

The `eventB` package*

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

February 27, 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	2
3.1	Package Loading	2
3.2	Helper Macros	2
3.3	Declaration of Options for the Package	4
3.4	Commands for Pretty-Print Event-B Models	7

1 Introduction

This package was developed in order to ease the typesetting of Event-B models in L^AT_EX.

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` v1.1.1, dated 2012/02/21.

3 Implementation

3.1 Package Loading

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

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

3.2 Helper Macros

We define same basic helper macros that will be used to defined other macros.

```
\B@keywordbase Basic macro for Event-B keywords.
4 \newcommand{\B@keywordbase}[1]{\mathbf{#1}}

\B@identifierbase Basic macro for Event-B identifiers.
5 \newcommand{\B@identifierbase}[1]{\mathit{#1}}

\B@declarationbase Basic macro for Event-B declarations (e.g., variables, constants, etc.).
6 \newcommand{\B@declarationbase}[2]{
7   \begin{array}{l@{\B@tab}l}
8     \B@keyword{#1:} & #2
9   \end{array}
10 }

\B@sectionbase Basic macro for Event-B sections (e.g., invariants, axioms, etc.).
11 \newcommand{\B@sectionbase}[3][]{
12   \ifstrequal{#1}{}{
13     \begin{array}{l}
14       \B@keyword{#2:} \\
15       \begin{array}{l@{\B@tab}l}
16         #3
17       \end{array}
18     \end{array}
19   }{
20     \begin{array}{l@{\B@tab}l}
21       #3
22     \end{array}
23   }
24 }

\B@ifstrequal A wrapper for ake sure that the first argument is properly expanded.
25 \newcommand{\B@ifstrequal}{\expandafter\ifstrequal\expandafter}

\B@eventbase Basic macro for Event-B events.
26 \newcommand{\B@eventbase}[7][]{%
27   { % BEGIN group
28     \newcommand\evt@sts{#1}% Event status
29     \newcommand\evt@label{#2}% Event label
30     \newcommand\evt@absevt{#3}% Abstract event
```

```

31 \newcommand\evt@pars{#4}% Event parameters
32 \newcommand\evt@grds{#5}% Event guards
33 \newcommand\evt@wits{#6}% Event witnesses
34 \newcommand\evt@acts{#7}% Event actions

```

The convergence status is skipped if empty.

```

35 \B@ifstrequal{\evt@sts}{-}{
36   \newcommand\pretty@sts{}
37 }{
38   \newcommand\pretty@sts{\B@tab\Bstatus \B@tab \evt@sts \\\}
39 }

```

The refines clause is skipped if there are no abstract events.

```

40 \B@ifstrequal{\evt@absepts}{-}{
41   \newcommand\pretty@absepts{}
42 }{
43   \newcommand\pretty@absepts{\B@tab\Brefines \B@tab \evt@absepts{} \\\}
44 }

```

The parameters is skipped if there are none.

```

45 \B@ifstrequal{\evt@pars}{-}{
46   \newcommand\pretty@pars{}
47 }{
48   \newcommand\pretty@pars{\B@tab\Bany \B@tab \evt@pars \B@tab \Bwhere \\\}
49 }

```

The keywords for guards also depends on if there are parameters or not.

```

50 \B@ifstrequal{\evt@grds}{-}{
51   \newcommand\pretty@grds{}
52 }{
53   \newcommand\pretty@grds@tmp{
54     \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
55       \evt@grds
56     \end{array}\\\
57   }
58   \B@ifstrequal{\evt@pars}{-}{
59     \newcommand\pretty@grds{
60       \B@tab \Bwhen \\\
61       \pretty@grds@tmp
62     }
63   }{
64     \newcommand\pretty@grds{\pretty@grds@tmp}
65   }
66 }

