

The `eventB` package*

Thai Son Hoang
ETH-Zurich
<htson at inf dot ethz dot ch>

February 22, 2013

Abstract

This class provides a template for typesetting Event-B models. It was developed at the Swiss Federal Institute of Technology Zurich (ETH-Zurich).

Contents

1	Introduction	1
2	Usage	1
2.1	Package Options	1
3	Implementation	1
3.1	Package Loading	1
3.2	Declaration of Options for the Package	2
3.3	Commands for Pretty-Print Event-B Models	4

1 Introduction

This package was developed in order to ease the typesetting of Event-B models in \LaTeX .

2 Usage

See `sample-eventB.tex` for an example of how to use the package.

2.1 Package Options

The package offers the following options:

- `nobox`: to disable to bounding boxes for the Event-B modelling elements,
- `small`, `compact`, `tiny`: options for font size,
- `colour` (or `color`): to colour several modelling elements.

*This document corresponds to `eventB` v1.1, dated 2012/02/21.

3 Implementation

3.1 Package Loading

We begin by loading the required package `xspace` and `xcolor`.

```
1 \RequirePackage{xspace}
2 \RequirePackage{xcolor}
```

3.2 Declaration of Options for the Package

In this part various options for the package are defined.

Option for rounding boxes By default, Event-B modelling elements, e.g., invariants, events, etc., are displayed in a rounding box. This option enables them to be displayed without the rounding box.

```
3 \newlength{\B@oldfboxsep}
4 \newcommand{\event}[7][]{
5   \setlength{\B@oldfboxsep}{\fboxsep}
6   \setlength{\fboxsep}{2ex}
7   \fbox{
8     \ensuremath{
9       \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
10    }
11  }
12  \setlength{\fboxsep}{\B@oldfboxsep}
13 }
14
15 \newcommand{\Bdeclaration}[2]{
16   \fbox{
17     \ensuremath{
18       \B@declaration{#1}{#2}
19     }
20   }
21 }
22
23 \newcommand{\Bsection}[3][]{
24   \setlength{\B@oldfboxsep}{\fboxsep}
25   \setlength{\fboxsep}{2ex}
26   \fbox{
27     \ensuremath{
28       \B@section[#1]{#2}{#3}
29     }
30   }
31   \setlength{\fboxsep}{\B@oldfboxsep}
32 }
33
34 \DeclareOption{nobox}{
35   \renewcommand{\event}[7][]{
36     \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
37   }
38
39   \renewcommand{\Bdeclaration}[2]{
40     \B@declaration{#1}{#2}
```



```

83 \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
84 \setBLabelColour{green!50!black}
85 \newcommand{\setBP0Colour}[1]{\colorlet{B@pocolor}{#1}}
86 \setBP0Colour{red}
87 \renewcommand{\Bkeyword}[1]{
88   \ensuremath{\textcolor{B@keywordcolor}{\B@keyword{#1}}}\xspace
89 }
90 \renewcommand{\Bidentifier}[1]{
91   \ensuremath{\textcolor{B@identifiercolor}{\B@identifier{#1}}}\xspace
92 }
93 \renewcommand{\Blabel}[2][]{
94   \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
95 }
96 \renewcommand{\Bpo}[1]{
97   \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
98 }
99 }
100 \DeclareOption{color}{
101   \ExecuteOptions{colour}
102 }
103

```

After declaration of options, we execute them accordingly.

```

104 \ProcessOptions

```

3.3 Commands for Pretty-Print Event-B Models

We start with the definition of the `\eventB` macro.

```

105 \newcommand{\eventB}{Event-B\xspace}

```

The `Bcode` environment for displaying Event-B models. The environment has an optional argument for specifying the font size. By default, it is the same as the `\B@fontsize` controlled by the package option.

```

106 \newenvironment{Bcode}[1][\B@fontsize]{\begin{center}#1}{\end{center}}

