

# Package ‘texPreview’

December 5, 2019

**Type** Package

**Title** Compile and Preview Snippets of 'LaTeX' in 'RStudio'

**Version** 1.4.2

**Date** 2019-12-01

**Maintainer** Jonathan Sidi <yonis@metrumrg.com>

**Description** Compile and preview snippets of 'LaTeX'. Can be used directly from the R console, from 'RStudio', in Shiny apps and R Markdown documents. Must have 'pdflatex' or 'xelatex' or 'lualatex' in 'PATH'.

**License** GPL-2 | GPL-3

**URL** <https://github.com/metrumresearchgroup/texPreview>

**BugReports** <https://github.com/metrumresearchgroup/texPreview/issues>

**Depends** R (>= 3.3.0)

**Imports** base64enc,

fs,  
htmltools,  
knitr,  
lifecycle,  
magick,  
magrittr,  
rematch2,  
rstudioapi,  
svgPanZoom,  
utils,  
whisker,  
xml2

**Suggests** covr,  
kableExtra,  
nlme,  
rmarkdown,  
shiny,  
slickR,  
testthat,  
texreg,  
xtable,  
details,  
pdfutils

**VignetteBuilder** knitr  
**RdMacros** details,  
          lifecycle  
**LazyData** false  
**NeedsCompilation** no  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 6.1.1

**R topics documented:**

|                            |    |
|----------------------------|----|
| as.kable . . . . .         | 2  |
| buildUsepackage . . . . .  | 3  |
| build_usepackage . . . . . | 4  |
| getTexPackages . . . . .   | 5  |
| get_texpackages . . . . .  | 5  |
| texPreview . . . . .       | 6  |
| tex_opts . . . . .         | 7  |
| tex_preview . . . . .      | 8  |
| %>% . . . . .              | 10 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>11</b> |
|--------------|-----------|

---

|          |  |
|----------|--|
| as.kable | <i>Try to coerce an object into a knitr_kable object</i> |
|----------|--|

---

**Description**

coerce objects into a knitr\_kable class object with a latex format

**Usage**

as.kable(x)

**Arguments**

x                      object

**Value**

an object of class knitr\_kable

**Examples**

```
tex <- '\\begin{tabular}{llr}
\\hline
\\multicolumn{2}{c}{Item} \\
\\cline{1-2}
Animal      & Description & Price (\\$) \\
\\hline
Gnat        & per gram    & 13.65      \\
\\end{tabular}
```

```

& each      & 0.01      \\\
Gnu         & stuffed   & 92.50      \\\
Emu         & stuffed   & 33.33      \\\
Armadillo   & frozen    & 8.99      \\\
\\hline
\\end{tabular}'

ktex <- as.kable(tex)

class(ktex)

attributes(ktex)

ktex

# using an input call

x <- tex_preview(tex,returnType = 'input')

x

ktex_input <- as.kable(x)

class(ktex_input)

attributes(ktex_input)

ktex_input

# file path

x <- tex_preview(tex,returnType = 'input')

ktex_path <- as.kable(file.path(tempdir(),'tex_temp.tex'))

class(ktex_path)

attributes(ktex_path)

ktex_path

```

---

buildUsepackage

*Build usepackage command for TeX document* **Deprecated**


---

### Description

This function has been deprecated, use [build\\_usepackage](#) instead.

### Usage

```

buildUsepackage(pkg, options = NULL, uselibrary = NULL,
  chk.inst = FALSE)

```

**Arguments**

|            |   |
|------------|---|
| pkg        | character, name of TeX package  |
| options    | character, name(s) of options to use in the package                                     |
| uselibrary | character, part of document preamble to specify a uselibrary call related to package    |
| chk.inst   | logical, invokes a check to see if pkg is currently installed on system (default FALSE) |

---

|                  |  |
|------------------|--|
| build_usepackage | <i>Build usepackage command for TeX document</i> |
|------------------|--|

---

**Description**

input TeX package name and optional package functions to create usepackage call

**Usage**

```
build_usepackage(pkg, options = NULL, uselibrary = NULL,
  chk.inst = FALSE)
```

**Arguments**

|            |   |
|------------|---|
| pkg        | character, name of TeX package  |
| options    | character, name(s) of options to use in the package                                     |
| uselibrary | character, part of document preamble to specify a uselibrary call related to package    |
| chk.inst   | logical, invokes a check to see if pkg is currently installed on system (default FALSE) |

**Details**

if options and uselibrary are NULL (default) then only the call for the package is returned. See the TeX wikibook for more information [https://en.wikibooks.org/wiki/LaTeX/Document\\_Structure#Packages](https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Packages) on the usepackage command. If chk.inst finds that the package is not installed on system function returns NULL.