```

The witnesses are skipped if there are none.

```

67 \B@ifstrequal{\evt@wits}{-}{
68   \newcommand\pretty@wits{}
69 }{
70   \newcommand\pretty@wits{
71     \B@tab\Bwith\\\
72     \begin{array}{@{\B@tab\B@tab}l}
73       \evt@wits
74     \end{array}\\\
75   }
76 }

```

When there are no actions, SKIP is used. The keyword is changed depending on whether the event has parameters or not.

```

77 \B@ifstrequal{\evt@acts}{}{
78 \renewcommand\evt@acts{\SKIP}
79 }{}
80 \newcommand\pretty@acts@tmp{
81 \begin{array}{@{\B@tab\B@tab}l@{\B@tab}l}
82 \evt@acts
83 \end{array}\\
84 }
85 \newcommand\pretty@acts@keyword{\B@tab\Bthen \\\}
86 \B@ifstrequal{\evt@pars}{}{
87 \B@ifstrequal{\evt@grds}{}{
88 \renewcommand\pretty@acts@keyword{\B@tab\Bbegin \\\}
89 }{}
90 }{}
91 \newcommand\pretty@acts{
92 \pretty@acts@keyword
93 \pretty@acts@tmp
94 }

```

Finally we put all the pretty-print pieces together.

```

95 \begin{array}{l}
96 \Bevt{\evt@label} \\\
97 \pretty@sts
98 \pretty@absepts
99 \pretty@pars
100 \pretty@grds
101 \pretty@wits
102 \pretty@acts
103 \B@tab\Bend
104 \end{array}
105 } % END group
106 }

```

3.3 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.

```

107 \newlength{\B@tmplength}
108 \newcommand{\B@makebox}[1]{
109 {
110 \setlength{\B@tmplength}{\fboxsep}
111 \setlength{\fboxsep}{2ex}
112 \fbox{#1}
113 \setlength{\fboxsep}{\B@tmplength}
114 }
115 }

```

\B@event

```

116 \newcommand{\B@event}[7] [] {
117   \B@makebox{
118     \ensuremath{
119       \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
120     }
121   }
122 }

\B@declaration

123 \newcommand{\B@declaration}[2] {
124   \B@makebox{
125     \ensuremath{
126       \B@declarationbase[#1]{#2}
127     }
128   }
129 }

\B@section

130 \newcommand{\B@section}[3] [] {
131   \B@makebox{
132     \ensuremath{
133       \B@sectionbase[#1]{#2}{#3}
134     }
135   }
136 }

137 \DeclareOption{nobox}{
138   \renewcommand{\B@event}[7] [] {
139     \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
140   }
141
142   \renewcommand{\B@declaration}[2] {
143     \B@declarationbase[#1]{#2}
144   }
145
146   \renewcommand{\B@section}[3] [] {
147     \B@sectionbase[#1]{#2}{#3}
148   }
149 }
150

```

Options for font size and spacing We define the default values for font size and some spacing commands, and how they are redefined according to options `small`, `compact`, and `tiny`. In particular, option `compact` and `tiny` implies option `nobox`.

```

151 \newcommand{\B@fontsize}{\normalsize} % The font size used in Bcode environment
152 \newcommand{\Bvspace}[1] [2ex]{\[\[ #1]} % Vertical space
153 \newcommand{\Bhspace}[1] [2em]{\hspace{#1}} % Horizontal space
154 \newcommand{\B@tab}{\quad} % A small separation space
155
156 \DeclareOption{small}{
157   \renewcommand{\B@fontsize}{\small}
158   \renewcommand{\Bvspace}[1] [1ex]{\[\[ #1]}

