

# 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>1</b>
2.1	Package Options . . . . .	1
<b>3</b>	<b>Implementation</b>	<b>2</b>
3.1	Package Loading . . . . .	2
3.2	Helper Macros . . . . .	2
3.3	Declaration of Options for the Package . . . . .	6
3.4	Commands for Pretty-Print Event-B Models . . . . .	8

## 1 Introduction

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

## 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` v1.1.1, 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 pretty-print Event-B events.

```
26 \newcommand{\B@eventbase}[7][]{%
27   { % BEGIN group
```

We first save the arguments to local variables.

```
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     \Bvt{\evt@label} \\\
97     \pretty@sts
98     \pretty@absevs
99     \pretty@pars
100    \pretty@grds
101    \pretty@wits
102    \pretty@acts
103    \B@tab\Bend
104  \end{array}
105 } % END group
106 }

```

**\B@inlineeventbase** Basic macro for pretty-print Event-B events inline.

```

107 \newcommand{\B@inlineeventbase}[7][]{
108   { % BEGIN group

```

We first save the arguments to local variables.

```

109     \newcommand\evt@sts{#1}% Event status
110     \newcommand\evt@label{#2}% Event label
111     \newcommand\evt@absevs{#3}% Abstract event
112     \newcommand\evt@pars{#4}% Event parameters
113     \newcommand\evt@grds{#5}% Event guards
114     \newcommand\evt@wits{#6}% Event witnesses
115     \newcommand\evt@acts{#7}% Event actions

```

The convergence status is skipped if empty.

```

116   \B@ifstrequal{\evt@sts}{}{
117     \newcommand\pretty@sts{}
118   }{
119     \newcommand\pretty@sts{(\evt@sts)}
120   }

```

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

```

121 \B@ifstrequal{\evt@absevt}{-}{-}{
122 \newcommand\pretty@absevt{}
123 }{
124 \newcommand\pretty@absevt{\Brefines~\evt@absevt}
125 }
```

The `parameters` is skipped if there are none.

```

126 \B@ifstrequal{\evt@pars}{-}{-}{
127 \newcommand\pretty@pars{}
128 }{
129 \newcommand\pretty@pars{\Bany~\evt@pars~\Bwhere~}
130 }
```

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

```

131 \B@ifstrequal{\evt@grds}{-}{-}{
132 \newcommand\pretty@grds{}
133 }{
134 \newcommand\pretty@grds@tmp{
135 \evt@grds~
136 }
137 \B@ifstrequal{\evt@pars}{-}{-}{
138 \Bwhen~\pretty@grds@tmp
139 }{
140 \newcommand\pretty@grds{\pretty@grds@tmp}
141 }
142 }
```

The `witnesses` are skipped if there are none.

```

143 \B@ifstrequal{\evt@wits}{-}{-}{
144 \newcommand\pretty@wits{}
145 }{
146 \newcommand\pretty@wits{
147 \Bwith~
148 \evt@wits~
149 }
150 }
```

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

```

151 \B@ifstrequal{\evt@acts}{-}{-}{
152 \renewcommand\evt@acts{\SKIP}
153 }{}
154 \newcommand\pretty@acts@tmp{
155 \evt@acts
156 }
157 \newcommand\pretty@acts@keyword{\Bthen}
158 \B@ifstrequal{\evt@pars}{-}{-}{
159 \B@ifstrequal{\evt@grds}{-}{-}{
160 \renewcommand\pretty@acts@keyword{\Bbegin}
161 }{}
162 }{}
163 \newcommand\pretty@acts{
164 \pretty@acts@keyword~
165 \pretty@acts@tmp~
166 }
```

Finally we put all the pretty-print pieces together.

```

167   \begin{array}{l}
168     \Bevt{\evt@label}\pretty@sts\pretty@absepts~\widehat{=}~
169     \pretty@pars
170     \pretty@grds
171     \pretty@wits
172     \pretty@acts
173     \Bend
174   \end{array}
175 } % END group
176 }

```

**\B@makebox** A wrapper macro to make a fbox with the boundary adjusted.

```

177 \newlength{\B@tmplength}
178 \newcommand{\B@makebox}[1]{
179   {
180     \setlength{\B@tmplength}{\fboxsep}
181     \setlength{\fboxsep}{2ex}
182     \fbox{#1}
183     \setlength{\fboxsep}{\B@tmplength}
184   }
185 }

```

### 3.3 Declaration of Options for the Package

In this part various options for the package are defined.

