

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
0001 0
0002 0 MODULE PATMAIN ( MAIN = PAT$MAIN,           ! DRIVER FOR PATCH
L 0003 0     %IF %VARIANT EQL 1
0004 0     %THEN
0005 0     ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
0006 0     %FI
0007 0     IDENT = 'V04-000'
0008 0     ) =
0009 1 BEGIN
0010 1
0011 1
0012 1 *****
0013 1 *
0014 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0015 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0016 1 *  ALL RIGHTS RESERVED.
0017 1 *
0018 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0019 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0020 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0021 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0022 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0023 1 *  TRANSFERRED.
0024 1 *
0025 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0026 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0027 1 *  CORPORATION.
0028 1 *
0029 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0030 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0031 1 *
0032 1 *
0033 1 *****
0034 1
0035 1
0036 1 **
0037 1 FACILITY:  IMAGE FILE PATCH UTILITY, PATCH
0038 1
0039 1 ABSTRACT:  THIS ROUTINE DRIVES PATCH.  IT PERFORMS THE COMMAND LINE
0040 1           INTERFACE, THE FILE HANDLING, AND THE COMMAND PROCESSING LOOP.
0041 1
0042 1 ENVIRONMENT:
0043 1
0044 1 AUTHOR:   K.D. MORSE   , CREATION DATE:  29-SEP-77
0045 1
0046 1 MODIFIED BY:
0047 1
0048 1     V03-012 MCN0180      Maria del C. Nasr      12-Jul-1984
0049 1     Process qualifiers that take a value properly.  First
0050 1     check if the qualifier is present, and if true, get the
0051 1     value.  In that way, it will not process the value if the
0052 1     qualifier has been negated.
0053 1
0054 1     V03-011 MCN0172      Maria del C. Nasr      12-Jun-1984
0055 1     Set new version number and date displayed by PATCH.
0056 1
0057 1     V03-010 MCN0157      Maria del C. Nasr      20-Feb-1984
```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

0058 1
0059 1
0060 1
0061 1
0062 1
0063 1
0064 1
0065 1
0066 1
0067 1
0068 1
0069 1
0070 1
0071 1
0072 1
0073 1
0074 1
0075 1
0076 1
0077 1
0078 1
0079 1
0080 1
0081 1
0082 1
0083 1
0084 1
0085 1
0086 1
0087 1
0088 1
0089 1
0090 1
0091 1
0092 1
0093 1
0094 1
0095 1
0096 1
0097 1
0098 1
0099 1
0100 1
0101 1
0102 1
0103 1
0104 1
0105 1
0106 1
0107 1
0108 1
0109 1
0110 1
0111 1
0112 1
0113 1
0114 1

When we open the input file, save the channel, but do not connect stream. We are doing a user file open now to use the image activator routines for image processing.

- V03-009 MTR0025 Mike Rhodes 8-Aug-1983
Add /ABSOLUTE qualifier to allow patching ANY file via absolute (virtual) addresses. Restrictions: The file's size may not grow (eg. default patch area may NOT be created/expanded, etc.). This is due to our inability to determine if we'll end up breaking an image or other such file which has internal size declarations within its header.
- Additionally, routine EXIT_HANDLER has been added, which will be declared when the file is patched in place (/ABSOLUTE /NONEW_VERSION).
- V03-008 MTR0024 Mike Rhodes 24-Jul-1983
Correct default action of JOURNAL file processing.
- V03-007 MTR0022 Mike Rhodes 22-Feb-1983
Fix command file processing (/COMMAND and CREATE).
- V03-006 MTR0020 Mike Rhodes 4-Feb-1983
Convert to the new CLI interface, advance PATCH's welcome message to reflect this change.
- V03-005 MTR0019 Mike Rhodes 5-Jan-1983
Advance PATCH's ident to reflect inclusion of code to handle Debug Module/psect Table (DMT) in PATWRT.B32.
- V03-004 MTR0018 Mike Rhodes 15-Nov-1982
Bump PATCH's ident message.
- V03-003 MTR0009 Mike Rhodes 08-Jul-1982
Advance PATCH's ID to reflect the inclusion of 2 operand instruction alias support.
- V03-002 MTR0007 Mike Rhodes 14-Jun-1982
Use shared system messages. Affected modules include: DYNMEM.B32, PATBAS.B32, PATCMD.B32, PATIHD.B32, PATINT.B32, PATIO.B32, PATMAI.B32, PATMSG.MSG, PATWRT.B32, and PATSPA.B32.

The shared messages are defined by PATERR.B32 and we simply link against these symbols. They are declared as external literals below.

Clock the welcome message ident to 3-01 15-Jun-1982.
- V03-001 MTR0003 Mike Rhodes 19-Mar-1982
Modified routine OPEN_INPUT. It passed the status value incorrectly from the RAB instead of the FAB.
Fixes QAR 734.
- V03-000 MTR0002 Mike Rhodes 15-Mar-1982
Advance PATCH welcome message version number to

115	0115	1	3-00 and the date to 15-Mar-1982.
116	0116	1	
117	0117	1	V02-035 MTR0002 Mike Rhodes 01-Oct-1981
118	0118	1	Change PATCH welcome message version number to
119	0119	1	2.33 and the date to 01-Oct-1981.
120	0120	1	
121	0121	1	V02-034 MTR0001 Mike Rhodes 17-Sep-1981
122	0122	1	Change PATCH welcome message version number to
123	0123	1	2.32 and the date to 17-Sep-1981.
124	0124	1	
125	0125	1	V02-033 KDM0052 Kathleen D. Morse 28-Apr-1981
126	0126	1	Change PATCH version number to 2.31 and date to 28-Apr-1981.
127	0127	1	
128	0128	1	V02-032 BLS0043 Benn Schreiber 10-Apr-1981
129	0129	1	Get input file size from FHC XAB
130	0130	1	
131	0131	1	V02-031 KDM0045 Kathleen D. Morse 04-Mar-1981
132	0132	1	Change PATCH version number to 2.30 and date to 04-Mar-1981.
133	0133	1	
134	0134	1	V02-030 KDM0044 Kathleen D. Morse 04-Mar-1981
135	0135	1	Fix some comment lines.
136	0136	1	
137	0137	1	V0229 PCG0001 Peter George 02-FEB-1981
138	0138	1	Add require statement for LIB\$:PATDEF.REQ
139	0139	1	
140	0140	1	
141	0141	1	--

```
143 0142 1 |
144 0143 1 | TABLE OF CONTENTS:
145 0144 1 |
146 0145 1 |
147 0146 1 | FORWARD ROUTINE
148 0147 1 |   PROCESS_COMMAND : NOVALUE,
149 0148 1 |   VOLUME_PARSE : NOVALUE,
150 0149 1 |   UPDATE_PARSE : NOVALUE,
151 0150 1 |   OPEN_INPUT : NOVALUE,
152 0151 1 |   EXIT_HANDLER : NOVALUE,
153 0152 1 |   OPEN_OUTPUT : NOVALUE,
154 0153 1 |   OPEN_JOURNAL : NOVALUE,
155 0154 1 |   PAT$OPEN_COMFIL : NOVALUE;
156 0155 1 |
157 0156 1 |
158 0157 1 | INCLUDE FILES:
159 0158 1 |
160 0159 1 |
161 0160 1 | LIBRARY 'SYSS$LIBRARY:STARLET.L32';
162 0161 1 | LIBRARY 'SYSS$LIBRARY:CLIMAC.L32';
163 0162 1 | REQUIRE 'SRC$:PATPCT.REQ';
164 0202 1 | REQUIRE 'SRC$:PREFIX.REQ';
165 0390 1 | REQUIRE 'SRC$:PATPRE.REQ';
166 0553 1 | REQUIRE 'LIB$:PATDEF.REQ';
167 0607 1 | REQUIRE 'LIB$:PATMSG.REQ';
168 0781 1 | REQUIRE 'SRC$:VXSMAC.REQ';
169 0846 1 | REQUIRE 'SRC$:SYSLIT.REQ';
170 0896 1 | REQUIRE 'SRC$:SYSSER.REQ';

! Process the command line
! VOLUME qualifier action routine
! UPDATE qualifier action routine.
! OPENS THE INPUT IMAGE FILE
! Handle exit conditions for patching in pla
! INITIALIZES FOR FUTURE OUTPUT IMAGE FILE 0
! OPENS THE JOURNAL FILE
! OPENS THE COMMAND FILE

! SYSTEM STRUCTURE DEFINITIONS
! CLI STRUCTURE ALLOCATION
! DEFINE PSECTS
! UTILITY MACROS
! DEFINES PATCH LITERALS
! Defines literals
! DEFINES PATCH ERROR MESSAGE NUMBERS
! DEFINE LITERALS
! DEFINES SYSTEM LITERALS
! DEFINES FAL OUTPUT MACROS
```

PATMAIN
V04-000

J 11
16-Sep-1984 00:03:48
15-Sep-1984 22:50:49

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

Page 5
(1)

: R0928 1
: R0929 1
: R0930 1
: R0931 1
: R0932 1

SWITCHES LIST (SOURCE);

EXTERNAL ROUTINE
PAT\$fao_out;

! formats a line and outputs to the terminal

```

171 0978 1
172 0979 1
173 0980 1  EXTERNAL REFERENCES:
174 0981 1
175 0982 1
176 0983 1  EXTERNAL ROUTINE
177 0984 1      CLISPRESNT      : ADDRESSING_MODE (GENERAL),
178 0985 1      CLISGET_VALUE : ADDRESSING_MODE (GENERAL),
179 0986 1      LIB$CVT_DX_DX  : ADDRESSING_MODE (GENERAL),
180 0987 1      PAT$FIND_DST,
181 0988 1      pat$com_proc,
182 0989 1      PAT$FAO_PUT,
183 0990 1      PAT$INIT,
184 0991 1      PAT$GET_IHD,
185 0992 1      PAT$CLOSEFILES,
186 0993 1      PAT$WRITEFILE,
187 0994 1      PAT$ERR_HANDLER,
188 0995 1      GETFILDSC;
189 0996 1
190 0997 1  EXTERNAL
191 0998 1      PAT$GB_EXEC_CMD : BYTE,
192 0999 1      PAT$CP_OUT_STR : REF VECTOR[.BYTE],
193 1000 1      PAT$GL_BUF_SIZ,
194 1001 1      PAT$GL_CHANUM,
195 1002 1      PAT$GB_TAKE_CMD: BYTE,
196 1003 1      PAT$GL_ERRCODE,
197 1004 1      PAT$GL_HEAD_LST,
198 1005 1      PAT$GL_IMGBCKS,
199 1006 1      PAT$GW_IMGVOL : WORD,
200 1007 1      PAT$GL_ECO_UPD,
201 1008 1      PAT$GL_FLAGS : BITVECTOR [32],
202 1009 1      PAT$GL_OLDXABFH : BLOCK[.BYTE],
203 1010 1      PAT$GL_OLDFAB : BLOCK[.BYTE],
204 1011 1      PAT$GL_OLDRAB : BLOCK[.BYTE],
205 1012 1      PAT$GL_OLDNBK : BLOCK[.BYTE],
206 1013 1      PAT$GL_NEWFAB : BLOCK[.BYTE],
207 1014 1      PAT$GL_NEWRAB : BLOCK[.BYTE],
208 1015 1      PAT$GL_NEWNBK : BLOCK[.BYTE],
209 1016 1      PAT$GL_NEWXABALL : BLOCK[.BYTE],
210 1017 1      PAT$GL_JNLFAB : BLOCK[.BYTE],
211 1018 1      PAT$GL_JNLRAB : BLOCK[.BYTE],
212 1019 1      PAT$GL_JNLNBK : BLOCK[.BYTE],
213 1020 1      PAT$GL_COMFAB : BLOCK[.BYTE],
214 1021 1      PAT$GL_COMRAB : BLOCK[.BYTE],
215 1022 1      PAT$GL_COMNBK : BLOCK[.BYTE],
216 1023 1      PAT$GB_OLDNAME,
217 1024 1      PAT$GB_NEWNAME,
218 1025 1      PAT$GB_JNLNAME,
219 1026 1      PAT$GB_COMNAME;
220 1027 1
221 1028 1  EXTERNAL LITERAL
222 1029 1
223 1030 1  Define shared message references. (resolved @ link time)
224 1031 1
225 1032 1      PAT$_CLOSEIN,
226 1033 1      PAT$_CLOSEOUT,
227 1034 1      PAT$_OPENIN,

```