```

Declarations and Collections Event-B modelling elements are organised into declarations (e.g., variables, constants, etc.) or collections (e.g., invariants, axioms). For each declaration, the input is a comma-separated list of elements. For each collection, the input is a newly(`\`)-separated list of elements.

```

107 \newcommand{\carriersets}[1]{
108   \Bdeclaration{sets}{#1}
109 }
110 \newcommand{\constants}[1]{
111   \Bdeclaration{constants}{#1}
112 }
113 \newcommand{\axioms}[2][]{
114   \Bsection{#1}{axioms}{#2}
115 }
116 \newcommand{\variables}[1]{
117   \Bdeclaration{variables}{#1}
118 }

```

```

119 \newcommand{\invariants}[2][]{
120   \Bsection[#1]{invariants}{#2}
121 }
122 \newcommand{\variant}[1]{
123   \Bdeclaration{variant}{#1}
124 }

```

Event-B keywords We define the keywords for pretty-print Event-B models.

```

125 \newcommand{\Bany}{\Bkeyword{any}}
126 \newcommand{\Bbegin}{\Bkeyword{begin}}
127 \newcommand{\Bend}{\Bkeyword{end}}
128 \newcommand{\Brefines}{\Bkeyword{refines}}
129 \newcommand{\Bstatus}{\Bkeyword{status}}
130 \newcommand{\Bthen}{\Bkeyword{then}}
131 \newcommand{\Bwhen}{\Bkeyword{when}}
132 \newcommand{\Bwhere}{\Bkeyword{where}}
133 \newcommand{\Bwith}{\Bkeyword{with}}

```

Event-B modelling elements We define several macros for pretty-print Event-B modelling elements.

```

134 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
135 \newcommand{\Bset}[1]{\Bidentifier{#1}}
136 \newcommand{\Bcst}[1]{\Bidentifier{#1}}
137 \newcommand{\Baxm}[1]{\Blabel{#1}}
138 \newcommand{\Bthm}[1]{\Blabel{thm}{#1}}
139
140 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
141 \newcommand{\Bvrb}[1]{\Bidentifier{#1}}
142 \newcommand{\Binv}[1]{\Blabel{#1}}
143 \newcommand{\Bvt}[1]{\Blabel{#1}}
144 \newcommand{\Bpar}[1]{\Bidentifier{#1}}
145 \newcommand{\Bact}[1]{\Blabel{#1}}
146 \newcommand{\Bgrd}[1]{\Blabel{#1}}
147 \newcommand{\Bbap}[1]{\hbox{\sl\bfseries #1}}

```

Meta-macros for creating macros for modelling elements We define meta-macros to create macros for different modelling elements.

```

148 \newcommand{\B@newmacro}[3][]{
149   \def\input@macro{#1}
150   \ifx\input@macro\empty
151     \expandafter\def\csname #2\endcsname{#3{#2}}
152   \else
153     \expandafter\def\csname #1\endcsname{#3{#2}}
154   \fi
155 }
156 \newcommand{\newBctx}[2][]{\B@newmacro{#1}{#2}{\Bctx}}
157 \newcommand{\newBset}[2][]{\B@newmacro{#1}{#2}{\Bset}}
158 \newcommand{\newBcst}[2][]{\B@newmacro{#1}{#2}{\Bcst}}
159 \newcommand{\newBaxm}[2][]{\B@newmacro{#1}{#2}{\Baxm}}

