

# Package ‘matahari’

February 6, 2020

**Type** Package

**Title** Spy on Your R Session

**Version** 0.1.3

**Description** Conveniently log everything you type into the R console. Logs are stored as tidy data frames which can then be analyzed using 'tidyverse' style tools.

**Depends** R (>= 3.5.0)

**Imports** rstudioapi, tibble, readr, jsonlite, clipr, purrr, rlang

**Suggests** testthat, knitr, rmarkdown

**License** MIT + file LICENSE

**URL** <https://github.com/jhudsl/matahari>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**BugReports** <https://github.com/jhudsl/matahari/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Sean Kross [aut, cre] (<<https://orcid.org/0000-0001-5215-0316>>),  
Lucy D'Agostino McGowan [aut] (<<https://orcid.org/0000-0001-7297-9359>>),  
Jeff Leek [ldr] (<<https://orcid.org/0000-0002-2873-2671>>)

**Maintainer** Sean Kross <[sean@seankross.com](mailto:sean@seankross.com)>

**Repository** CRAN

**Date/Publication** 2020-02-06 10:30:02 UTC

## R topics documented:

dance_recital	2
dance_remove	3
dance_report	4

dance_save	5
dance_start	5
dance_stop	6
dance_tbl	7

<b>Index</b>	<b>9</b>
--------------	----------

---

dance_recital	<i>Create a matahari-esque data frame from a file.</i>
---------------	--

---

## Description

Create a matahari-esque data frame from a file.

## Usage

```
dance_recital(code, evaluate = TRUE)
```

## Arguments

code	A string or the path to a file containing R code.
evaluate	Logical, indicating whether to evaluate the code, default is TRUE

## Examples

```
library(knitr)

# Evaluate a string of R code
kable(dance_recital("x <- 4; x *7"))

## |expr   |value |error |output |warnings   |messages   |
## |:-----|:-----|:-----|:-----|:-----|:-----|
## |x <- 4 |4     |NULL  |      |character(0)|character(0)|
## |x * 7  |28    |NULL  |      |character(0)|character(0)|

# Evaluate an R script. We have provided an R script for testing purposes.
code_file <- system.file("test", "sample_code.R", package = "matahari")
kable(dance_recital(code_file)[,1:3])

## |expr           |value |error
## |:-----|:-----|:-----|
## |4 + 4          |8     |NULL
## |wow!           |wow!  |NULL
## |mean(1:10)     |5.5   |NULL
## |stop("Error!") |NULL  |list(message = "Error!", call = .f(...))
## |warning("Warning!") |Warning! |NULL
## |message("Hello?") |NULL  |NULL
## |cat("Welcome!")  |NULL  |NULL
```

---

dance_remove	<i>Remove your logging history.</i>
--------------	-------------------------------------

---

**Description**

Remove your logging history.

**Usage**

```
dance_remove()
```

**Value**

Either TRUE if the log was removed or FALSE if the log does not exist (invisibly).

**Examples**

```
## Not run:
library(knitr)

# Start recording
dance_start(value = TRUE)

# Execute some expressions
"Hello!"
4 + 4

# Stop recording
dance_stop()

# Access log of your session
kable(dance_tbl()[3:4, 1:5])

## |expr   |value |path |contents |selection |
## |:-----|:-----|:----|:-----|:-----|
## |Hello! |Hello! |NA   |NA       |NA        |
## |4 + 4  |8      |NA   |NA       |NA        |

# Deletes the lgo of your session
dance_remove()

# With no log, dance_tbl() returns NULL invisibly.
withVisible(dance_tbl())

## $value
## NULL
##
## $visible
## FALSE
```

```
## End(Not run)
```

---

```
dance_report          Copy the log to your clipboard.
```

---

### Description

Copy the log to your clipboard.

### Usage

```
dance_report(...)
```

### Arguments

```
...          Developer options.
```

### Examples

```
## Not run:  
  
# Start recording  
dance_start(value = TRUE)  
  
# Execute some expressions  
"Hello!"  
4 + 4  
  
# Stop recording  
dance_stop()  
  
# Assign a base64 encoded tibble of your log to a variable  
report_code <- dance_report()  
substr(report_code, 1, 10)  
  
## "WAoAAAADAA"  
  
nchar(report_code)  
  
## 397432  
  
## End(Not run)
```

---

dance_save	<i>Save the log as an rds file.</i>
------------	-------------------------------------

---

**Description**

Save the log as an rds file.

**Usage**

```
dance_save(path)
```

**Arguments**

path	The path to the rds file.
------	---------------------------

**Examples**

```
## Not run:  
  
# Start recording  
dance_start(value = TRUE)  
  
# Execute some expressions  
"Hello!"  
4 + 4  
  
# Stop recording  
dance_stop()  
  
# Save your log locally  
dance_save("session.rds")  
  
## End(Not run)
```

---

dance_start	<i>Start Spying</i>
-------------	---------------------

---

**Description**

Start logging your R console activity. If you're using RStudio the contents of your current editor tab will also be tracked. All logging takes place when R code is executed in the R console.

**Usage**

```
dance_start(  
  expr = TRUE,  
  value = FALSE,  
  path = FALSE,  
  contents = FALSE,  
  selection = FALSE  
)
```

**Arguments**

expr	The R expressions typed into the R console will be logged unless this is set to FALSE.
value	Values that are computed on the R console will be logged unless this is set to FALSE.
path	The path to the file in focus on the RStudio editor will be logged unless this is set to FALSE.
contents	The file contents of the RStudio editor tab in focus will be logged unless this is set to FALSE.
selection	The text that is highlighted in the RStudio editor tab in focus will be logged unless this is set to FALSE.

**Examples**

```
## Not run:  
# Begin recording your R session.  
dance_start()  
  
# Each of the following expressions adds a row to the tibble that tracks  
# the execution of your R code.  
"Hello!"  
4 + 4  
x <- 7  
x^2  
rm(x)  
x  
  
# This stops recording your R session.  
dance_stop()  
  
## End(Not run)
```

---

dance\_stop

*Stop Spying*

---

**Description**

Pause the current logging session.

**Usage**

```
dance_stop()
```

**Value**

TRUE if logging was taking place, otherwise FALSE (invisibly).

**Examples**

```
## Not run:  
# Begin recording your R session.  
dance_start()  
  
# Each expression adds a row to the tibble that tracks the execution of  
# your R code.  
x <- 7  
x * 4  
  
# This stops recording your R session.  
dance_stop()  
  
## End(Not run)
```

---

dance_tbl	<i>Get the log as a tibble</i>
-----------	--------------------------------

---

**Description**

Get the log as a tibble

**Usage**

```
dance_tbl()
```

**Value**

Either a `tibble::tibble()` containing your logged history or NULL, if there is no log.

**Examples**

```
## Not run:  
  
library(knitr)  
  
# Start recording  
dance_start(value = TRUE)  
  
# Execute some expressions  
"Hello!"
```

```
4 + 4

# Stop recording
dance_stop()

# Access log of your session
dance_tbl()

# Display the log nicely
kable(dance_tbl()[3:4, 1:5])

## |expr   |value |path |contents |selection |
## |:-----|:-----|:----|:-----|:-----|
## |Hello! |Hello! |NA   |NA       |NA       |
## |4 + 4  |8      |NA   |NA       |NA       |

## End(Not run)
```



# Index

dance\_recital, [2](#)  
dance\_remove, [3](#)  
dance\_report, [4](#)  
dance\_save, [5](#)  
dance\_start, [5](#)  
dance\_stop, [6](#)  
dance\_tbl, [7](#)  
  
tibble::tibble(), [7](#)