Package: freestiler (via r-universe)

May 24, 2026

Title Create Vector Tiles from Spatial Data

Version 0.1.7

Date 2026-05-11

Description Create vector tile archives in 'PMTiles' format from 'sf' spatial data frames. Supports 'Mapbox Vector Tile' ('MVT') and 'MapLibre Tile' ('MLT') output formats. Uses a 'Rust' backend via 'extendr' for fast, in-memory tiling with zero external system dependencies.

License MIT + file LICENSE

URL https://walker-data.com/freestiler/, https://github.com/walkerke/freestiler/

BugReports https://github.com/walkerke/freestiler/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports sf

Suggests arrow, DBI, duckdb, httpuv, jsonlite, mapgl, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/rextendr/version 0.4.2.9000

SystemRequirements Cargo (Rust's package manager), rustc >= 1.77.2

Depends R (>= 4.2)

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

Repository https://walkerke.r-universe.dev

Date/Publication 2026-05-12 20:58:31 UTC

RemoteUrl https://github.com/walkerke/freestiler

RemoteRef HEAD

RemoteSha 5db19a4b56b63958e88d6fa178d8c06b9cf46c58

Contents

`freestile` *Create vector tiles from spatial data*

Description

Creates a PMTiles archive containing vector tiles from one or more sf data frames. Supports both Mapbox Vector Tile (MVT) and MapLibre Tile (MLT) formats, multi-layer output, feature dropping, point clustering, and feature coalescing.

Usage

```
freestile(
  input,
  output,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  generate_ids = TRUE,
  overwrite = TRUE,
  quiet = FALSE
)
```

Arguments

`input`	An sf data frame, or a named list of sf/freestile_layer objects for multi-layer output.
`output`	Character. Path for the output .pmtiles file.
`layer_name`	Character. Name for the tile layer. If NULL, derived from the output filename. Only used for single-layer input.

`tile_format` Character. Tile encoding format: `"mvt"` (default) for Mapbox Vector Tiles or `"mlt"` for MapLibre Tiles.

`min_zoom` Integer. Minimum zoom level (default 0).

`max_zoom` Integer. Maximum zoom level (default 14).

`base_zoom` Integer. Zoom level at and above which all features are present (no dropping). NULL (default) uses each layer's own max_zoom. The drop-rate curve is also computed relative to base_zoom, so lowering it produces gentler thinning at low zooms. Inspired by tippecanoe's `-B` / `--base-zoom`.

`drop_rate` Numeric. Exponential drop rate for feature thinning (e.g. 2.5). At each zoom level below base_zoom, features are retained at a rate of 1/drop_rate^(base_zoom - zoom). Points are thinned using spatial ordering; polygons/lines are thinned by area. NULL (default) disables drop-rate thinning.

`cluster_distance` Numeric. Pixel distance for point clustering. Points within this radius are merged into cluster features with a `point_count` attribute. NULL (default) disables clustering.

`cluster_maxzoom` Integer. Maximum zoom level for clustering. Above this zoom, individual points are shown. Default is max_zoom - 1.

`coalesce` Logical. Whether to merge features with identical attributes within each tile (default FALSE). Lines sharing endpoints are merged; polygons are grouped into MultiPolygons.

`simplification` Logical. Whether to snap geometries to the tile pixel grid at each zoom level (default TRUE). This provides zoom-adaptive simplification and prevents slivers between adjacent polygons.

`generate_ids` Logical. Whether to assign sequential feature IDs (default TRUE).

`overwrite` Logical. Whether to overwrite existing output file (default TRUE).

`quiet` Logical. Whether to suppress progress messages (default FALSE).

Details

Input data in any coordinate reference system (CRS) is automatically reprojected to WGS84 (EPSG:4326) before tiling.

Value

The output file path (invisibly).

Examples

```
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))

# Single layer
freestile(nc, "nc.pmtiles", layer_name = "counties")
```

```
# Multi-layer
pts <- st_centroid(nc)
freestile(
  list(counties = nc, centroids = pts),
  "nc_layers.pmtiles"
)

# With dropping and coalescing
freestile(nc, "nc_drop.pmtiles", drop_rate = 2.5, coalesce = TRUE)

# With point clustering
freestile(pts, "pts.pmtiles", cluster_distance = 50, cluster_maxzoom = 8)

## End(Not run)
```

`freestile_file` *Create vector tiles from a spatial file*

Description

Reads a GeoParquet, GeoPackage, Shapefile, or other spatial file directly into the tiling engine. Input data in any coordinate reference system is automatically reprojected to WGS84 (EPSG:4326) before tiling.