```

! Check for an entities presence in the comm
! Obtain the value associated with a qualifi
! General conversion routine.
! INITIALIZES SYMBOL TABLES
! PATCH COMMAND PROCESSOR
! FORMATS OUTPUT MESSAGE
! OPEN COMMUNICATION CHANNELS
! PROCESS IMAGE HEADER INFORMATION
! CLOSE ALL FILES
! OUTPUTS MESSAGE TO A FILE
! ERROR MESSAGE HANDLER
! Returns the address of a file name descrip

! INDICATOR WHETHER OR NOT TO EXECUTE PATCH
! POINTER TO OUTPUT BUFFER
! SIZE OF MESSAGE IN OUTPUT BUFFER
! CHANNEL NUMBER
! INDICATOR TO PROCESC COMMANDS
! ERROR CODE
! HEAD OF PATCH COMMAND ARGUMENT LIST
! NUMBER OF BLOCKS IN NEW IMAGE
! Relative Volume Number for new image
! ECU LEVELS TO UPDATE
! CLI PARSE FLAGS
! INPUT IMAGE FILE FILE HEADER CHARS. XAB
! INPUT IMAGE FILE FAB
! INPUT IMAGE FILE RAB
! INPUT IMAGE FILE NAME BLOCK
! OUTPUT IMAGE FILE FAB
! OUTPUT IMAGE FILE RAB
! OUTPUT IMAGE FILE NAME BLOCK
! Output image ALlocation XAB
! JOURNAL FILE FAB
! JOURNAL FILE RAB
! JOURNAL FILE NAME BLOCK
! OUTPUT COMMAND FILE FAB
! OUTPUT COMMAND FILE RAB
! OUTPUT COMMAND FILE NAME BLOCK
! OLD IMAGE FILE NAME
! NEW IMAGE FILE NAME
! JOURNAL FILE NAME
! COMMAND FILE NAME

```

```

! Error closing input file.
! Error closing output file.
! Error opening input file.

```



```

: 228      1035 1      PAT$_OPENOUT,      ! Error opening output file.
: 229      1036 1      PAT$_OVERLAY,      ! File is being overwritten.
: 230      1037 1      PAT$_READERR,      ! Error reading from file.
: 231      1038 1      PAT$_SYSERROR,      ! System Service error.
: 232      1039 1      PAT$_WRITEERR;      ! Error writing to file.
: 233      1040 1
: 234      1041 1
: 235      1042 1      MACROS:
: 236      1043 1
: 237      1044 1
: 238      1045 1
: 239      1046 1      EQUATED SYMBOLS:
: 240      1047 1
: 241      1048 1
: 242      1049 1
: 243      1050 1      OWN STORAGE:
: 244      1051 1
: 245      1052 1      OWN
: 246      1053 1      exit_cb : VECTOR [4, LONG],      ! EXIT Control Block.
: 247      1054 1      input_desc : $BBLOCK [dsc$c_s_bln];
: 248      1055 1
: 249      1056 1      BIND
: 250      1057 1      !++
: 251      1058 1      ! Qualifier and Parameter name descriptors for command line parsing.
: 252      1059 1      !--
: 253      1060 1      absolute= $descriptor ('ABSOLUTE'),      ! /ABSOLUTE qualifier
: 254      1061 1      cmdline = $descriptor ('$LINE'),      ! Command line identifier for CLI call back.
: 255      1062 1      command = $descriptor ('COMMAND'),      ! /COMMAND qualifier
: 256      1063 1      file = $descriptor ('FILE'),      ! Input file parameter (required).
: 257      1064 1      journal = $descriptor ('JOURNAL'),      ! /JOURNAL qualifier
: 258      1065 1      new_ver = $descriptor ('NEW VERSION'),      ! /NEW VERSION qualifier
: 259      1066 1      output = $descriptor ('OUTPUT'),      ! /OUTPUT qualifier
: 260      1067 1      update = $descriptor ('UPDATE'),      ! /UPDATE qualifier
: 261      1068 1      volume = $descriptor ('VOLUME'),      ! /VOLUME qualifier
: 262      1069 1
: 263      1070 1      !++
: 264      1071 1      ! FAO control string for output of command line to command file.
: 265      1072 1      !--
: 266      1073 1      COM_STR      = UPLIT ( %ASCIC 'COMMAND FILE:      '!AD'''),
: 267      1074 1      COM_LINE_STR      = UPLIT (%ASCIC '!AS');

```

```

: 269 1075 1 GLOBAL ROUTINE PAT$MAIN : NOVALUE = ! MAIN PATCH DRIVER
: 270 1076 1
: 271 1077 1 !++
: 272 1078 1 ! FUNCTIONAL DESCRIPTION:
: 273 1079 1
: 274 1080 1 ! THIS MODULE IS THE DRIVER FOR THE IMAGE FILE PATCH UTILTIY.
: 275 1081 1 ! IT PERFORMS THE FOLLOWING:
: 276 1082 1
: 277 1083 1 ! 1. REPEATEDLY CALLS BACK THE CLI REQUESTING THE PIECES OF THE
: 278 1084 1 ! COMMAND LINE
: 279 1085 1 ! 2. OPENS THE FILES FOR INPUT AND OUTPUT
: 280 1086 1 ! 3. CALLS THE ROUTINE TO MAP THE INPUT IMAGE FILE
: 281 1087 1 ! 4. LOOPS, CALLING APPROPRIATE ROUTINES TO HANDLE PATCH COMMANDS
: 282 1088 1 ! 5. CLOSES ALL FILES
: 283 1089 1
: 284 1090 1
: 285 1091 1 ! FORMAL PARAMETERS:
: 286 1092 1
: 287 1093 1 ! NONE
: 288 1094 1
: 289 1095 1 ! IMPLICIT INPUTS:
: 290 1096 1
: 291 1097 1 ! CLI HAS THE COMMAND LINE.
: 292 1098 1 ! THE FABS AND RABS ARE INITIALIZED, SAVE FOR FILE NAMES.
: 293 1099 1
: 294 1100 1 ! IMPLICIT OUTPUTS:
: 295 1101 1
: 296 1102 1 ! NONE
: 297 1103 1
: 298 1104 1 ! ROUTINE VALUE:
: 299 1105 1
: 300 1106 1 ! NONE
: 301 1107 1
: 302 1108 1 ! COMPLETION CODES:
: 303 1109 1
: 304 1110 1 ! NONE
: 305 1111 1
: 306 1112 1 ! SIDE EFFECTS:
: 307 1113 1
: 308 1114 1 ! THE INPUT AND OUTPUT FILES ARE OPENED. THE COMMAND LINE IS PROCESSED.
: 309 1115 1 ! THE PATCH COMMANDS ARE EXECUTED.
: 310 1116 1
: 311 1117 1 ! --
: 312 1118 1
: 313 1119 2 BEGIN
: 314 1120 2 BIND
: 315 1121 2 NEW_IMG_STR = UPLIT ( %ASCIC 'IMAGE FILE BEING PATCHED: '!'AD'''),
: 316 1122 2 JNL_STR = UPLIT ( %ASCIC 'JOURNAL FILE: '!'AD'''),
: 317 1123 2 UPDATE_STR = UPLIT ( %ASCIC 'UPDATE QUALIFIER VALUES: '),
: 318 1124 2 UPD_VAC_STR = UPLIT ( %ASCIC '!UL,');
: 319 1125 2
: 320 1126 2 LOCAL
: 321 1127 2 BIT_NUMBER, ! ECO LEVEL BIT COUNTER
: 322 1128 2 OUT_BUFFER : VECTOR[TTY_OUT_WIDTH,BYTE]; ! OUTPUT BUFFER FOR JOURNAL MESSAGES
: 323 1129 2
: 324 1130 2 ENABLE PAT$ERR_HANDLER; ! DECLARE ERROR HANDLER
: 325 1131 2

```

```

326 1132 2 |++
327 1133 2 | OPEN COMMUNICATION CHANNELS AND INITIALIZE CONTEXT DATA,ETC.
328 1134 2 |--
329 1135 2 PAT$INIT(); ! OPEN CHANNELS
330 1136 2 |--
331 1137 2 |++
332 1138 2 | PROCESS THE COMMAND LINE ET. AL.
333 1139 2 |--
334 1140 2 PROCESS_COMMAND ();
335 1141 2 |--
336 1142 2 |++
337 1143 2 | OUTPUT WELCOME MESSAGE.
338 1144 2 |--
339 1145 2 $FAO TT_OUT('!//!// PATCH Version 4-00 15-Sep-1984!//');
340 1146 2 |--
341 1147 2 |++
342 1148 2 | OUTPUT COMMAND INFORMATION TO JOURNAL FILE.
343 1149 2 |--
344 1150 2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
345 1151 2 PAT$GL_BUF_SIZ = 0;
346 1152 2 PAT$FAO PUT(NEW_IMG_STR, .PAT$GL_OLDNBK[NAM$B_RSL], PAT$GB_OLDNAME);
347 1153 2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
348 1154 2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
349 1155 2 PAT$GL_BUF_SIZ = 0;
350 1156 2 PAT$FAO PUT(JNL_STR, .PAT$GL_JNLNBK[NAM$B_RSL], PAT$GB_JNLNAME);
351 1157 2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
352 1158 2 ! IF (.PAT$GL_FLAGS AND PAT$M_COMMAND) NEQ 0
353 1159 2 ! THEN
354 1160 2 BEGIN
355 1161 2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
356 1162 2 PAT$GL_BUF_SIZ = 0;
357 1163 2 PAT$FAO PUT(COM_STR, .PAT$GL_COMNBK[NAM$B_RSL], PAT$GB_COMNAME);
358 1164 2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
359 1165 2 END;
360 1166 2 |--
361 1167 2 |++
362 1168 2 | Attempt to write the input file spec to the output command file. This
363 1169 2 | places it in the appended patch command text buffer even if no output
364 1170 2 | command file is being written.
365 1171 2 |--
366 1172 2 PAT$WRITEFILE(.PAT$GL_OLDFAB [FAB$B_FNS], .PAT$GL_OLDFAB [FAB$L_FNA], PAT$GL_COMRAB);
367 1173 2 IF .PAT$GL_FLAGS [PAT$S_UPDATE]
368 1174 2 THEN
369 1175 2 BEGIN
370 1176 2 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
371 1177 2 PAT$GL_BUF_SIZ = 0;
372 1178 2 PAT$FAO PUT(UPDATE_STR);
373 1179 2 BIT_NUMBER = PAT$K_MIN_ECO;
374 1180 2 WHILE .BIT_NUMBER [EQ PAT$K_MAX_ECO
375 1181 2 DO
376 1182 2 BEGIN
377 1183 2 IF .PAT$GL_ECO_UPD<.BIT_NUMBER-1, 1>
378 1184 2 THEN
379 1185 2 PAT$FAO PUT(UPD_VAL_STR, .BIT_NUMBER);
380 1186 2 BIT_NUMBER = .BIT_NUMBER + 1;
381 1187 2 END;
382 1188 2 PAT$WRITEFILE(.PAT$GL_BUF_SIZ-1, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);

```

