

Package ‘rstudiothemes’

June 19, 2026

Title Create and Install Custom 'RStudio' Themes

Version 1.1.2

Description Create, convert and install custom 'RStudio' editor themes from 'Visual Studio Code', 'Positron' and 'TextMate' theme files. Convert themes between 'TextMate', 'Visual Studio Code' and 'Positron' formats and install bundled ports of popular themes for use in 'RStudio'.

License MIT + file LICENSE

URL <https://dieghernan.github.io/rstudiothemes/>,
<https://github.com/dieghernan/rstudiothemes>

BugReports <https://github.com/dieghernan/rstudiothemes/issues>

Depends R (>= 4.1.0)

Imports cli, colorspace, dplyr, grDevices, jsonlite, rstudioapi, sass, tidy, tools, utils, xml2

Suggests knitr, quarto, testthat (>= 3.0.0), uuid

VignetteBuilder quarto

Config/Needs/coverage covr

Config/Needs/website dieghernan/gitdevr, ragg, styler, gadenbuie/rstthemes

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

X-schema.org-keywords r-package, rstudio-themes, textmate-themes, vscode-themes, positron-themes, cran, cran-r

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-8457-4658>>),
Garrick Aden-Buie [cph] (for the rstthemes::try_rstthemes() function)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2026-06-19 19:20:02 UTC

Contents

convert_tm_to_vs_theme	2
convert_to_rstudio_theme	3
convert_vs_to_tm_theme	5
generate_uuid	6
on_rstudio	7
read_tm_theme	8
read_vs_theme	9
rstudiothemes-actions	9

Index **12**

convert_tm_to_vs_theme

Convert a TextMate theme file to Visual Studio Code or Positron

Description

Convert a .tmTheme file representing a TextMate theme and write the equivalent Visual Studio Code theme file (.json).

convert_tm_to_positron_theme() is an alias of convert_tm_to_vs_theme().

Usage

```
convert_tm_to_vs_theme(
  path,
  outfile = tempfile(fileext = ".json"),
  name = NULL,
  author = NULL
)
```

```
convert_tm_to_positron_theme(
  path,
  outfile = tempfile(fileext = ".json"),
  name = NULL,
  author = NULL
)
```

Arguments

path	Path or URL to a TextMate theme file, in .tmTheme format.
outfile	Path where the resulting file will be written. Defaults to a temporary file created with <code>tempfile()</code> .
name	Theme name. If NULL, the name from the input file is used.
author	Theme author. If NULL, it attempts to extract the author from the input file, otherwise it defaults to "rstudiothemes R package".

Value

This function is called for its side effects. It writes a .json theme file to outfile and returns the path.

See Also

Theme converters: [convert_to_rstudio_theme\(\)](#), [convert_vs_to_tm_theme\(\)](#)

Examples

```
tmtheme <- system.file("ext/test.tmTheme",
  package = "rstudiothemes"
)
path <- convert_tm_to_vs_theme(tmtheme)

readLines(path) |>
  head(50) |>
  cat(sep = "\n")
```

convert_to_rstudio_theme

Convert a theme file to RStudio

Description

Convert a .tmTheme or .json file that defines a TextMate or Visual Studio Code theme and write the equivalent RStudio .rstheme file.

Optionally, the generated theme can be installed and applied to the RStudio IDE.

Important: This function only works in RStudio. It returns NULL when called from other IDEs.

Usage

```

convert_to_rstudio_theme(
  path,
  outfile = tempfile(fileext = ".rstheme"),
  name = NULL,
  use_italics = TRUE,
  output_style = "expanded",
  force = FALSE,
  apply = FALSE
)

```

Arguments

path	Path or URL to a TextMate theme file (.tmTheme format) or a Visual Studio Code theme file (.json format).
outfile	Path where the resulting file will be written. Defaults to a temporary file created with <code>tempfile()</code> .
name	Theme name. If NULL, the name from the input file is used.
use_italics	Logical. Use italics in the resulting theme. The default is TRUE, although some themes may look better without italics.
output_style	Bracketing and formatting style of the CSS output. Possible styles: "nested", "expanded", "compact", and "compressed".
force	Whether to force the operation and overwrite an existing file with the same name. Default: FALSE.
apply	Logical. Apply the theme with <code>rstudioapi::applyTheme()</code> .

Details

RStudio supports custom editor themes in two formats, .tmTheme and .rstheme. The .tmTheme format originated with TextMate and has become a common theme format. **This tmTheme editor** hosts a large collection of .tmTheme files. The .rstheme format is specific to RStudio.

To switch editor themes, go to Tools > Global Options > Appearance > Add and use the Editor theme selector.

For more information, see <https://docs.posit.co/ide/user/ide/guide/ui/appearance.html>.

Value

This function is called for its side effects. It writes a .rstheme file to outfile and returns the path. If force or apply is TRUE, it installs the theme. If apply is TRUE, it also applies the theme to your RStudio IDE.

