMINE_BIT] = FALSE;
1225 2 PAT$GL_CONTEXT [VERIFY_BIT] = FALSE;
1226 2 PAT$GL_CONTEXT [INSERT_BIT] = FALSE;
1227 2 PAT$GL_CONTEXT [DELETE_BIT] = FALSE;
1228 2 PAT$GL_CONTEXT [SET_ECO] = FALSE;
1229 2 PAT$GL_CONTEXT [SCOPE_BIT] = FALSE;
1230 2 PAT$GL_CONTEXT [MODULE_BIT] = FALSE;
1231 2 PAT$GL_CONTEXT [ALIGN_LONG] = FALSE;
1232 2 PAT$GL_CONTEXT [ALIGN_WORD] = FALSE;
1233 2 PAT$GL_CONTEXT [ALIGN_QUAD] = FALSE;
1234 2 PAT$GL_CONTEXT [ALIGN_BYTE] = FALSE;
1235 2 PAT$GL_CONTEXT [ALIGN_PAGE] = FALSE;
1236 2 PAT$GL_CONTEXT [PAT_AREA_BIT] = FALSE;
1237 2 PAT$GL_CONTEXT [INST_SUBST] = FALSE;
```


PATBAS
V04-000

F 3
16-Sep-1984 00:08:26
14-Sep-1984 12:52:28

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATBAS.B32;1 Page 5 (3)

```
: 209      1238 2 PAT$GL_CONTEXT [LITERAL_BIT] = FALSE;  
: 210      1239 2 PAT$GL_CONTEXT [OPN_COM_FILE] = FALSE;  
: 211      1240 2 PAT$GL_CONTEXT [I_HAT_SEEN] = FALSE;  
: 212      1241 1 END;
```

.TITLE PATBAS
.IDENT \V04-000\

ISE\$C_SIZE== 20
TXT\$C_SIZE== 4
PAL\$C_SIZE== 16
ASD\$C_SIZE== 9
FWR\$C_SIZE== 24

.EXTRN PAT\$FREEZ, PAT\$FREEINIT
.EXTRN PAT\$RESET DEF, PAT\$GB_ECOLVL
.EXTRN PAT\$GB_ERRNAME, PAT\$GB_INPNAME
.EXTRN PAT\$GB_OUTNAME, PAT\$GL_CONTEXT
.EXTRN PAT\$GL_INPFAB, PAT\$GL_INPRAB
.EXTRN PAT\$GL_OUTFAB, PAT\$GL_OUTRAB
.EXTRN PAT\$GL_ERRFAB, PAT\$GL_ERRRAB
.EXTRN PAT\$GL_SYMHED, PAT\$GL_SYMTBPTR
.EXTRN PAT\$GL_OLDLABLS
.EXTRN PAT\$GL_NEWLABLS
.EXTRN PAT\$GL_RLCLABLS
.EXTRN PAT\$GL_ERRCODE, PAT\$GL_TXTFREE
.EXTRN PAT\$GL_TXTLHD, PAT\$GL_TXTTAIL
.EXTRN PAT\$K_ERRNAMLNG
.EXTRN PAT\$K_OUTNAMLNG
.EXTRN PAT\$K_INPNAMLNG
.EXTRN PAT\$CLOSEIN, PAT\$CLOSEOUT
.EXTRN PAT\$OPENIN, PAT\$OPENOUT
.EXTRN PAT\$READERR, PAT\$_SYSERROR
.EXTRN PAT\$WRITEERR
.WEAK ACCESS_CHECK

.PSECT _PAT\$CODE, NOWRT, 2

.ENTRY PAT\$SET_CONTEXT, Save nothing
BICL2 #134022655, PAT\$GL_CONTEXT
RET

: 1181
: 1240
: 1241

00000000G EF 07FD05FF 8F 0000 00000
CA 00002
04 0000D

; Routine Size: 14 bytes, Routine Base: _PAT\$CODE + 0000

```
214 1242 1 GLOBAL ROUTINE PAT$INIT: NOVALUE =
215 1243 1
216 1244 1 ++
217 1245 1 FUNCTIONAL DESCRIPTION:
218 1246 1
219 1247 1     Sets up local context for PATCH.  If the logical devices SYSS$INPUT,
220 1248 1     SYSS$OUTPUT, or SYSS$ERROR cannot be set up, the routine SIGNALS an error;
221 1249 1     otherwise it returns.  Initializes symbol tables and PATCH command
222 1250 1     text buffers.