```

383 1189 2      END;
384 1190 2
385 1191 2      ++
386 1192 2      ! READ THE IMAGE HEADER AND SET UP THE IMAGE SECTION TABLE.
387 1193 2      --
388 1194 2      PAT$GET_IHD();                                ! GET INFO FROM IMAGE HEADER
389 1195 2
390 1196 2      ++
391 1197 2      ! INITIALIZE SYMBOL TABLES FROM RST AND GST IN OLD IMAGE FILE.
392 1198 2      --
393 1199 2      PAT$FIND_DST();
394 1200 2
395 1201 2      ++
396 1202 2      ! THIS IS THE LOOP WHICH CAUSES PATCH COMMANDS TO BE PROCESSED.
397 1203 2      ! AS LONG AS THERE ARE COMMANDS TO PROCESS, THE COMMAND FLAG IS ON.
398 1204 2      ! IT IS TURNED OFF WHEN EOF IS ENCOUNTERED.
399 1205 2      --
400 1206 2      PAT$GB_TAKE_CMD = TRUE;                        ! PROCESS COMMANDS
401 1207 2      WHILE .PAT$GB_TAKE_CMD DO
402 1208 2
403 1209 2          PAT$COM_PROC();
404 1210 2
405 1211 2      ++
406 1212 2      ! CLOSE ALL FILES.
407 1213 2      --
408 1214 2      PAT$CLOSEFILES();                            ! FINISH UP
409 1215 2
410 1216 1      END; ! END OF PAT$MAIN

```

```

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

```

```

.TITLE PATMAIN
.IDENT \V04-000\

.PSECT _PAT$PLIT,NOWRT,NOEXE,0

45 54 55 4C 4F 53 42 41 00000 P.AAB: .ASCII \ABSOLUTE\
00000008 00008 P.AAA: .LONG 8
00000000 0000C P.AAA: .ADDRESS P.AAB
45 4E 49 4C 24 00010 P.AAD: .ASCII \SLINE\
00015 P.AAD: .BLKB 3
00000005 00018 P.AAC: .LONG 5
00000000 0001C P.AAC: .ADDRESS P.AAD
44 4E 41 4D 4D 4F 43 00020 P.AAF: .ASCII \COMMAND\
0C027 P.AAF: .BLKB 1
00000007 0C028 P.AAE: .LONG 7
00000000 0002C P.AAE: .ADDRESS P.AAF
45 4C 49 46 00030 P.AAH: .ASCII \FILE\
00000004 00034 P.AAG: .LONG 4
00000000 00038 P.AAG: .ADDRESS P.AAH
4C 41 4E 52 55 4F 4A 0003C P.AAJ: .ASCII \JOURNAL\
00043 P.AAJ: .BLKB 1
00000007 00044 P.AAI: .LONG 7
00000000 00048 P.AAI: .ADDRESS P.AAJ
4E 4F 49 53 52 45 56 5F 57 45 4E 0004C P.AAL: .ASCII \NEW_VERSION\
00057 P.AAL: .BLKB 1

```

```

0000000B, 00058 P.AAK: .LONG 11
00000000, 0005C .ADDRESS P.AAL
54 55 50 54 55 4F 00060 P.AAN: .ASCII \OUTPUT\
00066 .BLKB 2
00000006, 00068 P.AAM: .LONG 6
00000000, 0006C .ADDRESS P.AAN
45 54 41 44 50 55 00070 P.AAP: .ASCII \UPDATE\
00076 .BLKB 2
00000006, 00078 P.AAO: .LONG 6
00000000, 0007C .ADDRESS P.AAP
45 4D 55 4C 4F 56 00080 P.AAR: .ASCII \VOLUME\
00086 .BLKB 2
00000006, 00088 P.AAQ: .LONG 6
00000000, 0008C .ADDRESS P.AAR
09 3A 45 4C 49 46 20 44 4E 41 4D 4D 4F 43 15 00090 P.AAS: .ASCII <21>\COMMAND FILE:\<9><9><9>\'\
21 22 09 09 0009F .ASCII <21>\COMMAND FILE:\<9><9><9>\'\
00 00 22 44 41 000A3 .ASCII \AD''\<0><0>
53 41 21 03 000A8 P.AAT: .ASCII <3>\!AS\
49 45 42 20 45 4C 49 46 20 45 47 41 4D 49 1F 000AC P.AAU: .ASCII <31>\IMAGE FILE BEING PATCHED:\<9>\'\
21 22 09 3A 44 45 48 43 54 41 50 20 47 4E 000BB .ASCII <31>\IMAGE FILE BEING PATCHED:\<9>\'\
09 3A 45 4C 49 46 20 4C 41 4E 52 55 4F 4A 15 000C9 .ASCII \AD''\
000CC P.AAV: .ASCII <21>\JOURNAL FILE:\<9><9><9>\'\
21 22 09 09 000DB .ASCII <21>\JOURNAL FILE:\<9><9><9>\'\
49 46 49 4C 41 55 51 20 45 54 41 44 50 55 19 000E4 P.AAW: .ASCII \AD''\<0><0>
000F3 .ASCII <25>\UPDATE QUALIFIER VALUES:\<9><0>
00 00 2C 4C 55 21 20 05 00100 P.AAX: .ASCII <0>
00 00 2C 4C 55 21 20 05 00108 P.AAY: .ASCII <5>\ !UL,\<0><0>
65 56 20 20 48 43 54 41 50 20 20 2F 21 2F 21 00109 .ASCII <0>
31 20 20 20 20 30 30 2D 34 20 6E 6F 69 73 72 00118 .ASCII \!// PATCH Version 4-00 15-Sep-1984\
34 38 39 31 2D 70 65 53 2D 35 00127 .ASCII \!// PATCH Version 4-00 15-Sep-1984\
2F 21 00131 .ASCII \!//

```

```

00000 EXIT_CB: .BLKB 16
00010 INPUT_DESC:
.BLKB 8

```

```

ISE$C_SIZE== 20
TXT$C_SIZE== 4
PAL$C_SIZE== 16
ASD$C_SIZE== 9
FWR$C_SIZE== 24
ABSOLUTE= P.AAA
CMDLINE= P.AAC
COMMAND= P.AAE
FILE= P.AAG
JOURNAL= P.AAI
NEW_VER= P.AAK
OUTPUT= P.AAM
UPDATE= P.AAO
VOLUME= P.AAQ
COM_STR= P.AAS
COM_LINE_STR= P.AAT
NEW_IMG_STR= P.AAU

```

```

.PSECT _PAT$OWN,NOEXE,2

```

```

JNL_STR= P.AAV
UPDATE_STR= P.AAW
UPD_VAL_STR= P.AAX
.EXTRN PAT$FAO_OUT, CLISPRESNT
.EXTRN CLISGET-VALUE, LIB$CVT_DX_DX
.EXTRN PAT$FIND_DST, PAT$COM_PROC
.EXTRN PAT$FAO_PUT, PAT$INIT
.EXTRN PAT$GET_IHD, PAT$CLOSEFILES
.EXTRN PAT$WRITEFILE, PAT$ERR_HANDLER
.EXTRN GETFILDSC, PAT$GB_EXEC_CMD
.EXTRN PAT$CP_OUT_STR, PAT$GL_BUF_SIZ
.EXTRN PAT$GL_CHARUM, PAT$GB_TAKE_CMD
.EXTRN PAT$GL_ERRCODE, PAT$GC_HEAD_LST
.EXTRN PAT$GL_IMGBLKS, PAT$GW_IMGVOL
.EXTRN PAT$GL_ECO_UPD, PAT$GL_FLAGS
.EXTRN PAT$GL_OLDRABFH
.EXTRN PAT$GL_OLDFAB, PAT$GL_OLDRAB
.EXTRN PAT$GL_OLDNBK, PAT$GL_NEWFAB
.EXTRN PAT$GL_NEWRAB, PAT$GL_NEWNBK
.EXTRN PAT$GL_NEWXABALL
.EXTRN PAT$GL_JNLFAB, PAT$GL_JNLRAB
.EXTRN PAT$GL_JNLNBK, PAT$GL_COMFAB
.EXTRN PAT$GL_COMRAB, PAT$GL_COMNBK
.EXTRN PAT$GB_OLDNAME, PAT$GB_NEWNAME
.EXTRN PAT$GB_JNLNAME, PAT$GB_COMNAME
.EXTRN PAT$CLOSEIN, PAT$CLOSEOUT
.EXTRN PAT$OPENIN, PAT$OPENOUT
.EXTRN PAT$OVERLAY, PAT$READERR
.EXTRN PAT$SYSERROR, PAT$WRITEERR

```

.PSECT _PAT\$CODE, NOWRT, 2

