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.

Contents

1 Introduction

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

```
\defbeamertemplate{itemize item}{square}{\hbox{\vrule width 2ex height 2
ex}}
```

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.
 ...

l.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 `\defbeamertemplate`. 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 beamertools 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

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{beamertools}[2025/01/09 v0.1 A collection of programming tools for beamer]
\newcommand<>\redefbeamertemplate{
  \ifblank{#1}
  {\def\beamer@@oar{<presentation>}}
  {\def\beamer@@oar{#1}}\beamer@redefbeamertemplate}

\def\beamer@redefbeamertemplate{\@ifstar{\beamer@sbttrue\beamer@redef@sbtoo}{\beamer@sbtfalse\

\def\beamer@redef@sbtoo#1#2{\@ifnextchar[{\beamer@redef@sbto{#1}{#2}}{\beamer@redef@sbto{#1}{#

\def\beamer@redef@sbto#1#2[#3]{\@ifnextchar[{\beamer@@redef@sbto{#1}{#2}[{#3}]}{\def\beamer@sb

\long\def\beamer@@redef@sbto#1#2[#3][#4]{\long\def\beamer@sbtotemp{[{#4}]}\beamer@@@redef@sbto

\long\def\beamer@@@redef@sbto#1#2[#3]#4{%
  \beamer@sbttoks={\beamer@@@@redef@sbto{#1}{#2}[{#3}]{#4}}%
  \@ifnextchar[{\the\beamer@sbttoks}{\the\beamer@sbttoks[action]{}}%
}

\long\def\beamer@@@@redef@sbto#1#2[#3]#4[#5]#6{%
  \edef\beamer@@sbtoarg{\expandafter\noexpand\csname beamer@@tmpop@#1@#2\endcsname[{#3}]}%
\@ifundefined{beamer@@tmpop@#1@#2}{%
  \PackageError{beamertools}{beamer template with element name `#1' and predefined option `#2
}
\expandafter\expandafter\expandafter\renewcommand\expandafter\beamer@@sbtoarg\beamer@sbtotemp
    \expandafter\long\expandafter\def\csname beamer@@tmpl@#1\endcsname{#4}#6}%
  \let\beamer@@next=\beamer@resetsbtoarg%
  \expandafter\mode\beamer@@oar{\let\beamer@@next=\relax}%
  \beamer@@next%
  \ifbeamer@sbt\setbeamertemplate{#1}[{#2}]\relax\relax\relax\relax\relax\relax\relax\relax\re
}
```

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.