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

**\B@event** Default definition displays Event-B events in a box.

```

186 \newcommand{\B@event}[7] []{
187   \B@makebox{
188     \ensuremath{
189       \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
190     }
191   }
192 }

```

**\B@declaration** Default definition displays Event-B declarations in a box.

```

193 \newcommand{\B@declaration}[2]{
194   \B@makebox{
195     \ensuremath{
196       \B@declarationbase{#1}{#2}
197     }
198   }
199 }

```

**\B@section** Default definition displays Event-B sections in a box

```

200 \newcommand{\B@section}[3] []{
201   \B@makebox{
202     \ensuremath{

```

```

203     \B@sectionbase[#1]{#2}{#3}
204   }
205 }
206 }

```

The above commands are redefined accordingly when option `nobox` is enabled.

```

207 \DeclareOption{nobox}{

```

`\B@event` Redefine the definition without the bounding box.

```

208   \renewcommand{\B@event}[7][]{
209     \B@eventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
210   }

```

`\B@declaration` Redefine the definition without the bounding box.

```

211   \renewcommand{\B@declaration}[2]{
212     \B@declarationbase[#1]{#2}
213   }

```

`\B@section` Redefine the definition without the bounding box.

```

214   \renewcommand{\B@section}[3][]{
215     \B@sectionbase[#1]{#2}{#3}
216   }
217 }

```

**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`.

```

218 \newcommand{\B@fontsize}{\normalsize} % The font size used in Bcode environment
219 \newcommand{\Bvspace}[1][2ex]{\vspace{#1}} % Vertical space
220 \newcommand{\Bhspace}[1][2em]{\hspace{#1}} % Horizontal space
221 \newcommand{\B@tab}{\quad} % A small separation space
222
223 \DeclareOption{small}{
224   \renewcommand{\B@fontsize}{\small}
225   \renewcommand{\Bvspace}[1][1ex]{\vspace{#1}}
226   \renewcommand{\Bhspace}[1][1em]{\hspace{#1}}
227   \renewcommand{\B@tab}{\ }
228 }
229 \DeclareOption{compact}{
230   \renewcommand{\B@fontsize}{\footnotesize}
231   \renewcommand{\Bvspace}[1][0ex]{\vspace{#1}}
232   \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
233   \renewcommand{\B@tab}{\ }
234   \ExecuteOptions{nobox}
235 }
236 \DeclareOption{tiny}{
237   \renewcommand{\B@fontsize}{\scriptsize}
238   \renewcommand{\Bvspace}[1][-0.5ex]{\vspace{#1}}
239   \renewcommand{\Bhspace}[1][0.5em]{\hspace{#1}}
240   \renewcommand{\B@tab}{\ }
241   \ExecuteOptions{nobox}
242 }
243

```

**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.

```

244 \eventB
245 % Default definition for Event-B keywords
246 \eventB
247 \newcommand{\B@keyword}[1]{\ensuremath{\B@keywordbase{#1}}\xspace}
248 \newcommand{\B@identifier}[1]{\ensuremath{\B@identifierbase{#1}}\xspace}
249 % Default definition for Event-B identifiers
250
251 \newcommand{\B@label}[2][ ]{\ensuremath{\B@label{#1}{#2}}\xspace}
252 \newcommand{\B@po}[1]{\ensuremath{\B@po{#1}}\xspace}
253 \DeclareOption{colour}{
254   \newcommand{\setBKeywordColour}[1]{\colorlet{B@keywordcolor}{#1}}
255   \setBKeywordColour{blue}
256   \newcommand{\setBIdentifierColour}[1]{\colorlet{B@identifiercolor}{#1}}
257   \setBIdentifierColour{blue!50!red}
258   \newcommand{\setBLabelColour}[1]{\colorlet{B@labelcolor}{#1}}
259   \setBLabelColour{green!50!black}
260   \newcommand{\setBPOLColour}[1]{\colorlet{B@pocolor}{#1}}
261   \setBPOLColour{red}
262   \renewcommand{\B@keyword}[1]{
263     \ensuremath{\textcolor{B@keywordcolor}{\B@keywordbase{#1}}}\xspace
264   }
265   \renewcommand{\B@identifier}[1]{
266     \ensuremath{\textcolor{B@identifiercolor}{\B@identifierbase{#1}}}\xspace
267   }
268   \renewcommand{\B@label}[2][ ]{
269     \ensuremath{\textcolor{B@labelcolor}{\B@label{#1}{#2}}}\xspace
270   }
271   \renewcommand{\B@po}[1]{
272     \ensuremath{\textcolor{B@pocolor}{\B@po{#1}}}\xspace
273   }
274 }
275 \DeclareOption{color}{
276   \ExecuteOptions{colour}
277 }
278