See Also

`rstudioapi::addTheme()`, `rstudioapi::applyTheme()`

Theme converters: `convert_tm_to_vs_theme()`, `convert_vs_to_tm_theme()`

Examples

```

if (on_rstudio() && interactive()) {
  vstheme <- system.file("ext/skeleton-syntax-color-theme.json",
    package = "rstudiothemes"
  )

  # Apply the theme for 10 seconds to demonstrate the effect.
  current_theme <- rstudioapi::getThemeInfo()$editor

  # Print the current theme name.
  current_theme
  new_rs_theme <- convert_to_rstudio_theme(vstheme,
    name = "A testing theme",
    apply = TRUE, force = TRUE
  )

  Sys.sleep(10)

  rstudioapi::applyTheme(current_theme)
  rstudioapi::removeTheme("A testing theme")
}

```

 convert_vs_to_tm_theme

Convert a Visual Studio Code or Positron theme file to TextMate

Description

Convert a .json file representing a Visual Studio Code or Positron theme and write the equivalent TextMate theme file (.tmTheme).

convert_positron_to_tm_theme() is an alias of convert_vs_to_tm_theme().

Usage

```

convert_vs_to_tm_theme(
  path,
  outfile = tempfile(fileext = ".tmTheme"),
  name = NULL,
  author = NULL
)

convert_positron_to_tm_theme(
  path,
  outfile = tempfile(fileext = ".tmTheme"),
  name = NULL,
  author = NULL
)

```

Arguments

path	Path or URL to a Visual Studio Code or Positron theme file, in .json format.
outfile	Path where the resulting file will be written. Defaults to a temporary file created with <code>tempfile()</code> .
name	Theme name. If NULL, the name from the input file is used.
author	Theme author. If NULL, it attempts to extract the author from the input file, otherwise it defaults to "rstudiothemes R package".

Value

This function is called for its side effects. It writes a .tmTheme file to outfile and returns the file path.

See Also

Theme converters: `convert_tm_to_vs_theme()`, `convert_to_rstudio_theme()`

Examples

```

vstheme <- system.file("ext/test-simple-color-theme.json",
  package = "rstudiothemes"
)
path <- convert_vs_to_tm_theme(vstheme)

readLines(path) |>
  head(50) |>
  cat(sep = "\n")

```

generate_uuid

Generate random UUIDs

Description

Generate version 4 pseudo-random Universally Unique Identifiers (UUIDs).

Usage

```
generate_uuid(hint = NULL)
```

Arguments

hint	Optional character string or object coercible with <code>as.character()</code> . Used as a random seed.
------	---

Details

This helper generates a **UUID** for identifying generated theme versions.

Value

A character string representing a valid UUID that can be validated with `uuid::UUIDvalidate()`.

Source

Heavily based on an unreleased version of `ids::uuid()`.

References

Davis KR, Peabody B and Leach P (2024). "Universally Unique Identifiers (UUIDs)." RFC 9562. doi:10.17487/RFC9562, <https://www.rfc-editor.org/info/rfc9562>.

See Also

Helper functions: `on_rstudio()`

Examples

```
# Random UUID.
generate_uuid()

generate_uuid()

# Persistent UUID with `hint`.
hint <- "something as seed"

generate_uuid(hint)

generate_uuid(hint)
```

`on_rstudio`*Check whether the session is running in RStudio*

Description

Detect whether the current R session is running in RStudio, which is used to decide whether themes can be applied to the IDE.

Usage

```
on_rstudio()
```

Value

TRUE if running in RStudio, FALSE otherwise.

See Also

Helper functions: `generate_uuid()`

Examples

```
on_rstudio()
```

read_tm_theme	<i>Read and parse a TextMate theme file</i>
---------------	---

Description

Read a .tmTheme XML file representing a TextMate or Sublime Text theme.

Usage

```
read_tm_theme(path)
```

Arguments

path Path or URL to a TextMate theme file, in .tmTheme format.

Value

A [tibble](#) containing the theme data.

See Also

Theme readers: [read_vs_theme\(\)](#)

Examples

```
the_theme <- system.file("ext/test-color-theme.json",
  package = "rstudiothemes"
) |>
# Convert the Visual Studio Code theme to TextMate format.
convert_vs_to_tm_theme()

# Check the converted theme.
readLines(the_theme) |>
  head(10) |>
  cat(sep = "\n")

read_tm_theme(the_theme)
```

read_vs_theme	<i>Read and parse a Visual Studio Code or Positron theme file</i>
---------------	---

Description

Read a .json file representing a Visual Studio Code or Positron theme.
read_positron_theme() is an alias of read_vs_theme().

Usage

```
read_vs_theme(path)
read_positron_theme(path)
```

Arguments

path Path or URL to a Visual Studio Code or Positron theme file, in .json format.

Value

A [tibble](#) containing the theme data.

See Also

Theme readers: [read_tm_theme\(\)](#)

Examples

```
vstheme <- system.file("ext/test-color-theme.json",
  package = "rstudiothemes"
)
read_vs_theme(vstheme)
```

rstudiothemes-actions	<i>Manage RStudio themes</i>
-----------------------	------------------------------

Description

Install, list, preview or remove the RStudio themes included in **rstudiothemes**. These functions are adapted from selected **rsthemes** functions. [MIT License](#) Copyright © rsthemes authors.

