
carto Documentation

Release

Carto contributors

Jul 22, 2017

Contents:

1	Installation & Usage	3
1.1	Installation	3
1.2	Usage from the command line	3
1.3	Usage from JavaScript (Carto API)	4
2	Language Elements	9
2.1	Color	9
2.2	Float	10
2.3	URI	10
2.4	String	11
2.5	Boolean	11
2.6	Expressions	11
2.7	Numbers	11
2.8	Percentages	11
2.9	Functions	12
2.10	Mapnik Render-Time Variables	12
2.11	Controlling output of symbolizers and symbolizer attributes	12
3	Styling Concepts	15
3.1	Attachments and Instances	15
3.2	Variables & Expressions	16
3.3	Nested Styles	16
3.4	FontSets	16
3.5	Filters	17
4	MML File Structure	19
4.1	Properties Overview	19
4.2	Layer property	21
4.3	Stylesheet property	25
4.4	Example	26
5	Mapnik API documentation	29
5.1	3.0.6	29
5.2	3.0.3	74
5.3	3.0.0	120
5.4	2.3.0	166
5.5	2.2.0	204

5.6	2.1.1	241
5.7	2.1.0	271
5.8	2.0.2	302
5.9	2.0.1	324
5.10	2.0.0	346

CartoCSS (short: Carto) is a language for map design. It is similar in syntax to CSS, but builds upon it with specific abilities to filter map data and by providing things like variables. It targets the [Mapnik renderer](#) and is able to generate Mapnik XML or a JSON variant of Mapnik XML. It can run from the command line or in the browser by using a bundler like Browserify.

Carto is an evolution of the [Cascadenik](#) idea and language, with an emphasis on speed and flexibility.

Installation

If you are using a map design application like Kosmtik, Mapbox Studio Classic or Tilemill you already have CartoCSS installed and might be more interested in the language reference.

Else you can install the *carto* binary with NPM by running:

```
npm install -g carto
```

You should consider using a Node.js version manager like [NVM](#).

Optionally you may also want to install *millstone* which is required for resolving data in the same way as Mapbox Studio Classic does:

```
npm install -g millstone
```

Having *millstone* installed specifically enables support for localizing external resources (URLs and local files) referenced in your MML file, and detecting projections (using [node-srs](#)).

Usage from the command line

Now that Carto is installed you should have a *carto* command line tool available that can be run on a CartoCSS project:

```
carto project.mml > mapnik.xml
```

The following command line options are available:

- a / -api VERSION** Specify Mapnik API version (e.g. `-api 3.0.10`) (default: latest Mapnik API version)
- b / -benchmark** Outputs total compile time
- f / -file** Outputs to the specified file instead of stdout

- h / -help** Display help message
- l / -localize** Use millstone to localize resources when loading an MML (default: off)
- n / -nosymlink** Use absolute paths instead of symlinking files
- o / -output** Specify output format, possible values are `mapnik` and `json` (default: `mapnik`)
- ppi RESOLUTION** Pixels per inch used to convert m, mm, cm, in, pt, pc to pixels (default: 90.714)
- q / -quiet** Do not output any warnings (default: off)
- v / -version** Display version information

Usage from JavaScript (Carto API)

Alternatively, Carto can be used from JavaScript. While you theoretically could instantiate almost all of the classes the main outward facing stable interfaces are the `Renderer` and the `MML` interface. We start with an example and describe the details afterwards:

```
// defined variables:
// - input (the name or identifier of the file being parsed)
var carto = require('carto');

try {
  var data = fs.readFileSync(input, 'utf-8');
  var mml = new carto.MML({});
  mml.load(path.dirname(input), data, function (err, data) {
    var output = {};

    if (!err) {
      output = new carto.Renderer({
        filename: input
      }).render(data);
    }

    if (output.msg) {
      output.msg.forEach(function (v) {
        if (v.type === 'error') {
          console.error(carto.Util.getMessageToPrint(v));
        }
        else if (v.type === 'warning') {
          console.warn(carto.Util.getMessageToPrint(v));
        }
      });
    }

    // output content (if no errors)
    if (output.data) {
      console.log(output.data);
    }
  });
} catch (err) {
  // program failures
  ...
}
```


MML

The MML interface loads and processes a MML file (see [MML File Structure](#) for details). You instantiate the class with `carto.MML`. The constructor takes a options object with the following possible attributes:

- `localize` *boolean* (same as `-l / --localize` on the command line) - this uses `millstone` to localize stylesheet resources
- `nosymlink` *boolean* (same as `-n / --nosymlink` on the command line) - for millstone, tells it to use unmodified paths instead of symlinking files

By calling `load(basedir, data, callback)` the MML file is loaded and processed. This method does not perform reading from a file, so you have to read the contents of the file yourself and provide it as string to the method via the `data` parameter to the load function. The `basedir` parameter is used to resolve stylesheet references. When the processing is finished the specified `callback` function is called, which has the following signature:

```
function (err, data) {}
```

If an error occurred you find the message within `err` and `data` is `null`. When successful you find the processed MML data structure in `data` and `err` is `null`. The structure within `data` is expected by the `Renderer` interface's `render` method.

Note: If you want to use Carto within the browser you should not use MML loading via `carto.MML.load`. Instead you should supply the JSON of the MML including the Stylesheet strings directly to `carto.Renderer.render`.

Renderer

The `Renderer` interface performs the parsing and transformation for rendering from a MML file string (either self loaded or loaded through the MML interface) or from a MSS file string (without layers). You instantiate the class with `carto.Renderer`. The constructor takes a options object with the following possible attributes:

- `benchmark` *boolean* (similar to `-b / --benchmark` on the command line) - specifies if `carto` should run in benchmarking mode
- `effects` *array* - a container for side-effects limited to `FontSets`
- `filename` *string* - name of the input file, used to format errors and warnings
- `outputFormat` *string [mapnik|json]* (similar to `-o / --output` on the command line) - specifies which format the output should have, either Mapnik XML or JSON similar to Mapnik XML
- `ppi` *float* (similar to `-ppi` on the command line) - Pixels per inch used to convert m, mm, cm, in, pt, pc to pixels
- `quiet` *boolean* (similar to `-q / --quiet` on the command line) - if `carto` should output warnings or not
- `reference` *class* - `carto` uses a reference to validate input. You can specify your own which has to adhere to the specification. (see [Using a custom reference](#))
- `validationData` *object*
 - `fonts` *array* - a list of fonts that `carto` should use to validate if used fonts are valid/present
- `version` *string (semver)* (similar to `-a / --api` on the command line) - specify which Mapnik API version `carto` should use

`carto.Renderer` offers two methods for actual rendering. You can either use `render(data)` or `renderMSS(data)`. Both accept a string of either a processed MML file or a MSS style fragment. The `render` method produces a full-featured style output while the `renderMSS` outputs only a style fragment. Both return the following object:

```
{
  msg: [],
  data: ''
}
```

If errors or warnings occurred during rendering you will find them in `msg` and `data` will be `null` (in case of errors). The actual output is found in `data` if no errors occurred.

Util

Carto provides a `Util` class to assist you with e.g. message formatting. Like in the example you can call `getMessageToPrint` with a received message object to output it nicely formatted as string.

Using a custom reference

Carto uses a reference to validate input. This reference specifies which rules and functions are valid and which types a rule can take. It also describes how rules are transformed for the output. By default carto uses [mapnik-reference](#) as reference, but you can also use your own. It has to adhere to the following specification:

```
{
  versions: [], // array of versions (semver) as strings
  latest: '', // latest version (semver) as strings
  load: function (version) {} // return data structure for specified version
}
```

The data structure returned by `load` has to look like this:

```
{
  version: '', // version (semver) as string
  style: {}, // rules that apply to the style as a whole
  layer: {}, // rules that apply to a layer as a whole
  symbolizers: {}, // rules that apply to different elements of the renderer, this_
  ↪elements make up the map
  colors: {}, // color names and their mapping to RGB values
  datasources: {} // possible data sources for the rendering library and their_
  ↪parameters
}
```

Note: `datasources` is not yet used by carto for validation.

All entries that contain rules are objects where their attributes are named after a color, symbolizer or rule. `style` and `layer` have the same inner structure. Here is an example:

```
{
  'filter-mode': {
    type: {},
    doc: '',
    'default-value': '',
    'default-meaning': ''
  }
  ...
}
```

symbolizer first contains the possible symbolizers and then their rules:

```
{
  polygon: {
    fill: {},
    'fill-opacity': {}
    ...
  }
  ...
}
```

colors maps color names to their RGB values:

```
{
  aliceblue: [ 240, 248, 255 ],
  antiquewhite: [ 250, 235, 215 ]
  ...
}
```

datasources is similar to symbolizers and contains first the possible data sources and then their possible parameters:

```
{
  csv: {
    file: {},
    base: {}
    ...
  }
}
```

Rules (all the parts that where specified with {} with a little preview at filter-mode) can have several attributes that are evaluated:

```
name: {
  css: '', // rule name which is used in CartoCSS
  default-meaning: '', // meaning of the default value
  default-value: '', // default value of the rule
  doc: '', // documentation about the rule
  expression: bool, // whether this rule is a expression or not
  functions: [], // array of arrays that contain function name and # of params e.g.
↳ ["matrix", 6]
  range: '', // range of values that are allowed e.g. 0-1
  required: bool, // if this rule is required
  status: '[unstable|experimental|deprecated]', // if omitted it means stable
  type: '[bbox|boolean|color|float|functions|numbers|string|uri]', // type can also
↳ be an array of keywords
}
```

Caution: Adherence to the specification is not assessed in-depth because that would be too resource intensive. If you don't adhere to the specification it is quite likely that you hit runtime errors.

Below is a list of values and an explanation of any expression that can be applied to properties in CartoCSS.

Color

CartoCSS accepts a variety of syntaxes for colors - HTML-style hex values, rgb, rgba, hsl, hsla, hsluv, and hsluva. It also supports the predefined HTML colors names, like *yellow* and *blue*.

```
#line {
  line-color: #ff0;
  line-color: #ffff00;
  line-color: rgb(255, 255, 0);
  line-color: rgba(255, 255, 0, 1);
  line-color: hsl(100, 50%, 50%);
  line-color: hsla(100, 50%, 50%, 1);
  line-color: hsluv(100, 50%, 50%); // same values yield different color than HSL
  line-color: hsluva(100, 50%, 50%, 1);
  line-color: yellow;
}
```

Especially of note is the support for HSL and HSLuv, which can be easier to reason about than RGB. CartoCSS also includes several color operation functions borrowed from [LessCSS](#):

```
// lighten and darken colors
lighten(#ace, 10%);
darken(#ace, 10%);

// saturate and desaturate
saturate(#550000, 10%);
desaturate(#00ff00, 10%);

// increase or decrease the opacity of a color
fadein(#fafafa, 10%);
```

```
fadeout(#fefefe, 14%);

// spin rotates a color around the color wheel by degrees
spin(#ff00ff, 10);

// mix generates a color in between two other colors.
mix(#fff, #000, 50%);

// get color components
hue(#ff00ff);
saturation(#ff00ff);
lightness(#ff00ff);
alpha(hsla(100, 50%, 50%, 0.5));
```

These functions all take arguments which can be color variables, literal colors, or the results of other functions operating on colors. All the above mentioned functions also come in a *functionp*-variant (e.g. `lightenp`), which force a given color into HSLuv color space.