```

03FC 00000 .ENTRY PAT$MAIN, Save R2,R3,R4,R5,R6,R7,R8,R9 : 1075
59 00000000G EF 9E 00002 MOVAB PAT$GB_TAKE_CMD, R9
58 00000000G EF 9E 00009 MOVAB PAT$GL_JNLRAB, R8
57 00000000G EF 9E 00010 MOVAB PAT$CP_OUT_STR, R7
56 00000000G EF 9E 00017 MOVAB PAT$WRITEFILE, R6
55 00000000G EF 9E 0001E MOVAB PAT$FAO_PUT, R5
54 00000000' EF 9E 00025 MOVAB P.AAY, R4
53 00000000G EF 9E 0002C MOVAB PAT$GL_BUF_SIZ, R3
5E FF7C CE 9E 00033 MOVAB -132(SP), SP
6D 00DC CF DE 00038 MOVAL 7$, (FP) : 1119
00000000G EF 00 FB 0003D CALLS #0, PAT$INIT : 1135
00000000V EF 00 FB 00044 CALLS #0, PROCESS_COMMAND : 1140
7E D4 0004B CLRL -(SP) : 1145
54 DD 0004D PUSHL R4
00000000G EF 02 FB 0004F CALLS #2, PAT$FAO_OUT
67 6E 9E 00056 MOVAB OUT_BUFFER, -PAT$CP_OUT_STR : 1150
63 D4 00059 CLRL PAT$GL_BUF_SIZ : 1151
00000000G EF 9F 0005B PUSHAB PAT$GB_OLDNAME : 1152
7E 00000000G EF 9A 00061 MOVZBL PAT$GL_OLDNBK+3, -(SP)
A4 A4 9F 00068 PUSHAB NEW_IMG_STR
65 03 FB 0006B CALLS #3, -PAT$FAO_PUT
58 DD 0006E PUSHL R8 : 1153
04 AE 9F 00070 PUSHAB OUT_BUFFER
63 DD 00073 PUSHL PAT$GL_BUF_SIZ
66 03 FB 00075 CALLS #3, PAT$WRITEFILE

```

67		6E	9E	00078	MOVAB	OUT_BUFFER, PAT\$CP_OUT_STR	1154	
		63	D4	0007B	CLRL	PAT\$GL_BUF_SIZ	1155	
	00000000G	EF	9F	0007D	PUSHAB	PAT\$GB_JNLNAME	1156	
7E	00000000G	EF	9A	00083	MOVZBL	PAT\$GL_JNLNBK+3, -(SP)		
	C4	A4	9F	0008A	PUSHAB	JNL_STR		
65		03	FB	0008D	CALLS	#3, PAT\$FAO_PUT		
		58	DD	00090	PUSHL	R8	1157	
	04	AE	9F	00092	PUSHAB	OUT_BUFFER		
		63	DD	00095	PUSHL	PAT\$GL_BUF_SIZ		
66		03	FB	00097	CALLS	#3, PAT\$WRITEFILE		
	00000000G	EF	9F	0009A	PUSHAB	PAT\$GL_COMRAB	1172	
	00000000G	EF	DD	000A0	PUSHL	PAT\$GL_OLDFAB+44		
7E	00000000G	EF	9A	000A6	MOVZBL	PAT\$GL_OLDFAB+52, -(SP)		
66		03	FB	000AD	CALLS	#3, PAT\$WRITEFILE		
3B	00000000G	EF	04	E1	BBC	#4, PAT\$GL_FLAGS, 4\$	1173	
67		6E	9E	000B8	MOVAB	OUT_BUFFER, PAT\$CP_OUT_STR	1176	
		63	D4	000BB	CLRL	PAT\$GL_BUF_SIZ	1177	
		A4	9F	000BD	PUSHAB	UPDATE_STR	1178	
65		01	FB	000C0	CALLS	#1, PAT\$FAO_PUT		
52		01	D0	000C3	MOVL	#1, BIT_NUMBER	1179	
00000080		8F	D1	000C6	1\$:	CMPL	BIT_NUMBER, #128	1180
		18	14	000CD	BGTR	3\$		
50		FF	A2	9E	MOVAB	-1(R2), R0	1183	
08	00000000G	EF	50	E1	BBC	R0, PAT\$GL_ECO_UPD, 2\$		
		52	DD	000DB	PUSHL	BIT_NUMBER	1185	
		FB	A4	9F	PUSHAB	UPD_VAL_STR		
65		02	FB	000E0	CALLS	#2, PAT\$FAO_PUT		
		52	D6	000E3	2\$:	INCL	BIT_NUMBER	1186
		DF	11	000E5	BRB	1\$	1180	
		58	DD	000E7	3\$:	PUSHL	R8	1188
		04	AE	9F	PUSHAB	OUT_BUFFER		
7E		63	01	C3	SUBL3	#1, PAT\$GL_BUF_SIZ, -(SP)		
	00000000G	66	03	FB	CALLS	#3, PAT\$WRITEFILE		
	00000000G	EF	00	FB	CALLS	#0, PAT\$GET_IHD	1194	
		EF	00	FB	CALLS	#0, PAT\$FIND_DST	1199	
		69	01	90	MOVAB	#1, PAT\$GB_TAKE_CMD	1206	
	00000000G	09	69	E9	BLBC	PAT\$GB_TAKE_CMD, 6\$	1207	
		EF	00	FB	CALLS	#0, PAT\$COM_PROC	1209	
	00000000G	EF	F4	11	BRB	5\$		
		00	FB	00110	CALLS	#0, PAT\$CLOSEFILES	1214	
			04	00117	RET		1216	
			0000	00118	7\$:	.WORD	Save nothing	1119
		7E	D4	0011A	CLRL	-(SP)		
		5E	DD	0011C	PUSHL	SP		
	00000000G	7E	AC	7D	MOVQ	4(AP), -(SP)		
		EF	03	FB	CALLS	#3, PAT\$ERR_HANDLER		
			04	00129	RET			

; Routine Size: 298 bytes, Routine Base: _PAT\$CODE + 0000

; 411 1217 1

```

: 413 1218 1 ROUTINE PROCESS_COMMAND : NOVALUE =
: 414 1219 1 ++
: 415 1220 1 Functional Description:
: 416 1221 1
: 417 1222 1     Call slave procedures to process specific portions
: 418 1223 1     of the command line.
: 419 1224 1
: 420 1225 1 Implicit Outputs:
: 421 1226 1
: 422 1227 1     The command line and its associated qualifiers
: 423 1228 1     have been parsed and processed.
: 424 1229 1
: 425 1230 1 Side Effects:
: 426 1231 1
: 427 1232 1     Errors will be SIGNALLed by the CLI.
: 428 1233 1
: 429 1234 1 --
: 430 1235 1
: 431 1236 2 BEGIN
: 432 1237 2
: 433 1238 2 CH$FILL (0, dsc$c_s_bln, input_desc); ! Initialize the input string descriptor.
: 434 1239 2 input_desc [dsc$b_dftype]=dsc$k_dtype_t; ! Define the data type as text.
: 435 1240 2 input_desc [dsc$b_class]=dsc$k_class_d; ! Define the descriptor as a dynamic string descriptor.
: 436 1241 2
: 437 1242 2 open_input (); ! Obtain the INPUT file spec and open the file, process /ABS and /NE
: 438 1243 2 open_output (); ! Obtain the OUTPUT file spec and open the file.
: 439 1244 2 open_journal (); ! Obtain the JOURNAL file spec and open the file.
: 440 1245 2 volume_parse (); ! Process the /VOLUME qualifier.
: 441 1246 2 update_parse (); ! Process the /UPDATE qualifier.
: 442 1247 2 IF CLIS$PRESENT (command) ! If /COMMAND has been specified...
: 443 1248 2 THEN pat$open_comfil (1); ! obtain the COMMAND file spec and open the file (if any).
: 444 1249 2
: 445 1250 1 END; ! of ROUTINE process_command

```

				003C 0000	PROCESS_COMMAND:			
08	00	6E		00 2C 00002	.WORD	Save R2,R3,R4,R5		: 1218
			00000000'	EF 00007	MOVCS	#0, (SP), #0, #8, INPUT_DESC		: 1238
			00000000' EF 020E	8F B0 0000C	MOVW	#526, INPUT_DESC+2		: 1239
			00000000V EF	00 FB 00015	CALLS	#0, OPEN_INPUT		: 1242
			00000000V EF	00 FB 0001C	CALLS	#0, OPEN_OUTPUT		: 1243
			00000000V EF	00 FB 00023	CALLS	#0, OPEN_JOURNAL		: 1244
			00000000V EF	00 FB 0002A	CALLS	#0, VOLUME_PARSE		: 1245
			00000000V EF	00 FB 00031	CALLS	#0, UPDATE_PARSE		: 1246
			00000000' EF	9F 00038	PUSHAB	COMMAND		: 1247
		00		01 FB 0003E	CALLS	#1, CLIS\$PRESENT		
		09		50 E9 00045	BLBC	R0, 1\$		
				01 DD 00048	PUSHL	#1		: 1248
			00000000V EF	01 FB 0004A	CALLS	#1, PAT\$OPEN_COMFIL		: 1248
				04 00051 1\$:	RET			: 1250

; Routine Size: 82 bytes, Routine Base: _PAT\$CODE + 012A

PATMAIN
V04-000

G 12
16-Sep-1984 00:03:48
14-Sep-1984 12:52:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATMAI.B32;1 Page 15 (4)

: 446

1251 1

P
VI

```

448 1252 1 ROUTINE OPEN_INPUT : NOVALUE = ! ROUTINE TO OPEN INPUT FILE
449 1253 1
450 1254 1
451 1255 1 ++
452 1256 1 FUNCTIONAL DESCRIPTION:
453 1257 1 PROCESS THE INPUT FILE PARAMETER IN THE COMMAND LINE.
454 1258 1 THE FAB IS SET UP AND THE INPUT IMAGE FILE OPENED.
455 1259 1
456 1260 1 IMPLICIT INPUTS:
457 1261 1
458 1262 1 CLI HAS THE COMMAND LINE.
459 1263 1 THE FABs AND RABs ARE INITIALIZED, SAVE FOR FILE NAMES.
460 1264 1
461 1265 1 SIDE EFFECTS:
462 1266 1
463 1267 1 THE INPUT FILE IS OPENED, AND MAY BE CLOSED DUE
464 1268 1 TO THE USER'S REQUEST TO PATCH THE FILE IN PLACE
465 1269 1 VIA A COMBINATION OF THE COMMAND QUALIFIERS
466 1270 1 /ABSOLUTE AND /NEW_VERSION.
467 1271 1
468 1272 1 ALSO, DECLARE AN EXIT HANDLER WHICH WILL WARN THE USER
469 1273 1 THAT UPON IMAGE RUNDOWN THE FILE BEING PATCHED IS OVERWRITTEN.
470 1274 1 THIS IS A DIRECT RESULT OF SECTION MAPPING, AND THE PATCH
471 1275 1 UPDATE COMMAND IS MORE OF A CHECK POINT OPERATION (EG. WRITE
472 1276 1 WHAT I'VE DONE SO FAR, BACK TO THE SECTION FILE) THAN ANYTHING
473 1277 1 ELSE.
474 1278 1
475 1279 1 --
476 1280 1
477 1281 2 BEGIN
478 1282 2 PAT$GL_FLAGS [PAT$$ABSOLUTE] = CLIPRESENT (absolute); ! See if we should patch via absolute (virtu
479 1283 2 PAT$GL_FLAGS [PAT$$NEW_VERSION] = CLIPRESENT (new_ver); ! See if a new version of the file should be
480 1284 2 PAT$GL_FLAGS [PAT$$INPUT] = CLIPRESENT (input_desc); ! Get the INPUT file specification.
481 1285 2 PAT$GL_OLDFAB[FAB$L_FNA]=.INPUT_DESC[DSC$A_POINTER]; ! SET FILE NAME ADDRESS
482 1286 2 PAT$GL_OLDFAB[FAB$L_FNS]=.INPUT_DESC[DSC$W_LENGTH]; ! SET FILE NAME LENGTH
483 1287 2 PAT$GL_ERRCODE=$OPEN(FAB=PAT$GL_OLDFAB); ! OPEN INPUT FILE
484 1288 2 INPUT_DESC[DSC$A_POINTER] = 0; ! Force allocation of a new string buffer.
485 1289 2 IF NOT .PAT$GL_ERRCODE ! SUCCESS ON OPEN?
486 1290 2 THEN
487 1291 3 BEGIN
488 1292 3 PAT$GL_FLAGS [PAT$$INPUT] = 0; ! SET FLAG FILE NOT OPEN
489 1293 3 SIGNAL(PAT$_OPENIN,T,GETFILDSC(PAT$GL_OLDFAB),.PAT$GL_OLDFAB[FAB$L_STS],.PAT$GL_OLDFAB[FAB$L_STV]);
490 1294 3 END
491 1295 2 ELSE
492 1296 3 BEGIN
493 1297 3 PAT$GL_CHANUM = .PAT$GL_OLDFAB [FAB$L_STV]; ! Save channel number
494 1298 3 IF .PAT$GL_FLAGS [PAT$$ABSOLUTE] AND NOT .PAT$GL_FLAGS [PAT$$NEW_VERSION]
495 1299 3 THEN
496 1300 3 ! Patching the file in place. The mapping will take care of accessing the file and updating
497 1301 3 !
498 1302 4 BEGIN
499 1303 4 PAT$GL_ERRCODE = $ASSGN (CHAN=.PAT$GL_CHANUM);
500 1304 4 IF NOT .PAT$GL_ERRCODE
501 1305 4 THEN
502 1306 4 SIGNAL ( PAT$_CLOSEIN, 1, GETFILDSC(PAT$GL_OLDFAB), .PAT$GL_ERRCODE, 0);
503 1307 4 PAT$GL_CHANUM = 0; ! No channel assigned anymore
504 1308 4 PAT$GL_FLAGS [PAT$$INPUT] = 0; ! Set file not open flag.

```