Usage

```
freestile_file(
  input,
  output,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  overwrite = TRUE,
  quiet = FALSE,
  engine = "geoparquet"
)
```

Arguments

input	Character. Path to the input spatial file.
output	Character. Path for the output .pmtiles file.
layer_name	Character. Name for the tile layer. If NULL, derived from the output filename.
tile_format	Character. "mvt" (default) or "mlt".
min_zoom	Integer. Minimum zoom level (default 0).
max_zoom	Integer. Maximum zoom level (default 14).
base_zoom	Integer. Zoom level at and above which all features are present. NULL (default) uses max_zoom.
drop_rate	Numeric. Exponential drop rate. NULL (default) disables.
cluster_distance	Numeric. Pixel distance for clustering. NULL disables.
cluster_maxzoom	Integer. Max zoom for clustering. Default max_zoom - 1.
coalesce	Logical. Whether to merge features with identical attributes (default FALSE).
simplification	Logical. Whether to snap geometries to the tile pixel grid (default TRUE).
overwrite	Logical. Whether to overwrite existing output (default TRUE).
quiet	Logical. Whether to suppress progress (default FALSE).
engine	Character. Backend engine: "geoparquet" (default, for GeoParquet files) or "duckdb" (for any file format DuckDB supports).

Details

The GeoParquet engine requires compilation with FREESTILER_GEOPARQUET=true. The DuckDB engine uses the Rust DuckDB backend when included in the build (enabled by default for native builds), or falls back to the R duckdb package. Control backend selection with options(freestiler.duckdb_backend = "auto"|"rust"|"r").

Value

The output file path (invisibly).

Examples

```
## Not run:
freestile_file("data.parquet", "output.pmtiles")
freestile_file("data.gpkg", "output.pmtiles", engine = "duckdb")

## End(Not run)
```

`freestile_layer` *Create a layer specification with per-layer zoom range*

Description

Wraps an sf object with optional per-layer zoom range overrides for use in multi-layer tile generation.

Usage

```
freestile_layer(input, min_zoom = NULL, max_zoom = NULL)
```

Arguments

`input`	An sf data frame.
`min_zoom`	Integer. Minimum zoom level for this layer. If NULL, uses the global min_zoom from `freestile()`.
`max_zoom`	Integer. Maximum zoom level for this layer. If NULL, uses the global max_zoom from `freestile()`.

Value

A freestile_layer object (list with class attribute).

Examples

```
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
roads <- st_read("roads.shp")

freestile(
  list(
    counties = freestile_layer(nc, min_zoom = 0, max_zoom = 10),
    roads = freestile_layer(roads, min_zoom = 8, max_zoom = 14)
  ),
  "layers.pmtiles"
)

## End(Not run)
```

`freestile_query` *Create vector tiles from a DuckDB SQL query*

Description

Executes a SQL query via DuckDB's spatial extension and pipes the results into the tiling engine. Uses the Rust DuckDB backend when included in the build (enabled by default for native builds), or falls back to the R duckdb package. Control backend selection with `options(freestiler.duckdb_backend = "auto"|"rust"|"r")`.

Usage

```
freestile_query(
  query,
  output,
  db_path = NULL,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  overwrite = TRUE,
  quiet = FALSE,
  source_crs = NULL,
  streaming = "auto"
)
```

Arguments

`query`	Character. A SQL query that returns a geometry column. DuckDB spatial functions like `ST_Read()` and `read_parquet()` are available. Multi-statement SQL is supported: setup statements (e.g., `LOAD h3;`) are executed first, then the final SELECT is used for tiling.
`output`	Character. Path for the output .pmtiles file.
`db_path`	Character. Path to a DuckDB database file, or NULL (default) for an in-memory database.
`layer_name`	Character. Name for the tile layer. If NULL, derived from the output filename.
`tile_format`	Character. `"mvt"` (default) or `"mlt"`.
`min_zoom`	Integer. Minimum zoom level (default 0).
`max_zoom`	Integer. Maximum zoom level (default 14).

base_zoom	Integer. Zoom level at and above which all features are present. NULL (default) uses max_zoom.
drop_rate	Numeric. Exponential drop rate. NULL (default) disables.
cluster_distance	Numeric. Pixel distance for clustering. NULL disables.
cluster_maxzoom	Integer. Max zoom for clustering. Default max_zoom - 1.
coalesce	Logical. Whether to merge features with identical attributes (default FALSE).
simplification	Logical. Whether to snap geometries to the tile pixel grid (default TRUE).
overwrite	Logical. Whether to overwrite existing output (default TRUE).
quiet	Logical. Whether to suppress progress (default FALSE).
source_crs	Character or NULL. CRS of the geometry returned by query, for example "EPSG:4326" or "EPSG:4267". Used only by the R duckdb fallback; ignored by the Rust DuckDB backend.
streaming	Character. DuckDB query execution mode: "auto" (default) enables the streaming point pipeline for large queries, "always" forces it, and "never" uses the existing in-memory path.