223 1251 1
224 1252 1 CALLING SEQUENCE:
225 1253 1
226 1254 1     PAT$INIT()
227 1255 1
228 1256 1 INPUTS:
229 1257 1
230 1258 1     none
231 1259 1
232 1260 1 IMPLICIT INPUTS:
233 1261 1
234 1262 1     The FABs and RABs for input and output.
235 1263 1     Addresses of context blocks that are to be set by the initialization.
236 1264 1     PAT$GL_TXTFREE - Pointer to first free byte of text buffer in last block
237 1265 1     PAT$GL_TXTLHD - Pointer to first block of text buffer
238 1266 1     PAT$GL_TXTTAIL - Pointer to last block of text buffer
239 1267 1
240 1268 1 OUTPUTS:
241 1269 1
242 1270 1     none
243 1271 1
244 1272 1 IMPLICIT OUTPUTS:
245 1273 1
246 1274 1     The communication channels are open.
247 1275 1
248 1276 1 ROUTINE VALUE:
249 1277 1
250 1278 1     NOVALUE
251 1279 1
252 1280 1
253 1281 1 SIDE EFFECTS:
254 1282 1
255 1283 1     Context, input and output modes, etc. are set up.
256 1284 1     The symbol table listheads are set up and the first block for
257 1285 1     appended patch command text is allocated.
258 1286 1 --
259 1287 1
260 1288 2 BEGIN
261 1289 2
262 1290 2 ! Define local descriptors with file names for error signaling.
263 1291 2 !
264 1292 2
265 1293 2 LOCAL
266 1294 2     PAT$ERRNAM_DESC : VECTOR [2],
267 1295 2     PAT$INPNAM_DESC : VECTOR [2],
268 1296 2     PAT$OUTNAM_DESC : VECTOR [2];
269 1297 2
270 1298 2 PAT$ERRNAM_DESC [0] = PAT$K_ERRNAMLNG;
```



```
271 1299 2 PAT$ERRNAM_DESC [1] = PAT$GB_ERRNAME;
272 1300 2 PAT$INPNAM_DESC [0] = PAT$K_INPNAMLNG;
273 1301 2 PAT$INPNAM_DESC [1] = PAT$GB_INPNAME;
274 1302 2 PAT$OUTNAM_DESC [0] = PAT$K_OUTNAMLNG;
275 1303 2 PAT$OUTNAM_DESC [1] = PAT$GB_OUTNAME;
276 1304 2
277 1305 2 !++
278 1306 2 ! If the OPENS and CONNECTs cannot be done successfully for logical devices
279 1307 2 ! SYS$ERROR, SYS$INPUT, and SYS$OUTPUT, then SIGNAL a fatal error.
280 1308 2 ! This causes a return to the command line interpreter.
281 1309 2 !--
282 1310 2
283 1311 2 PAT$GL_ERRCODE = $OPEN (FAB = PAT$GL_OUTFAB);
284 1312 2 IF NOT .PAT$GL_ERRCODE
285 1313 2 THEN
286 1314 2     SIGNAL_STOP (PAT$ OPENOUT, 1, PAT$OUTNAM_DESC,
287 1315 2     .PAT$GL_OUTFAB[FAB$L_STS], .PAT$GL_OUTFAB[FAB$L_STV]);
288 1316 2
289 1317 2 PAT$GL_ERRCODE = $CONNECT (RAB = PAT$GL_OUTRAB);
290 1318 2 IF NOT .PAT$GL_ERRCODE
291 1319 2 THEN
292 1320 2     SIGNAL_STOP (PAT$ OPENOUT, 1, PAT$OUTNAM_DESC,
293 1321 2     .PAT$GL_OUTRAB[RAB$L_STS], .PAT$GL_OUTRAB[RAB$L_STV]);
294 1322 2
295 1323 2 PAT$GL_ERRCODE = $OPEN (FAB = PAT$GL_ERRFAB);
296 1324 2 IF NOT .PAT$GL_ERRCODE
297 1325 2 THEN
298 1326 2     SIGNAL_STOP (PAT$ OPENOUT, 1, PAT$ERRNAM_DESC,
299 1327 2     .PAT$GL_ERRFAB[FAB$L_STS], .PAT$GL_ERRFAB[FAB$L_STV]);
