

# Package ‘whirl’

April 15, 2025

**Title** Log Execution of Scripts

**Version** 0.2.0

**Description** Logging of scripts suitable for clinical trials using 'Quarto' to create nice human readable logs. 'whirl' enables execution of scripts in batch, while simultaneously creating logs for the execution of each script, and providing an overview summary log of the entire batch execution.

**License** Apache License (>= 2)

**URL** <https://novonordisk-opensource.github.io/whirl/>,  
<https://github.com/novonordisk-opensource/whirl>

**BugReports** <https://github.com/NovoNordisk-OpenSource/whirl/issues>

**Depends** R (>= 4.1)

**Imports** callr, cli, dplyr, jsonlite, kableExtra, knitr, quarto, R6 (>= 2.4.0), reticulate, rlang, sessioninfo, stringr, tibble, tidyr, unglue, utils, withr, yaml, zephyr (>= 0.1.1)

**Suggests** ggplot2, renv, rstudioapi, testthat (>= 3.0.0), usethis

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**SystemRequirements** Quarto command line tool  
(<<https://github.com/quarto-dev/quarto-cli>>).

**NeedsCompilation** no

**Author** Aksel Thomsen [aut, cre],  
Lovemore Gakava [aut],  
Cervan Girard [aut],  
Kristian Troejelsgaard [aut],  
Steffen Falgreen Larsen [aut],  
Vladimir Obucina [aut],  
Novo Nordisk A/S [cph]

**Maintainer** Aksel Thomsen <oath@novonordisk.com>

**Repository** CRAN

**Date/Publication** 2025-04-15 20:10:01 UTC

## Contents

custom_logging . . . . .	2
run . . . . .	3
use_whirl . . . . .	5
whirl-options . . . . .	5
<b>Index</b>	<b>8</b>

---

custom_logging	<i>Helper function to log custom messages</i>
----------------	---

---

## Description

Useful for e.g. read and write operations on databases etc. that are not automatically captured.

## Usage

```
log_read(file, log = Sys.getenv("WHIRL_LOG_MSG"))
log_write(file, log = Sys.getenv("WHIRL_LOG_MSG"))
log_delete(file, log = Sys.getenv("WHIRL_LOG_MSG"))
```

## Arguments

file	<code>character()</code> description of the file that was read, written or deleted.
log	<code>character()</code> path to the log file.

## Details

The default environment variable WHIRL\_LOG\_MSG is set in the session used to log scripts, and input is automatically captured in the resulting log.

If run outside of whirl, meaning when the above environment variable is unset, the operations are streamed to `stdout()`. By default the console.

## Examples

```
# Stream logs to console since `WHIRL_LOG_MSG` is not set:
log_read("my/folder/input.txt")
log_write("my/folder/output.txt")
log_delete("my/folder/old_output.txt")
```

---

run *Execute single or multiple R, R Markdown, and Quarto scripts*

---

### Description

Executes and logs the execution of the scripts. Logs for each script are stored in the same folder as the script.

The way the execution is logged is configurable through several options for e.g. the verbosity of the logs. See [whirl-options](#) on how to configure these.

### Usage

```
run(
  input = "_whirl.yml",
  steps = NULL,
  summary_file = "summary.html",
  n_workers = zephyr::get_option("n_workers", "whirl"),
  check_renv = zephyr::get_option("check_renv", "whirl"),
  verbosity_level = zephyr::get_verbosity_level("whirl"),
  track_files = zephyr::get_option("track_files", "whirl"),
  out_formats = zephyr::get_option("out_formats", "whirl"),
  log_dir = zephyr::get_option("log_dir", "whirl")
)
```

### Arguments

input	A character vector of file path(s) to R, R Markdown, Quarto scripts, or files in a folder using regular expression, or to to a whirl config file. The input can also be structured in a list where each element will be executed sequentially, while scripts within each element can be executed in parallel.
steps	An optional argument that can be used if only certain steps within a config files (or list) is to be executed. Should be equivalent to the names of the steps found in the config file. If kept as NULL (default) then all steps listed in the config file will be executed.
summary_file	A character string specifying the file path where the summary log will be stored.
n_workers	Number of simultaneous workers used in the run function. A maximum of 128 workers is allowed.. Default: 1.
check_renv	Should the projects renv status be checked?. Default: FALSE.
verbosity_level	Verbosity level for functions in whirl. See <a href="#">zephyr::verbosity_level</a> for details.. Default: NA_character_.
track_files	Should files read and written be tracked? Currently only supported on Linux.. Default: FALSE.
out_formats	Which log format(s) to produce. Possibilities are html, json, and markdown formats: gfm, commonmark, and markua.. Default: "html".

`log_dir` The output directory of the log files. Default is the folder of the executed script. `log_dir` can be a path as a character or it can be a function that takes the script path as input and returns the log directory. For more information see the examples of `run()` or `vignette('whirl')`. Default: `function(x) dirname(x)`.

## Value

A tibble containing the execution results for all the scripts.