```

```

160 \newcommand{\newBthm}[2] [] {\B@newmacro{#1}{#2}{\Bthm}}
161 \newcommand{\newBmch}[2] [] {\B@newmacro{#1}{#2}{\Bmch}}
162 \newcommand{\newBvrb}[2] [] {\B@newmacro{#1}{#2}{\Bvrb}}
163 \newcommand{\newBinvt}[2] [] {\B@newmacro{#1}{#2}{\Binvt}}
164 \newcommand{\newBevt}[2] [] {\B@newmacro{#1}{#2}{\Bevt}}
165 \newcommand{\newBpar}[2] [] {\B@newmacro{#1}{#2}{\Bpar}}
166 \newcommand{\newBgrd}[2] [] {\B@newmacro{#1}{#2}{\Bgrd}}
167 \newcommand{\newBact}[2] [] {\B@newmacro{#1}{#2}{\Bact}}
168
169 %%%% Theorem Proof Obligation
170 %%%% Print the theorem proof obligation, given the theorem label.
171 %%%% Arguments:
172 %%%% 1. Theorem label
173 %%%%
174 %%%% Usage:
175 %%%% - \thmpo{thm} will produce "thm/THM"
176 \newcommand{\thmpo}[1] {\Bthm{#1}/\Bpo{THM}}
177
178 %%%% Axiom Well-definedness Proof Obligation
179 %%%% Print the axiom well-definedness proof obligation, given the
180 %%%% axiom label.
181 %%%% Arguments:
182 %%%% 1. Axiom label
183 %%%%
184 %%%% Usage:
185 %%%% - \axmwdpo{axm} will produce "axm/WD"
186 \newcommand{\axmwdpo}[1] {\Baxm{#1}/\Bpo{WD}}
187
188 %%%% Invariant Proof Obligation
189 %%%% Print the invariant proof obligation, given the event name and
190 %%%% invariant label
191 %%%% Arguments:
192 %%%% 1. Event name
193 %%%% 2. Invariant label
194 %%%%
195 %%%% Usage:
196 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
197 \newcommand{\invpo}[2] {\Bevt{#1}/\Binvt{#2}/\Bpo{INV}}
198
199 %%%% Theorem (in guard) Proof Obligation
200 %%%% Print the simulation proof obligation, given the event name and
201 %%%% the theorem (in guard) label.
202 %%%% Arguments:
203 %%%% 1. Event name
204 %%%% 2. Theorem (in guard) label
205 %%%%
206 %%%% Usage:
207 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
208 \newcommand{\grdthmpo}[2] {\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
209

```

```

210 %%%% Feasibility Proof Obligation
211 %%%% Print the feasibility proof obligation, given the event name and
212 %%%% the action label
213 %%%% Arguments:
214 %%%% 1. Event name
215 %%%% 2. Action label
216 %%%%
217 %%%% Usage:
218 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
219 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
220
221 %%%% Variant finiteness Proof Obligation
222 %%%% Print the Variant finiteness proof obligation
223 %%%% Arguments: No arguments
224 %%%%
225 %%%% Usage:
226 %%%% - \finpo will produce "FIN"
227 \newcommand{\finpo}{\Bpo{FIN}}
228
229 %%%% Variant Proof Obligation
230 %%%% Print the guard strengthen proof obligation, given the event name
231 %%%% Arguments:
232 %%%% 1. Event name
233 %%%%
234 %%%% Usage:
235 %%%% - \grdpo{evt} will produce "evt/VAR"
236 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
237
238 %%%% Simulation Proof Obligation
239 %%%% Print the simulation proof obligation, given the event name and
240 %%%% the action label.
241 %%%% Arguments:
242 %%%% 1. Event name
243 %%%% 2. Action label
244 %%%%
245 %%%% Usage:
246 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
247 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
248
249 %%%% Guard Strengthen Proof Obligation
250 %%%% Print the guard strengthen proof obligation, given the event
251 %%%% name and the guard label
252 %%%% Arguments:
253 %%%% 1. (Abstract) Event name
254 %%%% 2. (Abstract) Guard label
255 %%%%
256 %%%% Usage:
257 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
258 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
259
260 %%%% Variant Natural Number Proof Obligation
261 %%%% Print the Variant Natural Number proof obligation, given the event name