300 1328 2
301 1329 2 PAT$GL_ERRCODE = $CONNECT (RAB = PAT$GL_ERRRAB);
302 1330 2 IF NOT .PAT$GL_ERRCODE
303 1331 2 THEN
304 1332 2     SIGNAL_STOP (PAT$ OPENOUT, 1, PAT$ERRNAM_DESC,
305 1333 2     .PAT$GL_ERRRAB[RAB$L_STS], .PAT$GL_ERRRAB[RAB$L_STV]);
306 1334 2
307 1335 2 PAT$GL_ERRCODE = $OPEN (FAB = PAT$GL_INPFAB);
308 1336 2 IF NOT .PAT$GL_ERRCODE
309 1337 2 THEN
310 1338 2     SIGNAL_STOP (PAT$ OPENIN, 1, PAT$INPNAM_DESC,
311 1339 2     .PAT$GL_INPFAB[FAB$L_STS], .PAT$GL_INPFAB[FAB$L_STV]);
312 1340 2
313 1341 2 PAT$GL_ERRCODE = $CONNECT (RAB = PAT$GL_INPRAB);
314 1342 2 IF NOT .PAT$GL_ERRCODE
315 1343 2 THEN
316 1344 2     SIGNAL_STOP (PAT$ OPENIN, 1, PAT$INPNAM_DESC,
317 1345 2     .PAT$GL_INPRAB[RAB$L_STS], .PAT$GL_INPRAB[RAB$L_STV]);
318 1346 2
319 1347 2
320 1348 2 !++
321 1349 2 ! Initialize an area of free storage, and give the name of the
322 1350 2 ! PATCH error reporting mechanism as the name of the routine that
323 1351 2 ! the free storage package can call to report errors.
324 1352 2 !--
325 1353 2 PAT$FREEINIT ();
326 1354 2
327 1355 2 !++
```

```

328 1356 2 | Set the default mode settings.
329 1357 2 |--
330 1358 2 PAT$RESET_DEF ();
331 1359 2 |--
332 1360 2 |++
333 1361 2 | Set all the single command context bits to FALSE. These bits
334 1362 2 | refer to context that is valid only during a single command,
335 1363 2 | not across multiple commands.
336 1364 2 |--
337 1365 2 PAT$SET_CONTEXT ();
338 1366 2 |--
339 1367 2 |++
340 1368 2 | Initialize the symbol chain by allocating storage for the header
341 1369 2 | link and setting the pointers to point to the link.
342 1370 2 |--
343 1371 2 PAT$GSL_SYMHEAD = PAT$FREEZ (OVERHEAD_SYM);
344 1372 2 DLL_RLINK (.PAT$GSL_SYMHEAD) = .PAT$GSL_SYMHEAD;
345 1373 2 DLL_LLINK (.PAT$GSL_SYMHEAD) = .PAT$GSL_SYMHEAD;
346 1374 2 SYM_VALUE (.PAT$GSL_SYMHEAD) = 0;
347 1375 2 SYM_CSTRING (.PAT$GSL_SYMHEAD) = 0;
348 1376 2 PAT$GSL_SYMTBPTR = .PAT$GSL_SYMHEAD;
349 1377 2 |--
350 1378 2 |++
351 1379 2 | Initialize the old contents label symbol chain by allocating storage for
352 1380 2 | the header link and setting the pointers to point to the link.
353 1381 2 |--
354 1382 2 PAT$GSL_OLDLABLS = PAT$FREEZ (OVERHEAD_SYM);
355 1383 2 DLL_RLINK (.PAT$GSL_OLDLABLS) = .PAT$GSL_OLDLABLS;
356 1384 2 DLL_LLINK (.PAT$GSL_OLDLABLS) = .PAT$GSL_OLDLABLS;
357 1385 2 SYM_VALUE (.PAT$GSL_OLDLABLS) = 0;
358 1386 2 SYM_CSTRING (.PAT$GSL_OLDLABLS) = 0;
359 1387 2 |--
360 1388 2 |++
361 1389 2 | Initialize the new contents un-relocated label symbol chain by allocating
362 1390 2 | storage for the header link and setting the pointers to point to the link.