**Value**

character

**Examples**

```
build_usepackage(pkg = 'xcolor')
build_usepackage(pkg = 'xcolor', options = 'usenames')

#build many at once using mapply

geom.opts=c('paperwidth=35cm', 'paperheight=35cm', 'left=2.5cm', 'top=2.5cm')
use.opts="\usetikzlibrary{mindmap,backgrounds}"
```

```

unlist(mapply(build_usepackage,
pkg =      list('times','geometry','tikz'),
options=    list(NULL,geom.opts ,NULL),
uselibrary = list(NULL, NULL, use.opts)
))

```

---

getTexPackages

*Get list of TeX packages installed on System* **Deprecated**


---

### Description

This function has been deprecated, use [get\\_texpackages](#) instead.

### Usage

```
getTexPackages()
```

---

get\_texpackages

*Get list of TeX packages installed on System*


---

### Description

Fetch all TeX packages currently installed on system

### Usage

```
get_texpackages()
```

### Details

If OS is Windows function checks against MikTeX else function checks against TeXLive.

### Value

character

### Examples

```
## Not run: head(get_texpackages())
```

texPreview

*Render and Preview snippets of TeX in R Viewer* **Deprecated****Description**

This function has been deprecated, use [tex\\_preview](#) instead.

**Usage**

```
texPreview(obj, tex_lines = NULL, stem = "tex_temp",
  overwrite = TRUE, keep_pdf = FALSE, tex_message = FALSE,
  fileDir = tex_opts$get("fileDir"), margin = tex_opts$get("margin"),
  imgFormat = tex_opts$get("imgFormat"),
  returnType = tex_opts$get("returnType"),
  resizebox = tex_opts$get("resizebox"),
  usrPackages = tex_opts$get("usrPackages"),
  engine = tex_opts$get("engine"), cleanup = tex_opts$get("cleanup"),
  density = tex_opts$get("density"), svg_max = tex_opts$get("svg_max"),
  print.xtable.opts = tex_opts$get("print.xtable.opts"),
  opts.html = tex_opts$get("opts.html"), markers = interactive(), ...)
```

**Arguments**

|             |  |
|-------------|--|
| obj         | object to convert to TeX script  |
| tex_lines   | vector of character, in case of special needs, instead of asking texPreview to build up, you may choose to pass in the contents of the complete LaTeX file directly. It should be a vector of character with each element as a line of raw TeX code. |
| stem        | character, name to use in output files, Default: "tex_temp"  |
| overwrite   | logical, controls if overwriting of output stem* files given their existences, Default: TRUE   |
| keep_pdf    | logical, controls if the rendered pdf file should be kept or deleted, Default: FALSE   |
| tex_message | logical, controls if latex executing messages are displayed in console. Default: FALSE   |
| fileDir     | character, output destination. If NULL a temp.dir() will be used and no output will be saved, Default: tex_opts\$get('fileDir')  |
| margin      | table margin for pdflatex call, Default: tex_opts\$get('margin')   |
| imgFormat   | character, defines the type of image the PDF is converted to Default: tex_opts\$get('imgFormat')   |
| returnType  | character, one of "viewer", "html", or "tex" determining appropriate return type for the rendering process, Default: tex_opts\$get('returnType')   |
| resizebox   | logical, forces a tabular tex object to be constrained on the margins of the document, Default: tex_opts\$get('resizebox')   |
| usrPackages | character, vector of usepackage commands, see details for string format  |
| engine      | character, specifies which latex to pdf engine to use ('pdflatex', 'xelatex', 'lualatex'), Default: tex_opts\$get('engine')  |
| cleanup     | character, vector of file extensions to clean up after building pdf, Default: tex_opts\$get('cleanup')   |

