

# Package: ezknitr (via r-universe)

July 12, 2024

**Title** Avoid the Typical Working Directory Pain When Using 'knitr'

**Version** 0.6.3

**Description** An extension of 'knitr' that adds flexibility in several ways. One common source of frustration with 'knitr' is that it assumes the directory where the source file lives should be the working directory, which is often not true. 'ezknitr' addresses this problem by giving you complete control over where all the inputs and outputs are, and adds several other convenient features to make rendering markdown/HTML documents easier.

**URL** <https://docs.ropensci.org/ezknitr/>,  
<https://github.com/ropensci/ezknitr>

**BugReports** <https://github.com/ropensci/ezknitr/issues>

**Depends** R (>= 3.0.2)

**Imports** knitr (>= 1.7), markdown (>= 0.7), R.utils (>= 1.34.0)

**Suggests** testthat (>= 0.9.1), rmarkdown

**License** MIT + file LICENSE

**Encoding** UTF-8

**SystemRequirements** pandoc with https support

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**Repository** <https://ropensci.r-universe.dev>

**RemoteUrl** <https://github.com/ropensci/ezknitr>

**RemoteRef** master

**RemoteSha** 4bd1832a7bea661816bc9c1777b2f85d1dd7a594

## Contents

ezknitr_core . . . . .	2
open_output_dir . . . . .	5
setup_ezspin_test . . . . .	5
set_default_params . . . . .	6

---

ezknitr_core	<i>Knit Rmd or spin R files without the typical pain of working directories</i>
--------------	---

---

## Description

ezknitr is an extension of knitr that adds flexibility in several ways. One common source of frustration with knitr is that it assumes the directory where the source file lives should be the working directory, which is often not true. ezknitr addresses this problem by giving you complete control over where all the inputs and outputs are, and adds several other convenient features. The two main functions are ezknit and ezspin, which are wrappers around knitr's knit and spin, used to make rendering markdown/HTML documents easier.

## Usage

```
ezspin(  
  file,  
  wd,  
  out_dir,  
  fig_dir,  
  out_suffix,  
  params = list(),  
  verbose = FALSE,  
  chunk_opts = list(tidy = FALSE),  
  keep_rmd = FALSE,  
  keep_md = TRUE,  
  keep_html = TRUE,  
  move_intermediate_file = TRUE,  
  ...  
)  
  
ezknit(  
  file,  
  wd,  
  out_dir,  
  fig_dir,  
  out_suffix,  
  params = list(),  
  verbose = FALSE,  
  chunk_opts = list(tidy = FALSE),  
  keep_md = TRUE,  
  keep_html = TRUE  
)
```

## Arguments

file	The path to the input file (.Rmd file if using ezknit or .R script if using ezspin). If wd is provided, then this path is relative to wd.
------	---

<code>wd</code>	The working directory to be used in the Rmd/R script. Defaults to the current working directory (note that this is not the same behaviour as <code>knitr</code> ). See the 'Detailed Arguments' section for more details.
<code>out_dir</code>	The output directory for the rendered markdown or HTML files (if <code>wd</code> is provided, then this path is relative to <code>wd</code> ). Defaults to the directory containing the input file.
<code>fig_dir</code>	The name (or path) of the directory where figures should be generated. See the 'Detailed Arguments' section for more details.
<code>out_suffix</code>	A suffix to add to the output files. Can be used to differentiate outputs from runs with different parameters. The name of the output files is the name of the input file appended by <code>out_suffix</code> , separated by a dash.
<code>params</code>	A named list of parameters to be passed to use in the input Rmd/R file. For example, if the script to execute assumes that there is a variable named <code>DATASET_NAME</code> , then you can use <code>params = list('DATASET_NAME' = 'oct10dat')</code> .
<code>verbose</code>	If <code>TRUE</code> , then show the progress of knitting the document.
<code>chunk_opts</code>	List of <code>knitr</code> chunk options to use. See <code>?knitr::opts_chunk</code> for a list of available chunk options.
<code>keep_rmd, keep_md</code>	Should intermediate Rmd or md files be kept ( <code>TRUE</code> ) or deleted ( <code>FALSE</code> )?
<code>keep_html</code>	Should the final html file be kept ( <code>TRUE</code> ) or deleted ( <code>FALSE</code> )?
<code>move_intermediate_file</code>	Should the intermediate Rmd file be moved to the output folder ( <code>TRUE</code> ) or stay in the same folder as the source R file ( <code>FALSE</code> )?
<code>...</code>	Any extra parameters that should be passed to <code>knitr::spin</code> .

## Details

If you have a very simple project with a flat directory structure, then `knitr` works great. But even something as simple as trying to knit a document that reads a file from a different directory or placing the output rendered files in a different folder cannot be easily done with `knitr`.

`ezknitr` improves basic `knitr` functionality in a few ways. You get to decide:

- What the working directory of the source file is
- Where the output files will go
- Where the figures used in the markdown will go
- Any parameters to pass to the source file

## Value

The path to the output directory (invisibly).

## Detailed Arguments

All paths given in the arguments can be either absolute or relative.

