

Package ‘tl’

June 30, 2026

Type Package

Title Tiny Logging Interface to 'rspdlite' Wrapping 'spdlog' C++20
Logging

Version 0.0.2

Date 2026-06-30

Description Just how 'spdlog' provides a nice and consistent interface to 'spdlog' (via 'RcppSpdlog'), this package does so for 'spdlog', the lightweight header-only C++20 logging library that provides a lighter version of 'spdlog'. This package is essentially a thin shim around it for a more compact interface from both R and C++.

URL <https://github.com/eddelbuettel/tl>

BugReports <https://github.com/eddelbuettel/tl/issues>

License GPL (>= 2)

Imports rspdlite

Suggests tinytest

NeedsCompilation no

Author Dirk Eddelbuettel [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6419-907X>>)

Maintainer Dirk Eddelbuettel <edd@debian.org>

Repository CRAN

Date/Publication 2026-06-30 16:00:02 UTC

Contents

tl-package	2
trace	2
Index	5

tl-package	<i>Tiny Logging Interface to 'rspdlite' Wrapping 'spdlog' C++20 Logging</i>
------------	---

Description

Just how 'spdlog' provides a nice and consistent interface to 'spdlog' (via 'RcppSpdlog'), this package does so for 'spdlog', the lightweight header-only C++-20 logging library that provides a lighter version of 'spdlog'. This package is essentially a thin shim around it for a more compact interface from both R and C++.

Package Content

Index of help topics:

tl-package	Tiny Logging Interface to 'rspdlite' Wrapping 'spdlog' C++20 Logging
trace	Tiny logging wrapper for 'rspdlite'

Maintainer

Dirk Eddelbuettel <edd@debian.org>

Author(s)

Dirk Eddelbuettel [aut, cre] (ORCID: <<https://orcid.org/0000-0001-6419-907X>>)

trace	<i>Tiny logging wrapper for 'rspdlite'</i>
-------	--

Description

Logging support from R and C++ routines based on **rspdlite**

Usage

trace(...)

debug(...)

info(...)

warn(...)

error(...)

critical(...)

```

set_level(...)

get_level()

set_name(...)

get_name()

set_format(utc = FALSE, show_date = TRUE, show_thread_id = FALSE,
           precision = "ms")

init()

```

Arguments

...	Argument(s) passed along
utc	Boolean flag to select display of current time in UTC rather than local, default is off
show_date	Boolean flag to select display of date part of current, default is on
show_thread_id	Boolean flag to select display of current thread, default is off
precision	Character value for selected time precision: one of "ms" (the default format), "us", "ns" or "none"

Details

These functions all pass on their arguments to the corresponding function in the **rspdlite** package implementing them. The core purpose of these functions is to provide a 'tighter' interface via the `tl::` prefix from both R and C++, i.e. `tl::debug("Condition met, value {}")` works from both. See the **rspdlite** package for more.

The `init()` function passes a logging level from the environment variable `TL_LEVEL`, or a global option `getOption("tl.level")`, and sets it. Otherwise level 'info' remains the default.

Value

In general, nothing is returned as the functions are invoked for their side effect of logging.

See Also

rspdlite

Examples

```

lvl <- tl::get_level()
tl::debug("This message is ignored by the default level 'info'.")
tl::info("This message is show by the default level.")
tl::set_level("warn")
tl::info("Now this message at 'info' is ignored too.")
tl::warn("A warning messages passes at level warning. {}", 42L)

```

```
tl::set_name("my_logger")
tl::error("Error messages also pass, and see the name set")
tl::set_format(show_thread_id=TRUE, precision="ns")
tl::error("Warning message under changed formatting")
tl::set_level(lvl) # revert to prior level
tl::set_name("") # revert to no name
tl::set_format() # revert to default format
```

Index

* package

- tl-package, [2](#)
- critical (trace), [2](#)
- debug (trace), [2](#)
- error (trace), [2](#)
- get_level (trace), [2](#)
- get_name (trace), [2](#)
- info (trace), [2](#)
- init (trace), [2](#)
- set_format (trace), [2](#)
- set_level (trace), [2](#)
- set_name (trace), [2](#)
- tl (tl-package), [2](#)
- tl-package, [2](#)
- trace, [2](#)
- warn (trace), [2](#)