```

: 505      1309  4
: 506      1310  4
: 507      1311  4
: 508      1312  4
: 509      1313  4
: 510      1314  4
: 511      1315  5
: 512      1316  4
: 513      1317  4
: 514      1318  3
: 515      1319  2
: 516      1320  2
: 517      1321  1

```

```

EXIT_CB [0] = 0;
EXIT_CB [1] = EXIT_HANDLER;
EXIT_CB [2] = 1;
EXIT_CB [3] = PAT$GL_ERRCODE;

IF NOT (PAT$GL_ERRCODE = $DCLEXH (DESBK = EXIT_CB)) ! Declare the EXIT handler.
THEN
  SIGNAL (PAT$_SYSERROR, 0, .PAT$GL_ERRCODE);
END;

END;
END; ! END OF OPEN_INPUT

```

```

.EXTRN SYSS$OPEN, SYSS$DASSGN
.EXTRN SYSS$DCLEXH

```

07FC 0000 OPEN_INPUT:

					.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10	: 1252		
		5A	00000000G	EF 9E 00002	MOVAB	GETFILDSC, R10			
		59	00000000G	00 9E 00009	MOVAB	CLIS\$PRESENT, R9			
		58	00000000'	EF 9E 00010	MOVAB	ABSOLUTE, R8			
		57	00000000G	EF 9E 00017	MOVAB	PAT\$GL_CHANUM, R7			
		56	00000000G	00 9E 0001E	MOVAB	LIB\$SIGNAL, R6			
		55	00000000G	EF 9E 00025	MOVAB	PAT\$GL_OLDFAB, R5			
		54	00000000G	EF 9E 0002C	MOVAB	PAT\$GL_FLAGS, R4			
		53	00000000G	EF 9E 00033	MOVAB	PAT\$GL_ERRCODE, R3			
		52	00000000'	EF 9E 0003A	MOVAB	INPUT_DESC, R2			
				58	DD 00041	PUSHL	R8	: 1282	
64	01	69		01	FB 00043	CALLS	#1, CLIS\$PRESENT		
		06		50	FO 00046	INSV	R0, #6, #1, PAT\$GL_FLAGS		
			50	A8	9F 0004B	PUSHAB	NEW_VER	: 1283	
64	01	69		01	FB 0004E	CALLS	#1, CLIS\$PRESENT		
		07		50	FO 00051	INSV	R0, #7, #1, PAT\$GL_FLAGS		
			2C	52	DD 00056	PUSHL	R2	: 1284	
				A8	9F 00058	PUSHAB	FILE		
64	01	00000000G	00	02	FB 0005B	CALLS	#2, CLIS\$GET_VALUE		
			02	50	FO 00062	INSV	R0, #2, #1, PAT\$GL_FLAGS		
		2C	A5	04	A2	DO 00067	MOVL	INPUT_DESC+4, PAT\$GL_OLDFAB+44	: 1285
		34	A5	62	90 0006C	MOVB	INPUT_DESC, PAT\$GL_OLDFAB+52	: 1286	
				55	DD 00070	PUSHL	R5	: 1287	
		00000000G	00	01	FB 00072	CALLS	#1, SYSS\$OPEN		
			63	50	DO 00079	MOVL	R0, PAT\$GL_ERRCODE		
				04	A2	D4 0007C	CLRL	INPUT_DESC+4	: 1288
		1A		63	E8 0007F	BLBS	PAT\$GL_ERRCODE, 1\$: 1289	
		64		04	8A 00082	BICB2	#4, PAT\$GL_FLAGS	: 1292	
		7E		08	A5	7D 00085	MOVQ	PAT\$GL_OLDFAB+8, -(SP)	: 1293
				55	DD 00089	PUSHL	R5		
		6A		01	FB 0008B	CALLS	#1, GETFILDSC		
				50	DD 0008E	PUSHL	R0		
				01	DD 00090	PUSHL	#1		
			00000000G	8F	DD 00092	PUSHL	#PAT\$ OPENIN		
		66		05	FB 00098	CALLS	#5, LIB\$SIGNAL		
				04	0009B	RET		: 1289	
		67	0C	A5	DO 0009C 1\$:	MOVL	PAT\$GL_OLDFAB+12, PAT\$GL_CHANUM	: 1297	

5E	64	06	E1	000A0	BBC	#6, PAT\$GL_FLAGS, 3\$	1298
		64	95	000A4	TSTB	PAT\$GL_FLAGS	
		5A	19	000A6	BLSS	3\$	
		67	DD	000A8	PUSHL	PAT\$GL_CHANUM	1303
00000000G	00	01	FB	000AA	CALLS	#1, SYSSDASSGN	
	63	50	D0	000B1	MOVL	R0, PAT\$GL_ERRCODE	
	16	63	E8	000B4	BLBS	PAT\$GL_ERRCODE, 2\$	1304
		7E	D4	000B7	CLRL	-(SP)	1306
		63	DD	000B9	PUSHL	PAT\$GL_ERRCODE	
		55	DD	000BB	PUSHL	R5	
	6A	01	FB	000BD	CALLS	#1, GETFILDSC	
		50	DD	000C0	PUSHL	R0	
		01	DD	000C2	PUSHL	#1	
	00000000G	8F	DD	000C4	PUSHL	#PAT\$ CLOSEIN	
	66	05	FB	000CA	CALLS	#5, LIB\$SIGNAL	
		67	D4	000CD	CLRL	PAT\$GL_CHANUM	1307
	64	04	8A	000CF	BICB2	#4, PAT\$GL_FLAGS	1308
		A2	D4	000D2	CLRL	EXIT_CB	1310
	F0	EF	9E	000D5	MOVAB	EXIT_HANDLER, EXIT_CB+4	1311
F4	A2	01	D0	000DD	MOVL	#1, EXIT_CB+8	1312
F8	A2	63	9E	000E1	MOVAB	PAT\$GL_ERRCODE, EXIT_CB+12	1313
FC	A2	A2	9F	000E5	PUSHAB	EXIT_CB	1315
	F0	01	FB	000E8	CALLS	#1, SYSSDCLEXH	
00000000G	00	50	D0	000EF	MOVL	R0, PAT\$GL_ERRCODE	
	63	50	E8	000F2	BLBS	R0, 3\$	
	0D	63	DD	000F5	PUSHL	PAT\$GL_ERRCODE	1317
		7E	D4	000F7	CLRL	-(SP)	
	00000000G	8F	DD	000F9	PUSHL	#PAT\$ SYSERROR	
	66	03	FB	000FF	CALLS	#3, LIB\$SIGNAL	
		04	00102	3\$:	RET		1321

; Routine Size: 259 bytes. Routine Base: _PAT\$CODE + 017C

; 518 1322 1

```

: 520      1323 1 ROUTINE EXIT_HANDLER (status_adr) : NOVALUE =
: 521      1324 1 ++
: 522      1325 1
: 523      1326 1 Functional Description:
: 524      1327 1
: 525      1328 1 Process exit conditions when patching a file in absolute mode
: 526      1329 1 AND in place (/ABSOLUTE /NONEW_VERSION). The only action for
: 527      1330 1 this routine currently is to issue an informational message to
: 528      1331 1 the user indicating that the file being pathed is being overwritten.
: 529      1332 1 This is a direct result of the normal handling of mapped disk sections.
: 530      1333 1
: 531      1334 1 Inputs:
: 532      1335 1
: 533      1336 1 status_adr      addr.rl      The address of the status longword
: 534      1337 1 indicating the reason for image exit.
: 535      1338 1 (NOTE: This value is ignored!)
: 536      1339 1
: 537      1340 1 Side Effects:
: 538      1341 1
: 539      1342 1 The file being patched may have modified pages which are written
: 540      1343 1 back to the section file file during image rundown.
: 541      1344 1
: 542      1345 1 --
: 543      1346 2 BEGIN
: 544      1347 2
: 545      1348 2 LOCAL
: 546      1349 2 OLD_FILE : $BBLOCK [DSC$C_S_BLN];
: 547      1350 2
: 548      1351 2 OLD_FILE [DSC$W_LENGTH] = .PAT$GL_OLDNBK [NAM$B_RSL];
: 549      1352 2 OLD_FILE [DSC$A_POINTER] = PAT$GB_OLDNAME;
: 550      1353 2 SIGNAL (PAT$_OVERLAY, 1, OLD_FILE);
: 551      1354 2
: 552      1355 1 END;      ! of ROUTINE exit_handler

```

```

                                0000 00000 EXIT_HANDLER:
                                .WORD      Save nothing      : 1323
                                5E          08 C2 00002        SUBL2      #8, SP
                                04 6E 00000000G EF 9B 00005    MOVZBW     PAT$GL_OLDNBK+3, OLD_FILE : 1351
                                AE 00000000G EF 9E 0000C    MOVAB     PAT$GB_OLDNAME, OLD_FILE+4 : 1352
                                5E DD 00014        PUSHL     SP : 1353
                                01 DD 00016        PUSHL     #1
                                00000000G 8F DD 00018        PUSHL     #PAT$_OVERLAY
                                00000000G 00 03 FB 0001E    CALLS    #3, LIB$SIGNAL
                                04 00025        RET

```

: Routine Size: 38 bytes, Routine Base: _PAT\$CODE + 027F

: 553 1356 1