```

```

159 \renewcommand{\Bhspace}[1][1em]{\hspace{#1}}
160 \renewcommand{\B@tab}{\ }
161 }
162 \DeclareOption{compact}{
163 \renewcommand{\B@fontsize}{\footnotesize}
164 \renewcommand{\Bvspace}[1][0ex]{\[\[ #1]}
165 \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
166 \renewcommand{\B@tab}{\ }
167 \ExecuteOptions{nobox}
168 }
169 \DeclareOption{tiny}{
170 \renewcommand{\B@fontsize}{\scriptsize}
171 \renewcommand{\Bvspace}[1][-0.5ex]{\[\[ #1]}
172 \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
173 \renewcommand{\B@tab}{\ }
174 \ExecuteOptions{nobox}
175 }
176

```

Options for colouring Keywords, labels and identifiers in Event-B can be coloured. We define several commands and redefine them accordingly for colouring. When `colour` (or `color`) option is enabled, one can customise the colours for Event-B keywords, labels or identifier or proof obligation labels.

```

177 <eventB>
178 % Default definition for Event-B keywords
179 </eventB>
180 \newcommand{\B@keyword}[1]{\ensuremath{\B@keywordbase{#1}}\xspace}
181 \newcommand{\B@identifier}[1]{\ensuremath{\B@identifierbase{#1}}\xspace}
182 % Default definition for Event-B identifiers
183
184 \newcommand{\B@label}[2][ ]{\ensuremath{\B@label{#1}{#2}}\xspace}
185 \newcommand{\B@po}[1]{\ensuremath{\B@po{#1}}\xspace}
186 \DeclareOption{colour}{
187 \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{#1}}
188 \setBKeywordColour{blue}
189 \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{#1}}
190 \setBIdentifierColour{blue!50!red}
191 \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
192 \setBLabelColour{green!50!black}
193 \newcommand{\setBPOLColour}[1]{\colorlet{B@pocolor}{#1}}
194 \setBPOLColour{red}
195 \renewcommand{\B@keyword}[1]{
196 \ensuremath{\textcolor{B@keywordcolor}{\B@keywordbase{#1}}}\xspace
197 }
198 \renewcommand{\B@identifier}[1]{
199 \ensuremath{\textcolor{B@identifiercolor}{\B@identifierbase{#1}}}\xspace
200 }
201 \renewcommand{\B@label}[2][ ]{
202 \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
203 }
204 \renewcommand{\B@po}[1]{
205 \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
206 }

```

```

207 }
208 \DeclareOption{color}{
209   \ExecuteOptions{colour}
210 }
211

```

After declaration of options, we execute them accordingly.

```

212 \ProcessOptions

```

3.4 Commands for Pretty-Print Event-B Models

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

```

213 \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.

```

214 \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.

```

215 \newcommand{\carriersets}[1]{
216   \B@declaration{sets}{#1}
217 }
218 \newcommand{\constants}[1]{
219   \B@declaration{constants}{#1}
220 }
221 \newcommand{\axioms}[2][ ]{
222   \B@section[#1]{axioms}{#2}
223 }
224 \newcommand{\variables}[1]{
225   \B@declaration{variables}{#1}
226 }
227 \newcommand{\invariants}[2][ ]{
228   \B@section[#1]{invariants}{#2}
229 }
230 \newcommand{\variant}[1]{
231   \B@declaration{variant}{#1}
232 }

```

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

```

233 \newcommand{\Bany}{\B@keyword{any}}
234 \newcommand{\Bbegin}{\B@keyword{begin}}
235 \newcommand{\Bend}{\B@keyword{end}}
236 \newcommand{\Brefines}{\B@keyword{refines}}
237 \newcommand{\Bstatus}{\B@keyword{status}}
238 \newcommand{\Bthen}{\B@keyword{then}}
239 \newcommand{\Bwhen}{\B@keyword{when}}
240 \newcommand{\Bwhere}{\B@keyword{where}}
241 \newcommand{\Bwith}{\B@keyword{with}}