Important: These functions only work in RStudio and return NULL when called from other IDEs. The exception is `list_rstudiothemes(list_installed = FALSE)`.

Usage

```
install_rstudiothemes(
  style = c("all", "dark", "light"),
  themes = NULL,
  destdir = NULL
)

remove_rstudiothemes(style = c("all", "dark", "light"))

list_rstudiothemes(style = c("all", "dark", "light"), list_installed = TRUE)

try_rstudiothemes(style = c("all", "dark", "light"), themes = NULL, delay = 0)
```

Arguments

style	Theme group: "all", "dark" or "light".
themes	Optional character vector of theme names. If provided, only these themes are used and style is ignored.
destdir	Optional directory for .rstheme files. By default, it uses <code>rstudioapi::addTheme()</code> , but this argument allows installation to non-standard directories.
list_installed	Should the installed rstudiothemes themes be listed (default). If FALSE, the available themes in the rstudiothemes package are listed instead.
delay	Number of seconds to wait between themes. Set to 0 to be prompted to continue after each theme.

Value

`install_rstudiothemes()` and `remove_rstudiothemes()` return NULL invisibly.

`list_rstudiothemes()` returns a character vector of theme names.

`try_rstudiothemes()` has side effects: it starts a widget that allows users to try different themes. The widget can be exited by following the prompts, which restore the original theme.

Functions

- `install_rstudiothemes()`: Install RStudio themes.
- `remove_rstudiothemes()`: Remove bundled **rstudiothemes** themes from RStudio.
- `list_rstudiothemes()`: List installed or available themes.
- `try_rstudiothemes()`: Preview each bundled **rstudiothemes** RStudio theme.

Bundled themes

rstudiothemes includes RStudio themes based on the following editor themes:

- Ayu Theme by teabyii.
- Andromeda Theme by Eliver Lara.
- Catppuccin Theme by **Catppuccin**.

- Cobalt2 Theme by Wes Bos.
- CRAN Theme by dieghernan, based on the CRAN (R Project) website theme created with Pandoc.
- Dracula Theme by [Dracula](#).
- GitHub Dark and Light Themes by GitHub.
- JellyFish Theme by Pawel Borkar.
- Matcha Theme by Luca Falasco.
- Matrix Theme by UstymUkhman.
- Night Owl Dark and Light Themes (no italics) by Sarah Drasner.
- Nord Theme by Arctic Ice Studio.
- OKSolar Theme by dieghernan.
- One Dark Pro Theme by binaryify.
- Overflow Theme by dieghernan.
- Panda Theme by Panda Theme.
- Selenized Themes by dieghernan.
- Skeletor Syntax Theme by dieghernan.
- SynthWave '84 Theme by Robb Owen.
- Tokyo Night Theme by Enkia.
- Winter is Coming Theme by John Papa.

Author(s)

Garrick Aden-Buie <https://github.com/gadenbuie>

References

Aden-Buie G (2026). *rsthemes: Full Themes for RStudio v1.2+*. R package version 0.5.1, commit 48fc078f772e5e63669bc9773eabc8e9cdc7f699, <https://github.com/gadenbuie/rsthemes>.

Examples

```
list_rstudiothemes(list_installed = FALSE)
```

Index

- * **converters**
 - convert_tm_to_vs_theme, 2
 - convert_to_rstudio_theme, 3
 - convert_vs_to_tm_theme, 5
- * **helpers**
 - generate_uuid, 6
 - on_rstudio, 7
- * **readers**
 - read_tm_theme, 8
 - read_vs_theme, 9

as.character(), 6

convert_positron_to_tm_theme
(convert_vs_to_tm_theme), 5

convert_tm_to_positron_theme
(convert_tm_to_vs_theme), 2

convert_tm_to_vs_theme, 2

convert_tm_to_vs_theme(), 4, 6

convert_to_rstudio_theme, 3

convert_to_rstudio_theme(), 3, 6

convert_vs_to_tm_theme, 5

convert_vs_to_tm_theme(), 3, 4

generate_uuid, 6

generate_uuid(), 7

install_rstudiothemes
(rstudiothemes-actions), 9

list_rstudiothemes
(rstudiothemes-actions), 9

on_rstudio, 7

on_rstudio(), 7

read_positron_theme (read_vs_theme), 9

read_tm_theme, 8

read_tm_theme(), 9

read_vs_theme, 9

read_vs_theme(), 8

remove_rstudiothemes
(rstudiothemes-actions), 9

rstudioapi::addTheme(), 4, 10

rstudioapi::applyTheme(), 4

rstudiothemes-actions, 9

tempfile(), 3, 4, 6

tibble, 8, 9

try_rstudiothemes
(rstudiothemes-actions), 9

UUID, 6

uuid::UUIDvalidate(), 7