Details

When using the R fallback, source_crs must be supplied explicitly so the query result can be interpreted or reprojected correctly. Pass "EPSG:4326" if the SQL already returns WGS84 geometry, or the source CRS string (for example "EPSG:4267") to have DuckDB reproject to WGS84 before tiling. For file-based input where the CRS is embedded in the file, use freestile_file() with engine = "duckdb" instead, which auto-detects the source CRS.

Value

The output file path (invisibly).

Examples

```
## Not run:
# Query a GeoParquet file
freestile_query(
  "SELECT * FROM read_parquet('data.parquet') WHERE pop > 50000",
  "output.pmtiles"
)

# Query a Shapefile
freestile_query(
  "SELECT * FROM ST_Read('counties.shp')",
  "counties.pmtiles"
)

# Query with an existing DuckDB database
freestile_query(
  "SELECT * FROM my_table WHERE region = 'West'",
```

```
  "west.pmtiles",
  db_path = "my_database.duckdb"
)

## End(Not run)
```

`serve_tiles` *Serve PMTiles files via local HTTP server with CORS*

Description

Start a local HTTP server to serve PMTiles files with CORS headers and HTTP range request support. This allows PMTiles to be consumed by mapgl and MapLibre GL JS. The server runs in the background and can be stopped with `stop_server()`.

Usage

```
serve_tiles(path, port = 8080)
```

Arguments

`path`	Path to a directory containing PMTiles files, or a single PMTiles file. If a single file, its directory will be served.
`port`	Port number for the HTTP server. Default is 8080.

Details

If a server is already running on the requested port, it is stopped first.

The server uses httpuv (a dependency of Shiny) to serve static files with the CORS and range-request headers that PMTiles requires. Works well for files up to ~1 GB. For larger files, consider an external server like `npx http-server /path --cors -c-1`.

Value

Invisibly returns a list with `url`, `port`, and `dir`. The server handle is stored internally so it can be stopped with `stop_server()`.

See Also

`stop_server()`, `view_tiles()`

Examples

```
## Not run:
# Serve a directory
serve_tiles("/tmp/tiles")

# Serve a single file (its directory is served)
serve_tiles("us_bgs.pmtiles")

# Stop when done
stop_server()

## End(Not run)
```

`stop_server` *Stop a local tile server*

Description

Stop a local tile server

Usage

```
stop_server(port = NULL)
```

Arguments

`port`	Port number to stop, or `NULL` to stop all running servers.

Value

Invisibly returns `TRUE` if a server was stopped.

See Also

`serve_tiles()`

Examples

```
## Not run:
serve_tiles("tiles/")
stop_server()        # stop all
stop_server(8080)    # stop specific port

## End(Not run)
```

`view_tiles` *Quickly view a PMTiles file on an interactive map*

Description

Starts a local tile server (if needed) and creates an interactive mapgl map showing the tileset. Layer type and styling are auto-detected from the PMTiles metadata when possible.

Usage

```
view_tiles(
  input,
  layer = NULL,
  layer_type = NULL,
  color = NULL,
  opacity = 0.5,
  port = 8080,
  promote_id = NULL
)
```

Arguments

`input`	Path to a local `.pmtiles` file.
`layer`	Character. Source layer name to display. If `NULL` (default), the first layer in the tileset is used.
`layer_type`	Character. Map layer type: `"fill"`, `"line"`, or `"circle"`. If `NULL`, auto-detected from the PMTiles metadata geometry type.
`color`	Fill, line, or circle color. Default is `"navy"` for fill and line layers, `"steelblue"` for circle layers.
`opacity`	Numeric opacity (0–1). Default is 0.5.
`port`	Port for the local tile server. Default is 8080.
`promote_id`	Character. Property name to use as the feature ID for hover interactivity. If `NULL`, no feature promotion is used.

Value

A mapgl map object (can be piped into further mapgl operations).

See Also

`serve_tiles()`, `freestile()`

Examples

```
## Not run:
freestile(nc, "nc.pmtiles", layer_name = "counties")
view_tiles("nc.pmtiles")

# Override auto-detection
view_tiles("roads.pmtiles", layer_type = "line", color = "red")

# Point data
view_tiles("airports.pmtiles", layer_type = "circle", color = "orange")

## End(Not run)
```

Index