The `wd` argument is very important and is set to the current working directory by default. The path of the input file and the path of the output directory are both relative to `wd` (unless they are absolute paths). Moreover, any code in the R script that reads or writes files will use `wd` as the working directory.

The `fig_dir` argument is relative to the output directory, since the figures accompanying a markdown file should be placed in the same directory. It is recommended to either leave `fig_dir` as default or set it to a different name but not to a different directory. Because of the way `knitr` works, there are a few known minor issues if `fig_dir` is set to a different directory.

## Difference between `ezknit` and `ezspin`

`ezknit` is a wrapper around `knitr::knit` while `ezspin` is a wrapper around `ezspin`. The two functions are very similar. `knit` is the more popular and well-known function. It is used to render a markdown/HTML document from an Rmarkdown source. `spin` takes an R script as its input, produces an Rmarkdown document from the R script, and then calls `knit` on it.

## See Also

[open\\_output\\_dir](#) [setup\\_ezknit\\_test](#) [setup\\_ezspin\\_test](#) [set\\_default\\_params](#) [knit](#) [spin](#)

## Examples

```
## Not run:
tmp <- setup_ezknit_test()
ezknit("R/ezknit_test.Rmd", wd = "ezknitr_test")
ezknit("R/ezknit_test.Rmd", wd = "ezknitr_test",
      out_dir = "output", fig_dir = "coolplots",
      params = list(numPoints = 50))
open_output_dir()
unlink(tmp, recursive = TRUE, force = TRUE)

tmp <- setup_ezspin_test()
ezspin("R/ezspin_test.R", wd = "ezknitr_test")
ezspin("R/ezspin_test.R", wd = "ezknitr_test",
      out_dir = "output", fig_dir = "coolplots",
      params = list(numPoints = 50), keep_rmd = TRUE)
open_output_dir()
unlink(tmp, recursive = TRUE, force = TRUE)

## End(Not run)
```

---

open_output_dir	<i>Open the directory containing the output from the last ezknitr command</i>
-----------------	---

---

### Description

Call this function after running [ezspin](#) or [ezknitr](#) to open the resulting output directory in your file browser. This is simply a convenience function so that if you want to see the results you don't need to navigate to the appropriate folder manually.

### Usage

```
open_output_dir()
```

### Examples

```
## Not run:  
library(ezknitr)  
setup_ezspin_test()  
ezspin("R/ezspin_test.R", wd = "ezknitr_test")  
open_output_dir()  
  
## End(Not run)
```

---

setup_ezspin_test	<i>Set up a test directory to experiment with ezspin or ezknitr</i>
-------------------	---

---

### Description

Create a few directories that try to mimic a real data-analysis project structure, and add a data file and a simple R script (for ezspin) or Rmarkdown file (for ezknitr).

After setting up these files and directories, you can run ezknitr commands and their equivalent knitr commands to compare and see the benefits of using ezknitr.

More specific instructions on how to interact with this test directory will be printed to the console.

### Usage

```
setup_ezspin_test()
```

```
setup_ezknitr_test()
```

### Value

The path to the test directory.

**See Also**

[ezspin spin ezknit knit open\\_output\\_dir](#)

**Examples**

```
## Not run:
library(ezknitr)

# setup the test directory structures and run naive spin
setup_ezspin_test()
knitr::spin("ezknitr_test/R/ezspin_test.R")
file.remove(c("ezspin_test.md", "ezspin_test.html"))

# setup the test directory structures and run simple ezspin
setup_ezspin_test()
ezspin("R/ezspin_test.R", wd = "ezknitr_test")

# setup the test directory structures and run ezspin with more parameters
tmp <- setup_ezspin_test()
ezspin("R/ezspin_test.R", wd = "ezknitr_test",
      out_dir = "output", fig_dir = "coolplots")
unlink(tmp, recursive = TRUE, force = TRUE)

## End(Not run)
```

---

set\_default\_params      *Set default parameters*

---

**Description**

Create variables with the given values only if these variables do not currently exist.

**Usage**

```
set_default_params(params)
```

**Arguments**

params              List of parameters.

**Details**

Sometimes it may be useful to define a variable only if it hasn't been defined yet. One example where this can be useful is when you have an Rmd script that uses some variables and you want to be able to use custom values for these variables, but also give them a default value in the script in case they are not set beforehand.

**Examples**

```
exists("foo")
exists("bar")
foo <- 5
set_default_params(list(foo = 10, bar = 20))
print(foo)
print(bar)
```

# Index

ezkmit, [5](#), [6](#)  
ezkmit (ezknitr\_core), [2](#)  
ezknitr\_core, [2](#)  
ezspin, [5](#), [6](#)  
ezspin (ezknitr\_core), [2](#)

knit, [4](#), [6](#)

open\_output\_dir, [4](#), [5](#), [6](#)

set\_default\_params, [4](#), [6](#)  
setup\_ezkmit\_test, [4](#)  
setup\_ezkmit\_test (setup\_ezspin\_test), [5](#)  
setup\_ezspin\_test, [4](#), [5](#)  
setup\_test (setup\_ezspin\_test), [5](#)  
spin, [4](#), [6](#)