```

555 1357 1 ROUTINE OPEN_OUTPUT : NOVALUE = . ROUTINE TO INIT FOR OUTPUT FILE OPEN
556 1358 1
557 1359 1 !++
558 1360 1 FUNCTIONAL D=SCRIPTION:
559 1361 1
560 1362 1     OBTAIN THE OUTPUT FILE PARAMETER IN THE COMMAND LINE.
561 1363 1     THE FAB IS SET UP AND THE NUMBER OF IMAGE FILE BLOCKS IS INITIALIZED.
562 1364 1     THE IMAGE FILE WILL BE OPENED JUST BEFORE IT IS WRITTEN AS THE NUMBER OF
563 1365 1     BLOCKS THAT MUST BE ALLOCATED CONTIGUOUSLY, WILL NOT BE KNOWN UNTIL THEN.
564 1366 1
565 1367 1 IMPLICIT INPUTS:
566 1368 1
567 1369 1     CLI HAS THE COMMAND LINE.
568 1370 1     THE OUTPUT IMAGE FAB ARE INITIALIZED.
569 1371 1
570 1372 1 IMPLICIT OUTPUTS:
571 1373 1
572 1374 1     NONE
573 1375 1
574 1376 1 ROUTINE VALUE:          NONE
575 1377 1
576 1378 1 COMPLETION CODES:
577 1379 1
578 1380 1     NONE
579 1381 1
580 1382 1 SIDE EFFECTS:
581 1383 1
582 1384 1     THE NUMBER OF IMAGE BLOCKS IS INITIALIZED AS IS THE FILENAME IN THE FAB.
583 1385 1
584 1386 1 !--
585 1387 1
586 1388 2 BEGIN
587 1389 2
588 1390 2 IF CLISPRESNT (output)
589 1391 2 THEN
590 1392 2     CLISGET_VALUE (output, input_desc);           ! Get the OUTPUT file specification.
591 1393 2
592 1394 2 PAT$GL_IMGBLKS = .PAT$GL_OLDXABFH[XAB$L EBK] +       ! INITIALIZE NUMBER OF BLOCKS IN FILE
593 1395 2     (.PAT$GL_OLDXABFH[XAB$W FFB] NEQ 0);
594 1396 2 PAT$GL_NEWFAB[FAB$L_FNA]=.INPUT_DESC[DSC$A_POINTER]; ! SET FILE NAME ADDRESS
595 1397 2 PAT$GL_NEWFAB[FAB$B_FNS]=.INPUT_DESC[DSC$W_LENGTH]; ! SET FILE NAME LENGTH
596 1398 2 PAT$GL_FLAGS [PAT$S_OUTPUT] = 0;                   ! SET FLAG FILE NOT OPEN
597 1399 2 INPUT_DESC [DSC$A_POINTER] = 0;                   ! Force allocation of a new buffer
598 1400 2                                                    ! to preserve the OUTPUT file spec.
599 1401 1 END;      ! END OF OPEN_OUTPUT

```

```

                                000C 0000 OPEN_OUTPUT:
                                .WORD      Save R2,R3           : 1357
                                MOVAB      OUTPUT, R3
                                MOVAB      INPUT_DESC, R2
                                PUSHL      R3                   : 1390
                                CALLS     #1, CLISPRESNT
                                BLBC      R0, 1$

```

PATMAIN
V04-000

M 12
16-Sep-1984 00:03:48
14-Sep-1984 12:52:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATMAI.B32;1 Page 21
(7)

		52	DD	0001C		PUSHL	R2		: 1392	
		53	DD	0001E		PUSHL	R3		: 1393	
00000000G	00	02	FB	00020		CALLS	#2, CLISGET_VALUE		: 1394	
		50	D4	00027	1\$:	CLRL	R0		: 1395	
		00000000G	EF	B5	00029	TSTW	PAT\$GL_OLDXABFH+20		: 1396	
		02	13	0002F		BEQL	2\$: 1397	
00000000G	EF	00000000G	FF	40	9E	00033	2\$:	MOVAB	@PAT\$GL_OLDXABFH+16[R0], PAT\$GL_IMGBLKS	: 1398
00000000G	EF		04	A2	D0	0003F		MOVL	INPUT_DESC+4, PAT\$GL_NEWFAB+44	: 1399
00000000G	EF			62	90	00047		MOVB	INPUT_DESC, PAT\$GL_NEWFAB+52	: 1400
00000000G	EF			08	8A	0004E		BICB2	#8, PAT\$GL_FLAGS	: 1401
		04	A2	D4	00055	CLRL	INPUT_DESC+4		: 1402	
				04	00058	RET			: 1403	

: Routine Size: 89 bytes, Routine Base: _PAT\$CODE + 02A5

: 600 1402 1

```

1403 1 ROUTINE OPEN_JOURNAL : NOVALUE = ! ROUTINE TO OPEN JOURNAL FILE
1404 1
1405 1 ++
1406 1 FUNCTIONAL DESCRIPTION:
1407 1
1408 1 PROCESS THE JOURNAL FILE PARAMETER IN THE COMMAND LINE.
1409 1 THE FAB IS SET UP AND THE JOURNAL FILE OPENED.
1410 1
1411 1 IMPLICIT JOURNALS:
1412 1
1413 1 CLI HAS THE COMMAND LINE.
1414 1 THE JOURNAL FAB AND RAB ARE INITIALIZED.
1415 1
1416 1 IMPLICIT OUTPUTS:
1417 1
1418 1 NONE
1419 1
1420 1 ROUTINE VALUE: NONE
1421 1
1422 1 COMPLETION CODES:
1423 1
1424 1 NONE
1425 1
1426 1 SIDE EFFECTS:
1427 1
1428 1 THE JOURNAL FILE IS OPENED.
1429 1
1430 1 --
1431 1
1432 2 BEGIN
1433 2
1434 2 ! If the /JOURNAL qualifier is present, get the file specification.
1435 2
1436 2 IF CLISPRESNT (journal)
1437 2 THEN
1438 2 PAT$GL_FLAGS [PAT$$_JOURNAL] = CLISGET_VALUE (journal, input_desc);
1439 2
1440 2 PAT$GL_JNLFAB[FAB$FNA]=.INPUT_DESC [DSC$A_POINTER]; ! SET FILE NAME ADDRESS
1441 2 PAT$GL_JNLFAB[FAB$B_FNS]=.INPUT_DESC [DSC$W_LENGTH]; ! SET FILE NAME LENGTH
1442 2 PAT$GL_ERRCODE=$CREATE(FAB=PAT$GL_JNLFAB); ! OPEN JOURNAL FILE
1443 2 IF NOT .PAT$GL_ERRCODE ! SUCCESS ON OPEN?
1444 2 THEN
1445 2 BEGIN
1446 2 PAT$GL_FLAGS [PAT$$_JOURNAL] = 0; ! SET FLAG FILE NOT OPEN
1447 2 SIGNAL(PAT$ OPENOUT,1,GETFILDSC(PAT$GL_JNLFAB),.PAT$GL_JNLFAB[FAB$FNA],.PAT$GL_JNLFAB[FAB$B_FNS]);
1448 2 END
1449 2 ELSE
1450 2 BEGIN
1451 2 PAT$GL_ERRCODE=$CONNECT(RAB=PAT$GL_JNLRAB); ! CONNECT JOURNAL FILE
1452 2 IF NOT .PAT$GL_ERRCODE ! SUCCESS ON CONNECT?
1453 2 THEN SIGNAL(PAT$ OPENOUT,1,GETFILDSC(PAT$GL_JNLFAB),.PAT$GL_JNLRAB[RAB$FNA],.PAT$GL_JNLRAB[RAB$B_FNS]);
1454 2 ELSE PAT$GL_FLAGS [PAT$$_JOURNAL] = 1; ! SET FILE OPEN FLAG, DATA STREAM ACTIVE.
1455 2 END;
1456 1 END; ! END OF OPEN_JOURNAL

```


.EXTRN SYS\$CREATE, SYS\$CONNECT

		00FC 00000 OPEN_JOURNAL:				
				.WORD	Save R2,R3,R4,R5,R6,R7	1403
		57	00000000'	EF 9E 00002	MOVAB JOURNAL, R7	
		56	00000000G	EF 9E 00009	MOVAB PAT\$GL_JNLRAB+12, R6	
		55	00000000G	EF 9E 00010	MOVAB PAT\$GL_FLAGS, R5	
		54	00000000'	EF 9E 00017	MOVAB INPUT_DESC, R4	
		53	00000000G	EF 9E 0001E	MOVAB PAT\$GL_ERRCODE, R3	
		52	00000000G	EF 9E 00025	MOVAB PAT\$GL_JNLFAB, R2	
				57	DD 0002C	PUSHL R7
	00000000G	00		01	FB 0002E	CALLS #1, CLISPRESENT
		10		50	E9 00035	BLBC R0, 1\$
				54	DD 00038	PUSHL R4
				57	DD 0003A	PUSHL R7
	00000000G	00		02	FB 0003C	CALLS #2, CLISGET_VALUE
65	01	01		50	FO 00043	INSV R0, #1, #1, -PAT\$GL_FLAGS
		2C	A2	04	A4	DO 00048 1\$:
		34	A2		64	90 0004D
					52	DD 00051
	00000000G	00		01	FB 00053	CALLS #1, SYS\$CREATE
		63		50	DO 0005A	MOVL R0, PAT\$GL_ERRCODE
		0A		63	E8 00C5D	BLBS PAT\$GL_ERRCODE, 2\$
		65		02	8A 00060	BICB2 #2, PAT\$GL_FLAGS
				66	DD 00063	PUSHL PAT\$GL_JNLRAB+12
			08	A2	DD 00065	PUSHL PAT\$GL_JNLFAB+8
				14	11 00068	BRB 3\$
			F4	A6	9F 0006A 2\$:	PUSHAB PAT\$GL_JNLRAB
	00000000G	00		C1	FB 0006D	CALLS #1, SYS\$CONNECT
		63		50	DO 00074	MOVL R0, PAT\$GL_ERRCODE
		1F		63	E8 00077	BLBS PAT\$GL_ERRCODE, 4\$
		7E	FC	A6	7D 0007A	MOVQ PAT\$GL_JNLRAB+8, -(SP)
				52	DD 0007E 3\$:	PUSHL R2
	00000000G	EF		01	FB 00080	CALLS #1, GETFILDSC
				50	DD 00087	PUSHL R0
				01	DD 00089	PUSHL #1
	00000000G	00	00000000G	8F	DD 0008B	PUSHL #PAT\$ OPENOUT
				05	FB 00091	CALLS #5, LIB\$SIGNAL
				04	00098	RET
		65		02	88 00099 4\$:	BISB2 #2, PAT\$GL_FLAGS
				04	0009C	RET

: Routine Size: 157 bytes, Routine Base: _PAT\$CODE + 02FE

: 656 1457 1