363 1391 2 |--
364 1392 2 PAT$GSL_NEWLABLS = PAT$FREEZ (OVERHEAD_SYM);
365 1393 2 DLL_RLINK (.PAT$GSL_NEWLABLS) = .PAT$GSL_NEWLABLS;
366 1394 2 DLL_LLINK (.PAT$GSL_NEWLABLS) = .PAT$GSL_NEWLABLS;
367 1395 2 SYM_VALUE (.PAT$GSL_NEWLABLS) = 0;
368 1396 2 SYM_CSTRING (.PAT$GSL_NEWLABLS) = 0;
369 1397 2 |--
370 1398 2 |++
371 1399 2 | Initialize the new contents relocated label symbol chain by allocating
372 1400 2 | storage for the header link and setting the pointers to point to the link.
373 1401 2 |--
374 1402 2 PAT$GSL_RLCLABLS = PAT$FREEZ (OVERHEAD_SYM);
375 1403 2 DLL_RLINK (.PAT$GSL_RLCLABLS) = .PAT$GSL_RLCLABLS;
376 1404 2 DLL_LLINK (.PAT$GSL_RLCLABLS) = .PAT$GSL_RLCLABLS;
377 1405 2 SYM_VALUE (.PAT$GSL_RLCLABLS) = 0;
378 1406 2 SYM_CSTRING (.PAT$GSL_RLCLABLS) = 0;
379 1407 2 |--
380 1408 2 |++
381 1409 2 | Allocate the first buffer for the appended patch command text and
382 1410 2 | initialize the pointers.
383 1411 2 |--
384 1412 2 PAT$GSL_TXTLHD = PAT$FREEZ((A_PAGE + 3)/4);
```



```

: 385
: 386
: 387
: 388
: 389
: 390
: 391
: 392
: 393
: 394
: 395
: 396
: 397
: 398
: 399

1413 2 PAT$GL_TXTTAIL = .PAT$GL_TXTLHD;
1414 2 PAT$GL_TXTFREE = .PAT$GL_TXTLHD + TXT$C_SIZE;
1415 2
1416 2 !++
1417 2 Initialize the ECO level for this patch. This ECO level byte is
1418 2 set to an ECO level by the "SET ECO" command and re-initialized by
1419 2 the "UPDATE" command.
1420 2 --
1421 2 PAT$GB_ECOLVL = 0;
1422 2
1423 2 !++
1424 2 Initialization is complete and successful. Return TRUE.
1425 2 --
1426 2 RETURN(TRUE);
1427 2 END;
```

```

                                .EXTRN  SYSS$OPEN, SYSS$CONNECT
                                .ENTRY   PAT$INIT, Save R2,R3,R4,R5,R6,R7,R8,R9
                                : 1242
59 00000000G 8F D0 00002    MOVL    #PAT$ OPENIN, R9
58 00000000G EF 9E 00009    MOVAB   PAT$GL_TXTLHD, R8
57 00000000G 00 9E 00010    MOVAB   SYSS$CONNECT, R7
56 00000000G 00 9E 00017    MOVAB   SYSS$OPEN, R6
55 00000000G 8F D0 0001E    MOVL    #PAT$ OPENOUT, R5
54 00000000G EF 9E 00025    MOVAB   PAT$FREEZ, R4
53 00000000G 00 9E 0002C    MOVAB   LIB$STOP, R3
52 00000000G EF 9E 00033    MOVAB   PAT$GL_ERRCODE, R2
5E 00000000G 14 C2 0003A    SUBL2   #20, SP
0C AE 00000000G 8F D0 0003D    MOVL    #PAT$K_ERRNAMLNG, PAT$ERRNAM_DESC
10 AE 00000000G EF 9E 00045    MOVAB   PAT$GB_ERRNAME, PAT$ERRNAM_DESC+4
04 AE 00000000G 8F D0 0004D    MOVL    #PAT$K_INPNAMLNG, PAT$INPNAM_DESC
08 AE 00000000G EF 9E 00055    MOVAB   PAT$GB_INPNAME, PAT$INPNAM_DESC+4
    00000000G 8F DD 0005D    PUSHL   #PAT$K_OUTNAMLNG
04 AE 00000000G EF 9E 00063    MOVAB   PAT$GB_OUTNAME, PAT$OUTNAM_DESC+4
    00000000G EF 9F 0006B    PUSHAB  PAT$GL_OUTFAB