## Examples

```
# Copy example scripts:
file.copy(
  from = system.file("examples", c("success.R", "warning.R", "error.R")),
  package = "whirl"
),
to = tempdir()
)

# Run a single script and create log:
run(file.path(tempdir(), "success.R"))

# Run several scripts in parallel on up to 2 workers:
run(
  input = file.path(tempdir(), c("success.R", "warning.R", "error.R")),
  n_workers = 2
)

# Run several scripts in two steps by providing them as list elements:
run(
  list(
    file.path(tempdir(), c("success.R", "warning.R")),
    file.path(tempdir(), "error.R")
  )
)

# Re-directing the logs to a sub-folder by utilizing the log_dir argument in
# run(). This will require that the sub-folder exists.

# Specifying the path using a manually defined character
run(file.path(tempdir(), "success.R"), log_dir = tempdir())

# Specifying the path with a generic function that can handle the scripts
# individually.
run(
  input = file.path(tempdir(), "success.R"),
  log_dir = function(x) {paste0(dirname(x), "/logs")}
)
```

---

use_whirl	<i>Use whirl</i>
-----------	------------------

---

### Description

Utility function to setup execution with whirl in your project:

1. Creates configuration file (default `_whirl.yaml`)
2. Updates `.gitignore` to not include log files

See `vignette("whirl")` for how to specify paths inside the configuration file.

### Usage

```
use_whirl(config_file = "_whirl.yaml")
```

### Arguments

`config_file` Path to the whirl config file, relative to the project

---

whirl-options	<i>Options for whirl</i>
---------------	--------------------------

---

### Description

#### **verbosity\_level:**

Verbosity level for functions in whirl. See [zephyr::verbosity\\_level](#) for details.

- Default: `NA_character_`
- Option: `whirl.verbosity_level`
- Environment: `R_WHIRL_VERBOSITY_LEVEL`

#### **out\_formats:**

Which log format(s) to produce. Possibilities are `html`, `json`, and markdown formats: `gfm`, `commonmark`, and `markua`.

- Default: `"html"`
- Option: `whirl.out_formats`
- Environment: `R_WHIRL_OUT_FORMATS`

#### **track\_files:**

Should files read and written be tracked? Currently only supported on Linux.

- Default: `FALSE`
- Option: `whirl.track_files`
- Environment: `R_WHIRL_TRACK_FILES`

**check\_renv:**

Should the projects renv status be checked?

- Default: FALSE
- Option: `whirl.check_renv`
- Environment: `R_WHIRL_CHECK_RENV`

**track\_files\_discards:**

List of file naming patterns not be tracked when `track_files = TRUE`

- Default: `c("^/lib", "^/etc", "^/lib64", "^/usr", "^/var", "^/opt", "^/sys", "^/proc",  
"/tmp", "/null", "/urandom", "/.cache")`
- Option: `whirl.track_files_discards`
- Environment: `R_WHIRL_TRACK_FILES_DISCARDS`

**track\_files\_keep:**

List of file naming patterns always to be tracked when `track_files = TRUE`

- Default: NULL
- Option: `whirl.track_files_keep`
- Environment: `R_WHIRL_TRACK_FILES_KEEP`

**approved\_pkgs\_folder:**

Approved folder library packages

- Default: NULL
- Option: `whirl.approved_pkgs_folder`
- Environment: `R_WHIRL_APPROVED_PKGS_FOLDER`

**approved\_pkgs\_url:**

Approved URL library packages

- Default: NULL
- Option: `whirl.approved_pkgs_url`
- Environment: `R_WHIRL_APPROVED_PKGS_URL`

**n\_workers:**

Number of simultaneous workers used in the `run` function. A maximum of 128 workers is allowed.

- Default: 1
- Option: `whirl.n_workers`
- Environment: `R_WHIRL_N_WORKERS`

**log\_dir:**

The output directory of the log files. Default is the folder of the executed script. `log_dir` can be a path as a character or it can be a function that takes the script path as input and returns the log directory. For more information see the examples of `run()` or `vignette('whirl')`.

- Default: `function(x) dirname(x)`
- Option: `whirl.log_dir`
- Environment: `R_WHIRL_LOG_DIR`

**execute\_dir:**

The working directory of the process executing each script. Default us to execute R files from the working directory when calling `run()` and all other functions from the directory of the script. To change provide a character path (used for all scripts) or a function that takes the script as input and returns the execution directory.

- Default: NULL
- Option: `whirl.execute_dir`
- Environment: `R_WHIRL_EXECUTE_DIR`

**wait\_timeout:**

Timeout for waiting for the R process from `callr::r_session` to start, in milliseconds.

- Default: 9000
- Option: `whirl.wait_timeout`
- Environment: `R_WHIRL_WAIT_TIMEOUT`

# Index

`character()`, [2](#)

`custom_logging`, [2](#)

`log_delete (custom_logging)`, [2](#)

`log_read (custom_logging)`, [2](#)

`log_write (custom_logging)`, [2](#)

`run`, [3](#)

`use_whirl`, [5](#)

`whirl-options`, [3](#), [5](#)

`zephyr::verbosity_level`, [3](#), [5](#)