|                   |  |
|-------------------|--|
| density           | numeric, controls the density of the image. Default is 150: <code>tex_opts\$get('density')</code>  |
| svg_max           | numeric, maximum svg file size allowable to preview, Default: <code>tex_opts\$get('svg_max')</code>  |
| print.xtable.opts | list, contains arguments to pass to <code>print.table</code> , relevant only if <code>xtable</code> is used as the input, Default: <code>tex_opts\$get('print.xtable.opts')</code> |
| opts.html         | list, html options, Default: <code>tex_opts\$get('opts.html')</code>   |
| markers           | logical, if TRUE then RStudio markers will be invoked to create links for the log file on rendering errors, Default: <code>interactive()</code>                                    |
| ...               | passed to <a href="#">system2</a>  |

---

|          |  |
|----------|--|
| tex_opts | <i>Default and current tex options</i> |
|----------|--|

---

## Description

Options for functions in the `texPreview` package. When running R code, the object `tex_opts` (default options) is not modified by chunk headers (local chunk options are merged with default options), whereas `tex_opts_current` (current options) changes with different chunk headers and it always reflects the options for the current chunk.

## Usage

`tex_opts`

`tex_opts_current`

## Format

An object of class `list` of length 5.

## Details

Normally we set up the global options once in the first code chunk in a document using `tex_opts$set()`, so that all *latter* chunks will use these options. Note the global options set in one chunk will not affect the options in this chunk itself, and that is why we often need to set global options in a separate chunk.

Below is a list of default chunk options, retrieved via `tex_opts$get()`:

These options correspond to fields in the direct call to [tex\\_preview](#), which are listed in explained in the help manual.

## Note

`tex_opts_current` is read-only in the sense that it does nothing if you call `tex_opts_current$set()`; you can only query the options via `tex_opts_current$get()`.

## Examples

```
tex_opts$get()
```

tex\_preview

*Render and Preview snippets of TeX in R Viewer***Description**

input TeX script into the function and it renders a pdf and converts it an image which is sent to Viewer.

**Usage**

```
tex_preview(obj, tex_lines = NULL, stem = "tex_temp",
  overwrite = TRUE, keep_pdf = FALSE, tex_message = FALSE,
  fileDir = tex_opts$get("fileDir"), margin = tex_opts$get("margin"),
  imgFormat = tex_opts$get("imgFormat"),
  returnType = tex_opts$get("returnType"),
  resizebox = tex_opts$get("resizebox"),
  usrPackages = tex_opts$get("usrPackages"),
  engine = tex_opts$get("engine"), cleanup = tex_opts$get("cleanup"),
  density = tex_opts$get("density"), svg_max = tex_opts$get("svg_max"),
  print.xtable.opts = tex_opts$get("print.xtable.opts"),
  opts.html = tex_opts$get("opts.html"), markers = interactive(), ...)
```

**Arguments**

|             |  |
|-------------|--|
| obj         | object to convert to TeX script  |
| tex_lines   | vector of character, in case of special needs, instead of asking texPreview to build up, you may choose to pass in the contents of the complete LaTeX file directly. It should be a vector of character with each element as a line of raw TeX code. |
| stem        | character, name to use in output files, Default: "tex_temp"  |
| overwrite   | logical, controls if overwriting of output stem* files given their existences, Default: TRUE   |
| keep_pdf    | logical, controls if the rendered pdf file should be kept or deleted, Default: FALSE   |
| tex_message | logical, controls if latex executing messages are displayed in console. Default: FALSE   |
| fileDir     | character, output destination. If NULL a temp.dir() will be used and no output will be saved, Default: tex_opts\$get('fileDir')  |
| margin      | table margin for pdflatex call, Default: tex_opts\$get('margin')   |
| imgFormat   | character, defines the type of image the PDF is converted to Default: tex_opts\$get('imgFormat')   |
| returnType  | character, one of "viewer", "html", or "tex" determining appropriate return type for the rendering process, Default: tex_opts\$get('returnType')   |
| resizebox   | logical, forces a tabular tex object to be constrained on the margins of the document, Default: tex_opts\$get('resizebox')   |
| usrPackages | character, vector of usepackage commands, see details for string format  |
| engine      | character, specifies which latex to pdf engine to use ('pdflatex', 'xelatex', 'lualatex'), Default: tex_opts\$get('engine')  |