```

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

```

242 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{\#1}}\xspace}
243 \newcommand{\Bset}[1]{\Bidentifier{\#1}}
244 \newcommand{\Bcst}[1]{\Bidentifier{\#1}}
245 \newcommand{\Baxm}[1]{\Blabel{\#1}}
246 \newcommand{\Bthm}[1]{\Blabel[thm]{\#1}}
247
248 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{\#1}}\xspace}
249 \newcommand{\Bvrb}[1]{\Bidentifier{\#1}}
250 \newcommand{\Binv}[1]{\Blabel{\#1}}
251 \newcommand{\Bevt}[1]{\Blabel{\#1}}
252 \newcommand{\Bpar}[1]{\Bidentifier{\#1}}
253 \newcommand{\Bact}[1]{\Blabel{\#1}}
254 \newcommand{\Bgrd}[1]{\Blabel{\#1}}
255 \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.

```

256 \newcommand{\B@newmacro}[3][]{
257   \ifstrequal{\#1}{}{
258     \expandafter\def\csname #2\endcsname{\#3{\#2}}
259   }{
260     \expandafter\def\csname #1\endcsname{\#3{\#2}}
261   }
262 }

263 \newcommand{\newBctx}[2][]{\B@newmacro{\#1}{\#2}{\Bctx}}
264 \newcommand{\newBset}[2][]{\B@newmacro{\#1}{\#2}{\Bset}}
265 \newcommand{\newBcst}[2][]{\B@newmacro{\#1}{\#2}{\Bcst}}
266 \newcommand{\newBaxm}[2][]{\B@newmacro{\#1}{\#2}{\Baxm}}
267 \newcommand{\newBthm}[2][]{\B@newmacro{\#1}{\#2}{\Bthm}}
268 \newcommand{\newBmch}[2][]{\B@newmacro{\#1}{\#2}{\Bmch}}
269 \newcommand{\newBvrb}[2][]{\B@newmacro{\#1}{\#2}{\Bvrb}}
270 \newcommand{\newBinv}[2][]{\B@newmacro{\#1}{\#2}{\Binv}}
271 \newcommand{\newBevt}[2][]{\B@newmacro{\#1}{\#2}{\Bevt}}
272 \newcommand{\newBpar}[2][]{\B@newmacro{\#1}{\#2}{\Bpar}}
273 \newcommand{\newBgrd}[2][]{\B@newmacro{\#1}{\#2}{\Bgrd}}
274 \newcommand{\newBact}[2][]{\B@newmacro{\#1}{\#2}{\Bact}}
275
276 %%%% Theorem Proof Obligation
277 %%%% Print the theorem proof obligation, given the theorem label.
278 %%%% Arguments:
279 %%%% 1. Theorem label
280 %%%%
281 %%%% Usage:
282 %%%% - \thmpo{thm} will produce "thm/THM"
283 \newcommand{\thmpo}[1]{\Bthm{\#1}/\Bpo{THM}}
284

```



```

285 %%%% Axiom Well-definedness Proof Obligation
286 %%%% Print the axiom well-definedness proof obligation, given the
287 %%%% axiom label.
288 %%%% Arguments:
289 %%%% 1. Axiom label
290 %%%%
291 %%%% Usage:
292 %%%% - \axmwdpo{axm} will produce "axm/WD"
293 \newcommand{\axmwdpo}[1]{\Baxm{#1}/\Bpo{WD}}
294
295 %%%% Invariant Proof Obligation
296 %%%% Print the invariant proof obligation, given the event name and
297 %%%% invariant label
298 %%%% Arguments:
299 %%%% 1. Event name
300 %%%% 2. Invariant label
301 %%%%
302 %%%% Usage:
303 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
304 \newcommand{\invpo}[2]{\Bevt{#1}/\Binv{#2}/\Bpo{INV}}
305
306 %%%% Theorem (in guard) Proof Obligation
307 %%%% Print the simulation proof obligation, given the event name and
308 %%%% the theorem (in guard) label.
309 %%%% Arguments:
310 %%%% 1. Event name
311 %%%% 2. Theorem (in guard) label
312 %%%%
313 %%%% Usage:
314 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
315 \newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
316
317 %%%% Feasibility Proof Obligation
318 %%%% Print the feasibility proof obligation, given the event name and
319 %%%% the action label
320 %%%% Arguments:
321 %%%% 1. Event name
322 %%%% 2. Action label
323 %%%%
324 %%%% Usage:
325 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
326 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
327
328 %%%% Variant finiteness Proof Obligation
329 %%%% Print the Variant finiteness proof obligation
330 %%%% Arguments: No arguments
331 %%%%
332 %%%% Usage:
333 %%%% - \finpo will produce "FIN"
334 \newcommand{\finpo}{\Bpo{FIN}}
335
336 %%%% Variant Proof Obligation

```

