

# Package ‘rlog’

July 23, 2025

**Title** A Simple, Opinionated Logging Utility

**Version** 0.1.0

**Description** A very lightweight package that writes out log messages in an opinionated way. Simpler and lighter than other logging packages, 'rlog' provides a compact feature set that focuses on getting the job done in a Unix-like way.

**URL** <https://github.com/sellorm/rlog>

**BugReports** <https://github.com/sellorm/rlog/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Mark Sellors [aut, cre]

**Maintainer** Mark Sellors <rstats@5vcc.com>

**Repository** CRAN

**Date/Publication** 2021-02-24 09:20:05 UTC

## Contents

|                     |   |
|---------------------|---|
| log_debug . . . . . | 2 |
| log_error . . . . . | 2 |
| log_fatal . . . . . | 3 |
| log_info . . . . .  | 4 |
| log_trace . . . . . | 4 |
| log_warn . . . . .  | 5 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>6</b> |
|--------------|----------|

---

|           |                            |
|-----------|----------------------------|
| log_debug | <i>Log a debug message</i> |
|-----------|----------------------------|

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_debug(message)
```

**Arguments**

|         |                     |
|---------|---------------------|
| message | your message to log |
|---------|---------------------|

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:  
log_debug("This is a debug message")  
Sys.setenv("LOG_LEVEL" = "TRACE")  
log_debug("This is a debug message")  
  
## End(Not run)
```

---

|           |                             |
|-----------|-----------------------------|
| log_error | <i>Log an error message</i> |
|-----------|-----------------------------|

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_error(message)
```

**Arguments**

|         |                     |
|---------|---------------------|
| message | your message to log |
|---------|---------------------|

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:  
log_error("This is an error message")  
Sys.setenv("LOG_LEVEL" = "TRACE")  
log_error("This is an error message")  
  
## End(Not run)
```

---

log\_fatal

*Log a fatal message*

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_fatal(message)
```

**Arguments**

message            your message to log

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:  
log_fatal("This is a fatal message")  
Sys.setenv("LOG_LEVEL" = "TRACE")  
log_fatal("This is a fatal message")  
  
## End(Not run)
```

---

`log_info`*Log an info message*

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_info(message)
```

**Arguments**

message            your message to log

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:
log_info("This is an info message")
Sys.setenv("LOG_LEVEL" = "TRACE")
log_info("This is an info message")

## End(Not run)
```

---

`log_trace`*Log a trace message*

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_trace(message)
```

**Arguments**

message            your message to log

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:  
log_trace("This is a trace message")  
Sys.setenv("LOG_LEVEL" = "TRACE")  
log_trace("This is a trace message")  
  
## End(Not run)
```

---

log\_warn

*Log a warning message*

---

**Description**

Log messages will only be emitted if the log priority matches or is higher than the priority of your message

**Usage**

```
log_warn(message)
```

**Arguments**

message            your message to log

**Value**

invisibly returns TRUE/FALSE

**Examples**

```
## Not run:  
log_warn("This is a warning message")  
Sys.setenv("LOG_LEVEL" = "TRACE")  
log_warn("This is a warning message")  
  
## End(Not run)
```

# Index

log\_debug, [2](#)  
log\_error, [2](#)  
log\_fatal, [3](#)  
log\_info, [4](#)  
log\_trace, [4](#)  
log\_warn, [5](#)