|                   |  |
|-------------------|--|
| cleanup           | character, vector of file extensions to clean up after building pdf, Default: <code>tex_opts\$get('cleanup')</code>  |
| density           | numeric, controls the density of the image. Default is 150: <code>tex_opts\$get('density')</code>  |
| svg_max           | numeric, maximum svg file size allowable to preview, Default: <code>tex_opts\$get('svg_max')</code>  |
| print.xtable.opts | list, contains arguments to pass to <code>print.table</code> , relevant only if <code>xtable</code> is used as the input, Default: <code>tex_opts\$get('print.xtable.opts')</code> |
| opts.html         | list, html options, Default: <code>tex_opts\$get('opts.html')</code>   |
| markers           | logical, if TRUE then RStudio markers will be invoked to create links for the log file on rendering errors, Default: <code>interactive()</code>                                    |
| ...               | passed to <a href="#">system2</a>  |

## Details

`tex_preview` is an S3 method that can be used to preview TeX output from different object classes.

Built-in support includes:

- character (tex lines)
- `knitr_kable` (kable/kableExtra)
- xtable
- texreg
- equatimatic

The function assumes the system has `pdflatex` installed and it is defined in the PATH.

To add packages to the tex file on render there are two options

- Use [build\\_usepackage](#) and use the input argument `usrPackages`.
- Append to the input object `\usepackage{...}` calls, they will be parsed and added the to rendering.
- An image file of the name stem with the extension specified in `imgFormat`.
- The default extension is png.
- The function writes two files to disk in the `fileDir`
  - Image file
  - TeX script
- The rendering files are removed up from the `fileDir`. This can be controlled using the `cleanup` argument or `tex_opts$get('cleanup')`

## Value

The output of the function is dependent on the value of `returnType`:

- viewer: NULL
  - magick image is printed in the internal viewer
- tex:
  - character, TeX lines
  - printed 'asis' in RMarkdown
- input: character

- path to the file containing the tex wrapped in an input call
- printed 'asis' in RMarkdown
- html: magick image
  - Printed as an HTML document in the internal viewer
  - Printed as an image in RMarkdown

## Examples

```
data('iris')
if(interactive()){

# Raw TeX

tex <- '\\begin{tabular}{llr}
\\hline
\\multicolumn{2}{c}{Item} \\
\\cline{1-2}
Animal    & Description & Price (\\$) \\
\\hline
Gnat      & per gram    & 13.65      \\
& each        & 0.01       \\
Gnu       & stuffed     & 92.50      \\
Emu       & stuffed     & 33.33      \\
Armadillo & frozen      & 8.99       \\
\\hline
\\end{tabular}'

# knitr kable

mtcars%>%
  head()%>%
  knitr::kable("latex")%>%
  tex_preview()

# with svg output pan/zoom is enabled in the internal viewer

tex_preview(obj = tex,stem = 'eq',imgFormat = 'svg')

# use tex_lines parameter to pass full document

tikz_path <- system.file(
  'examples/tikz/credit_rationing.tex',
  package = 'texPreview'
)

tex_preview(tex_lines = readLines(tikz_path))

}
```

---

%>%

*re-export magrittr pipe operators*

---

## Description

re-export magrittr pipe operators

# Index

\*Topic **datasets**

tex\_opts, [7](#)

%>%, [10](#)

as.kable, [2](#)

build\_usepackage, [3](#), [4](#), [9](#)

buildUsepackage, [3](#)

get\_texpackages, [5](#), [5](#)

getTexPackages, [5](#)

system2, [7](#), [9](#)

tex\_opts, [7](#)

tex\_opts\_current (tex\_opts), [7](#)

tex\_preview, [6](#), [7](#), [8](#)

texPreview, [6](#)