```

```

262 %%%% Arguments:
263 %%%% 1. Event name
264 %%%%
265 %%%% Usage:
266 %%%% - \natpo{evt} will produce "evt/NAT"
267 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
268
269 \newcommand{\B@keyword}[1]{\mathbf{#1}}
270 \newcommand{\B@identifier}[1]{\mathit{#1}}
271 \newcommand{\B@label}[2][]{
272   \def\is@thm{#1}
273   \ifx\is@thm\@empty
274     \mathsf{#2}
275   \else
276     \mathit{#2}
277   \fi
278 }
279
280
281
282 \newcommand{\eventinline}[7][]{
283   \setlength{\B@oldfboxsep}{\fboxsep}
284   \setlength{\fboxsep}{2ex}
285   \fbox{
286     \ensuremath{
287       \B@eventinline[#1]{#2}{#3}{#4}{#5}{#6}{#7}
288     }
289   }
290   \setlength{\fboxsep}{\B@oldfboxsep}
291 }
292
293
294 \newcommand{\B@declaration}[2]{
295   \begin{array}{l@{\Bsep}l}
296     \Bkeyword{#1:} & #2
297   \end{array}
298 }
299
300 \newcommand{\B@section}[3][]{
301   \def\no@title{#1}
302   \ifx\no@title\@empty
303     \begin{array}{l}
304       \Bkeyword{#2:} \\\
305       \begin{array}{l@{\Bsep}l}
306         #3
307       \end{array}
308     \end{array}
309   \else
310     \begin{array}{l@{\Bsep}l}
311       #3
312     \end{array}
313   \fi
314 }
315

```



```

316 \newcommand{\B@po}[1]{\ensuremath{\mathsf{\#1}}\xspace}
317
318 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%
319 \newcommand{\SKIP}{\textsc{skip}}
320
321
322 %%%% Pretty print an general Event-B event
323 %%%% Arguments:
324 %%%% 1. (Optional) convergence status.
325 %%%% 2. Name of the event.
326 %%%% 3. Name of the abstract event.
327 %%%% 4. (Comma-separated) list of parameters.
328 %%%% 5. (Newline\\)-separated list of guards.
329 %%%% 6. (Newline\\)-separated list of witness predicates.
330 %%%% 7. (Newline\\)-separated list of assignments.
331 %%%%
332 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\\G2(x,y)}{W1\\W2}{S1(v,x,y)\\S2(w,x,y)}
333 %%%% will produce the following
334 %%%%
335 %%%% conc
336 %%%% refines abs
337 %%%% status conv
338 %%%% any x, y where
339 %%%% G1(x, y)
340 %%%% G2(x, y)
341 %%%% with
342 %%%% W1
343 %%%% W2
344 %%%% then
345 %%%% S1(v, x, y)
346 %%%% S2(w, x, y)
347 %%%% end
348 %%%%
349 %%%% Special case:
350 %%%% - Empty abstract event --> refines clause is omitted.
351 %%%% - Empty convergence status --> status clause is omitted.
352 %%%% - Empty witness --> with clause is omitted.
353 %%%% - Empty parameters, empty guards --> begin ... end
354 %%%% - Empty parameters --> when ... then ... end
355 %%%% - Empty actions --> \SKIP
356 \newcommand{\B@event}[7][]{
357   \def\evt@sts{\#1}
358   \def\evt@name{\#2}
359   \def\evt@absevt{\#3}
360   \def\evt@pars{\#4}
361   \def\evt@grds{\#5}
362   \def\evt@wits{\#6}
363   \def\evt@acts{\#7}
364   %% Pretty-print convergence status
365   \ifx\evt@sts\@empty
366     \def\pretty@sts{}
367   \else
368     \def\pretty@sts{\Bsep\Bstatus \Bsep \evt@sts \\\}

