

The beamertools Package

Vafa Khalighi

Version 0.1 January 9, 2025

If you want to report any bugs or typos and corrections in the documentation, or ask for any new features, or suggest any improvements, or ask any questions about the package, then please use the issue tracker:

<https://github.com/xepersian/beamertools/issues>

In doing so, please always explain your issue well enough, and always include a minimal working example showing the issue.

You may also have conversations, ask questions and post answers without opening issues using the Discussions space:

<https://github.com/xepersian/beamertools/discussions>

The announcements for the new releases of the package will also appear in the Discussions space under the Announcements category.

Copyright (c) 2025 Vafa Khalighi

Permission is granted to distribute and/or modify *both the documentation and the code* under the conditions of the L^AT_EX Project Public License, either version 1.3c of this license or (at your option) any later version.

Contents

1	Introduction	2
2	Using the package	2
2.1	Loading The Package	2
2.2	The user interface	3
3	Implementation	3
4	Index	3

1 Introduction

If you want to redefine any predefined beamer template and try something like

```
\defbeamer@template{itemize item}{square}{\hbox{\vrule width 2ex height 2ex}}
```

then beamer will give the following unhelpful error message

```
./test.tex:4: LaTeX Error: Command \beamer@tmpop@itemize item@square
already d
efined.

          Or name \end... illegal, see p.192 of the manual.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

1.4 \begin
      {document}
?
```

and one is puzzled as what to do.

beamer does not provide a control sequence for redefining a predefined beamer template. One way to get around this issue is to undefine the control sequence shown above and then to redefine the beamer template using the control sequence `\defbeamer@template`. But this is really ugly and unpleasant.

Of course, this is just a silly example for illustrating the issue. I found myself redefining a lot of predefined beamer templates in my `bidi`¹ package and the workaround above is just ugly and painful.

The `beamer@tools` package solves this exact issue. It provides a control sequence for redefining a predefined beamer template and unlike beamer unhelpful error message, it provides an informative error message if the beamer template you are trying to redefine is not predefined.

2 Using the package

2.1 Loading The Package

You can load the package in the ordinary way:

¹<https://ctan.org/pkg/bidi>

```
\usepackage{beamertools}
```

2.2 The user interface

The packages provides the control sequence `\redefbeamertemplate` for redefining a predefined beamer template and has exactly the same syntax as the control sequence `\defbeamertemplate` that beamer provides.

3 Implementation

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{beamertools}[2025/01/09 v0.1 A collection of programming tools for beamer]
3 \newcommand<>\redefbeamertemplate{
4   \ifblank{#1}
5     {\def\beamer@oar{<presentation>}}
6     {\def\beamer@oar{#1}}\beamer@redefbeamertemplate}
7
8 \def\beamer@redefbeamertemplate{\@ifstar{\beamer@sbttrue\beamer@redef@sbtoo}{\beamer@sbtfalse}
9
10 \def\beamer@redef@sbtoo#1#2{\@ifnextchar[{\beamer@redef@sbto{#1}{#2}}{\beamer@redef@sbto{#1}{f
11
12 \def\beamer@redef@sbto#1#2[#3]{\@ifnextchar[{\beamer@@redef@sbto{#1}{#2}][#3]}{\def\beamer@sb
13
14 \long\def\beamer@@redef@sbto#1#2[#3][#4]{\long\def\beamer@sbtotemp{[#4]}\beamer@@redef@sbto
15
16 \long\def\beamer@@redef@sbto#1#2[#3]#4{%
17   \beamer@sbttoks={\beamer@@@redef@sbto{#1}{#2} [#3] [#4]}%
18   \@ifnextchar[{\the\beamer@sbttoks}{\the\beamer@sbttoks[action] {}}%
19 }
20
21 \long\def\beamer@@@redef@sbto#1#2[#3]#4[#5]#6{%
22   \edef\beamer@@sbtoarg{\expandafter\noexpand\csname beamer@tmpop@#1@#2\endcsname[#3]}%
23   \@ifundefined{beamer@tmpop@#1@#2}{%
24     \PackageError{beamertools}{beamer template with element name ‘#1’ and predefined option ‘#2
25 }
26 \expandafter\expandafter\expandafter\renewcommand\expandafter\beamer@@sbtoarg\beamer@sbtotemp
27   \expandafter\long\expandafter\def\csname beamer@tmp1@#1\endcsname{#4}#6}%
28   \let\beamer@@next=\beamer@resetsbtoarg%
29   \expandafter\mode\beamer@oar{\let\beamer@@next=\relax}%
30   \beamer@@next%
31   \ifbeamer@sbt\setbeamertemplate{#1} [#2]\relax\relax\relax\relax\relax\relax\relax\relax\relax\relax
32 }

```

4 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	\@ifstar	8	B
\@ehc	<u>24</u>	\beamer@@@redef@sbto
\@ifnextchar	. 10, 12, 18	\@ifundefined 17, 21

\beamer@@redef@sbto			
.....	12, 14, 16		
\beamer@@next	28, 29, 30		
\beamer@@oar	... 5, 6, 29		
\beamer@redef@sbto			
.....	12, 14		
\beamer@@sbtoarg	22, 26		
\beamer@redef@sbto			
.....	10, 12		
\beamer@redef@sbtoo			
.....	8, 10		
\beamer@redefbeamertemplate			
.....	6, 8		
\beamer@resetsbtoarg	28		
\beamer@sbtfalse	... 8		
\beamer@sbtotemp	...		
.....	12, 14, 26		
\beamer@sbttoks	17, 18		
\beamer@sbttrue	... 8		
		C	
\csname 22, 27		
		D	
\def 5, 6, 8, 10, 12, 14, 16, 21, 27		
		E	
\edef 22		
\endcsname 22, 27		
\expandafter 22, 26, 27, 29		
		F	
\fi 31		
		I	
\ifbeamer@sbt 31		
\ifblank 4		
		L	
\let 28, 29		
\long 14, 16, 21, 27		
		M	
\mode 29		
		N	
\NeedsTeXFormat 1		
\newcommand 3		
\noexpand 22		
		P	
\PackageError 24		
\ProvidesPackage 2		
		R	
\redefbeamertemplate	3		
\relax 29, 31		
\renewcommand 26		
		S	
\setbeamertemplate	.. 31		
		T	
\the 18		