Float

Float is a fancy way of saying ‘number’. In CartoCSS, you specify *just a number* - unlike CSS, there are no units, but everything is specified in pixels.

```
#line {
  line-width: 2;
}
```

It’s also possible to do simple math with number values:

```
#line {
  line-width: 4 / 2; // division
  line-width: 4 + 2; // addition
  line-width: 4 - 2; // subtraction
  line-width: 4 * 2; // multiplication
  line-width: 4 % 2; // modulus
}
```

URI

URI is a fancy way of saying URL. When an argument is a URI, you use the same kind of `url('place.png')` notation that you would with HTML. Quotes around the URL aren’t required, but are highly recommended. URIs can be paths to places on your computer, or on the internet.

```
#markers {
  marker-file: url('marker.png');
}
```

String

A string is basically just text. In the case of CartoCSS, you're going to put it in quotes. Strings can be anything, though pay attention to the cases of `text-name` and `shield-name` - they actually will refer to features, which you refer to by putting them in brackets, as seen in the example below.

```
#labels {
  text-name: "[MY_FIELD]";
}
```

Boolean

Boolean means yes or no, so it accepts the values `true` or `false`.

```
#markers {
  marker-allow-overlap: true;
}
```

Expressions

Expressions are statements that can include fields, numbers, and other types in a really flexible way. You have run into expressions before, in the realm of 'fields', where you'd specify "`[FIELD]`", but expressions allow you to drop the quotes and also do quick addition, division, multiplication, and concatenation from within CartoCSS syntax.

```
#buildings {
  building-height: [HEIGHT_FIELD] * 10;
}
```

Numbers

Numbers are comma-separated lists of one or more number in a specific order. They're used in line dash arrays, in which the numbers specify intervals of line, break, and line again.

```
#disputedboundary {
  line-dasharray: 1, 4, 2;
}
```

Percentages

In CartoCSS, the percentage symbol, `%` universally means `value/100`. It's meant to be used with ratio-related properties, like opacity rules.

Attention: You should not use percentages as widths, heights, or other properties - unlike CSS, percentages are not relative to cascaded classes or page size, they're, as stated, simply the value divided by one hundred.

In an example:

```
#world {
  // this syntax
  polygon-opacity: 50%;

  // is equivalent to
  polygon-opacity: 0.5;
}
```

Functions

Functions are comma-separated lists of one or more functions. For instance, transforms use the `functions` type to allow for transforms within CartoCSS, which are optionally chainable.

```
#point {
  point-transform: scale(2, 2);
}
```

Mapnik Render-Time Variables

Mapnik >= 3.0.0 supports variables of the form `@var`. These can be used from CartoCSS by specifying them as strings. For example:

```
#layer {
  line-width: '@zoom';
}
```

For this to have any effect you have to pass the variables to Mapnik at render time in a hashmap of the form `variable_name:variable_value`.

Controlling output of symbolizers and symbolizer attributes

You can control symbolizer output by using rules that work on the whole symbolizer. E.g. `line` works on the line symbolizer. By using the keywords `none` or `auto` you can either suppress the symbolizer or output it with default values. The keyword `auto` does not work on shield and text symbolizers because they have attributes without default values. Here is an example how this works:

```
#layer {
  line: none;
  line-width: 2;
  [feature = 'redfeature'] {
    line-color: red;
  }
  [feature = 'bluefeature'] {
    line-color: blue;
  }
}
```


Without `line: none` carto would output a line symbolizer with default values for all features other than `redfeature` and `bluefeature`, that is a black line with width 1. In contrast, you can quickly output a symbolizer with default value by using `auto`:

```
#layer {
  [feature = 'quickfeature'] {
    marker: auto;
  }
}
```

This outputs a default markers symbolizer for all `quickfeature` features.

You can also control the output of individual symbolizer properties by specifying them with the keyword `none` e.g. `line-color: none`. They will then be removed from the symbolizer thus using their default value or not using them at all. This does not work or makes sense for all properties like e.g. not for `text-face-name` as it does not have a default value. For an overview over properties where this works or makes sense see [this list](#). In this case the use of `none` and `auto` is equivalent. In both cases the default value will be used as Mapnik uses the default value automatically when the property is not present.

Attachments and Instances

In CSS, a certain object can only have one instance of a property. A `<div>` has a specific border width and color, rules that match better than others (`#id` instead of `.class`) override previous definitions. CartoCSS acts the same way normally for the sake of familiarity and organization, but Mapnik itself is more powerful.

Layers in Mapnik can have multiple `borders` and multiple copies of other attributes. This ability is useful in drawing line outlines, like in the case of road borders or ‘glow’ effects around coasts. CartoCSS makes this accessible by allowing attachments to styles:

```
#world {
  line-color: #fff;
  line-width: 3;
}

#world::outline {
  line-color: #000;
  line-width: 6;
}
```

Attachments are optional.

While attachments allow creating implicit “layers” with the same data, using **instances** allows you to create multiple symbolizers in the same style/layer:

```
#roads {
  casing/line-width: 6;
  casing/line-color: #333;
  line-width: 4;
  line-color: #666;
}
```

This makes Mapnik first draw the line of color `#333` with a width of 6, and then immediately afterwards, it draws the same line again with width 4 and color `#666`. Contrast that to attachments: Mapnik would first draw all casings before

proceeding to the actual lines.

Variables & Expressions

CartoCSS inherits from its basis in [LessCSS](#) some new features in CSS. One can define variables in stylesheets, and use expressions to modify them.

```
@mybackground: #2B4D2D;

Map {
  background-color: @mybackground
}

#world {
  polygon-fill: @mybackground + #222;
  line-color: darken(@mybackground, 10%);
}
```

Nested Styles

CartoCSS also inherits nesting of rules from LessCSS.

```
/* Applies to all layers with .land class */
.land {
  line-color: #ccc;
  line-width: 0.5;
  polygon-fill: #eee;
  /* Applies to #lakes.land */
  #lakes {
    polygon-fill: #000;
  }
}
```

This can be a convenient way to group style changes by zoom level:

```
[zoom > 1] {
  /* Applies to all layers at zoom > 1 */
  polygon-gamma: 0.3;
  #world {
    polygon-fill: #323;
  }
  #lakes {
    polygon-fill: #144;
  }
}
```

FontSets

By defining multiple fonts in a `text-face-name` definition, you create [FontSets](#) in CartoCSS. These are useful for supporting multiple character sets and fallback fonts for distributed styles.

This CartoCSS code

```
#world {
  text-name: "[NAME]";
  text-size: 11;
  text-face-name: "Georgia Regular", "Arial Italic";
}
```

becomes this XML code

```
<FontSet name="fontset-0">
  <Font face-name="Georgia Regular">
  <Font face-name="Arial Italic">
</FontSet>
<Style name="world-text">
  <Rule>
    <TextSymbolizer fontset-name="fontset-0"
      size="11"
      name="[NAME]">
  </Rule>
</Style>
```

Filters

CartoCSS supports a variety of filter styles:

Numeric comparisons:

```
#world[population > 100]
#world[population < 100]
#world[population >= 100]
#world[population <= 100]
```

General comparisons:

```
#world[population = 100]
#world[population != 100]
```

String comparisons:

```
/* a regular expression over name */
#world[name =~ "A.*"]
```

More complex expressions:

```
#world[[population] % 50 = 0]
#world[[population] * 2 < 1000]
```

MML File Structure

The MML file (here: Map Markup Language file) is a **YAML** or **JSON** file containing layer definitions and stylesheet references. It is the central part of a CartoCSS stylesheet. If you generate Mapnik XML from a stylesheet this is the file you feed to carto and from which and its references the XML is produced.

You may wonder whether you should use YAML or JSON. Internally, JSON is used because it is a language that is easy to understand for machines. YAML is a superset of JSON and is easier to read and write for humans and is also less verbose. If you edit a stylesheet collaboratively and use version control YAML might make it easier to resolve version conflicts. Carto fully understands both forms.

Note: YAML makes it less tedious to specify repeating properties like database information for PostGIS datasources or extents by using a language feature called **anchors** or **aliases**.

Properties Overview

What follows is an overview over properties a MML file can have. This list is not complete. There may be other (also undocumented) options that Mapnik understands. Simple properties are described here in alphabetical order while more complex ones get their own section.

center

Type: Array

Specifies the initial center coordinates and zoom level of the map. Longitude, latitude and zoom level in that order. Example (WGS84): [-1.28398, 47.08997, 17]

bounds

Type: Array

Defines a bounding box of the map extent. Lower left coordinates (longitude, latitude) and upper right coordinates (longitude, latitude) in that order. Example (WGS84): [-179, -80, 179, 80]

description

Type: `String`

A description for the stylesheet (usually a bit longer than the name)

format

Type: `Keyword`

Possible values: `png`, `jpeg`, `tiff`, `webp`

Specifies the output format Mapnik should use. For more details see the [Mapnik wiki page on Image IO](#).

Layer

Lists all layers used in the project (see *Layer property*)

maxzoom

Type: `Integer`

Specifies the maximum zoom level of the map

metatile

Type: `Integer`

Specifies the number of tiles that make up one side of a metatile. For example, if the number is 2 then the metatile is 2 tiles wide and tall and consists of 4 individual tiles. For efficiency reasons Mapnik generates metatiles before splitting them into individual tiles.

minzoom

Type: `Integer`

Specifies the minimum zoom level of the map

name

Type: `String`

A name for the stylesheet

`_properties`

Type: `Object`

This is the same as the `properties` property for layers, but on a global level with a bit different structure. It is used when you do not specify the layers in the MML itself but only reference them. This is used in vector tile styles where the style and the data are separate.

First specify the layer name and then below it specify its properties as you would do in the `properties` property of the specific layer.

`scale`

Type: `Integer`

Specifies pixel scaling for the output. For example, a scale of 2 means that there are two pixels for each map pixel in the output.

`srs`

Type: `String`

Specifies the projection used by Mapnik using the PROJ.4 format (SRS means *Spatial reference system*). The format can be determined by e.g. using spatialreference.org.

`Stylesheet`

Lists all styles or style files in the project (see *Stylesheet property*)

Layer property

Type: `Array`

Beneath this property layer objects are referenced that are the building blocks of the map style. The order of specification is important as it constitutes the drawing order of layers used by Mapnik. Layers specified first are drawn first and those specified later are drawn afterwards.

Layers have different properties and their data can come from different data sources such as shape files or relational databases like PostgreSQL/PostGIS.

A layer object can have the following properties (there may be more that Mapnik understands, also undocumented ones).

`class`

Type: `String`

One or more classes associated with this layer separated by blanks. In style selectors a class can be referenced by `.classname` if class contains `classname` similar to CSS.

Datasource

Mapnik supports different datasources. Beneath this property you specify the type of the datasource and additional properties depending on the type.

Not all possible configuration options for each datasource are listed here. For further information see the page for the datasource type on the Mapnik Wiki: e.g. [PostGIS](#), [PgRaster](#), [ShapeFile](#), [GDAL](#), [OGR](#), [OsmPlugin](#).

type

Type: `Keyword`

Possible values: `shape`, `postgis`, `pgraster`, `raster`, `(gdal)`, `(ogr)`, `(osm)`

Specifies the format of the data source. Types in parenthesis are *not build by default* according to the [Mapnik Wiki](#).

band (gdal, pgraster)

Type: `Integer`

With this property you can request a specific raster band index (1-based). By specifying `-1` (gdal) / `0` (pgraster) you request to read all bands.

dbname (postgis, pgraster)

Type: `String`

Specifies the database name of the PostgreSQL database.

encoding (ogr, postgis, shape)

Type: `String`

Specifies the encoding of the database or shapefile e.g. `utf-8` or `latin1`.

extent (ogr, postgis, pgraster)

Type: `String`

Specifies the maximum extent of the geometries or raster data. Lower left coordinates (longitude, latitude) and upper right coordinates (longitude, latitude) in that order. By default this is deduced from the metadata of the table.

file (gdal, ogr, osm, raster, shape)

Type: `String`

Path and file name.

geometry_field (postgis)

Type: `String`

Specifies the name of the column that contains the geometry. Normally this will be deduced from the query but sometimes it can be necessary to specify it manually e.g. when there is more than one column with geometry.

host (postgis, pgraster)

Type: `String`

Specifies the hostname of the PostgreSQL database.

layer (ogr)

Type: `String`

The name of the layer to display.

layer_by_index (ogr)

Type: `Integer`

The index of the layer to display (mandatory if no layer name specified).

layer_by_sql (ogr)

Type: `String`

SQL-Statement to execute against the OGR datasource.

password (postgis, pgraster)

Type: `String`

Specifies the password for connecting to the PostgreSQL database.

port (postgis, pgraster)

Type: `String`

Specifies the port of the PostgreSQL database.

raster_field (pgraster)

Type: `String`

Specifies the name of the column that contains the raster data. Normally this will be deduced from the query but sometimes it can be necessary to specify it manually e.g. when there is more than one column with raster data.

simplify_geometries (postgis)

Type: `Boolean`

Specify if input vertices should be automatically reduced or not.

table (postgis, pgraster)

Type: `String`

Either the name of the table to fetch or a sub query (...) *AS queryname*.

user (postgis, pgraster)

Type: `String`

Specifies the username for connecting to the PostgreSQL database.

extent

Type: `Array`

Defines a bounding box of the layer extent. Lower left coordinates (longitude, latitude) and upper right coordinates (longitude, latitude) in that order. Example (WGS84): `[-179, -80, 179, 80]`

geometry

Type: `Keyword`

Possible values: `linestring`, `point`, `polygon`, `raster`

Specifies the geometry type for (the datasource of) this layer.

id

Type: `String`

A unique identifier for this layer. In style selectors it can be referenced with `#layerid` if the id is `layerid` similar to CSS.

properties

Type: `Object`

This property basically adds any sub-property as attribute to the Mapnik layer. So available values depend greatly on what Mapnik allows. Here is an (incomplete) list:

abstract

Type: `String`

A short description of this layer (typically longer than the title).

group-by

Type: `String`

Enables [grouped rendering](#) for Mapnik by specifying the field of the datasource that should be used for grouping. Mapnik then renders all styles of the layer for those features that have the same value for that field before moving on to other features that have different value.

maxzoom

Type: `Integer`

Specifies the zoom level until which the layer is visible.

minzoom

Type: `Integer`

Specifies the zoom level from which the layer is visible.

status

Type: `Integer`

Specifies if the layer is active or not. 0 means inactive or off, 1 means active or on.

title

Type: `String`

The title of this layer. Probably more verbose than the ID.

srs

Type: `String`

Specifies the projection for this layer using the [PROJ.4](#) format (SRS means [Spatial reference system](#)). The format can be determined by e.g. using [spatialreference.org](#).

srs-name

Type: `String`

The name of this SRS.