```

337 %%%% Print the guard strengthen proof obligation, given the event name
338 %%%% Arguments:
339 %%%% 1. Event name
340 %%%%
341 %%%% Usage:
342 %%%% - \grdpo{evt} will produce "evt/VAR"
343 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
344
345 %%%% Simulation Proof Obligation
346 %%%% Print the simulation proof obligation, given the event name and
347 %%%% the action label.
348 %%%% Arguments:
349 %%%% 1. Event name
350 %%%% 2. Action label
351 %%%%
352 %%%% Usage:
353 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
354 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
355
356 %%%% Guard Strengthen Proof Obligation
357 %%%% Print the guard strengthen proof obligation, given the event
358 %%%% name and the guard label
359 %%%% Arguments:
360 %%%% 1. (Abstract) Event name
361 %%%% 2. (Abstract) Guard label
362 %%%%
363 %%%% Usage:
364 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
365 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
366
367 %%%% Variant Natural Number Proof Obligation
368 %%%% Print the Variant Natural Number proof obligation, given the event name
369 %%%% Arguments:
370 %%%% 1. Event name
371 %%%%
372 %%%% Usage:
373 %%%% - \natpo{evt} will produce "evt/NAT"
374 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
375

```

\inlineevent

```

376 \newcommand{\inlineevent}[7][]{
377   \B@inlineeventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
378 }

379 \newcommand{\B@label}[2][]{
380   \ifstrequal{#1}{}{
381     \mathsf{#2}
382   }{
383     \mathit{#2}
384   }
385 }
386

```

```

387
388
389
390
391
392 \newcommand{\B@po}[1]{\ensuremath{\mathsf{#1}}\xspace}
393
394 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
395 \newcommand{\SKIP}{\textsc{skip}\xspace}
396 %
397 %%%% Pretty print an general Event-B event
398 %%%% Arguments:
399 %%%% 1. (Optional) convergence status.
400 %%%% 2. Name of the event.
401 %%%% 3. Name of the abstract event.
402 %%%% 4. (Comma-separated) list of parameters.
403 %%%% 5. (\land-separated) list of guards.
404 %%%% 6. (\land-separated) list of witness predicates.
405 %%%% 7. (||-separated) list of assignments.
406 %%%%
407 %%%% Usage: \B@inlineeventbase[conv]{conc}{abs}{x,y}{G1 \land G2}{W1 \land W2}{S1 || S2}
408 %%%% will produce the following
409 %%%%
410 %%%% conc (conv) refines abs = any x,y where G1 \land G2 with W1 \land W2 then S1 || S2 end
411 %%%%
412 %%%% Special case:
413 %%%% - Empty abstract event --> refines clause is omitted.
414 %%%% - Empty convergence status --> status clause is omitted.
415 %%%% - Empty witness --> with clause is omitted.
416 %%%% - Empty parameters, empty guards --> begin ... end
417 %%%% - Empty parameters --> when ... then ... end
418 %%%% - Empty actions --> \SKIP
419 \newcommand{\B@inlineeventbase}[7][{}]{
420   { % BEGIN group
421     \newcommand\evt@sts{#1}% Event status
422     \newcommand\evt@label{#2}% Event label
423     \newcommand\evt@absevt{#3}% Abstract event
424     \newcommand\evt@pars{#4}% Event parameters
425     \newcommand\evt@grds{#5}% Event guards
426     \newcommand\evt@wits{#6}% Event witnesses
427     \newcommand\evt@acts{#7}% Event actions
428     %% Pretty-print convergence status
429     \B@ifstrequal{\evt@sts}{-}{
430       \newcommand\pretty@sts{}
431     }{
432       \newcommand\pretty@sts{(\evt@sts)}
433     }
434     % Pretty-print abstract events
435     \B@ifstrequal{\evt@absevt}{-}{
436       \newcommand\pretty@absevt{}
437     }{
438       \newcommand\pretty@absevt{\Brefines~\evt@absevt}
439     }

```