```

```

369 \fi
370 % Pretty-print abstract events
371 \ifx\evt@absevt\@empty
372 \def\pretty@absevt{}
373 \else
374 \def\pretty@absevt{\Bsep\Brefines \Bsep \evt@absevt \\\}
375 \fi
376 % Pretty-print parameters
377 \ifx\evt@pars\@empty
378 \def\pretty@pars{}
379 \else
380 \def\pretty@pars{\Bsep\Bany \Bsep \evt@pars \Bsep \Bwhere \\\}
381 \fi
382 % Pretty-print guards
383 \ifx\evt@grds\@empty
384 \def\pretty@grds{}
385 \else
386 \def\evt@grds@tmp{
387   \begin{array}{@{\Bsep\Bsep}l@{\Bsep}l}
388     \evt@grds
389   \end{array}\\
390 }
391 \ifx\evt@pars\@empty
392 \def\pretty@grds{
393   \Bsep \Bwhen \\\
394   \evt@grds@tmp
395 }
396 \else
397 \def\pretty@grds{\evt@grds@tmp}
398 \fi
399 \fi
400 % Pretty-print witnesses
401 \ifx\evt@wits\@empty
402 \def\pretty@wits{}
403 \else
404 \def\pretty@wits{
405   \Bsep\Bwith\\
406   \begin{array}{@{\Bsep\Bsep}ll}
407     \evt@wits
408   \end{array}\\
409 }
410 \fi
411 % Pretty-print actions
412 \ifx\evt@acts\@empty
413 \def\evt@acts{\SKIP}
414 \else
415 \fi
416 \def\evt@acts@tmp{
417   \begin{array}{@{\Bsep\Bsep}l@{\Bsep}l}
418     \evt@acts
419   \end{array}\\
420 }
421 \def\evt@acts@keyword{\Bsep\Bthen \\\}
422 \ifx\evt@pars\@empty

```

```

423 \ifx\evt@grds\@empty
424 \def\evt@acts@keyword{\Bsep\Bbegin \\\}
425 \else
426 \fi
427 \else
428 \fi
429 \def\pretty@acts{
430   \evt@acts@keyword
431   \evt@acts@tmp
432 }
433 % Really do it now
434 \begin{array}{l}
435   \Bevt{\evt@name} \\\
436   \pretty@sts
437   \pretty@absevents
438   \pretty@pars
439   \pretty@grds
440   \pretty@wits
441   \pretty@acts
442   \Bsep\Bend
443 \end{array}
444 }
445
446 %%%% Pretty print an general Event-B event
447 %%%% Arguments:
448 %%%% 1. (Optional) convergence status.
449 %%%% 2. Name of the event.
450 %%%% 3. Name of the abstract event.
451 %%%% 4. (Comma-separated) list of parameters.
452 %%%% 5. (Newline(\)-separated) list of guards.
453 %%%% 6. (Newline(\)-separated) list of witness predicates.
454 %%%% 7. (Newline(\)-separated) list of assignments.
455 %%%%
456 %%%% Usage: \B@event[conv]{conc}{abs}{x,y}{G1(x,y)\G2(x,y)}{W1\W2}{S1(v,x,y)\S2(w,x,y)}
457 %%%% will produce the following
458 %%%%
459 %%%% conc
460 %%%% refines abs
461 %%%% status conv
462 %%%% any x, y where
463 %%%% G1(x, y)
464 %%%% G2(x, y)
465 %%%% with
466 %%%% W1
467 %%%% W2
468 %%%% then
469 %%%% S1(v, x, y)
470 %%%% S2(w, x, y)
471 %%%% end
472 %%%%
473 %%%% Special case:
474 %%%% - Empty abstract event --> refines clause is omitted.
475 %%%% - Empty convergence status --> status clause is omitted.
476 %%%% - Empty witness --> with clause is omitted.

```

```

477 %%%% - Empty parameters, empty guards --> begin ... end
478 %%%% - Empty parameters --> when ... then ... end
479 %%%% - Empty actions --> \SKIP
480 \newcommand{\B@eventinline}[7][{}]{
481   \def\evt@sts{#1}
482   \def\evt@name{#2}
483   \def\evt@absevt{#3}
484   \def\evt@pars{#4}
485   \def\evt@grds{#5}
486   \def\evt@wits{#6}
487   \def\evt@acts{#7}
488   %% Ignore convergence status
489   \def\pretty@sts{}
490   % Pretty-print abstract events
491   \ifx\evt@absevt\@empty
492     \def\pretty@absevt{}
493   \else
494     \def\pretty@absevt{\Brefines~\evt@absevt~}
495   \fi
496   % Pretty-print parameters
497   \ifx\evt@pars\@empty
498     \def\pretty@pars{}
499   \else
500     \def\pretty@pars{\Bany~\evt@pars~\Bwhere~}
501   \fi
502   % Pretty-print guards
503   \ifx\evt@grds\@empty
504     \def\pretty@grds{}
505   \else
506     \def\evt@grds@tmp{
507       \evt@grds
508     }
509     \ifx\evt@pars\@empty
510       \def\pretty@grds{
511         \Bwhen~
512         \evt@grds@tmp~
513       }
514     \else
515       \def\pretty@grds{\evt@grds@tmp~}
516     \fi
517   \fi
518   % Pretty-print witnesses
519   \ifx\evt@wits\@empty
520     \def\pretty@wits{}
521   \else
522     \def\pretty@wits{
523       \Bwith~
524       \evt@wits~
525     }
526   \fi
527   % Pretty-print actions
528   \ifx\evt@acts\@empty
529     \def\evt@acts{\SKIP}
530   \else