Stylesheet property

Type: `Array`

You have two options to specify the styles. Either you reference MSS files (here: Map Stylesheet files) or you specify style objects directly.

Referencing style files

Here you reference the style files used in an array of paths/file names. Carto understands relative as well as absolute paths. The order of style references is normally not important, but re-definition of variables can be affected by the order of the style files referenced.

Specifying style objects

Internally style file references are transformed into style objects anyway, so you can also specify them directly. This only makes sense if you generate them programmatically otherwise the notation could become a bit tedious. You specify an array of style objects. The order of the objects is normally not important, but re-definition of variables can be affected by the order of the style objects.

A style object consists of the following properties.

id

This is the identifier of the style object. When styles are being read from a style file this is usually the file name. The property is used when generating errors or warnings so it is advisable to set something recognizable here.

data

This contains the actual style in the form of a string.

Example

Here is a simple MML file example with two layers (one shapefile and one PostGIS layer) referencing one style file in YAML format. It has been modified from the MML file of [openstreetmap-carto](#).

```
scale: 1
metatile: 2
name: Example MML file
description: A example MML file to illustrate its options
bounds: &world
  - -180
  - -85.05112877980659
  - 180
  - 85.05112877980659
center:
  - 0
  - 0
  - 4
format: png
minzoom: 0
maxzoom: 19
srs: "+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0.0 +k=1.
↪0 +units=m +nadgrids=@null +wktext +no_defs +over"

# Various parts to be included later on
_parts:
  extents: &extents
  extent: *world
```

```

srs-name: "900913"
srs: "+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0.0_
↪+k=1.0 +units=m +nadgrids=@null +wktext +no_defs +over"
osm2pgsql: &osm2pgsql
  type: "postgis"
  dbname: "gis"
  key_field: ""
  geometry_field: "way"
  extent: "-20037508,-20037508,20037508,20037508"

Stylesheet:
- style_file.mss
Layer:
- id: world
  name: world
  geometry: polygon
  <<: *extents
  Datasource:
    file: data/simplified-land-polygons-complete-3857/simplified_land_polygons.shp
    type: shape
  properties:
    maxzoom: 9
- id: landcover-low-zoom
  name: landcover-low-zoom
  geometry: polygon
  <<: *extents
  Datasource:
    <<: *osm2pgsql
    table: |-
      (SELECT
        way, name, way_pixels,
        COALESCE(wetland, landuse, "natural") AS feature
      FROM (SELECT
        way, COALESCE(name, '') AS name,
        ('landuse_' || (CASE WHEN landuse IN ('forest', 'military') THEN_
↪landuse ELSE NULL END)) AS landuse,
        ('natural_' || (CASE WHEN "natural" IN ('wood', 'sand', 'scree',
↪'shingle', 'bare_rock') THEN "natural" ELSE NULL END)) AS "natural",
        ('wetland_' || (CASE WHEN "natural" IN ('wetland', 'mud') THEN (CASE_
↪WHEN "natural" IN ('mud') THEN "natural" ELSE tags->'wetland' END) ELSE NULL END))_
↪AS wetland,
        way_area/NULLIF(!pixel_width!::real*!pixel_height!::real,0) AS way_
↪pixels
      FROM planet_osm_polygon
      WHERE (landuse IN ('forest', 'military')
        OR "natural" IN ('wood', 'wetland', 'mud', 'sand', 'scree', 'shingle',
↪'bare_rock'))
        AND way_area > 0.01*!pixel_width!::real*!pixel_height!::real
        AND building IS NULL
      ORDER BY COALESCE(layer,0), way_area DESC
    ) AS features
    ) AS landcover_low_zoom
  properties:
    minzoom: 7
    maxzoom: 9

```


The following is a list of properties provided in CartoCSS that you can apply to map elements when using the Mapnik renderer. Each API version of Mapnik has its own subsection. Pick the one that applies to the version you are running. If there is no subsection for your specific version the next older listed version applies to this version.

3.0.6

Style

image-filters

Type: functions

Possible values: `agg-stack-blur` `emboss` `blur` `gray` `sobel` `edge-detect` `x-gradient` `y-gradient` `invert` `sharpen` `color-blind-protanope` `color-blind-deuteranope` `color-blind-tritanope` `colorize-alpha` `color-to-alpha` `scale-hsla`

Default Value: none (*no filters*)

A list of image filters that will be applied to the active rendering canvas for a given style. The presence of one more `image-filters` will trigger a new canvas to be created before starting to render a style and then this canvas will be composited back into the main canvas after rendering all features and after all `image-filters` have been applied. See `direct-image-filters` if you want to apply a filter directly to the main canvas.

image-filters-inflate

Type: boolean

Default Value: false (*No special handling will be done and image filters that blur data will only blur up to the edge of a tile boundary*)

A property that can be set to true to enable using an inflated image internally for seamless blurring across tiles (requires buffered data).

direct-image-filters

Type: functions

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen color-blind-protanope color-blind-deuteranope color-blind-tritanope colorize-alpha color-to-alpha scale-hsla`

Default Value: none (*no filters*)

A list of image filters to apply to the main canvas (see the `image-filters` doc for how they work on a separate canvas).

comp-op

Type: keyword

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: float

Default Value: 1 (*No separate buffer will be used and no alpha will be applied to the style after rendering.*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer).

Symbolizers

map

background-color

Type: color

Default Value: none (*Will be rendered transparent.*)

Map Background color.

background-image

Type: uri

Default Value: *(No background image will be used.)*

An image that is repeated below all features on a map as a background. Accepted formats: svg, jpg, png, tiff, and webp.

background-image-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over *(The background-image will be blended with the background normally (placed on top of any existing background-color).)*

Set the compositing operation used to blend the image into the background.

background-image-opacity

Type: float

Default Value: 1 *(The image opacity will not be changed when applied to the map background.)*

Set the opacity of the image.

srs

Type: string

Default Value: +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs *(The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space.)*

Map spatial reference (proj4 string).

buffer-size

Type: `float`

Default Value: 0 (*No buffer will be used.*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

maximum-extent

Type: `string`

Default Value: `-20037508.34,-20037508.34,20037508.34,20037508.34` (*All data will be clipped to global mercator extent (default is applied in Carto.js).*)

An extent to be used to limit the bounds used to query all layers during rendering. Should be `minx`, `miny`, `maxx`, `maxy` in the coordinates of the Map.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: `none` (*No map-specific fonts will be registered.*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon

Type: `keyword`

Status: **unstable**

Possible values: `auto none`

Default Value:

Allows omitting a polygon symbolizer rule or emitting it with default values.

polygon-fill

Type: `color`

Default Value: The color gray will be used for fill. (*Gray and fully opaque (alpha = 1), same as `rgb(128,128,128)` or `rgba(128,128,128,1)`.*)

Fill color to assign to a polygon.

polygon-opacity

Type: `float`

Default Value: 1 (*Color is fully opaque.*)

The opacity of the polygon.

polygon-gamma

Type: `float`

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon edges.

polygon-gamma-method

Type: `keyword`

Possible values: `power linear none threshold multiply`

Default Value: `power` (*$pow(x, gamma)$ is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

polygon-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extend outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

polygon-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

polygon-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function.

polygon-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a line symbolizer rule or emitting it with default values.

line-color

Type: color

Default Value: black (*black and fully opaque (alpha = 1), same as rgb(0,0,0) or rgba(0,0,0,1).*)

The color of a drawn line.

line-width

Type: float

Default Value: 1 (*The line will be rendered 1 pixel wide.*)

The width of a line in pixels.

line-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of a line.

line-join

Type: keyword

Possible values: miter miter-revert round bevel

Default Value: miter (*The line joins will be rendered using a miter look.*)

The behavior of lines when joining.

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt (*The line endings will be rendered using a butt look.*)

The display of line endings.

line-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of stroke line.

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*pow(x,gamma) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

line-dasharray

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

Valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled).

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'.*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

line-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

line-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: float

Status: unstable

Default Value: 0 (*Will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full (*The line will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker

Type: `keyword`

Status: **unstable**

Possible values: `auto none`

Default Value:

Allows omitting a marker symbolizer rule or emitting it with default values.

marker-file

Type: `uri`

Default Value: `none` (*An ellipse or circle, if width equals height.*)

A file that this marker shows at each placement. If no file is given, the marker will show an ellipse. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

marker-opacity

Type: `float`

Default Value: `1` (*The stroke-opacity and fill-opacity of the marker.*)

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The fill opacity of the marker. This property will also set the `fill-opacity` of elements in an SVG loaded from a file.

marker-line-color

Type: `color`

Default Value: `black` (*The marker will be drawn with a black outline.*)

The color of the stroke around the marker. This property will also set the `stroke` of elements in an SVG loaded from a file.

marker-line-width

Type: float

Default Value: 0.5 (*The marker will be drawn with an outline of .5 pixels wide.*)

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself. This property will also set the `stroke-width` of elements in an SVG loaded from a file.

marker-line-opacity

Type: float

Default Value: 1 (*Color is fully opaque. This property will also set the “stroke-opacity” of elements in an SVG loaded from a file.*)

The opacity of a line.

marker-placement

Type: keyword

Possible values: point line interior vertex-first vertex-last

Default Value: point (*Place markers at the center point (centroid) of the geometry.*)

Attempt to place markers on a point, in the center of a polygon, or if `markers-placement:line`, then multiple times along a line. ‘interior’ placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior. The ‘vertex-first’ and ‘vertex-last’ options can be used to place markers at the first or last vertex of lines or polygons.