```

658 1458 1 GLOBAL ROUTINE PAT$OPEN_COMFIL (COMMAND_LEVEL) : NOVALUE = ! ROUTINE TO OPEN COMMAND FILE
659 1459 1
660 1460 1 ++
661 1461 1 FUNCTIONAL DESCRIPTION:
662 1462 1
663 1463 1 PROCESS THE COMMAND FILE PARAMETER IN THE COMMAND LINE.
664 1464 1 THE FAB IS SET UP AND THE COMMAND FILE OPENED.
665 1465 1
666 1466 1 IT IS ALSO SERVING TO OPEN THE COMMAND FILE BECAUSE OF THE "CREATE" COMMAND.
667 1467 1 IN THIS CASE, THE INPUT PARAMETERS ARE ZEROS. THERE SHOULD BE ONLY ONE "CREATE"
668 1468 1 COMMAND PER PATCH SESSION. THE "CREATE" COMMAND MAY OR MAY NOT HAVE ONE
669 1469 1 ARGUMENT WHICH IS THE COMMAND FILE SPECIFICATION.
670 1470 1
671 1471 1 FORMAL PARAMETERS:
672 1472 1
673 1473 1 command_level val 0 Implies the PATCH CREATE command.
674 1474 1 1 Indicates that we are processing the command line.
675 1475 1
676 1476 1 IMPLICIT COMMANDS:
677 1477 1
678 1478 1 CLI HAS THE COMMAND LINE.
679 1479 1 THE OUTPUT COMMAND FAB AND RAB ARE INITIALIZED.
680 1480 1
681 1481 1 IMPLICIT OUTPUTS:
682 1482 1
683 1483 1 NONE
684 1484 1
685 1485 1 ROUTINE VALUE:
686 1486 1
687 1487 1 NONE
688 1488 1
689 1489 1 COMPLETION CODES:
690 1490 1
691 1491 1 NONE
692 1492 1
693 1493 1 SIDE EFFECTS:
694 1494 1
695 1495 1 THE COMMAND FILE IS OPENED.
696 1496 1
697 1497 1 --
698 1498 1
699 1499 2 BEGIN
700 1500 2
701 1501 2 LOCAL
702 1502 2 POINTER : REF VECTOR[ ,LONG], ! POINTER TO ADDRESS OFFILE NAME
703 1503 2 NAME_DSC : REF VECTOR[ ,BYTE], ! POINTER TO ASCIC FILE NAME
704 1504 2 OUT_BUFFER : VECTOR[TTY_OUT_WIDTH,BYTE]; ! OUTPUT BUFFER FOR JOURNAL MESSAGES
705 1505 2
706 1506 2
707 1507 2 IF .PAT$GL_FLAGS [PAT$S_COMMAND] ! Does a command file already exist?
708 1508 2 THEN ! If so, signal an error indicating
709 1509 3 BEGIN ! one command file per session!
710 1510 3 SIGNAL (PAT$_DUPCOMFIL+MSG$K_WARN,2,.PAT$GL_CCMNBK[NAM$B_RSL],PAT$GB_COMNAME);
711 1511 2 END;
712 1512 2
713 1513 2 POINTER = .PAT$GL_HEAD_LST;
714 1514 3 IF (.POINTER NEQA 0)

```

```

: 715      1515  2 THEN
: 716      1516  3 BEGIN
: 717      1517  3 NAME_DSC = .POINTER[1];           ! GET ADDRESS OF FILE NAME
: 718      1518  3 PAT$GL_COMFAB[FAB$L_FNA]=NAME_DSC[1];       ! SET FILE NAME ADDRESS
: 719      1519  3 PAT$GL_COMFAB[FAB$B_FNS]=.NAME_DSC[0];       ! SET FILE NAME LENGTH
: 720      1520  3 END
: 721      1521  2 ELSE
: 722      1522  3 BEGIN           ! See if a file spec was given
: 723      1523  4 IF (PAT$GL_FLAGS [PAT$$_COMMAND] = CLISGET_VALUE (command, input_desc))
: 724      1524  3 THEN
: 725      1525  4 BEGIN           ! If it was...
: 726      1526  4 PAT$GL_COMFAB[FAB$L_FNA]=.input_desc [dsc$a_pointer]; ! SET FILE NAME ADDRESS
: 727      1527  4 PAT$GL_COMFAB[FAB$B_FNS]=.input_desc [dsc$w_length]; ! SET FILE NAME LENGTH
: 728      1528  4 END
: 729      1529  3 ELSE           ! Otherwise use default...
: 730      1530  4 BEGIN           ! file name from RFLNAM.
: 731      1531  4 PAT$GL_COMFAB[FAB$L_FNA]=0;           ! SET FILE NAME ADDRESS
: 732      1532  4 PAT$GL_COMFAB[FAB$B_FNS]=0;           ! SET FILE NAME LENGTH
: 733      1533  3 END;
: 734      1534  2 END;
: 735      1535  2
: 736      1536  2 PAT$GL_ERRCODE=$CREATE(FAB=PAT$GL_COMFAB); ! OPEN OUTPUT COMMAND FILE
: 737      1537  2 IF NOT .PAT$GL_ERRCODE ! SUCCESS ON OPEN?
: 738      1538  2 THEN
: 739      1539  3 BEGIN
: 740      1540  3 PAT$GL_FLAGS [PAT$$_COMMAND] = 0; ! SET FLAG FILE NOT OPEN
: 741      1541  3 SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_COMFAB),.PAT$GL_COMFAB[FAB$L_STS],.PAT$GL_COMFAB[FAB$L_STV]);
: 742      1542  3 END
: 743      1543  2 ELSE
: 744      1544  3 BEGIN
: 745      1545  3 PAT$GL_ERRCODE=$CONNECT(RAB=PAT$GL_COMRAB); ! CONNECT COMMAND FILE
: 746      1546  3 IF NOT .PAT$GL_ERRCODE ! SUCCESS ON CONNECT?
: 747      1547  3 THEN
: 748      1548  3 SIGNAL(PAT$_OPENOUT,1,GETFILDSC(PAT$GL_COMFAB),.PAT$GL_COMRAB[RAB$L_STS],.PAT$GL_COMRAB[RAB$L_STV])
: 749      1549  3 ELSE
: 750      1550  4 BEGIN
: 751      1551  4 ++
: 752      1552  4 ! Now set the flag indicating the file is open. If the command
: 753      1553  4 ! file was opened due to a CREATE command, then write out the
: 754      1554  4 ! name of the command file to the journal file and write out
: 755      1555  4 ! the DCL PATCH command line to the command file.
: 756      1556  4 --
: 757      1557  4 PAT$GL_FLAGS [PAT$$_COMMAND] = 1; ! SET FILE OPEN BIT
: 758      1558  5 IF (.COMMAND_LEVEL EQL 0)
: 759      1559  4 THEN
: 760      1560  5 BEGIN
: 761      1561  5 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
: 762      1562  5 PAT$GL_BUF_SIZ = 0;
: 763      1563  5 PAT$FAD_PUT(COM_STR, .PAT$GL_COMNBK[NAM$B_RSL], PAT$GB_COMNAME);
: 764      1564  5 PAT$WRITEFILE(.PAT$GL_BUF_SIZ, CH$PTR(OUT_BUFFER, 0), PAT$GL_JNLRAB);
: 765      1565  5 PAT$CP_OUT_STR = CH$PTR(OUT_BUFFER, 0);
: 766      1566  5 PAT$GL_BUF_SIZ = 0;
: 767      1567  5 PAT$GL_COMRAB[RAB$W_RSZ] = .PAT$GL_OLDFAB [FAB$B_FNS],
: 768      1568  5 PAT$GL_COMRAB[RAB$L_RBF] = .PAT$GL_OLDFAB [FAB$L_FNA];
: 769      1569  6 IF NOT (PAT$GL_ERRCODE = $PUT(RAB=PAT$GL_COMRAB))
: 770      1570  5 THEN
: 771      1571  5 SIGNAL(PAT$_WRITEERR,1,GETFILDSC(PAT$GL_COMFAB),
```


		D4	A2	9F	000C9	6\$:	PUSHAB	PAT\$GL_COMFAB	:	
6B			01	FB	000CC		CALLS	#1, GETFILDSC	:	
			50	DD	000CF		PUSHL	R0	:	
			01	DD	000D1		PUSHL	#1	:	
	00000000G		8F	DD	000D3		PUSHL	#PAT\$_OPENOUT	:	
			71	11	000D9		BRB	8\$:	
65			01	88	000DB	7\$:	BISB2	#1, PAT\$GL_FLAGS	:	1557
		04	AC	D5	000DE		TSTL	COMMAND_LEVEL	:	1558
			6C	12	000E1		BNEQ	9\$:	
00000000G	EF		6E	9E	000E3		MOVAB	OUT_BUFFER, PAT\$CP_OUT_STR	:	1561
			67	D4	000EA		CLRL	PAT\$GL_BUF_SIZ	:	1562
			58	DD	000EC		PUSHL	R8	:	1563
7E			69	9A	000EE		MOVZBL	PAT\$GL_COMNBK+3, -(SP)	:	
00000000G	EF	00000000'	EF	9F	000F1		PUSHAB	COM_STR	:	
			03	FB	000F7		CALLS	#3, -PAT\$FAO_PUT	:	
	00000000G		EF	9F	000FE		PUSHAB	PAT\$GL_JNLRAB	:	1564
		04	AE	9F	00104		PUSHAB	OUT_BUFFER	:	
			67	DD	00107		PUSHL	PAT\$GL_BUF_SIZ	:	
00000000G	EF		03	FB	00109		CALLS	#3, PAT\$WRITEFILE	:	
00000000G	EF		6E	9E	00110		MOVAB	OUT_BUFFER, PAT\$CP_OUT_STR	:	1565
			67	D4	00117		CLRL	PAT\$GL_BUF_SIZ	:	1566
22	A3	00000000G	EF	9B	00119		MOVZBW	PAT\$GL_OLDFAB+52, PAT\$GL_COMRAB+34	:	1567
28	A3	00000000G	EF	D0	00121		MOVL	PAT\$GL_OLDFAB+44, PAT\$GL_COMRAB+40	:	1568
			53	DD	00129		PUSHL	R3	:	1569
00000000G	00		01	FB	0012B		CALLS	#1, SYSSPUT	:	
	64		50	D0	00132		MOVL	R0, PAT\$GL_ERRCODE	:	
	17		50	E8	00135		BLBS	R0, 9\$:	
	7E		A3	7D	00138		MOVQ	PAT\$GL_COMRAB+8, -(SP)	:	1572
			A2	9F	0013C		PUSHAB	PAT\$GL_COMFAB	:	1571
6B			01	FB	0013F		CALLS	#1, GETFILDSC	:	
			50	DD	00142		PUSHL	R0	:	
			01	DD	00144		PUSHL	#1	:	
	00000000G		8F	DD	00146		PUSHL	#PAT\$_WRITEERR	:	
6A			05	FB	0014C	8\$:	CALLS	#5, LIB\$SIGNAL	:	
			04	0014F	9\$:		RET		:	1576

; Routine Size: 336 bytes, Routine Base: _PAT\$CODE + 039B

; 777 1577 1