```

```

531 \fi
532 \def\evt@acts@tmp{
533   \evt@acts
534 }
535 \def\evt@acts@keyword{\Bthen}
536 \ifx\evt@pars\empty
537 \ifx\evt@grds\empty
538 \def\evt@acts@keyword{\Bbegin}
539 \else
540 \fi
541 \else
542 \fi
543 \def\pretty@acts{
544   \evt@acts@keyword~
545   \evt@acts@tmp~
546 }
547 % Really do it now
548 \begin{array}{l}
549   \Bevt{\evt@name}\widehat{=}\sim
550   \pretty@sts
551   \pretty@absevs
552   \pretty@pars
553   \pretty@grds
554   \pretty@wits
555   \pretty@acts
556   \Bend
557 \end{array}
558 }
559
560 %%%% INITIALISATION label
561 \newBevt{init}
562
563 %%%% Pretty print the initialisation: no ‘refines’ clause. no parameters, no
564 %%%% guards
565 %%%% Arguments:
566 %%%% 1. (Newline\\-separated) list of assignments.
567 %%%%
568 %%%% Usage: \init{S1(v,x,y)\S2(w,x,y)}
569 %%%% will produce the following
570 %%%%
571 %%%% init
572 %%%% begin
573 %%%% S1(v, x, y)
574 %%%% S2(w, x, y)
575 %%%% end
576 %%%%
577 \newcommand{\initialisation}[1]{
578   \event{\init}{-}{-}{-}{-}{#1}
579 }

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols		
<code>\@empty</code>	150,	
273, 302, 365,		
371, 377, 383,		
391, 401, 412,		
422, 423, 491,		
497, 503, 509,		
519, 528, 536, 537		
<code>\</code> 49, 55, 61, 68, 304,		
328–330, 332,		
368, 374, 380,		
389, 393, 405,		
408, 419, 421,		
424, 435, 452–		
454, 456, 566, 568		
<code>\sqcup</code>	57, 63, 70	
A		
<code>\axioms</code>	113	
<code>\axmwdpo</code>	185, 186	
B		
<code>\B@declaration</code>		
.	18, 40, 294	
<code>\B@event</code>		
9, 36, 332, 356, 456		
<code>\B@eventinline</code> 287, 480		
<code>\B@fontsize</code>		
48, 54, 60, 67, 106		
<code>\B@identifier</code> 75, 91, 270		
<code>\B@keyword</code> . 74, 88, 269		
<code>\B@label</code> . . . 76, 94, 271		
<code>\B@newmacro</code> 148, 156–167		
<code>\B@oldfboxsep</code> . 3, 5,		
12, 24, 31, 283, 290		
<code>\B@po</code> 77, 97, 316		
<code>\B@section</code> . 28, 44, 300		
<code>\Bact</code> . 145, 167, 219, 247		
<code>\Bany</code> 125, 380, 500		
<code>\Baxm</code> 137, 159, 186		
<code>\Bbap</code> 147		
<code>\Bbegin</code> . . . 126, 424, 538		
<code>\Bcst</code> 136, 158		
<code>\Bctx</code> 134, 156		
<code>\Bdeclaration</code> 15, 39,		
108, 111, 117, 123		
<code>\begin</code> 106, 295, 303,		
305, 310, 387,		
406, 417, 434, 548		
<code>\Bend</code> 127, 442, 556		
<code>\Bevt</code> 143,		
164, 197, 208,		
219, 236, 247,		
258, 267, 435, 549		
<code>\bfseries</code> 147		
<code>\Bgrd</code> 146, 166, 258		
<code>\Bhspace</code> . 50, 56, 62, 69		
<code>\Bidentifier</code> . 75, 90,		
135, 136, 141, 144		
<code>\Binv</code> 142, 163, 197		