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each.*)

A special setting to allow the user to control rendering behavior for ‘multi-geometries’ (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The ‘each’ policy is default and means all geometries will get a marker. The ‘whole’ policy means that the aggregate centroid between all geometries will be used. The ‘largest’ policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Status: deprecated

Possible values: `arrow` `ellipse`

Default Value: `ellipse` (*The marker shape is an ellipse.*)

The default marker-type. If a SVG file is not given as the `marker-file` parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width).

marker-width

Type: float

Default Value: `10` (*The marker width is 10 pixels.*)

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: `10` (*The marker height is 10 pixels.*)

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: `blue` (*The marker fill color is blue.*)

The color of the area of the marker. This property will also set the `fill` of elements in an SVG loaded from a file.

marker-allow-overlap

Type: boolean

Default Value: `false` (*Do not allow markers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-avoid-edges

Type: `boolean`

Default Value: `false` (*Markers will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing markers that intersect with tile boundaries.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Value to control whether the placement of the feature will prevent the placement of other features.

marker-spacing

Type: `float`

Default Value: `100` (*In the case of marker-placement:line then draw a marker every 100 pixels along a line.*)

Space between repeated markers in pixels. If the spacing is less than the marker size or larger than the line segment length then no marker will be placed. Any value less than 1 will be ignored and the default will be used instead.

marker-max-error

Type: `float`

Default Value: `0.2` (*N/A: not intended to be changed.*)

N/A: not intended to be changed.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform marker instance with specified function. Ignores map scale factor.

marker-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

marker-simplify

Type: `float`

Default Value: `0` (*Geometry will not be simplified.*)

geometries are simplified by the given tolerance.

marker-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

marker-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform marker geometry with specified function.

marker-offset

Type: float

Default Value: 0 (*Will not be offset.*)

Offsets a marker from a line a number of pixels parallel to its actual path. Positive values move the marker left, negative values move it right (relative to the directionality of the line).

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

marker-direction

Type: keyword

Possible values: auto auto-down left right left-only right-only up down

Default Value: right (*Markers are oriented to the right in the line direction.*)

How markers should be placed along lines. With the “auto” setting when marker is upside down the marker is automatically rotated by 180 degrees to keep it upright. The “auto-down” value places marker in the opposite orientation to “auto”. The “left” or “right” settings can be used to force marker to always be placed along a line in a given direction and therefore disables rotating if marker appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down markers rather than trying to flip it. The “up” and “down” settings don’t adjust marker’s orientation to the line direction.

shield

shield

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a shield symbolizer rule.

shield-name

Type: `string`

Default Value: *(No text label will be rendered with the shield.)*

Value to use for a shield's text label. Data columns are specified using brackets like `[column_name]`.

shield-file

Type: `uri`

Default Value: `none`

Image file to render behind the shield text. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

shield-face-name

Type: `string`

Default Value: `none`

Font name and style to use for the shield text.

shield-unlock-image

Type: `boolean`

Default Value: `false` *(text alignment relative to the shield image uses the center of the image as the anchor for text positioning.)*

This parameter should be set to `true` if you are trying to position text beside rather than on top of the shield image.

shield-size

Type: `float`

Default Value: `10` *(Font size of 10 will be used to render text.)*

The size of the shield text in pixels.

shield-fill

Type: color

Default Value: black (*The shield text will be rendered black.*)

The color of the shield text.

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this shield should be placed. Point placement places one shield on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

shield-avoid-edges

Type: boolean

Default Value: false (*Shields will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing shields that intersect with tile boundaries.

shield-allow-overlap

Type: boolean

Default Value: false (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a shield collides with any other text, shield, or marker.*)

Minimum distance that a shield can be placed from any other text, shield, or marker.

shield-repeat-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance between repeated shields. If set this will prevent shields being rendered nearby each other that contain the same text. Similar to shield-min-distance with the difference that it works the same no matter what placement strategy is used.

shield-min-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance to the next shield with the same text. Only works for line placement.

shield-spacing

Type: float

Default Value: 0 (*Only one shield per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated shields on a line.

shield-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a shield is nearby a tile boundary.*)

Minimum distance a shield will be placed from the edge of a tile. This option is similar to shield-avoid-edges:true except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

shield-label-position-tolerance

Type: float

Default Value: shield-spacing/2.0 (*If a shield cannot be placed then the renderer will advance by shield-spacing/2.0 to try placement again.*)

Allows the shield to be displaced from its ideal position by a number of pixels (only works with placement:line).

shield-wrap-width

Type: `unsigned`

Default Value: 0 (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

shield-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

shield-wrap-character

Type: `string`

Default Value: " " (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `white` (*The shield halo text will be rendered white.*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: 0 (*no halo.*)

Specify the radius of the halo in pixels.

shield-halo-rasterizer

Type: keyword

Possible values: full fast

Default Value: full (*The shield will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

shield-halo-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: (*No transformation.*)

Transform shield halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

shield-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield-halo-opacity

Type: float

Default Value: 1 (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text halo.

shield-character-spacing

Type: unsigned

Default Value: 0 (*The default character spacing of the font will be used.*)

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0 (*The default font spacing will be used.*)

Vertical spacing between lines of multiline labels (in pixels).

shield-text-dx

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the shield right.

shield-text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the shield down.

shield-dx

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right.

shield-dy

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down.

shield-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the image used for the shield.

shield-text-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the text placed on top of the shield.

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto (*TODO.*)

The shield's horizontal alignment from its centerpoint.

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle (*TODO.*)

The shield's vertical alignment from its centerpoint.

shield-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size shield to avoid overlaps. "simple" for basic algorithm (using shield-placements string,) "dummy" to turn this feature off.

shield-placements

Type: string

Default Value: *(No alternative placements will be used.)*

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is shield-placements: "E,NE,SE,W,NW,SW";.

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none *(No text transformation will be applied.)*

Transform the case of the characters.

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto *(TODO.)*

Define how text in a shield’s label is justified.

shield-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none *(No transformation.)*

Transform shield instance with specified function. Ignores map scale factor.

shield-clip

Type: boolean

Default Value: false *(The geometry will not be clipped to map bounds before rendering.)*

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

shield-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for shield placement by the given tolerance.

shield-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for shield placement by the given algorithm.

shield-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for shield placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a line pattern symbolizer rule or emitting it with default values.

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line. Accepted formats: svg, jpg, png, tiff, and webp.

line-pattern-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

line-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

line-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

line-pattern-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-offset

Type: float

Default Value: 0 (*The line will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-pattern-geometry-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function and apply pattern to transformed geometry.

line-pattern-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*No transformation.*)

Transform line pattern instance with specified function.

line-pattern-comp-op

Type: keyword

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a polygon pattern symbolizer rule or emitting it with default values.

polygon-pattern-file

Type: uri

Default Value: none

Image to use as a repeated pattern fill within a polygon. Accepted formats: svg, jpg, png, tiff, and webp.

polygon-pattern-alignment

Type: keyword

Possible values: global local

Default Value: global (*Patterns will be aligned to the map (or tile boundaries) when being repeated across polygons. This is ideal for seamless patterns in tiled rendering.*)

Specify whether to align pattern fills to the layer's geometry (local) or to the map (global).

polygon-pattern-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon pattern edges.

polygon-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

polygon-pattern-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-pattern-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

polygon-pattern-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

polygon-pattern-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function and apply pattern to transformed geometry.

polygon-pattern-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform polygon pattern instance with specified function.

polygon-pattern-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster

Type: `keyword`

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a raster symbolizer rule or emitting it with default values.

raster-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessel mitchell sinc lanczos blackman

Default Value: near (*Nearest neighbor resampling will be used to scale the image to the target size of the map.*)

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image.*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster-colorizer-default-mode

Type: keyword

Possible values: `discrete` `linear` `exact`

Default Value: `linear` (*A linear interpolation is used to generate colors between the two nearest stops.*)

This can be either `discrete`, `linear` or `exact`. If it is not specified then the default is `linear`.

raster-colorizer-default-color

Type: `color`

Default Value: `transparent` (*Pixels that are not colored by the colorizer stops will be transparent.*)

This can be any color. Sets the color that is applied to all values outside of the range of the colorizer-stops. If not supplied pixels will be fully transparent.

raster-colorizer-epsilon

Type: `float`

Default Value: `1.1920928955078125e-07` (*Pixels must very closely match the stop filter otherwise they will not be colored.*)

This can be any positive floating point value and will be used as a tolerance in floating point comparisons. The higher the value the more likely a stop will match and color data.

raster-colorizer-stops

Type: `tags`

Default Value: (*No colorization will happen without supplying stops.*)

Assigns raster data values to colors. Stops must be listed in ascending order, and contain at a minimum the value and the associated color. You can also include the color-mode as a third argument, like `stop(100, #fff, exact)`.

point

point

Type: keyword

Status: **unstable**

Possible values: `auto` `none`

Default Value:

Allows omitting a point symbolizer rule or emitting it with default values.

point-file

Type: `uri`

Default Value: `none` (*A 4x4 black square will be rendered.*)

Image file to represent a point. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

point-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Control whether the placement of the feature will prevent the placement of other features.

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A value from 0 to 1 to control the opacity of the point.

point-placement

Type: `keyword`

Possible values: `centroid` `interior`

Default Value: `centroid` (*The centroid of the geometry will be used to place the point.*)

Control how this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation.*)

Transform point instance with specified function. Ignores map scale factor.

point-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a text symbolizer rule.

text-name

Type: string

Default Value: none

Value to use for a text label. Data columns are specified using brackets like [column_name].

text-face-name

Type: `string`

Default Value: `none`

Font name and style to render a label in.

text-size

Type: `float`

Default Value: `10` (*Font size of 10 will be used to render text.*)

Text size in pixels.

text-ratio

Type: `unsigned`

Default Value: `0` (*TODO.*)

Define the amount of text (of the total) present on successive lines when wrapping occurs.

text-wrap-width

Type: `unsigned`

Default Value: `0` (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

text-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

text-wrap-character

Type: `string`

Default Value: `" "` (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long text.

text-repeat-wrap-character

Type: `boolean`

Status: **unstable**

Default Value: `false` (*Character will be removed when used to wrap a line.*)

Keep the character used to wrap a line instead of removing it, and repeat it on the new line.

text-spacing

Type: `unsigned`

Default Value: `0` (*Only one label per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated text labels on a line.

text-character-spacing

Type: `float`

Default Value: `0` (*The default character spacing of the font will be used.*)

Horizontal spacing adjustment between characters in pixels. This value is ignored when `horizontal-alignment` is set to `adjust`. Typographic ligatures are turned off when this value is greater than zero.

text-line-spacing

Type: `float`

Default Value: `0` (*The default font spacing will be used.*)

Vertical spacing adjustment between lines in pixels.

text-label-position-tolerance

Type: `float`

Default Value: `text-spacing/2.0` (*If a shield cannot be placed then the renderer will advance by `text-spacing/2.0` to try placement again.*)

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

text-max-char-angle-delta

Type: `float`

Default Value: `22.5` (*The label will not be placed if a character falls on a line with an angle sharper than 22.5 degrees.*)

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 * \text{math.pi} / 180.0$. The higher the value the fewer labels will be placed around around sharp corners.

text-fill

Type: `color`

Default Value: `black` (*The text will be rendered black.*)

Specifies the color for the text.

text-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text.

text-halo-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text halo.

text-halo-fill

Type: `color`

Default Value: `white` (*The halo will be rendered white.*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: `0` (*no halo.*)

Specify the radius of the halo in pixels.

text-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full` (*The text will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

text-halo-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation.*)

Transform text halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

text-dx

Type: `float`

Default Value: `0` (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the X axis. With “dummy” placement-type, a positive value displaces to the right. With “simple” placement-type, it is either left, right or unchanged, depending on the placement selected. Any non-zero value implies “horizontal-alignment” changes to “left” by default. Has no effect with ‘line’ text-placement-type.

text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the Y axis. With “dummy” placement-type, a positive value displaces downwards. With “simple” placement-type, it is either up, down or unchanged, depending on the placement selected. With “line” placement-type, a positive value displaces above the path.

