

# Package ‘tracer’

May 8, 2026

**Title** Slick Call Stacks

**Version** 1.0.0

**Description** Better looking call stacks after an error.

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/mangothecat/tracer#readme>

**BugReports** <https://github.com/mangothecat/tracer/issues>

**Imports** clisymbols, crayon, prettycode, utils

**Suggests** covr, mockery, testthat

**RoxygenNote** 5.0.1.9000

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Gabor Csardi [aut, cre]

**Maintainer** Gabor Csardi <[csardi.gabor@gmail.com](mailto:csardi.gabor@gmail.com)>

**Repository** CRAN

**Date/Publication** 2017-01-28 17:31:38

## Contents

tb .....	2
<b>Index</b>	<b>3</b>

---

tb *A better* `base::traceback()`

---

### Description

Compared to `base::traceback()`, `tb`

- shows the stack from top to bottom, the same way as `utils::recover()`,
- shows references to source files in a clickable form (in terminals that support this),
- cuts long lines of code,
- colors the output nicely (although this is a matter of taste),
- can show the source code where the error happened, with syntax highlighting, if the source code is available.

### Usage

```
tb(frame = NULL, context = 5)
```

```
bt(frame = NULL, context = 5)
```

### Arguments

frame	If NULL, the stack trace is shown. If it is an integer referring to a frame in the stack, the source code of the corresponding call is shown.
context	Number of source code lines to show before and after the current line.

### Details

`bt` is an alias to `tb`.

### The custom error handler

`tracer` defines an error handler when loaded, via a call to `options(error = tracer:::dumper)`. If you overwrite the error handler, then `tb` does not work properly. You can reinstate the `tracer:::dumper` error handler via an explicit call to `options()`.

# Index

`bt (tb)`, [2](#)

`tb`, [2](#)

`utils::recover()`, [2](#)