```

779 1578 1 ROUTINE VOLUME_PARSE : NOVALUE =
780 1579 1
781 1580 1 !++
782 1581 1 ! FUNCTIONAL DESCRIPTION:
783 1582 1
784 1583 1 ! Obtain a Relative Volume Number if specified on the command line,
785 1584 1 ! or when the VOLUME qualifier is specified without a value, to
786 1585 1 ! default the value from the old image.
787 1586 1
788 1587 1 ! IMPLICIT INPUTS:
789 1588 1
790 1589 1 ! PAT$GW_IMGVOL Relative Volume Number.
791 1590 1
792 1591 1 ! IMPLICIT OUTPUTS:
793 1592 1
794 1593 1 ! PAT$GW_IMGVOL is loaded with the appropriate Relative Volume Number.
795 1594 1
796 1595 1 ! ROUTINE VALUE:
797 1596 1
798 1597 1 ! NONE
799 1598 1
800 1599 1 ! COMPLETION CODES:
801 1600 1
802 1601 1 ! NONE
803 1602 1
804 1603 1 ! --
805 1604 2 BEGIN
806 1605 2 LOCAL
807 1606 2 volume_desc : $BBLOCK [dsc$c_s_bln], ! Declare a local descriptor for converting
808 1607 2 volume_number : INITIAL (0); ! the volume number specified from the comma
809 1608 2
810 1609 2 BIND
811 1610 2 OLD_FILE_ID = PAT$GL_OLDNBK[NAM$W_FID] : VECTOR [,WORD];
812 1611 2
813 1612 2 CH$FILL (0, dsc$c_s_bln, volume_desc); ! Initialize the descriptor.
814 1613 2 volume_desc [dsc$b_dtype] = dsc$k_dtype_lu; ! Data type is Longword Unsigned.
815 1614 2 volume_desc [dsc$b_class] = dsc$k_class_s; ! Scalar class.
816 1615 2 volume_desc [dsc$a_pointer] = volume_number; ! Set address of scalar.
817 1616 2
818 1617 2 IF CL$PRESENT (volume)
819 1618 2 THEN
820 1619 3 IF (PAT$GL_FLAGS [PAT$S_VOLUME] = CL$GET_VALUE (volume, input_desc))
821 1620 3 THEN
822 1621 3 BEGIN
823 1622 3 IF LIB$CVT_DX DX (input_desc, volume_desc) ! Convert the volume number,
824 1623 3 THEN PAT$GW_IMGVOL = .volume_number ! If its ok, then load the user supplied num
825 1624 3 ELSE PAT$GW_IMGVOL = .OLD_FILE_ID[2]; ! otherwise, default to the old Rel. Vol. No
826 1625 3 END
827 1626 2 ELSE
828 1627 2 PAT$GW_IMGVOL = .OLD_FILE_ID[2]; ! Default to the old Rel. Vol. No.
829 1628 2
830 1629 1 END; ! of ROUTINE volume_parse

```

		01FC 00000		VOLUME_PARSE:					
		58	00000000'	EF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	: 1578
		57	00000000G	EF	9E	00009	MOVAB	VOLUME, R8	:
		56	00000000'	EF	9E	00010	MOVAB	PAT\$GW_IMGVOL, R7	:
		5E		08	C2	00017	MOVAB	INPUT_DESC, R6	:
				7E	D4	0001A	SUBL2	#8, SP	:
08	00	6E		00	2C	0001C	CLRL	VOLUME_NUMBER	: 1604
			04	AE		00021	MOVCS	#0, (SP), #0, #8, VOLUME_DESC	: 1612
	06	AE	0104	8F	B0	00023		#260, VOLUME_DESC+2	: 1613
	08	AE		6E	9E	00029	MOVAB	VOLUME_NUMBER, VOLUME_DESC+4	: 1615
				58	DD	0002D	PUSHL	R8	: 1617
	00000000G	00		01	FB	0002F	CALLS	#1, CLISPRESENT	:
		31		50	E9	00036	BLBC	P0, 2\$:
				56	DD	00039	PUSHL	R6	: 1619
				58	DD	0003B	PUSHL	R8	:
00000000G	EF	00		02	FB	0003D	CALLS	#2, CLISGET_VALUE	:
		05		50	F0	00044	INSV	R0, #5, #1, -PAT\$GL_FLAGS	:
		13		50	E9	0004D	BLBC	R0, 1\$:
			04	AE	9F	00050	PUSHAB	VOLUME_DESC	: 1622
				56	DD	00053	PUSHL	R6	:
	00000000G	00		02	FB	00055	CALLS	#2, LIB\$CVT_DX_DX	:
		04		50	E9	0005C	BLBC	R0, 1\$:
		67		6E	B0	0005F	MOVW	VOLUME_NUMBER, PAT\$GW_IMGVOL	: 1623
				04		00062	RET		:
		67	00000000G	EF	B0	00063	MOVW	OLD_FILE_ID+4, PAT\$GW_IMGVOL	: 1627
				04		0006A	RET		: 1629

; Routine Size: 107 bytes, Routine Base: _PAT\$CODE + 04EB

; 831 1630 1

```

833 1631 1 ROUTINE UPDATE_PARSE : NOVALUE =
834 1632 1 +-
835 1633 1 Functional Description:
836 1634 1
837 1635 1 Obtain the values associated with the /UPDATE qualifier (if any).
838 1636 1
839 1637 1 Implicit Inputs:
840 1638 1
841 1639 1 Implicit Outputs:
842 1640 1
843 1641 1 Side Effects:
844 1642 1
845 1643 1 --
846 1644 2 BEGIN
847 1645 2 LOCAL
848 1646 2     eco_desc : $BBLOCK [dsc$c_s_bln],           ! Declare a local descriptor for converting
849 1647 2     eco_number : INITIAL (0),                 ! eco level(s) specified with the /UPDATE qu
850 1648 2     status;                                       ! Local status.
851 1649 2
852 1650 2 CHSFILL (0, dsc$c_s_bln, eco_desc);         ! Initialize the descriptor.
853 1651 2 eco_desc [dsc$b_dtype] = dsc$k_dtype_lu;    ! Data type is Longword Unsigned.
854 1652 2 eco_desc [dsc$b_class] = dsc$k_class_s;    ! Scalar class.
855 1653 2 eco_desc [dsc$a_pointer] = eco_number;      ! Point to scalar.
856 1654 2
857 1655 2 PAT$GL_FLAGS [PAT$$UPDATE] = status = CLISPRESENT (update); ! Was /UPDATE specified?
858 1656 2
859 1657 2 IF .PAT$GL_FLAGS [PAT$$UPDATE]                ! If it was then don't process any commands
860 1658 2 THEN     PAT$GB_EXEC_CMD = FALSE             ! we see the first SET ECO command.
861 1659 2 ELSE     PAT$GB_EXEC_CMD = TRUE;            ! Otherwise, process each command immediatel
862 1660 2
863 1661 2 WHILE .status DO                               ! While there are ECO level(s) still to proc
864 1662 3     BEGIN                                       ! get the eco number (ascii text) and conver
865 1663 4     IF (status = CLISGET_VALUE (update, input_desc)) ! convert it to its equivalent number.
866 1664 3     THEN                                       ! Once this is done, set the bit in the ECO
867 1665 4     BEGIN
868 1666 5     IF NOT (PAT$GL_ERRCODE = LIB$CVT_DX_DX (input_desc, eco_desc))
869 1667 4     THEN     SIGNAL (PAT$NUMVAL, 1, .PAT$GL_ERRCODE); ! Numeric conversion errors are signalled.
870 1668 4     PAT$GL_ECO_UPD <.eco_number-1,1> = 1;
871 1669 3     END;
872 1670 2     END;
873 1671 2
874 1672 1 END; ! of ROUTINE update_parse

```

```

                                07FC 00000 UPDATE_PARSE:
                                .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9,R10      : 1631
08      00      5A 00000000' EF 9E 00002      MOVAB    UPDATE, R10
                                59 00000000G EF 9E 00009      MOVAB    PAT$GL_ERRCODE, R9
                                58 00000000' EF 9E 00010      MOVAB    INPUT_DESC, R8
                                57 00000000G EF 9E 00017      MOVAB    PAT$GB_EXEC_CMD, R7
                                56 00000000G EF 9E 0001E      MOVAB    PAT$GL_FLAGS, R6
                                5E              08 C2 00025      SUBL2   #8, SP
                                7E D4 00028      CLRL    ECO_NUMBER
                                6E              00 2C 0002A      MOVCS   #0, (SP), #0, #8, ECO_DESC      : 1644
                                                : 1650

```

0022

			04	AE	0002F						
	06	AE	0104	8F	B0	00031	MOVW	#260, ECO_DESC+2		1651	
	08	AE		6E	9E	00037	MOVAB	ECO_NUMBER, ECO_DESC+4		1653	
				5A	DD	0003B	PUSHL	R10		1655	
	00000000G	00		01	FB	0003D	CALLS	#1, CLISPRESENT			
66		52		50	DO	00044	MOVL	R0, STATUS			
	01	04		52	FO	00047	INSV	STATUS, #4, #1, PAT\$GL_FLAGS			
	04	66		04	E1	0004C	BBC	#4, PAT\$GL_FLAGS, 1\$		1657	
				67	94	00050	CLRB	PAT\$GB_EXEC_CMD		1658	
				03	11	00052	BRB	2\$			
				67	01	90	00054	1\$: MOVB	#1, PAT\$GB_EXEC_CMD	1659	
				42	52	E9	00057	2\$: BLBC	STATUS, 4\$	1661	
					58	DD	0005A	PUSHL	R8	1663	
	00000000G	00		5A	DD	0005C	PUSHL	R10			
		52		02	FB	0005E	CALLS	#2, CLISGET_VALUE			
		31		50	DO	00065	MOVL	R0, STATUS			
			04	52	E9	00068	BLBC	STATUS, 4\$			
				AE	9F	0006B	PUSHAB	ECO_DESC		1666	
	00000000G	00		58	DD	0006E	PUSHL	R8			
		69		02	FB	00070	CALLS	#2, LIB\$CVT_DX_DX			
		11		50	DO	00077	MOVL	R0, PAT\$GL_ERRCODE			
				50	E8	0007A	BLBS	R0, 3\$			
				69	DD	0007D	PUSHL	PAT\$GL_ERRCODE		1667	
				01	DD	0007F	PUSHL	#1			
	00000000G	00	006D81A4	8F	DD	00081	PUSHL	#7176612			
		6E		03	FB	00087	CALLS	#3, LIB\$SIGNAL			
50				01	C3	0008E	3\$: SUBL3	#1, ECO_NUMBER, R0		1668	
BD	00000000G	EF		50	E2	00092	BBSS	R0, PAT\$GL_ECO_UPD, 2\$			
				BB	11	0009A	BRB	2\$		1661	
				04	0009C	4\$: RET				1672	

: Routine Size: 157 bytes, Routine Base: _PAT\$CODE + 0556

```

: 875          1673  1
: 876          1674  1 END
: 877          1675  0 ELUDOM

```

! END OF MODULE

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
-PAT\$OWN	24	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
-PAT\$PLIT	307	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(0)
-PAT\$CODE	1523	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
-ABS	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

PATMAIN
V04-000

K 13
16-Sep-1984 00:03:48
14-Sep-1984 12:52:39

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

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	36	0	581	00:01.0
_\$255\$DUA28:[SYSLIB]CLIMAC.L32;1	14	0	0	9	00:00.0

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

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS\$:PATMAI/OBJ=OBJS:PATMAI MSRCS:PATMAI/UPDATE=(ENHS:PATMAI)

: Size: 1523 code + 331 data bytes
: Run Time: 00:31.3
: Elapsed Time: 01:59.1
: Lines/CPU Min: 3210
: Lexemes/CPU-Min: 35415
: Memory Used: 182 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 messages, command prompts, and data listings. The overall appearance is that of a dense collection of system logs or command outputs from a VAX/VMS environment.