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; “top” for dy>0, “bottom” for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false (*Text will be potentially placed near tile edges and therefore may look cut off unless the same text label is rendered on each adjacent tile.*)

Avoid placing labels that intersect with tile boundaries.

text-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a label collides with any other text, shield, or marker.*)

Minimum distance that a label can be placed from any other text, shield, or marker.

text-repeat-distance

Type: float

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance between repeated text. If set this will prevent text labels being rendered nearby each other that contain the same text. Similar to text-min-distance with the difference that it works the same no matter what placement strategy is used.

text-min-distance

Type: float

Status: deprecated

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance to the next label with the same text. Only works for line placement. Deprecated: replaced by `text-repeat-distance` and `text-margin`

text-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a text label is nearby a tile boundary.*)

Minimum distance a text label will be placed from the edge of a tile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

text-min-path-length

Type: float

Default Value: 0 (*place labels on all geometries no matter how small they are.*)

Place labels only on polygons and lines with a bounding width longer than this value (in pixels).

text-allow-overlap

Type: boolean

Default Value: false (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: float

Default Value: 0 (*Text is not rotated and is displayed upright.*)

Rotate the text. (only works with `text-placement:point`).

text-rotate-displacement

Type: boolean

Default Value: false (*Label center is used for rotation.*)

Rotates the displacement around the placement origin by the angle given by “orientation”.

text-upright

Type: keyword

Possible values: auto auto-down left right left-only right-only

Default Value: auto (*Text will be positioned upright automatically.*)

How this label should be placed along lines. By default when more than half of a label’s characters are upside down the label is automatically flipped to keep it upright. By changing this parameter you can prevent this “auto-upright” behavior. The “auto-down” value places text in the opposite orientation to “auto”. The “left” or “right” settings can be used to force text to always be placed along a line in a given direction and therefore disables flipping if text appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down text rather than trying to flip it.

text-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this label should be placed. Point placement places one label on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

text-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: string

Default Value: *(No alternative placements will be used.)*

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements: "E, NE, SE, W, NW, SW";`.

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none *(Transform text instance with specified function. Ignores map scale factor.)*

Transform the case of the characters.

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto adjust

Default Value: auto *(TODO.)*

The text’s horizontal alignment from it’s centerpoint. If `placement` is set to `line`, then `adjust` can be set to `auto-fit` the text to the length of the path by dynamically calculating `character-spacing`.

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto *(Auto alignment means that text will be centered by default except when using the “placement-type” parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the “text-placements” directives.)*

Define how text is justified.

text-clip

Type: boolean

Default Value: false *(The geometry will not be clipped to map bounds before rendering.)*

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

text-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for text placement by the given tolerance.

text-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for text placement by the given algorithm.

text-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for text placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-font-feature-settings

Type: `string`

Default Value: *(Default set of typographic features recommended by OpenType specification. Ligatures are turned off by default when “character-spacing” is greater than zero.)*

Comma separated list of OpenType typographic features. The syntax and semantics conforms to `font-feature-settings` from W3C CSS.

text-largest-bbox-only

Type: `boolean`

Status: experimental

Default Value: `true` *(For multipolygons only polygon with largest bbox area is labeled (does not apply to other geometries).)*

Controls default labeling behavior on multipolygons. The default is `true` and means that only the polygon with largest bbox is labeled.

building

building

Type: `keyword`

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a building symbolizer rule or emitting it with default values.

building-fill

Type: `color`

Default Value: The color gray will be used for fill. *(Gray and fully opaque (alpha = 1), same as `rgb(128,128,128)` or `rgba(128,128,128,1)`.)*

The color of the buildings fill. Note: 0.8 will be used to multiply each color component to auto-generate a darkened wall color.

building-fill-opacity

Type: `float`

Default Value: 1 (*Color is fully opaque.*)

The opacity of the building as a whole, including all walls.

building-height

Type: `float`

Default Value: 0 (*Buildings will not have a visual height and will instead look like flat polygons.*)

The height of the building in pixels.

debug

debug-mode

Type: `keyword`

Possible values: `collision` `vertex`

Default Value: `collision` (*The otherwise invisible collision boxes will be rendered as squares on the map.*)

The mode for debug rendering.

dot

dot

Type: `keyword`

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a dot symbolizer rule or emitting it with default values.

dot-fill

Type: `color`

Default Value: `gray` (*The dot fill color is gray.*)

The color of the area of the dot.

dot-opacity

Type: float

Default Value: 1 (*The opacity of the dot.*)

The overall opacity of the dot.

dot-width

Type: float

Default Value: 1 (*The marker width is 1 pixel.*)

The width of the dot in pixels.

dot-height

Type: float

Default Value: 1 (*The marker height is 1 pixels.*)

The height of the dot in pixels.

dot-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current layer on top of other layers.*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

3.0.3

Style

image-filters

Type: functions

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen color-blind-protanope color-blind-deuteranope color-blind-tritanope colorize-alpha color-to-alpha scale-hsla`

Default Value: `none` (*no filters*)

A list of image filters that will be applied to the active rendering canvas for a given style. The presence of one more `image-filters` will trigger a new canvas to be created before starting to render a style and then this canvas will be composited back into the main canvas after rendering all features and after all `image-filters` have been applied. See `direct-image-filters` if you want to apply a filter directly to the main canvas.

image-filters-inflate

Type: `boolean`

Default Value: `false` (*No special handling will be done and image filters that blur data will only blur up to the edge of a tile boundary*)

A property that can be set to true to enable using an inflated image internally for seamless blurring across tiles (requires buffered data).

direct-image-filters

Type: `functions`

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen color-blind-protanope color-blind-deuteranope color-blind-tritanope colorize-alpha color-to-alpha scale-hsla`

Default Value: `none` (*no filters*)

A list of image filters to apply to the main canvas (see the `image-filters` doc for how they work on a separate canvas).

comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: `float`

Default Value: 1 (*No separate buffer will be used and no alpha will be applied to the style after rendering.*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer).

Symbolizers

map

background-color

Type: `color`

Default Value: none (*Will be rendered transparent.*)

Map Background color.

background-image

Type: `uri`

Default Value: (*No background image will be used.*)

An image that is repeated below all features on a map as a background. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

background-image-comp-op

Type: `keyword`

Possible values: `clear` `src` `dst` `src-over` `dst-over` `src-in` `dst-in` `src-out` `dst-out` `src-atop` `dst-atop` `xor` `plus` `minus` `multiply` `divide` `screen` `overlay` `darken` `lighten` `color-dodge` `color-burn` `linear-dodge` `linear-burn` `hard-light` `soft-light` `difference` `exclusion` `contrast` `invert` `invert-rgb` `grain-merge` `grain-extract` `hue` `saturation` `color` `value`

Default Value: `src-over` (*The background-image will be blended with the background normally (placed on top of any existing background-color).*)

Set the compositing operation used to blend the image into the background.

background-image-opacity

Type: `float`

Default Value: 1 (*The image opacity will not be changed when applied to the map background.*)

Set the opacity of the image.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space.*)

Map spatial reference (proj4 string).

buffer-size

Type: `float`

Default Value: 0 (*No buffer will be used.*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with "avoid-edges".

maximum-extent

Type: `string`

Default Value: `-20037508.34,-20037508.34,20037508.34,20037508.34` (*All data will be clipped to global mercator extent (default is applied in Carto.js).*)

An extent to be used to limit the bounds used to query all layers during rendering. Should be minx, miny, maxx, maxy in the coordinates of the Map.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: uri

Default Value: none (*No map-specific fonts will be registered.*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a polygon symbolizer rule or emitting it with default values.

polygon-fill

Type: color

Default Value: The color gray will be used for fill. (*Gray and fully opaque (alpha = 1), same as rgb(128,128,128) or rgba(128,128,128,1).*)

Fill color to assign to a polygon.

polygon-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the polygon.

polygon-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon edges.

polygon-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \text{gamma})$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

polygon-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extend outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

polygon-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

polygon-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function.

polygon-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line

Type: `keyword`

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a line symbolizer rule or emitting it with default values.

line-color

Type: `color`

Default Value: `black` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)` or `rgba(0,0,0,1)`.)*)

The color of a drawn line.

line-width

Type: float

Default Value: 1 (*The line will be rendered 1 pixel wide.*)

The width of a line in pixels.

line-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of a line.

line-join

Type: keyword

Possible values: miter miter-revert round bevel

Default Value: miter (*The line joins will be rendered using a miter look.*)

The behavior of lines when joining.

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt (*The line endings will be rendered using a butt look.*)

The display of line endings.

line-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of stroke line.

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \gamma)$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

line-dasharray

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

Valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled).

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'.*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extend outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

line-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

line-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: `float`

Status: unstable

Default Value: `0` (*Will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full (*The line will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a marker symbolizer rule or emitting it with default values.

marker-file

Type: `uri`

Default Value: `none` (*An ellipse or circle, if width equals height.*)

A file that this marker shows at each placement. If no file is given, the marker will show an ellipse. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

marker-opacity

Type: `float`

Default Value: `1` (*The stroke-opacity and fill-opacity of the marker.*)

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The fill opacity of the marker. This property will also set the `fill-opacity` of elements in an SVG loaded from a file.

marker-line-color

Type: `color`

Default Value: `black` (*The marker will be drawn with a black outline.*)

The color of the stroke around the marker. This property will also set the `stroke` of elements in an SVG loaded from a file.

marker-line-width

Type: `float`

Default Value: `0.5` (*The marker will be drawn with an outline of .5 pixels wide.*)

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself. This property will also set the `stroke-width` of elements in an SVG loaded from a file.

marker-line-opacity

Type: float

Default Value: 1 (*Color is fully opaque. This property will also set the ‘stroke-opacity’ of elements in an SVG loaded from a file.*)

The opacity of a line.

marker-placement

Type: keyword

Possible values: point line interior vertex-first vertex-last

Default Value: point (*Place markers at the center point (centroid) of the geometry.*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. ‘interior’ placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior. The ‘vertex-first’ and ‘vertex-last’ options can be used to place markers at the first or last vertex of lines or polygons.

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each.*)

A special setting to allow the user to control rendering behavior for ‘multi-geometries’ (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The ‘each’ policy is default and means all geometries will get a marker. The ‘whole’ policy means that the aggregate centroid between all geometries will be used. The ‘largest’ policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Status: deprecated

Possible values: arrow ellipse

Default Value: ellipse (*The marker shape is an ellipse.*)

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width).

marker-width

Type: `float`

Default Value: 10 (*The marker width is 10 pixels.*)

The width of the marker, if using one of the default types.

marker-height

Type: `float`

Default Value: 10 (*The marker height is 10 pixels.*)

The height of the marker, if using one of the default types.

marker-fill

Type: `color`

Default Value: blue (*The marker fill color is blue.*)

The color of the area of the marker. This property will also set the `fill` of elements in an SVG loaded from a file.

marker-allow-overlap

Type: `boolean`

Default Value: false (*Do not allow markers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-avoid-edges

Type: `boolean`

Default Value: false (*Markers will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing markers that intersect with tile boundaries.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Value to control whether the placement of the feature will prevent the placement of other features.

marker-spacing

Type: `float`

Default Value: `100` (*In the case of marker-placement:line then draw a marker every 100 pixels along a line.*)

Space between repeated markers in pixels. If the spacing is less than the marker size or larger than the line segment length then no marker will be placed. Any value less than 1 will be ignored and the default will be used instead.

marker-max-error

Type: `float`

Default Value: `0.2` (*N/A: not intended to be changed.*)

N/A: not intended to be changed.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform marker instance with specified function. Ignores map scale factor.

marker-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

marker-simplify

Type: float

Default Value: 0 (*Geometry will not be simplified.*)

geometries are simplified by the given tolerance.

marker-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

marker-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*The geometry will not be transformed.*)

Transform marker geometry with specified function.

marker-offset

Type: float

Default Value: 0 (*Will not be offset.*)

Offsets a marker from a line a number of pixels parallel to its actual path. Positive values move the marker left, negative values move it right (relative to the directionality of the line).

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

marker-direction

Type: keyword

Possible values: auto auto-down left right left-only right-only up down

Default Value: right (*Markers are oriented to the right in the line direction.*)

How markers should be placed along lines. With the “auto” setting when marker is upside down the marker is automatically rotated by 180 degrees to keep it upright. The “auto-down” value places marker in the opposite orientation to “auto”. The “left” or “right” settings can be used to force marker to always be placed along a line in a given direction and therefore disables rotating if marker appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down markers rather than trying to flip it. The “up” and “down” settings don’t adjust marker’s orientation to the line direction.

shield

shield

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a shield symbolizer rule.

shield-name

Type: string

Default Value: (*No text label will be rendered with the shield.*)

Value to use for a shield’s text label. Data columns are specified using brackets like [column_name].

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text. Accepted formats: svg, jpg, png, tiff, and webp.

shield-face-name

Type: string

Default Value: none

Font name and style to use for the shield text.

shield-unlock-image

Type: boolean

Default Value: false (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image.

shield-size

Type: float

Default Value: 10 (*Font size of 10 will be used to render text.*)

The size of the shield text in pixels.

shield-fill

Type: color

Default Value: black (*The shield text will be rendered black.*)

The color of the shield text.