66 00000000G 01 FB 00071    CALLS   #1, SYSS$OPEN
62 00000000G 50 D0 00074    MOVL    R0, PAT$GL_ERRCODE
11 00000000G 62 E8 00077    BLBS    PAT$GL_ERRCODE, 1$
7E 00000000G EF 7D 0007A    MOVQ    PAT$GL_OUTFAB+8, -(SP)
    08 AE 9F 00081    PUSHAB  PAT$OUTNAM_DESC
    01 DD 00084    PUSHL   #1
    55 DD 00086    PUSHL   R5
63 00000000G 05 FB 00088    CALLS   #5, LIB$STOP
    00000000G EF 9F 0008B 1$: PUSHAB  PAT$GL_OUTTAB
67 00000000G 01 FB 00091    CALLS   #1, SYSS$CONNECT
62 00000000G 50 D0 00094    MOVL    R0, PAT$GL_ERRCODE
11 00000000G 62 E8 00097    BLBS    PAT$GL_ERRCODE, 2$
7E 00000000G EF 7D 0009A    MOVQ    PAT$GL_OUTTAB+8, -(SP)
    08 AE 9F 000A1    PUSHAB  PAT$OUTNAM_DESC
    01 DD 000A4    PUSHL   #1
    55 DD 000A6    PUSHL   R5
63 00000000G 05 FB 000A8    CALLS   #5, LIB$STOP
    00000000G EF 9F 000AB 2$: PUSHAB  PAT$GL_ERRFAB
66 00000000G 01 FB 000B1    CALLS   #1, SYSS$OPEN
62 00000000G 50 D0 000B4    MOVL    R0, PAT$GL_ERRCODE
                                : 1298
                                : 1299
                                : 1300
                                : 1301
                                : 1302
                                : 1303
                                : 1311
                                : 1312
                                : 1315
                                : 1314
                                : 1317
                                : 1318
                                : 1321
                                : 1320
                                : 1323
```

11		62	E8	000B7	BLBS	PAT\$GL_ERRCODE, 3\$	1324
7E	00000000G	EF	7D	000BA	MOVQ	PAT\$GL-ERRFAB+8, -(SP)	1327
	18	AE	9F	000C1	PUSHAB	PAT\$ERRNAM_DESC	1326
		01	DD	000C4	PUSHL	#1	
		55	DD	000C6	PUSHL	R5	
63		05	FB	000C8	CALLS	#5, LIB\$STOP	
	00000000G	EF	9F	000CB	PUSHAB	PAT\$GL_ERRRAB	1329
67		01	FB	000D1	CALLS	#1, SY\$CONNECT	
62		50	D0	000D4	MOVL	R0, PAT\$GL_ERRCODE	
11		62	E8	000D7	BLBS	PAT\$GL_ERRCODE, 4\$	1330
7E	00000000G	EF	7D	000DA	MOVQ	PAT\$GL-ERRRAB+8, -(SP)	1333
	18	AE	9F	000E1	PUSHAB	PAT\$ERRNAM_DESC	1332
		01	DD	000E4	PUSHL	#1	
		55	DD	000E6	PUSHL	R5	
63		05	FB	000E8	CALLS	#5, LIB\$STOP	
	00000000G	EF	9F	000EB	PUSHAB	PAT\$GL_INPFAB	1335
66		01	FB	000F1	CALLS	#1, SY\$OPEN	
62		50	D0	000F4	MOVL	R0, PAT\$GL_ERRCODE	
11		62	E8	000F7	BLBS	PAT\$GL_ERRCODE, 5\$	1336
7E	00000000G	EF	7D	000FA	MOVQ	PAT\$GL-INPFAB+8, -(SP)	1339
	10	AE	9F	00101	PUSHAB	PAT\$INPNAM_DESC	1338
		01	DD	00104	PUSHL	#1	
		59	DD	00106	PUSHL	R9	
63		05	FB	00108	CALLS	#5, LIB\$STOP	
	00000000G	EF	9F	0010B	PUSHAB	PAT\$GL_INPRAB	1341
67		01	FB	00111	CALLS	#1, SY\$CONNECT	
62		50	D0	00114	MOVL	R0, PAT\$GL_ERRCODE	
11		62	E8	00117	BLBS	PAT\$GL_ERRCODE, 6\$	1342