```

440 % Pretty-print parameters
441 \B@ifstrequal{\evt@pars}{-}{
442   \newcommand\pretty@pars{
443 }{
444   \newcommand\pretty@pars{\Bany~\evt@pars~\Bwhere~}
445 }
446 % Pretty-print guards
447 \B@ifstrequal{\evt@grds}{-}{
448   \newcommand\pretty@grds{
449 }{
450   \newcommand\pretty@grds@tmp{
451     \evt@grds~
452   }
453   \B@ifstrequal{\evt@pars}{-}{
454     \Bwhen~\pretty@grds@tmp
455   }{
456     \newcommand\pretty@grds{\pretty@grds@tmp}
457   }
458 }
459 % Pretty-print witnesses
460 \B@ifstrequal{\evt@wits}{-}{
461   \newcommand\pretty@wits{
462 }{
463   \newcommand\pretty@wits{
464     \Bwith~
465     \evt@wits~
466   }
467 }
468 % Pretty-print actions
469 \B@ifstrequal{\evt@acts}{-}{
470   \renewcommand\evt@acts{\SKIP}
471 }{}
472 \newcommand\pretty@acts@tmp{
473   \evt@acts
474 }
475 \newcommand\pretty@acts@keyword{\Bthen}
476 \B@ifstrequal{\evt@pars}{-}{
477   \B@ifstrequal{\evt@grds}{-}{
478     \renewcommand\pretty@acts@keyword{\Bbegin}
479   }{}
480 }{}
481 \newcommand\pretty@acts{
482   \pretty@acts@keyword~
483   \pretty@acts@tmp~
484 }
485 % Really do it now
486 \begin{array}{l}
487   \Bevt{\evt@label}\pretty@sts\pretty@absepts~\widehat{=}\~
488   \pretty@pars
489   \pretty@grds
490   \pretty@wits
491   \pretty@acts
492   \Bend
493 \end{array}

```


<code>\Bact</code> . 253, 274, 326, 354	<code>\eventB</code> 213	<code>\newBset</code> 264
<code>\Bany</code> 48, 233, 444	<code>\evt@absepts</code> . . . 30,	<code>\newBthm</code> 267
<code>\Baxm</code> 245, 266, 293	40, 43, 423, 435, 438	<code>\newBvrb</code> 269
<code>\Bbap</code> 255	<code>\evt@acts</code>	
<code>\Bbegin</code> . . . 88, 234, 478	. 34, 77, 78, 82,	P
<code>\Bcst</code> 244, 265	427, 469, 470, 473	<code>\pretty@absepts</code> 41,
<code>\Bctx</code> 242, 263	<code>\evt@grds</code>	43, 98, 436, 438, 487
<code>\Bend</code> 103, 235, 492	. 32, 50, 55, 87,	<code>\pretty@acts</code>
<code>\Bevt</code> 96, 251, 271, 304,	425, 447, 451, 477	. 91, 102, 481, 491
315, 326, 343,	<code>\evt@label</code>	<code>\pretty@acts@keyword</code>
354, 365, 374, 487	. . 29, 96, 422, 487 85,
<code>\bfseries</code> 255	<code>\evt@pars</code> . . . 31, 45,	88, 92, 475, 478, 482
<code>\Bgrd</code> 254, 273, 365	48, 58, 86, 424,	<code>\pretty@acts@tmp</code> . .
<code>\Bhspace</code>	441, 444, 453, 476	. . 80, 93, 472, 483
. 153, 159, 165, 172	<code>\evt@sts</code> 28,	<code>\pretty@grds</code>
<code>\Bidentifier</code> 181, 198,	35, 38, 421, 429, 432	. . . 51, 59, 64,
