

Package ‘watcher’

February 10, 2025

Type Package

Title Watch the File System for Changes

Version 0.1.1

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

License MIT + file LICENSE

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

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

Encoding UTF-8

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

Depends R (>= 3.6)

Imports later, R6, rlang

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/Needs/website tidyverse/tidytemplate

NeedsCompilation yes

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

Maintainer Charlie Gao <charlie.gao@shikokuchuo.net>

Repository CRAN

Date/Publication 2025-02-10 08:10:02 UTC

Contents

watcher	2
Index	4

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. Defaults to the current working directory.
callback	A function or formula (see rlang::as_function), 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 the 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 does not mean that changes are polled for at this interval, these still rely on the optimal platform-specific monitor. The implementation of 'latency' is also platform-dependent.

Events are 'bubbled' such that a single change that triggers multiple event flags 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. Start and stop background monitoring using the `$start()` and `$stop()` methods - these return a logical value whether or not they have succeeded.

Examples

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

watcher

3

w

sys.sleep(1)

Index

`rlang::as_function`, 2

`watcher`, 2