7E	00000000G	EF	7D	0011A	MOVQ	PAT\$GL-INPRAB+8, -(SP)	1345
	10	AE	9F	00121	PUSHAB	PAT\$INPNAM_DESC	1344
		01	DD	00124	PUSHL	#1	
		59	DD	00126	PUSHL	R9	
		05	FB	00128	CALLS	#5, LIB\$STOP	
00000000G	63	00	FB	0012B	CALLS	#0, PAT\$FREEINIT	1353
00000000G	EF	00	FB	00132	CALLS	#0, PAT\$RESET_DEF	1358
FEB4	CF	00	FB	00139	CALLS	#0, PAT\$SET_CONTEXT	1365
		04	DD	0013E	PUSHL	#4	1371
		01	FB	00140	CALLS	#1, PAT\$FREEZ	
00000000G	64	50	D0	00143	MOVL	R0, PAT\$GL_SYMHED	
	EF	50	D0	0014A	MOVL	R0, (R0)	1372
	60	50	D0	0014D	MOVL	R0, 4(R0)	1373
04	A0	50	D0	00151	MOVL	R0, 8(R0)	1374
		A0	7C	00154	CLRQ	8(R0)	1376
00000000G	EF	50	D0	0015B	MOVL	R0, PAT\$GL_SYMTBPTR	1382
		04	DD	0015D	PUSHL	#4	
		01	FB	00160	CALLS	#1, PAT\$FREEZ	
00000000G	64	50	D0	00167	MOVL	R0, PAT\$GL_OLDLABLS	
	EF	50	D0	0016A	MOVL	R0, (R0)	1383
	60	50	D0	0016E	MOVL	R0, 4(R0)	1384
04	A0	50	D0	00171	MOVL	R0, 8(R0)	1385
		A0	7C	00173	CLRQ	8(R0)	1392
		04	DD	00176	PUSHL	#4	
		01	FB	0017D	CALLS	#1, PAT\$FREEZ	
00000000G	64	50	D0	00180	MOVL	R0, PAT\$GL_NEWLABLS	
	EF	50	D0	00184	MOVL	R0, (R0)	1393
	60	50	D0	00187	MOVL	R0, 4(R0)	1394
04	A0	50	D0	00189	MOVL	R0, 8(R0)	1395
		A0	7C		CLRQ	8(R0)	1402
		04	DD		PUSHL	#4	
		01	FB		CALLS	#1, PAT\$FREEZ	

PATBAS
V04-000

L 3
16-Sep-1984 00:08:26
14-Sep-1984 12:52:28

VAX-11 Bliss-32 V4.0-742
DISK\$VM\$MASTER:[PATCH.SRC]PATBAS.B32;1
Page 11
(4)

00000000G	EF	50	D0	0018C	MOVL	R0, PAT\$GL_RLCLABLS	:	
	60	50	D0	00193	MOVL	R0, (R0)	:	1403
04	A0	50	D0	00196	MOVL	R0, 4(R0)	:	1404
		08	A0	7C 0019A	CLRQ	8(R0)	:	1405
	7E	80	8F	9A 0019D	MOVZBL	#128, -(SP)	:	1412
	64		01	FB 001A1	CALLS	#1, PAT\$FREEZ	:	
	68		50	D0 001A4	MOVL	R0, PAT\$GL_TXTLHD	:	
00000000G	EF	68	D0	001A7	MOVL	PAT\$GL_TXTLHD, PAT\$GL_TXTTAIL	:	1413
	68		04	C1 001AE	ADDL3	#4, PAT\$GL_TXTLHD, PAT\$GL_TXTFREE	:	1414
00000000G	EF		94	001B6	CLRB	PAT\$GB_ECOEVL	:	1421
			04	001BC	RET		:	1427

; Routine Size: 445 bytes, Routine Base: _PAT\$CODE + 000E

PATBAS
V04-000

M 3
16-Sep-1984 00:08:26
14-Sep-1984 12:52:28

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATBAS.B32;1 Page 12 (5)

: 401 1428 1 END
: 402 1429 0 ELUDOM

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
PAT\$CODE	459	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
ABS	0	NOVEC,NOWRT,NORD ,NOEXE,NOSHR, LCL, ABS, CON,NOPI,ALIGN(0)

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)/VARIANT:1/LIS=LIS\$:PATBAS/OBJ=OBJ\$:PATBAS MSRC\$:PATBAS/UPDATE=(ENH\$:PATBAS)

: Size: 459 code + 0 data bytes
: Run Time: 00:21.6
: Elapsed Time: 01:06.1
: Lines/CPU Min: 3962
: Lexemes/CPU-Min: 49258
: Memory Used: 188 pages
: Compilation Complete

0300 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

