

# Package ‘watcher’

December 2, 2025

**Type** Package

**Title** Watch the File System for Changes

**Version** 0.1.5

**Description** R binding for 'libfswatch', a file system monitoring library.  
Watch files, or directories recursively, for changes in the  
background. Log activity, or call an R function, upon every change  
event.

**License** MIT + file LICENSE

**URL** <https://watcher.r-lib.org>, <https://github.com/r-lib/watcher>

**BugReports** <https://github.com/r-lib/watcher/issues>

**Depends** R (>= 3.5)

**Imports** later, R6, rlang

**Suggests** testthat (>= 3.0.0)

**Biarch** true

**Config/Needs/website** tidyverse/tidytemplate

**Config/testthat/edition** 3

**Config/usethis/last-upkeep** 2025-04-23

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**SystemRequirements** 'libfswatch', or 'cmake' to compile from package  
sources

**NeedsCompilation** yes

**Author** Charlie Gao [aut, cre] (ORCID: <<https://orcid.org/0000-0002-0750-061X>>),  
Posit Software, PBC [cph, fnd] (ROR: <<https://ror.org/03wc8by49>>)

**Maintainer** Charlie Gao <[charlie.gao@posit.co](mailto:charlie.gao@posit.co)>

**Repository** CRAN

**Date/Publication** 2025-12-01 23:30:02 UTC

## Contents

watcher . . . . .	2
<b>Index</b>	<b>4</b>

---

watcher	<i>Watch a Filesystem Location</i>
---------	------------------------------------

---

### Description

Create a 'Watcher' on a filesystem location to monitor for changes in the background.

### Usage

```
watcher(path = getwd(), callback = NULL, latency = 1)
```

### Arguments

path	Character path to a file, or directory to watch recursively, or a vector of paths. Defaults to the current working directory.
callback	A function or formula (see <a href="#">rlang::as_function</a> ), which takes at least one argument. It will be called back with a character vector comprising the paths of all files that have changed. The default, NULL, causes the paths that have changed to be written to stdout instead.
latency	Numeric latency in seconds for events to be reported or callbacks triggered. The default is 1s.

### Details

Uses an optimal event-driven API for each platform: 'ReadDirectoryChangesW' on Windows, 'FSEvents' on MacOS, 'inotify' on Linux, 'kqueue' on BSD, and 'File Events Notification' on Solaris/Illumos.

Note: the latency setting controls how often the changes are processed, and does not mean that changes are polled for at this interval. The changes are monitored in an event-driven fashion by the platform-specific monitor. Events are 'bubbled' such that a single change that triggers multiple filesystem events will cause the callback to be called only once.

It is possible to set a watch on a path that does not currently exist, and it will be monitored once created.

### Value

A 'Watcher' R6 class object.

**Watcher Methods**

A Watcher is an R6 class with the following methods:

- `$start()` starts background monitoring. Returns logical TRUE upon success, FALSE otherwise.
- `$stop()` stops background monitoring. Returns logical TRUE upon success, FALSE otherwise.
- `$get_path()` returns the watched path as a character string.
- `$is_running()` returns logical TRUE or FALSE depending on whether the monitor is running.

**Examples**

```
w <- watcher(tempdir())
w$start()
w
w$get_path()
w$stop()
w$is_running()

Sys.sleep(1)
```

# Index

`rlang::as_function`, [2](#)

`watcher`, [2](#)