243, 244, 249, 252	<code>\evt@wits</code> 33,	100, 448, 456, 489
<code>\Binv</code> 250, 270, 304	67, 73, 426, 460, 465	<code>\pretty@grds@tmp</code> 53,
<code>\Blabel</code> 184,		61, 64, 450, 454, 456
201, 245, 246,	F	<code>\pretty@pars</code> . . . 46,
250, 251, 253, 254	<code>\finpo</code> 333, 334	48, 99, 442, 444, 488
<code>\Bmch</code> 248, 268	<code>\fispo</code> 325, 326	<code>\pretty@sts</code> 36,
<code>\Bpar</code> 252, 272		38, 97, 430, 432, 487
<code>\Bpo</code> 185, 204,	G	<code>\pretty@wits</code> . 68, 70,
283, 293, 304,	<code>\grdpo</code> 342, 364, 365	101, 461, 463, 490
315, 326, 334,	<code>\grdthmpo</code> 314, 315	
343, 354, 365, 374		S
<code>\Brefines</code> . 43, 236, 438	I	<code>\setBIdentifierColour</code>
<code>\Bset</code> 243, 264	<code>\INITIALISATION</code> 496, 514 189, 190
<code>\Bstatus</code> 38, 237	<code>\initialisation</code> 504, 513	<code>\setBKeywordColour</code> .
<code>\Bthen</code> 85, 238, 475	<code>\inlineevent</code> 376 187, 188
<code>\Bthm</code> . 246, 267, 283, 315	<code>\invariants</code> 227	<code>\setBLabelColour</code> . .
<code>\Bvrb</code> 249, 269	<code>\invpo</code> 303, 304 191, 192
<code>\Bvspace</code>	N	<code>\setBP0Colour</code> . 193, 194
. 152, 158, 164, 171	<code>\natpo</code> 373, 374	<code>\simpo</code> 353, 354
<code>\Bwhen</code> 60, 239, 454	<code>\newBact</code> 274	<code>\SKIP</code> . . 78, 395, 418, 470
<code>\Bwhere</code> . . . 48, 240, 444	<code>\newBaxm</code> 266	<code>\sl</code> 255
<code>\Bwith</code> 71, 241, 464	<code>\newBcst</code> 265	
C	<code>\newBctx</code> 263	T
<code>\carriersets</code> 215	<code>\newBevt</code> 271, 497	<code>\thmpo</code> 282, 283
<code>\constants</code> 218	<code>\newBgrd</code> 273	
E	<code>\newBinv</code> 270	V
<code>\event</code> 514, 517	<code>\newBmch</code> 268	<code>\variables</code> 224
	<code>\newBpar</code> 272	<code>\variant</code> 230
		<code>\varpo</code> 343

Change History

v1.0	mode 5
General: Initial version 1	<code>\B@event</code> : Ensure math-mode 4
v1.0.1	<code>\B@section</code> : Ensure math-mode . . 5
<code>\B@declaration</code> : Ensure math-	

v1.1		use etoolbox instead of ifthen	1
	General: Re-implement how options are defined, added options	Updated documentation, added	
	'nobox'	DoNotIndex	1
v1.1.1		\INITIALISATION: Renamed from	
	General: Updated documentation .	init	13
v2.0		\inlineevent: Renamed from	
	General: Major re-implementation,	eventinline	10