```

After declaration of options, we execute them accordingly.

```

279 \ProcessOptions

```

### 3.4 Commands for Pretty-Print Event-B Models

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

```

280 \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.

```

281 \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( $\backslash\backslash$ )-separated list of elements.

```

282 \newcommand{\carriersets}[1]{
283   \B@declaration{sets}{#1}
284 }

285 \newcommand{\constants}[1]{
286   \B@declaration{constants}{#1}
287 }

288 \newcommand{\axioms}[2][]{
289   \B@section[#1]{axioms}{#2}
290 }

291 \newcommand{\variables}[1]{
292   \B@declaration{variables}{#1}
293 }

294 \newcommand{\invariants}[2][]{
295   \B@section[#1]{invariants}{#2}
296 }

297 \newcommand{\variant}[1]{
298   \B@declaration{variant}{#1}
299 }

```

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

```

300 \newcommand{\Bany}{\B@keyword{any}}
301 \newcommand{\Bbegin}{\B@keyword{begin}}
302 \newcommand{\Bend}{\B@keyword{end}}
303 \newcommand{\Brefines}{\B@keyword{refines}}
304 \newcommand{\Bstatus}{\B@keyword{status}}
305 \newcommand{\Bthen}{\B@keyword{then}}
306 \newcommand{\Bwhen}{\B@keyword{when}}
307 \newcommand{\Bwhere}{\B@keyword{where}}
308 \newcommand{\Bwith}{\B@keyword{with}}

```

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

```

309 \newcommand{\Bctx}[1]{\ensuremath{\mathbf{#1}}\xspace}
310 \newcommand{\Bset}[1]{\Bidentifier{#1}}
311 \newcommand{\Bcst}[1]{\Bidentifier{#1}}
312 \newcommand{\Baxm}[1]{\Blabel{#1}}
313 \newcommand{\Bthm}[1]{\Blabel{thm}{#1}}
314
315 \newcommand{\Bmch}[1]{\ensuremath{\mathbf{#1}}\xspace}
316 \newcommand{\Bvrb}[1]{\Bidentifier{#1}}
317 \newcommand{\Binv}[1]{\Blabel{#1}}
318 \newcommand{\Bvt}[1]{\Blabel{#1}}
319 \newcommand{\Bpar}[1]{\Bidentifier{#1}}
320 \newcommand{\Bact}[1]{\Blabel{#1}}
321 \newcommand{\Bgrd}[1]{\Blabel{#1}}
322 \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.

```

323 \newcommand{\B@newmacro}[3][]{
324   \ifstrequal{#1}{}{
325     \expandafter\def\csname #2\endcsname{#3{#2}}
326   }{
327     \expandafter\def\csname #1\endcsname{#3{#2}}
328   }
329 }

330 \newcommand{\newBctx}[2][]{\B@newmacro[#1]{#2}{\Bctx}}
331 \newcommand{\newBset}[2][]{\B@newmacro[#1]{#2}{\Bset}}
332 \newcommand{\newBcst}[2][]{\B@newmacro[#1]{#2}{\Bcst}}
333 \newcommand{\newBaxm}[2][]{\B@newmacro[#1]{#2}{\Baxm}}
334 \newcommand{\newBthm}[2][]{\B@newmacro[#1]{#2}{\Bthm}}
335 \newcommand{\newBmch}[2][]{\B@newmacro[#1]{#2}{\Bmch}}
336 \newcommand{\newBvrb}[2][]{\B@newmacro[#1]{#2}{\Bvrb}}
337 \newcommand{\newBinvt}[2][]{\B@newmacro[#1]{#2}{\Binvt}}
338 \newcommand{\newBevt}[2][]{\B@newmacro[#1]{#2}{\Bevt}}
339 \newcommand{\newBpar}[2][]{\B@newmacro[#1]{#2}{\Bpar}}
340 \newcommand{\newBgrd}[2][]{\B@newmacro[#1]{#2}{\Bgrd}}
341 \newcommand{\newBact}[2][]{\B@newmacro[#1]{#2}{\Bact}}