<code>\Bkeyword</code> . . . 74, 87,		
125–133, 296, 304		
<code>\Blabel</code>		
76, 93, 137, 138,		
142, 143, 145, 146		
<code>\Bmch</code> 140, 161		
<code>\Bpar</code> 144, 165		
<code>\Bpo</code> 77, 96,		
176, 186, 197,		
208, 219, 227,		
236, 247, 258, 267		
<code>\Brefines</code> . 128, 374, 494		
<code>\Bsection</code> 23, 43, 114, 120		
<code>\Bsep</code> 51,		
57, 63, 70, 295,		
305, 310, 368,		
374, 380, 387,		
393, 405, 406,		
417, 421, 424, 442		
<code>\Bset</code> 135, 157		
<code>\Bstatus</code> 129, 368		
<code>\Bthen</code> 130, 421, 535		
<code>\Bthm</code> . 138, 160, 176, 208		
<code>\Bvrb</code> 141, 162		
<code>\Bvspace</code> . 49, 55, 61, 68		
<code>\Bwhen</code> 131, 393, 511		
<code>\Bwhere</code> . . . 132, 380, 500		
<code>\Bwith</code> 133, 405, 523		
C		
<code>\carriersets</code> 107		
<code>\colorlet</code> 79, 81, 83, 85		
<code>\constants</code> 110		
<code>\csgname</code> 151, 153		
D		
<code>\DeclareOption</code> . 34,		
53, 59, 66, 78, 100		
<code>\def</code> 149, 151,		
153, 272, 301,		
357–363, 366,		
368, 372, 374,		
378, 380, 384,		
386, 392, 397,		
402, 404, 413,		
416, 421, 424,		
429, 481–487,		
489, 492, 494,		
498, 500, 504,		
506, 510, 515,		
520, 522, 529,		
532, 535, 538, 543		
E		
<code>\else</code> 152,		
275, 309, 367,		
373, 379, 385,		
396, 403, 414,		
425, 427, 493,		
499, 505, 514,		
521, 530, 539, 541		
<code>\end</code> . . 106, 297, 307,		
308, 312, 389,		
408, 419, 443, 557		
<code>\endcsname</code> . . . 151, 153		
<code>\ensuremath</code>		
8, 17, 27, 74–77,		
88, 91, 94, 97,		
134, 140, 286, 316		
<code>\event</code> 4, 35, 578		
<code>\eventB</code> 105		
<code>\eventinline</code> 282		
<code>\evt@absevt</code> 359, 371,		
374, 483, 491, 494		
<code>\evt@acts</code> 363,		
412, 413, 418,		
487, 528, 529, 533		
<code>\evt@acts@keyword</code> .		
. 421, 424,		
430, 535, 538, 544		
<code>\evt@acts@tmp</code>		
. 416, 431, 532, 545		

<code>\evt@grds</code> 361, 383, 388, 423, 485, 503, 507, 537	<code>\input@macro</code> . . 149, 150	<code>\pretty@pars</code> 378, 380, 438, 498, 500, 552
<code>\evt@grds@tmp</code> 386, 394, 397, 506, 512, 515	<code>\invariants</code> 119	<code>\pretty@sts</code> . . . 366, 368, 436, 489, 550
<code>\evt@name</code> 358, 435, 482, 549	<code>\invpo</code> 196, 197	<code>\pretty@wits</code> 402, 404, 440, 520, 522, 554
<code>\evt@pars</code> 360, 377, 380, 391, 422, 484, 497, 500, 509, 536	<code>\is@thm</code> 272, 273	<code>\ProcessOptions</code> . . 104
<code>\evt@sts</code> 357, 365, 368, 481	M	Q
<code>\evt@wits</code> . 362, 401, 407, 486, 519, 524	<code>\mathbf</code> . . . 134, 140, 269	<code>\quad</code> 51
