

# Package ‘cliff’

October 12, 2022

**Title** Execute Command Line Programs Interactively

**Version** 0.1.2

**Description**

Execute command line programs and format results for interactive use. It is based on the package 'processx' so it does not use shell to start up the process like system() and system2(). It also provides a simpler and cleaner interface than processx::run().

**License** MIT + file LICENSE

**URL** <https://github.com/RTagBot/cliff>

**BugReports** <https://github.com/RTagBot/cliff/issues>

**Suggests** withr, crayon, testthat

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** ellipsis, processx, rlang

**NeedsCompilation** no

**Author** Randy Lai [aut, cre, cph]

**Maintainer** Randy Lai <randy.cs.lai@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-11-02 08:20:09 UTC

## R topics documented:

run . . . . . 2

**Index** 3

---

run	<i>Run a command line program and wait until it terminates.</i>
-----	---

---

## Description

Run a command line program and wait until it terminates.

## Usage

```
run(  
  command,  
  ...,  
  input = NULL,  
  error_on_status = TRUE,  
  wd = NULL,  
  timeout = Inf,  
  env = NULL  
)
```

## Arguments

command	the command to run
...	the arguments pass to the program, supports the <b>big bang</b> operator !!!
input	text pass to stdin
error_on_status	raise an error if return code is not 0.
wd	working directory
timeout	throw an error after this amount of time in second
env	additional environment variables

## Value

The stdout of the program in a scalar character. It may contain a trailing newline. Use `trimws()` to ensure the trailing newline is trimmed.

## Examples

```
## Not run:  
git <- function(...) cliff::run("git", ...)  
git("log", git("rev-parse", "--abbrev-ref", "HEAD"), "-n1")  
  
## End(Not run)
```

# Index

run, [2](#)