342
343 %%%% Theorem Proof Obligation
344 %%%% Print the theorem proof obligation, given the theorem label.
345 %%%% Arguments:
346 %%%% 1. Theorem label
347 %%%%
348 %%%% Usage:
349 %%%% - \thmpo{thm} will produce "thm/THM"
350 \newcommand{\thmpo}[1]{\Bthm{#1}/\Bpo{THM}}
351
352 %%%% Axiom Well-definedness Proof Obligation
353 %%%% Print the axiom well-definedness proof obligation, given the
354 %%%% axiom label.
355 %%%% Arguments:
356 %%%% 1. Axiom label
357 %%%%
358 %%%% Usage:
359 %%%% - \axmwdpo{axm} will produce "axm/WD"
360 \newcommand{\axmwdpo}[1]{\Baxm{#1}/\Bpo{WD}}
361
362 %%%% Invariant Proof Obligation
363 %%%% Print the invariant proof obligation, given the event name and
364 %%%% invariant label
365 %%%% Arguments:
366 %%%% 1. Event name
367 %%%% 2. Invariant label

```

```

368 %%%%
369 %%%% Usage:
370 %%%% - \invpo{evt}{inv} will produce "evt/inv/INV"
371 \newcommand{\invpo}[2]{\Bevt{#1}/\Binv{#2}/\Bpo{INV}}
372
373 %%%% Theorem (in guard) Proof Obligation
374 %%%% Print the simulation proof obligation, given the event name and
375 %%%% the theorem (in guard) label.
376 %%%% Arguments:
377 %%%% 1. Event name
378 %%%% 2. Theorem (in guard) label
379 %%%%
380 %%%% Usage:
381 %%%% - \grdthmpo{evt}{thm} will produce "evt/thm/THM"
382 \newcommand{\grdthmpo}[2]{\Bevt{#1}/\Bthm{#2}/\Bpo{THM}}
383
384 %%%% Feasibility Proof Obligation
385 %%%% Print the feasibility proof obligation, given the event name and
386 %%%% the action label
387 %%%% Arguments:
388 %%%% 1. Event name
389 %%%% 2. Action label
390 %%%%
391 %%%% Usage:
392 %%%% - \fispo{evt}{act} will produce "evt/act/FIS"
393 \newcommand{\fispo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{FIS}}
394
395 %%%% Variant finiteness Proof Obligation
396 %%%% Print the Variant finiteness proof obligation
397 %%%% Arguments: No arguments
398 %%%%
399 %%%% Usage:
400 %%%% - \finpo will produce "FIN"
401 \newcommand{\finpo}{\Bpo{FIN}}
402
403 %%%% Variant Proof Obligation
404 %%%% Print the guard strengthen proof obligation, given the event name
405 %%%% Arguments:
406 %%%% 1. Event name
407 %%%%
408 %%%% Usage:
409 %%%% - \grdpo{evt} will produce "evt/VAR"
410 \newcommand{\varpo}[1]{\Bevt{#1}/\Bpo{VAR}}
411
412 %%%% Simulation Proof Obligation
413 %%%% Print the simulation proof obligation, given the event name and
414 %%%% the action label.
415 %%%% Arguments:
416 %%%% 1. Event name
417 %%%% 2. Action label
418 %%%%
419 %%%% Usage:

```

```

420 %%%% - \simpo{evt}{act} will produce "evt/act/SIM"
421 \newcommand{\simpo}[2]{\Bevt{#1}/\Bact{#2}/\Bpo{SIM}}
422
423 %%%% Guard Strengthen Proof Obligation
424 %%%% Print the guard strengthen proof obligation, given the event
425 %%%% name and the guard label
426 %%%% Arguments:
427 %%%% 1. (Abstract) Event name
428 %%%% 2. (Abstract) Guard label
429 %%%%
430 %%%% Usage:
431 %%%% - \grdpo{evt}{grd} will produce "evt/grd/GRD"
432 \newcommand{\grdpo}[2]{\Bevt{#1}/\Bgrd{#2}/\Bpo{GRD}}
433
434 %%%% Variant Natural Number Proof Obligation
435 %%%% Print the Variant Natural Number proof obligation, given the event name
436 %%%% Arguments:
437 %%%% 1. Event name
438 %%%%
439 %%%% Usage:
440 %%%% - \natpo{evt} will produce "evt/NAT"
441 \newcommand{\natpo}[1]{\Bevt{#1}/\Bpo{NAT}}
442

```

\inlineevent

