

# Package ‘whereami’

July 22, 2025

**Type** Package

**Title** Reliably Return the Source and Call Location of a Command

**Description** Robust and reliable functions to return informative outputs to console with the run or source location of a command. This can be from the 'RScript'/R terminal commands or 'RStudio' console, source editor, 'Rmarkdown' document and a Shiny application.

**Version** 0.2.0

**License** MIT + file LICENSE

**Depends** R (>= 3.2.0)

**Imports** utils, graphics, grDevices, stats, cli, rstudioapi, jsonlite

**URL** <https://github.com/yoniced/whereami>

**BugReports** <https://github.com/yoniced/whereami/issues>

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**Suggests** magrittr, ggplot2, kableExtra, testthat, covr, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Jonathan Sidi [aut, cre] (ORCID: <https://orcid.org/0000-0002-4222-1819>),  
Colin Fay [ctb] (ORCID: <https://orcid.org/0000-0001-7343-1846>),  
Kirill Müller [aut] (ORCID: <https://orcid.org/0000-0002-1416-3412>)

**Maintainer** Jonathan Sidi <[yoniced@gmail.com](mailto:yoniced@gmail.com)>

**Repository** CRAN

**Date/Publication** 2022-12-22 20:40:02 UTC

## Contents

cat_where . . . . .	2
counter_state . . . . .	3
set_whereami_log . . . . .	4
thisfile . . . . .	5
whereami . . . . .	6
<b>Index</b>	<b>8</b>

---

cat_where	<i>cat function for whereami</i>
-----------	----------------------------------

---

### Description

cat function for whereami output

### Usage

```
cat_where(
  where,
  type = c("rule", "boxx", "bullet", "line", "print"),
  color = c("none", "black", "blue", "cyan", "green", "magenta", "red", "white",
    "yellow", "grey", "silver"),
  ...
)
```

### Arguments

where	whereami output
type	cli cat_* method. Can be boxx, bullet, line, print, rule. Default is rule.
color	cli col_* method.
...	further arguments passed to cli::cat_*.

### Value

character

### Author(s)

Colin Fay

### Examples

```
cat_where(whereami())
cat_where(whereami(), type = "bullet", bullet = "tick")
cat_where(whereami(), color = "green")
```

---

counter_state	<i>whereami counter</i>
---------------	-------------------------

---

## Description

Interact with internal whereami counter

## Usage

```
counter_state(item = NULL, tag = NULL)
```

```
counter_reset(item = NULL, tag = NULL)
```

```
counter_names()
```

```
counter_tags()
```

```
counter_get()
```

## Arguments

`item` character, name of the counter to access, Default: NULL

`tag` character, tag of the counter to acces, if it was set at initial whereami call. This can be used instead of the item name, Default: NULL

## Details

`counter_names()` returns names of the active stored counters.

`counter_state()` returns current hit count for item, if NULL then all counters are returned.

`counter_reset()` will remove counter of item, if item is NULL then all counters are reset.

`counter_get()` returns a table with all the counters. These can then be plotted with the `plot` method.

A json log of the counter is written to `file.path(tempdir(), 'whereami.json')` by default The path can be set using [set\\_whereami\\_log](#)

## Author(s)

Jonathan Sidi

## See Also

Other counter: [set\\_whereami\\_log\(\)](#)

**Examples**

```
if(interactive()){  
  
  txt <- "  
  whereami::cat_where(whereami::whereami(tag = 'tag1'))  
  "  
  
  tf <- tempfile(fileext = '.R')  
  
  cat(txt,file = tf)  
  
  source(tf)  
  
  counter_state()  
  
  counter_state(tag = 'tag1')  
  
  counter_names()  
  
  counter_tags()  
  
  counters <- counter_get()  
  
  counters  
  
  plot(counter_get())  
  
  # read the json log  
  jsonlite::read_json(  
  file.path(tempdir(),'whereami.json'),  
  simplifyVector = TRUE)  
  
  # clear all counters  
  counter_reset()  
  
  #verify that there are no active counters  
  counter_state()  
  
  # cleanup  
  unlink(tf)  
  
}
```

---

set\_whereami\_log

*Set Log Path*

---

**Description**

Set the path to write the counter log as a json

**Usage**

```
set_whereami_log(path = tempdir())
```

**Arguments**

path                    character, path to write to, Default: tempdir()

**See Also**

Other counter: [counter\\_state\(\)](#)

---

thisfile

*Determines the path of the currently running script in terminal*

---

**Description**

R does not store nor export the path of the currently running script. This is an attempt to circumvent this limitation by applying heuristics (such as call stack and argument inspection) that work in many cases. **CAVEAT:** Use this function only if your workflow does not permit other solution: if a script needs to know its location, it should be set outside the context of the script if possible.

**Usage**

```
thisfile()
```

```
thisfile_source()
```

```
thisfile_r()
```

```
thisfile_rscript()
```

```
thisfile_knit()
```

**Details**

This functions currently work only if the script was sourced, processed with knitr, or run with Rscript or using the `--file` parameter to the R executable. For code run with Rscript, the exact value of the parameter passed to Rscript is returned.

**Value**

The path of the currently running script, NULL if it cannot be determined.

**Note**

These functions were migrated from [r-lib/rprojroot](#).

**Author(s)**

Kirill Müller, Hadley Wickham, Michael R. Head

**References**

<https://stackoverflow.com/q/1815606/946850>

**See Also**

`base::source()`, `utils::Rscript()`, `base::getwd()`

**Examples**

```
if( !interactive() )
  thisfile()

# using in terminal ( with pipe() )

# open a temp file
tf <- tempfile(fileext = '.R')

# call to write to temp file
f <- 'cat(whereami::thisfile(), "\n", sep = "")'

# write to the file
cat(f,file = tf)

# create an R call to terminal
fcmd <- sprintf('%s' --slave --vanilla --no-save -f %s',R.home('bin/R'),tf)

# run the call
p <- pipe(fcmd)

# read the output
readLines(p)

# cleanup
close(p)
unlink(tf)
```

---

whereami

*Return where the script is run from*

---

**Description**

Returns the file that a command is run from.

If traceback is available then the line that it was run from is also returned.

**Usage**

```
whereami(path_expand = FALSE, tag = NULL)
```

**Arguments**

path_expand	logical, expand relational path, Default: FALSE
tag	character, optional tag for the call, Default: NULL

**Value**

character

**Author(s)**

Jonathan Sidi

**Examples**

```
whereami()  
whereami(path_expand = TRUE)
```

# Index

- \* **counter**
  - counter\_state, 3
  - set\_whereami\_log, 4
- \* **main**
  - whereami, 6
- \* **print**
  - cat\_where, 2
- \* **this**
  - thisfile, 5

base::getwd(), 6  
base::source(), 6

cat\_where, 2  
counter\_get(counter\_state), 3  
counter\_names(counter\_state), 3  
counter\_reset(counter\_state), 3  
counter\_state, 3, 5  
counter\_tags(counter\_state), 3

set\_whereami\_log, 3, 4

thisfile, 5  
thisfile\_knit(thisfile), 5  
thisfile\_r(thisfile), 5  
thisfile\_rscript(thisfile), 5  
thisfile\_source(thisfile), 5

utils::Rscript(), 6

whereami, 6