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this shield should be placed. Point placement places one shield on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

shield-avoid-edges

Type: boolean

Default Value: false (*Shields will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing shields that intersect with tile boundaries.

shield-allow-overlap

Type: boolean

Default Value: false (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a shield collides with any other text, shield, or marker.*)

Minimum distance that a shield can be placed from any other text, shield, or marker.

shield-repeat-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance between repeated shields. If set this will prevent shields being rendered nearby each other that contain the same text. Similar to shield-min-distance with the difference that it works the same no matter what placement strategy is used.

shield-min-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance to the next shield with the same text. Only works for line placement.

shield-spacing

Type: float

Default Value: 0 (*Only one shield per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated shields on a line.

shield-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a shield is nearby a tile boundary.*)

Minimum distance a shield will be placed from the edge of a tile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

shield-label-position-tolerance

Type: float

Default Value: `shield-spacing/2.0` (*If a shield cannot be placed then the renderer will advance by `shield-spacing/2.0` to try placement again.*)

Allows the shield to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

shield-wrap-width

Type: unsigned

Default Value: 0 (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

shield-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

shield-wrap-character

Type: `string`

Default Value: `" "` (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `white` (*The shield halo text will be rendered white.*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: `0` (*no halo.*)

Specify the radius of the halo in pixels.

shield-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full` (*The shield will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

shield-halo-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: *(No transformation.)*

Transform shield halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

shield-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over *(Add the current symbolizer on top of other symbolizer.)*

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield-halo-opacity

Type: float

Default Value: 1 *(Fully opaque.)*

A number from 0 to 1 specifying the opacity for the text halo.

shield-character-spacing

Type: unsigned

Default Value: 0 *(The default character spacing of the font will be used.)*

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0 *(The default font spacing will be used.)*

Vertical spacing between lines of multiline labels (in pixels).

shield-text-dx

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the shield right.

shield-text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the shield down.

shield-dx

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right.

shield-dy

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down.

shield-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the image used for the shield.

shield-text-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the text placed on top of the shield.

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto (*TODO.*)

The shield's horizontal alignment from its centerpoint.

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle (*TODO.*)

The shield's vertical alignment from its centerpoint.

shield-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size shield to avoid overlaps. "simple" for basic algorithm (using shield-placements string,) "dummy" to turn this feature off.

shield-placements

Type: string

Default Value: (*No alternative placements will be used.*)

If "placement-type" is set to "simple", use this "POSITIONS,[SIZES]" string. An example is
shield-placements: "E,NE,SE,W,NW,SW";.

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none (*No text transformation will be applied.*)

Transform the case of the characters.

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto (*TODO.*)

Define how text in a shield's label is justified.

shield-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation.*)

Transform shield instance with specified function. Ignores map scale factor.

shield-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

shield-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for shield placement by the given tolerance.

shield-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for shield placement by the given algorithm.

shield-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for shield placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a line pattern symbolizer rule.

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line. Accepted formats: svg, jpg, png, tiff, and webp.

line-pattern-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

line-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

line-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

line-pattern-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-offset

Type: float

Default Value: 0 (*The line will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-pattern-geometry-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function and apply pattern to transformed geometry.

line-pattern-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*No transformation.*)

Transform line pattern instance with specified function.

line-pattern-comp-op

Type: keyword

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a polygon pattern symbolizer rule or emitting it with default values.

polygon-pattern-file

Type: uri

Default Value: none

Image to use as a repeated pattern fill within a polygon. Accepted formats: svg, jpg, png, tiff, and webp.

polygon-pattern-alignment

Type: keyword

Possible values: global local

Default Value: global (*Patterns will be aligned to the map (or tile boundaries) when being repeated across polygons. This is ideal for seamless patterns in tiled rendering.*)

Specify whether to align pattern fills to the layer's geometry (local) or to the map (global).

polygon-pattern-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon pattern edges.

polygon-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

polygon-pattern-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-pattern-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

polygon-pattern-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

polygon-pattern-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function and apply pattern to transformed geometry.

polygon-pattern-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform polygon pattern instance with specified function.

polygon-pattern-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster

Type: `keyword`

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a raster symbolizer rule or emitting it with default values.

raster-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessel mitchell sinc lanczos blackman

Default Value: near (*Nearest neighbor resampling will be used to scale the image to the target size of the map.*)

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image.*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster-colorizer-default-mode

Type: keyword

Possible values: discrete linear exact

Default Value: linear (*A linear interpolation is used to generate colors between the two nearest stops.*)

This can be either discrete, linear or exact. If it is not specified then the default is linear.

raster-colorizer-default-color

Type: color

Default Value: transparent (*Pixels that are not colored by the colorizer stops will be transparent.*)

This can be any color. Sets the color that is applied to all values outside of the range of the colorizer-stops. If not supplied pixels will be fully transparent.

raster-colorizer-epsilon

Type: float

Default Value: 1.1920928955078125e-07 (*Pixels must very closely match the stop filter otherwise they will not be colored.*)

This can be any positive floating point value and will be used as a tolerance in floating point comparisons. The higher the value the more likely a stop will match and color data.

raster-colorizer-stops

Type: tags

Default Value: (*No colorization will happen without supplying stops.*)

Assigns raster data values to colors. Stops must be listed in ascending order, and contain at a minimum the value and the associated color. You can also include the color-mode as a third argument, like `stop(100, #fff, exact)`.

point

point

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a point symbolizer rule or emitting it with default values.

point-file

Type: `uri`

Default Value: `none` (*A 4x4 black square will be rendered.*)

Image file to represent a point. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

point-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Control whether the placement of the feature will prevent the placement of other features.

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A value from 0 to 1 to control the opacity of the point.

point-placement

Type: `keyword`

Possible values: `centroid` `interior`

Default Value: `centroid` (*The centroid of the geometry will be used to place the point.*)

Control how this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation.*)

Transform point instance with specified function. Ignores map scale factor.

point-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a text symbolizer rule or emitting it with default values.

text-name

Type: string

Default Value: none

Value to use for a text label. Data columns are specified using brackets like [column_name].

text-face-name

Type: `string`

Default Value: `none`

Font name and style to render a label in.

text-size

Type: `float`

Default Value: `10` (*Font size of 10 will be used to render text.*)

Text size in pixels.

text-ratio

Type: `unsigned`

Default Value: `0` (*TODO.*)

Define the amount of text (of the total) present on successive lines when wrapping occurs.

text-wrap-width

Type: `unsigned`

Default Value: `0` (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

text-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

text-wrap-character

Type: `string`

Default Value: `" "` (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long text.

text-repeat-wrap-character

Type: `boolean`

Status: unstable

Default Value: `false` (*Character will be removed when used to wrap a line.*)

Keep the character used to wrap a line instead of removing it, and repeat it on the new line.

text-spacing

Type: `unsigned`

Default Value: `0` (*Only one label per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated text labels on a line.

text-character-spacing

Type: `float`

Default Value: `0` (*The default character spacing of the font will be used.*)

Horizontal spacing adjustment between characters in pixels. This value is ignored when `horizontal-alignment` is set to `adjust`. Typographic ligatures are turned off when this value is greater than zero.

text-line-spacing

Type: `float`

Default Value: `0` (*The default font spacing will be used.*)

Vertical spacing adjustment between lines in pixels.

text-label-position-tolerance

Type: `float`

Default Value: `text-spacing/2.0` (*If a shield cannot be placed then the renderer will advance by `text-spacing/2.0` to try placement again.*)

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

text-max-char-angle-delta

Type: `float`

Default Value: `22.5` (*The label will not be placed if a character falls on a line with an angle sharper than 22.5 degrees.*)

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 * \text{math.pi} / 180.0$. The higher the value the fewer labels will be placed around around sharp corners.

text-fill

Type: `color`

Default Value: `black` (*The text will be rendered black.*)

Specifies the color for the text.

text-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text.

text-halo-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text halo.

text-halo-fill

Type: `color`

Default Value: `white` (*The halo will be rendered white.*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: `0` (*no halo.*)

Specify the radius of the halo in pixels.

text-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full` (*The text will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

text-halo-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation.*)

Transform text halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

text-dx

Type: `float`

Default Value: `0` (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the X axis. With “dummy” placement-type, a positive value displaces to the right. With “simple” placement-type, it is either left, right or unchanged, depending on the placement selected. Any non-zero value implies “horizontal-alignment” changes to “left” by default. Has no effect with ‘line’ text-placement-type.

text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the Y axis. With “dummy” placement-type, a positive value displaces downwards. With “simple” placement-type, it is either up, down or unchanged, depending on the placement selected. With “line” placement-type, a positive value displaces above the path.

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; “top” for dy>0, “bottom” for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false (*Text will be potentially placed near tile edges and therefore may look cut off unless the same text label is rendered on each adjacent tile.*)

Avoid placing labels that intersect with tile boundaries.

text-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a label collides with any other text, shield, or marker.*)

Minimum distance that a label can be placed from any other text, shield, or marker.

text-repeat-distance

Type: float

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance between repeated text. If set this will prevent text labels being rendered nearby each other that contain the same text. Similar to text-min-distance with the difference that it works the same no matter what placement strategy is used.

text-min-distance

Type: float

Status: deprecated

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance to the next label with the same text. Only works for line placement. Deprecated: replaced by `text-repeat-distance` and `text-margin`

text-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a text label is nearby a tile boundary.*)

Minimum distance a text label will be placed from the edge of a tile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

text-min-path-length

Type: float

Default Value: 0 (*place labels on all geometries no matter how small they are.*)

Place labels only on polygons and lines with a bounding width longer than this value (in pixels).

text-allow-overlap

Type: boolean

Default Value: false (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: float

Default Value: 0 (*Text is not rotated and is displayed upright.*)

Rotate the text. (only works with `text-placement:point`).

text-rotate-displacement

Type: boolean

Default Value: false (*Label center is used for rotation.*)

Rotates the displacement around the placement origin by the angle given by “orientation”.

text-upright

Type: keyword

Possible values: auto auto-down left right left-only right-only

Default Value: auto (*Text will be positioned upright automatically.*)

How this label should be placed along lines. By default when more than half of a label’s characters are upside down the label is automatically flipped to keep it upright. By changing this parameter you can prevent this “auto-upright” behavior. The “auto-down” value places text in the opposite orientation to “auto”. The “left” or “right” settings can be used to force text to always be placed along a line in a given direction and therefore disables flipping if text appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down text rather than trying to flip it.

text-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this label should be placed. Point placement places one label on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

text-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: string

Default Value: *(No alternative placements will be used.)*

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements: "E, NE, SE, W, NW, SW";`.

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none *(Transform text instance with specified function. Ignores map scale factor.)*

Transform the case of the characters.

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto adjust

Default Value: auto *(TODO.)*

The text’s horizontal alignment from it’s centerpoint. If `placement` is set to `line`, then `adjust` can be set to `auto-fit` the text to the length of the path by dynamically calculating `character-spacing`.