```

443 \newcommand{\inlineevent}[7][]{
444   \B@inlineeventbase[#1]{#2}{#3}{#4}{#5}{#6}{#7}
445 }

446 \newcommand{\B@label}[2][]{
447   \ifstrequal{#1}{}{
448     \mathsf{#2}
449   }{
450     \mathit{#2}
451   }
452 }
453
454
455
456
457
458
459 \newcommand{\B@po}[1]{\ensuremath{\mathsf{#1}}\xspace}
460
461 %%%% (BEGIN) Macros for Pretty-Print Event-B Components %%%
462 \newcommand{\SKIP}{\textsc{skip}\xspace}
463 %

```

\INITIALISATION

```

464 %%%% INITIALISATION label
465 \newBevt{INITIALISATION}
466

```

```

467 %%% Pretty print the initialisation: no ‘‘refines’’ clause. no parameters, no
468 %%% guards
469 %%% Arguments:
470 %%% 1. (Newline(\)-separated) list of assignments.
471 %%%
472 %%% Usage: \initialisation{S1(v,x,y)\S2(w,x,y)}
473 %%% will produce the following
474 %%%
475 %%% init
476 %%% begin
477 %%% S1(v, x, y)
478 %%% S2(w, x, y)
479 %%% end
480 %%%
481 \newcommand{\initialisation}[1]{
482   \event{\INITIALISATION}{}{}{}{#1}
483 }
484
485 \newcommand{\event}[7][[] {
486   \B@event[#1]{#2}{#3}{#4}{#5}{#6}{#7}
487 }
488
```

# 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.

	126, 131, 137,	\Bact . 320, 341, 393, 421
\_ . . . . . 227, 233, 240	143, 151, 158, 159	\Bany . . . . . 48, 129, 300
	\B@inlineeventbase .	\Baxm . . . . . 312, 333, 360
<b>A</b>	. . . . . <u>107</u> , 444	\Bbap . . . . . 322
\axioms . . . . . 288	\B@keyword . . 8, 14,	\Bbegin . . . 88, 160, 301
\axmwdpo . . . . . 359, 360	247, 262, 300–308	\Bcst . . . . . 311, 332
	\B@keywordbase . . . .	\Bctx . . . . . 309, 330
<b>B</b>	. . . . . <u>4</u> , 247, 263	\Bend . . . . . 103, 173, 302
\B@declaration . . . .	\B@label . . 251, 269, 446	\Bevt 96, 168, 318, 338,
. . . . <u>193</u> , <u>211</u> ,	\B@makebox . . . . .	371, 382, 393,
283, 286, 292, 298	. <u>177</u> , 187, 194, 201	410, 421, 432, 441
\B@declarationbase .	\B@newmacro 323, 330–341	\bfseries . . . . . 322
. . . . . <u>6</u> , 196, 212	\B@po . . . . . 252, 272, 459	\Bgrd . . . . . 321, 340, 432
\B@event . . <u>186</u> , <u>208</u> , 486	\B@section . . . . .	\Bhspace . . . . .
\B@eventbase <u>26</u> , 189, 209	. <u>200</u> , <u>214</u> , 289, 295	. 220, 226, 232, 239
\B@fontsize . . . 218,	. . . . . <u>11</u> , 203, 215	\Bidentifier 248, 265,
224, 230, 237, 281	\B@tab . . . . . 7, 15,	310, 311, 316, 319
\B@identifierbase .	20, 38, 43, 48,	\Binv . . . . . 317, 337, 371
. . . . . <u>5</u> , 248, 266	54, 60, 71, 72,	\Blabel . . . . . 251,
\B@ifstrequal . . . .	81, 85, 88, 103,	268, 312, 313,
. <u>25</u> , 35, 40, 45,	221, 227, 233, 240	317, 318, 320, 321
50, 58, 67, 77,	\B@tmplength . . . . .	\Bmch . . . . . 315, 335
86, 87, 116, 121.	. . . . . 177, 180, 183	\Bpar . . . . . 319, 339