<code>\ExecuteOptions</code> 64, 71, 101	<code>\mathit</code> 270, 276	R
<code>\expandafter</code> . . 151, 153	<code>\mathsf</code> 274, 316	<code>\renewcommand</code> 35, 39, 43, 54– 57, 60–63, 67– 70, 87, 90, 93, 96
F	N	<code>\RequirePackage</code> . . 1, 2
<code>\fbox</code> 7, 16, 26, 285	<code>\natpo</code> 266, 267	S
<code>\fboxsep</code> 5, 6, 12, 24, 25, 31, 283, 284, 290	<code>\newBact</code> 167	<code>\scriptsize</code> 67
<code>\fi</code> 154, 277, 313, 369, 375, 381, 398, 399, 410, 415, 426, 428, 495, 501, 516, 517, 526, 531, 540, 542	<code>\newBaxm</code> 159	<code>\setBIdentifierColour</code> 81, 82
<code>\finpo</code> 226, 227	<code>\newBcst</code> 158	<code>\setBKeywordColour</code> 79, 80
<code>\fispo</code> 218, 219	<code>\newBctx</code> 156	<code>\setBLabelColour</code> 83, 84
<code>\footnotesize</code> 60	<code>\newBevt</code> 164, 561	<code>\setBPOColour</code> . . . 85, 86
G	<code>\newBgrd</code> 166	<code>\setlength</code> 5, 6, 12, 24, 25, 31, 283, 284, 290
<code>\grdpo</code> 235, 257, 258	<code>\newBin</code> 163	<code>\simpo</code> 246, 247
<code>\grdthmpo</code> 207, 208	<code>\newBmch</code> 161	<code>\SKIP</code> 319, 355, 413, 479, 529
H	<code>\newBpar</code> 165	<code>\sl</code> 147
<code>\hbox</code> 147	<code>\newBset</code> 157	<code>\small</code> 54
<code>\hspace</code> . . 50, 56, 62, 69	<code>\newBthm</code> 160	T
I	<code>\newBvrb</code> 162	<code>\textcolor</code> 88, 91, 94, 97
<code>\ifx</code> 150, 273, 302, 365, 371, 377, 383, 391, 401, 412, 422, 423, 491, 497, 503, 509, 519, 528, 536, 537	<code>\newcommand</code> 4, 15, 23, 48–51, 74–77, 79, 81, 83, 85, 105, 107, 110, 113, 116, 119, 122, 125– 138, 140–148, 156–167, 176, 186, 197, 208, 219, 227, 236, 247, 258, 267, 269–271, 282, 294, 300, 316, 319, 356, 480, 577	<code>\textsc</code> 319
<code>\init</code> 568, 578	<code>\newenvironment</code> . . . 106	<code>\thmpo</code> 175, 176
<code>\initialisation</code> . . . 577	<code>\newlength</code> 3	V
	<code>\no@title</code> 301, 302	<code>\variables</code> 116
	<code>\normalsize</code> 48	<code>\variant</code> 122
	P	<code>\varpo</code> 236
	<code>\pretty@absevt</code> 372, 374, 437, 492, 494, 551	W
	<code>\pretty@acts</code> 429, 441, 543, 555	<code>\widehat</code> 549
	<code>\pretty@grds</code> . . 384, 392, 397, 439, 504, 510, 515, 553	X
		<code>\xspace</code> 74–77, 88, 91, 94, 97, 105, 134, 140, 316

Change History

v1.0	labels are in math-mode	1
General: Initial version	1	v1.1
v1.0.1	General: Re-implement how options	
General: Ensure that the keywords,	are defined, added options 'box'	1