

Package: ggplot2qgis (via r-universe)

July 21, 2026

Title Export 'ggplot2' as 'QGIS' Project

Version 0.0.0.9000

Description Convert a 'ggplot2'-based map plot (e.g. 'geom_sf()') into a 'QGIS' project ('.qgs') file.

License MIT + file LICENSE

URL <https://github.com/yutannihilation/ggplot2qgis>,
<https://yutannihilation.github.io/ggplot2qgis/>

BugReports <https://github.com/yutannihilation/ggplot2qgis/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports ggplot2 (>= 4.0.0), grDevices, rlang, sf (>= 1.0.0), tools

Suggests testthat (>= 3.0.0), withr, xml2

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://yutannihilation.r-universe.dev>

Date/Publication 2026-07-21 12:12:49 UTC

RemoteUrl <https://github.com/yutannihilation/ggplot2qgis>

RemoteRef HEAD

RemoteSha 46bd3ef86798e9f749c151e3d6bcf82a302de41c

Contents

write_qgs	2
Index	5

write_qgs

*Write a ggplot2 map plot as a QGIS project***Description**

Converts a ggplot2 plot whose layers are drawn from sf objects (or, for `ggplot2::geom_point()`, `ggplot2::geom_path()`, `ggplot2::geom_line()` and `ggplot2::geom_polygon()`, from plain data frames) into a QGIS project (.qgs) file. The data of each layer is saved as a GeoPackage under `<path minus extension>_data/`, and the layer is styled after the plot's trained color scale:

Usage

```
write_qgs(
  plot,
  path,
  use_plot_crs = FALSE,
  gradient_style = c("graduated", "continuous"),
  overwrite = FALSE,
  layer_names = NULL,
  basemap = NULL
)
```

Arguments

- | | |
|----------------|---|
| plot | A ggplot object. Each layer must be backed by sf data, or be one of the supported data.frame geoms (see <i>Data frame layers</i>). |
| path | Path of the .qgs file to write. Tilde paths (e.g. ~/x.qgs) are expanded. |
| use_plot_crs | If TRUE, the project (map canvas) CRS is the plot's CRS, resolved the way <code>ggplot2::coord_sf()</code> does: its <code>crs</code> argument if specified, otherwise the CRS of the first layer that defines one. If FALSE (the default), the project CRS is EPSG:3857 (Web Mercator). Either way the layers keep the CRS of their data; QGIS reprojects them on the fly. |
| gradient_style | How a continuous fill/colour scale is rendered: <ul style="list-style-type: none"> • "graduated" (the default): a graduated renderer with 25 equal-interval classes. The gradient is slightly banded, but the legend shows the classes with their value ranges. • "continuous": the exact ggplot2 look. The color is interpolated per feature by a data-defined expression on the symbol color (<code>ramp_color(create_ramp(...), ...)</code>). Caveats: QGIS cannot display a color ramp in the legend for a data-defined color, so the legend is a single swatch without any value labels, and the gradient is only discoverable in the layer styling panel behind the data-defined override of the symbol color, not in the renderer dropdown. <p>Binned scales are unaffected: their bins are exact in a graduated renderer, so there is nothing to trade off. Requesting "continuous" for a layer with a binned scale keeps the bins, with a warning.</p> |

overwrite	If FALSE (the default), writing to a path that already exists is an error. Set to TRUE to overwrite it.
layer_names	Names for the layers, used for the GeoPackage files and in the QGIS layer tree: a character vector with one name per layer, bottom-most first. /, \, , :, *, ?, ", <, > and control characters cannot be used (the name becomes a file name). If NULL (the default), each layer is named after the first of these that applies: • the layer's own name (e.g. <code>geom_sf(name = "counties")</code>), • the variable its data came from (e.g. <code>nc</code> for <code>geom_sf(data = nc)</code>, or for <code>ggplot(nc)</code> when the variable is unambiguous), • the geom (e.g. <code>geom_sf</code>), with a numbered suffix (<code>nc_2</code>) on collision.
basemap	An XYZ tile layer to add below the vector layers, or NULL (the default) for none. Either a predefined key or an arbitrary XYZ URL template (a string containing the {z}, {x} and {y} placeholders, e.g. " https://tile.openstreetmap.org/{z}/{x}/{y}.png "). The predefined keys are: • "osm": OpenStreetMap. • "gsi_standard": the GSI standard map (https://cyberjapandata.gsi.go.jp/xyz/std/{z}/{x}/{y}.png). • "gsi_pale": the GSI pale map (https://cyberjapandata.gsi.go.jp/xyz/pale/{z}/{x}/{y}.png). XYZ tiles are in EPSG:3857; QGIS reprojects them to the project CRS on the fly.

Details

- a continuous fill/colour scale becomes a graduated renderer with fine-grained equal-interval classes (or a continuously interpolated color, see `gradient_style`),
- a binned one (e.g. `ggplot2::scale_fill_steps()`) becomes a graduated renderer with one class per bin, using the scale's exact bin boundaries and colors,
- a discrete one becomes a categorized renderer,
- a layer with no fill/colour mapping becomes a single symbol with the color ggplot2 would have used.

Following ggplot2's semantics for polygons, fill is the interior and colour is the border: a colour scale on a polygon layer colors the outlines while the interior keeps the constant fill. Constant outline colors and widths are taken from the plot as well. Mapping both fill and colour on the same layer is not supported.

Only a bare column name is supported for the fill/colour aesthetics; a constant or a computed expression (e.g. `aes(fill = AREA * 2)`) is an error.

Value

path, invisibly.

Data frame layers

A `geom_point()`, `geom_path()`, `geom_line()` or `geom_polygon()` layer drawn from a plain data frame is converted to an `sf` layer: one point per row, or one linestring/polygon per group (ggplot2's grouping — an explicit group aesthetic or the interaction of the discrete aesthetics; `geom_line()` orders each line by x like ggplot2 does, and polygon rings are closed). The plot must use `ggplot2::coord_sf()`, and the x/y values are taken to be coordinates in the panel CRS: `coord_sf()`'s `crs` argument if given, otherwise the CRS of the first `sf` layer (`coord_sf(default_crs = )` is not supported). Like `fill/colour`, the x/y aesthetics must be bare column names, and the layer must use the identity stat and position.

A point layer keeps every column of the data frame as attributes; a line/polygon layer keeps the columns that are constant within every group, one feature per group (a mapped `fill/colour` column must be constant within each group). `fill = NA/colour = NA` (including `geom_polygon()`'s default colour) render as "not drawn" in QGIS. The size, shape, linetype and alpha aesthetics are not carried over; those symbol properties keep the QGIS defaults.

The project opens zoomed to the plot's displayed range (the panel range, including the default expansion and any `ggplot2::coord_sf()` `xlim/ylim`), reprojected to the project CRS, rather than the whole world.

Examples

```
library(ggplot2)

nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
p <- ggplot(nc) +
  geom_sf(aes(fill = AREA))

write_qgs(p, tempfile(fileext = ".qgs"))
```

Index

`ggplot2::coord_sf()`, [2](#), [4](#)
`ggplot2::geom_line()`, [2](#)
`ggplot2::geom_path()`, [2](#)
`ggplot2::geom_point()`, [2](#)
`ggplot2::geom_polygon()`, [2](#)
`ggplot2::scale_fill_steps()`, [3](#)

`write_qgs`, [2](#)