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto *(Auto alignment means that text will be centered by default except when using the “placement-type” parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the “text-placements” directives.)*

Define how text is justified.

text-clip

Type: boolean

Default Value: false *(The geometry will not be clipped to map bounds before rendering.)*

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

text-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for text placement by the given tolerance.

text-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for text placement by the given algorithm.

text-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for text placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-font-feature-settings

Type: `string`

Default Value: *(Default set of typographic features recommended by OpenType specification. Ligatures are turned off by default when “character-spacing” is greater than zero.)*

Comma separated list of OpenType typographic features. The syntax and semantics conforms to `font-feature-settings` from W3C CSS.

text-largest-bbox-only

Type: `boolean`

Status: experimental

Default Value: `true` *(For multipolygons only polygon with largest bbox area is labeled (does not apply to other geometries).)*

Controls default labeling behavior on multipolygons. The default is `true` and means that only the polygon with largest bbox is labeled.

building

building

Type: `keyword`

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a building symbolizer rule or emitting it with default values.

building-fill

Type: `color`

Default Value: The color gray will be used for fill. *(Gray and fully opaque (alpha = 1), same as `rgb(128,128,128)` or `rgba(128,128,128,1)`.)*

The color of the buildings fill. Note: 0.8 will be used to multiply each color component to auto-generate a darkened wall color.

building-fill-opacity

Type: `float`

Default Value: 1 (*Color is fully opaque.*)

The opacity of the building as a whole, including all walls.

building-height

Type: `float`

Default Value: 0 (*Buildings will not have a visual height and will instead look like flat polygons.*)

The height of the building in pixels.

debug

debug-mode

Type: `keyword`

Possible values: `collision` `vertex`

Default Value: `collision` (*The otherwise invisible collision boxes will be rendered as squares on the map.*)

The mode for debug rendering.

dot

dot

Type: `keyword`

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a dot symbolizer rule or emitting it with default values.

dot-fill

Type: `color`

Default Value: `gray` (*The dot fill color is gray.*)

The color of the area of the dot.

dot-opacity

Type: float

Default Value: 1 (*The opacity of the dot.*)

The overall opacity of the dot.

dot-width

Type: float

Default Value: 1 (*The marker width is 1 pixel.*)

The width of the dot in pixels.

dot-height

Type: float

Default Value: 1 (*The marker height is 1 pixels.*)

The height of the dot in pixels.

dot-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current layer on top of other layers.*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

3.0.0

Style

image-filters

Type: functions

Possible values: agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha color-to-alpha scale-hsla

Default Value: none (*no filters*)

A list of image filters that will be applied to the active rendering canvas for a given style. The presence of one more `image-filters` will trigger a new canvas to be created before starting to render a style and then this canvas will be composited back into the main canvas after rendering all features and after all `image-filters` have been applied. See `direct-image-filters` if you want to apply a filter directly to the main canvas.

image-filters-inflate

Type: boolean

Default Value: false (*No special handling will be done and image filters that blur data will only blur up to the edge of a tile boundary*)

A property that can be set to true to enable using an inflated image internally for seamless blurring across tiles (requires buffered data).

direct-image-filters

Type: functions

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha color-to-alpha scale-hsla`

Default Value: none (*no filters*)

A list of image filters to apply to the main canvas (see the `image-filters` doc for how they work on a separate canvas).

comp-op

Type: keyword

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: float

Default Value: 1 (*No separate buffer will be used and no alpha will be applied to the style after rendering.*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer).

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*Will be rendered transparent.*)

Map Background color.

background-image

Type: `uri`

Default Value: (*No background image will be used.*)

An image that is repeated below all features on a map as a background. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

background-image-comp-op

Type: `keyword`

Possible values: `clear` `src` `dst` `src-over` `dst-over` `src-in` `dst-in` `src-out` `dst-out` `src-atop` `dst-atop` `xor` `plus` `minus` `multiply` `divide` `screen` `overlay` `darken` `lighten` `color-dodge` `color-burn` `linear-dodge` `linear-burn` `hard-light` `soft-light` `difference` `exclusion` `contrast` `invert` `invert-rgb` `grain-merge` `grain-extract` `hue` `saturation` `color` `value`

Default Value: `src-over` (*The background-image will be blended with the background normally (placed on top of any existing background-color).*)

Set the compositing operation used to blend the image into the background.

background-image-opacity

Type: `float`

Default Value: `1` (*The image opacity will not be changed when applied to the map background.*)

Set the opacity of the image.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space.*)

Map spatial reference (proj4 string).

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used.*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

maximum-extent

Type: `string`

Default Value: `-20037508.34,-20037508.34,20037508.34,20037508.34` (*All data will be clipped to global mercator extent (default is applied in Carto.js).*)

An extent to be used to limit the bounds used to query all layers during rendering. Should be `minx, miny, maxx, maxy` in the coordinates of the Map.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: uri

Default Value: none (*No map-specific fonts will be registered.*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a polygon symbolizer rule or emitting it with default values.

polygon-fill

Type: color

Default Value: The color gray will be used for fill. (*Gray and fully opaque (alpha = 1), same as rgb(128,128,128) or rgba(128,128,128,1).*)

Fill color to assign to a polygon.

polygon-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the polygon.

polygon-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon edges.

polygon-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \text{gamma})$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

polygon-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extend outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

polygon-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

polygon-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function.

polygon-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line

Type: `keyword`

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a line symbolizer rule or emitting it with default values.

line-color

Type: `color`

Default Value: `black` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)` or `rgba(0,0,0,1)`.)*)

The color of a drawn line.

line-width

Type: float

Default Value: 1 (*The line will be rendered 1 pixel wide.*)

The width of a line in pixels.

line-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of a line.

line-join

Type: keyword

Possible values: miter miter-revert round bevel

Default Value: miter (*The line joins will be rendered using a miter look.*)

The behavior of lines when joining.

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt (*The line endings will be rendered using a butt look.*)

The display of line endings.

line-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of stroke line.

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \gamma)$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA.)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

line-dasharray

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*The line will be drawn without dashes.*)

Valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled).

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'.*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extend outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

Simplify geometries by the given tolerance.

line-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

Simplify geometries by the given algorithm.

line-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: `float`

Status: unstable

Default Value: `0` (*Will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full (*The line will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a marker symbolizer rule or emitting it with default values.

marker-file

Type: `uri`

Default Value: `none` (*An ellipse or circle, if width equals height.*)

A file that this marker shows at each placement. If no file is given, the marker will show an ellipse. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

marker-opacity

Type: `float`

Default Value: `1` (*The stroke-opacity and fill-opacity of the marker.*)

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The fill opacity of the marker. This property will also set the `fill-opacity` of elements in an SVG loaded from a file.

marker-line-color

Type: `color`

Default Value: `black` (*The marker will be drawn with a black outline.*)

The color of the stroke around the marker. This property will also set the `stroke` of elements in an SVG loaded from a file.

marker-line-width

Type: `float`

Default Value: `0.5` (*The marker will be drawn with an outline of .5 pixels wide.*)

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself. This property will also set the `stroke-width` of elements in an SVG loaded from a file.

marker-line-opacity

Type: float

Default Value: 1 (*Color is fully opaque. This property will also set the ‘stroke-opacity’ of elements in an SVG loaded from a file.*)

The opacity of a line.

marker-placement

Type: keyword

Possible values: point line interior vertex-first vertex-last

Default Value: point (*Place markers at the center point (centroid) of the geometry.*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. ‘interior’ placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior. The ‘vertex-first’ and ‘vertex-last’ options can be used to place markers at the first or last vertex of lines or polygons.

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each.*)

A special setting to allow the user to control rendering behavior for ‘multi-geometries’ (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The ‘each’ policy is default and means all geometries will get a marker. The ‘whole’ policy means that the aggregate centroid between all geometries will be used. The ‘largest’ policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Status: deprecated

Possible values: arrow ellipse

Default Value: ellipse (*The marker shape is an ellipse.*)

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width).

marker-width

Type: `float`

Default Value: 10 (*The marker width is 10 pixels.*)

The width of the marker, if using one of the default types.

marker-height

Type: `float`

Default Value: 10 (*The marker height is 10 pixels.*)

The height of the marker, if using one of the default types.

marker-fill

Type: `color`

Default Value: blue (*The marker fill color is blue.*)

The color of the area of the marker. This property will also set the `fill` of elements in an SVG loaded from a file.

marker-allow-overlap

Type: `boolean`

Default Value: false (*Do not allow markers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-avoid-edges

Type: `boolean`

Default Value: false (*Markers will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing markers that intersect with tile boundaries.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Value to control whether the placement of the feature will prevent the placement of other features.

marker-spacing

Type: `float`

Default Value: `100` (*In the case of marker-placement:line then draw a marker every 100 pixels along a line.*)

Space between repeated markers in pixels. If the spacing is less than the marker size or larger than the line segment length then no marker will be placed. Any value less than 1 will be ignored and the default will be used instead.

marker-max-error

Type: `float`

Default Value: `0.2` (*N/A: not intended to be changed.*)

N/A: not intended to be changed.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform marker instance with specified function. Ignores map scale factor.

marker-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

marker-simplify

Type: float

Default Value: 0 (*Geometry will not be simplified.*)

geometries are simplified by the given tolerance.

marker-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

marker-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*The geometry will not be transformed.*)

Transform marker geometry with specified function.

marker-offset

Type: float

Default Value: 0 (*Will not be offset.*)

Offsets a marker from a line a number of pixels parallel to its actual path. Positive values move the marker left, negative values move it right (relative to the directionality of the line).

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

marker-direction

Type: keyword

Possible values: auto auto-down left right left-only right-only up down

Default Value: right (*Markers are oriented to the right in the line direction.*)

How markers should be placed along lines. With the “auto” setting when marker is upside down the marker is automatically rotated by 180 degrees to keep it upright. The “auto-down” value places marker in the opposite orientation to “auto”. The “left” or “right” settings can be used to force marker to always be placed along a line in a given direction and therefore disables rotating if marker appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down markers rather than trying to flip it. The “up” and “down” settings don’t adjust marker’s orientation to the line direction.

shield

shield

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a shield symbolizer.

shield-name

Type: string

Default Value: (*No text label will be rendered with the shield.*)

Value to use for a shield’s text label. Data columns are specified using brackets like [column_name].

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text. Accepted formats: svg, jpg, png, tiff, and webp.

shield-face-name

Type: string

Default Value: none

Font name and style to use for the shield text.

shield-unlock-image

Type: boolean

Default Value: false (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image.

shield-size

Type: float

Default Value: 10 (*Font size of 10 will be used to render text.*)

The size of the shield text in pixels.

shield-fill

Type: color

Default Value: black (*The shield text will be rendered black.*)

The color of the shield text.

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this shield should be placed. Point placement places one shield on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

shield-avoid-edges

Type: boolean

Default Value: false (*Shields will be potentially placed near tile edges and therefore may look cut off unless they are rendered on each adjacent tile.*)

Avoid placing shields that intersect with tile boundaries.

shield-allow-overlap

Type: boolean

Default Value: false (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a shield collides with any other text, shield, or marker.*)

Minimum distance that a shield can be placed from any other text, shield, or marker.

shield-repeat-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance between repeated shields. If set this will prevent shields being rendered nearby each other that contain the same text. Similar to shield-min-distance with the difference that it works the same no matter what placement strategy is used.

shield-min-distance

Type: float

Default Value: 0 (*Shields with the same text will be rendered without restriction.*)

Minimum distance to the next shield with the same text. Only works for line placement.

shield-spacing

Type: float

Default Value: 0 (*Only one shield per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated shields on a line.

shield-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a shield is nearby a tile boundary.*)

Minimum distance a shield will be placed from the edge of a tile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

shield-label-position-tolerance

Type: float

Default Value: `shield-spacing/2.0` (*If a shield cannot be placed then the renderer will advance by `shield-spacing/2.0` to try placement again.*)

Allows the shield to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

shield-wrap-width

Type: unsigned

Default Value: 0 (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

shield-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

shield-wrap-character

Type: `string`

Default Value: `" "` (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `white` (*The shield halo text will be rendered white.*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: `0` (*no halo.*)

Specify the radius of the halo in pixels.

shield-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full` (*The shield will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

shield-halo-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: *(No transformation.)*

Transform shield halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

shield-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over *(Add the current symbolizer on top of other symbolizer.)*

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield-halo-opacity

Type: float

Default Value: 1 *(Fully opaque.)*

A number from 0 to 1 specifying the opacity for the text halo.

shield-character-spacing

Type: unsigned

Default Value: 0 *(The default character spacing of the font will be used.)*

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0 *(The default font spacing will be used.)*

Vertical spacing between lines of multiline labels (in pixels).

shield-text-dx

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the shield right.

shield-text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the shield down.

shield-dx

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right.

shield-dy

Type: float

Default Value: 0 (*Shield will not be displaced.*)

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down.

shield-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the image used for the shield.

shield-text-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the text placed on top of the shield.

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto (*TODO.*)

The shield's horizontal alignment from its centerpoint.

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle (*TODO.*)

The shield's vertical alignment from its centerpoint.

shield-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size shield to avoid overlaps. "simple" for basic algorithm (using shield-placements string,) "dummy" to turn this feature off.

shield-placements

Type: string

Default Value: (*No alternative placements will be used.*)

If "placement-type" is set to "simple", use this "POSITIONS,[SIZES]" string. An example is shield-placements: "E,NE,SE,W,NW,SW";.