\Bpo . . . . .	252, 271, 350, 360, 371, 382, 393, 401, 410, 421, 432, 441	\evt@wits . . . . .	33, 67, 73, 114, 143, 148	\pretty@acts@keyword	85, 88, 92, 157, 160, 164
\Brefines . .	43, 124, 303	<b>F</b>		\pretty@acts@tmp . .	80, 93, 154, 165
\Bset . . . . .	310, 331	\finpo . . . . .	400, 401	\pretty@grds . . . . .	51, 59, 64, 100, 132, 140, 170
\Bstatus . . . . .	38, 304	\fispo . . . . .	392, 393	\pretty@grds@tmp	53, 61, 64, 134, 138, 140
\Bthen . . . .	85, 157, 305	<b>G</b>		\pretty@pars . . . .	46, 48, 99, 127, 129, 169
\Bthm . . . .	313, 334, 350, 382	\grdpo . . . .	409, 431, 432	\pretty@sts . . . .	36, 38, 97, 117, 119, 168
\Bvrb . . . . .	316, 336	\grdthmpo . . . .	381, 382	\pretty@wits . . . .	68, 70, 101, 144, 146, 171
\Bvspace . . . . .	219, 225, 231, 238	<b>I</b>		<b>S</b>	
\Bwhen . . . .	60, 138, 306	\INITIALISATION	464, 482	\setBIdentifierColour	256, 257
\Bwhere . . . .	48, 129, 307	\initialisation	472, 481	\setBKeywordColour .	254, 255
\Bwith . . . .	71, 147, 308	\inlineevent . . . . .	443	\setBLabelColour . .	258, 259
<b>C</b>		\invariants . . . . .	294	\setBP0Colour . . . .	260, 261
\carriersets . . . . .	282	\invpo . . . . .	370, 371	\simpo . . . . .	420, 421
\constants . . . . .	285	<b>N</b>		\SKIP . . . . .	78, 152, 462
<b>E</b>		\natpo . . . . .	440, 441	\sl . . . . .	322
\event . . . . .	482, 485	\newBact . . . . .	341	<b>T</b>	
\eventB . . . . .	280	\newBaxm . . . . .	333	\thmpo . . . . .	349, 350
\evt@absepts . . . .	30, 40, 43, 111, 121, 124	\newBcst . . . . .	332	<b>V</b>	
\evt@acts . . . . .	34, 77, 78, 82, 115, 151, 152, 155	\newBctx . . . . .	330	\variables . . . . .	291
\evt@grds . . . . .	32, 50, 55, 87, 113, 131, 135, 159	\newBevt . . . . .	338, 465	\variant . . . . .	297
\evt@label . . . . .	29, 96, 110, 168	\newBgrd . . . . .	340	\varpo . . . . .	410
\evt@pars . . . .	31, 45, 48, 58, 86, 112, 126, 129, 137, 158	\newBinvt . . . . .	337		
\evt@sts . . . . .	28, 35, 38, 109, 116, 119	\newBmch . . . . .	335		
		\newBpar . . . . .	339		
		\newBset . . . . .	331		
		\newBthm . . . . .	334		
		\newBvrb . . . . .	336		
		<b>P</b>			
		\pretty@absepts	41, 43, 98, 122, 124, 168		
		\pretty@acts . . . . .	91, 102, 163, 172		

## Change History

v1.0		v1.1.1	
General: Initial version . . . . .	1	General: Updated documentation . .	1
v1.0.1		v2.0	
\B@declaration: Ensure math-		General: Major re-implementation,	
mode . . . . .	6	use <b>etoolbox</b> instead of <b>ifthen</b>	1
\B@event: Ensure math-mode . . . .	6	Updated documentation, added	
\B@section: Ensure math-mode . .	6	<b>DoNotIndex</b> . . . . .	1
v1.1		\B@makebox: Added . . . . .	6
General: Re-implement how op-		\INITIALISATION: Renamed from	
tions are defined, added options		<b>init</b> . . . . .	12
'nobox' . . . . .	1	\inlineevent: Renamed from	
		<b>eventinline</b> . . . . .	12