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none (*No text transformation will be applied.*)

Transform the case of the characters.

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto (*TODO.*)

Define how text in a shield's label is justified.

shield-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation.*)

Transform shield instance with specified function. Ignores map scale factor.

shield-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

shield-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for shield placement by the given tolerance.

shield-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for shield placement by the given algorithm.

shield-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for shield placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a line pattern symbolizer rule or emitting it with default values.

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line. Accepted formats: svg, jpg, png, tiff, and webp.

line-pattern-clip

Type: boolean

Default Value: false (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

line-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

line-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

line-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

line-pattern-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-offset

Type: float

Default Value: 0 (*The line will not be offset.*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-pattern-geometry-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*The geometry will not be transformed.*)

Transform line geometry with specified function and apply pattern to transformed geometry.

line-pattern-transform

Type: functions

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: none (*No transformation.*)

Transform line pattern instance with specified function.

line-pattern-comp-op

Type: keyword

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a polygon pattern symbolizer rule or emitting it with default values.

polygon-pattern-file

Type: uri

Default Value: none

Image to use as a repeated pattern fill within a polygon. Accepted formats: svg, jpg, png, tiff, and webp.

polygon-pattern-alignment

Type: keyword

Possible values: global local

Default Value: global (*Patterns will be aligned to the map (or tile boundaries) when being repeated across polygons. This is ideal for seamless patterns in tiled rendering.*)

Specify whether to align pattern fills to the layer's geometry (local) or to the map (global).

polygon-pattern-gamma

Type: float

Default Value: 1 (*Fully antialiased.*) Range: 0-1 Level of antialiasing of polygon pattern edges.

polygon-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications.*)

Apply an opacity level to the image used for the pattern.

polygon-pattern-clip

Type: `boolean`

Default Value: `false` (*The geometry will not be clipped to map bounds before rendering.*)

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

polygon-pattern-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified.*)

geometries are simplified by the given tolerance.

polygon-pattern-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*The geometry will be simplified using the radial distance algorithm.*)

geometries are simplified by the given algorithm.

polygon-pattern-smooth

Type: `float`

Default Value: `0` (*No smoothing.*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*The geometry will not be transformed.*)

Transform polygon geometry with specified function and apply pattern to transformed geometry.

polygon-pattern-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation.*)

Transform polygon pattern instance with specified function.

polygon-pattern-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster

Type: `keyword`

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a raster symbolizer rule or emitting it with default values.

raster-opacity

Type: `float`

Default Value: `1` (*Color is fully opaque.*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessel mitchell sinc lanczos blackman

Default Value: near (*Nearest neighbor resampling will be used to scale the image to the target size of the map.*)

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image.*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster-colorizer-default-mode

Type: keyword

Possible values: discrete linear exact

Default Value: linear (*A linear interpolation is used to generate colors between the two nearest stops.*)

This can be either discrete, linear or exact. If it is not specified then the default is linear.

raster-colorizer-default-color

Type: color

Default Value: transparent (*Pixels that are not colored by the colorizer stops will be transparent.*)

This can be any color. Sets the color that is applied to all values outside of the range of the colorizer-stops. If not supplied pixels will be fully transparent.

raster-colorizer-epsilon

Type: float

Default Value: 1.1920928955078125e-07 (*Pixels must very closely match the stop filter otherwise they will not be colored.*)

This can be any positive floating point value and will be used as a tolerance in floating point comparisons. The higher the value the more likely a stop will match and color data.

raster-colorizer-stops

Type: tags

Default Value: (*No colorization will happen without supplying stops.*)

Assigns raster data values to colors. Stops must be listed in ascending order, and contain at a minimum the value and the associated color. You can also include the color-mode as a third argument, like `stop(100, #fff, exact)`.

point

point

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a point symbolizer rule or emitting it with default values.

point-file

Type: `uri`

Default Value: `none` (*A 4x4 black square will be rendered.*)

Image file to represent a point. Accepted formats: `svg`, `jpg`, `png`, `tiff`, and `webp`.

point-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache.*)

Control whether the placement of the feature will prevent the placement of other features.

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque.*)

A value from 0 to 1 to control the opacity of the point.

point-placement

Type: `keyword`

Possible values: `centroid` `interior`

Default Value: `centroid` (*The centroid of the geometry will be used to place the point.*)

Control how this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation.*)

Transform point instance with specified function. Ignores map scale factor.

point-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a text symbolizer rule.

text-name

Type: string

Default Value: none

Value to use for a text label. Data columns are specified using brackets like [column_name].

text-face-name

Type: `string`

Default Value: `none`

Font name and style to render a label in.

text-size

Type: `float`

Default Value: `10` (*Font size of 10 will be used to render text.*)

Text size in pixels.

text-ratio

Type: `unsigned`

Default Value: `0` (*TODO.*)

Define the amount of text (of the total) present on successive lines when wrapping occurs.

text-wrap-width

Type: `unsigned`

Default Value: `0` (*Text will not be wrapped.*)

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

text-wrap-before

Type: `boolean`

Default Value: `false` (*Wrapped lines will be a bit longer than wrap-width.*)

Wrap text before wrap-width is reached.

text-wrap-character

Type: `string`

Default Value: `”` (*Lines will be wrapped when whitespace is encountered.*)

Use this character instead of a space to wrap long text.

text-repeat-wrap-character

Type: `boolean`

Status: **unstable**

Default Value: `false` (*Character will be removed when used to wrap a line.*)

Keep the character used to wrap a line instead of removing it, and repeat it on the new line.

text-spacing

Type: `unsigned`

Default Value: `0` (*Only one label per line will attempt to be placed.*)

Distance the renderer should use to try to place repeated text labels on a line.

text-character-spacing

Type: `float`

Default Value: `0` (*The default character spacing of the font will be used.*)

Horizontal spacing adjustment between characters in pixels. This value is ignored when `horizontal-alignment` is set to `adjust`. Typographic ligatures are turned off when this value is greater than zero.

text-line-spacing

Type: `float`

Default Value: `0` (*The default font spacing will be used.*)

Vertical spacing adjustment between lines in pixels.

text-label-position-tolerance

Type: float

Default Value: `text-spacing/2.0` (*If a shield cannot be placed then the renderer will advance by `text-spacing/2.0` to try placement again.*)

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

text-max-char-angle-delta

Type: float

Default Value: `22.5` (*The label will not be placed if a character falls on a line with an angle sharper than 22.5 degrees.*)

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 * \pi / 180.0$. The higher the value the fewer labels will be placed around around sharp corners.

text-fill

Type: color

Default Value: `black` (*The text will be rendered black.*)

Specifies the color for the text.

text-opacity

Type: float

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text.

text-halo-opacity

Type: float

Default Value: `1` (*Fully opaque.*)

A number from 0 to 1 specifying the opacity for the text halo.

text-halo-fill

Type: `color`

Default Value: `white` (*The halo will be rendered white.*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: `0` (*no halo.*)

Specify the radius of the halo in pixels.

text-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full` (*The text will be rendered using the highest quality method rather than the fastest.*)

Exposes an alternate text halo rendering method that sacrifices quality for speed.

text-halo-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation.*)

Transform text halo relative to the actual text with specified function. Allows for shadow or embossed effects. Ignores map scale factor.

text-dx

Type: `float`

Default Value: `0` (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the X axis. With “dummy” placement-type, a positive value displaces to the right. With “simple” placement-type, it is either left, right or unchanged, depending on the placement selected. Any non-zero value implies “horizontal-alignment” changes to “left” by default. Has no effect with ‘line’ text-placement-type.

text-dy

Type: float

Default Value: 0 (*Text will not be displaced.*)

Displace text by fixed amount, in pixels, +/- along the Y axis. With “dummy” placement-type, a positive value displaces downwards. With “simple” placement-type, it is either up, down or unchanged, depending on the placement selected. With “line” placement-type, a positive value displaces above the path.

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; “top” for dy>0, “bottom” for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false (*Text will be potentially placed near tile edges and therefore may look cut off unless the same text label is rendered on each adjacent tile.*)

Avoid placing labels that intersect with tile boundaries.

text-margin

Type: float

Default Value: 0 (*No extra margin will be used to determine if a label collides with any other text, shield, or marker.*)

Minimum distance that a label can be placed from any other text, shield, or marker.

text-repeat-distance

Type: float

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance between repeated text. If set this will prevent text labels being rendered nearby each other that contain the same text. Similar to text-min-distance with the difference that it works the same no matter what placement strategy is used.

text-min-distance

Type: float

Status: deprecated

Default Value: 0 (*Labels with the same text will be rendered without restriction.*)

Minimum distance to the next label with the same text. Only works for line placement. Deprecated: replaced by `text-repeat-distance` and `text-margin`

text-min-padding

Type: float

Default Value: 0 (*No margin will be used to detect if a text label is nearby a tile boundary.*)

Minimum distance a text label will be placed from the edge of a tile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the tile.

text-min-path-length

Type: float

Default Value: 0 (*place labels on all geometries no matter how small they are.*)

Place labels only on polygons and lines with a bounding width longer than this value (in pixels).

text-allow-overlap

Type: boolean

Default Value: false (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: float

Default Value: 0 (*Text is not rotated and is displayed upright.*)

Rotate the text. (only works with `text-placement:point`).

text-rotate-displacement

Type: boolean

Default Value: false (*Label center is used for rotation.*)

Rotates the displacement around the placement origin by the angle given by “orientation”.

text-upright

Type: keyword

Possible values: auto auto-down left right left-only right-only

Default Value: auto (*Text will be positioned upright automatically.*)

How this label should be placed along lines. By default when more than half of a label’s characters are upside down the label is automatically flipped to keep it upright. By changing this parameter you can prevent this “auto-upright” behavior. The “auto-down” value places text in the opposite orientation to “auto”. The “left” or “right” settings can be used to force text to always be placed along a line in a given direction and therefore disables flipping if text appears upside down. The “left-only” or “right-only” properties also force a given direction but will discard upside down text rather than trying to flip it.

text-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point (*One shield will be placed per geometry.*)

How this label should be placed. Point placement places one label on top of a point geometry and at the centroid of a polygon or the middle point of a line, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of a polygon.

text-placement-type

Type: keyword

Possible values: dummy simple list

Default Value: dummy (*Alternative placements will not be enabled.*)

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: string

Default Value: *(No alternative placements will be used.)*

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements: "E, NE, SE, W, NW, SW";`.

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize reverse

Default Value: none *(Transform text instance with specified function. Ignores map scale factor.)*

Transform the case of the characters.

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto adjust

Default Value: auto *(TODO.)*

The text’s horizontal alignment from it’s centerpoint. If `placement` is set to `line`, then `adjust` can be set to `auto-fit` the text to the length of the path by dynamically calculating `character-spacing`.

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto *(Auto alignment means that text will be centered by default except when using the “placement-type” parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the “text-placements” directives.)*

Define how text is justified.

text-clip

Type: boolean

Default Value: false *(The geometry will not be clipped to map bounds before rendering.)*

Turning on clipping can help performance in the case that the boundaries of the geometry extent outside of tile extents. But clipping can result in undesirable rendering artifacts in rare cases.

text-simplify

Type: float

Default Value: 0 (*geometry will not be simplified.*)

Simplify the geometries used for text placement by the given tolerance.

text-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*The geometry will be simplified using the radial distance algorithm.*)

Simplify the geometries used for text placement by the given algorithm.

text-smooth

Type: float

Default Value: 0 (*No smoothing.*) Range: 0-1 Smooths out the angles of the geometry used for text placement. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-halo-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current symbolizer on top of other symbolizer.*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text-font-feature-settings

Type: `string`

Default Value: *(Default set of typographic features recommended by OpenType specification. Ligatures are turned off by default when “character-spacing” is greater than zero.)*

Comma separated list of OpenType typographic features. The syntax and semantics conforms to `font-feature-settings` from W3C CSS.

text-largest-bbox-only

Type: `boolean`

Status: experimental

Default Value: `true` *(For multipolygons only polygon with largest bbox area is labeled (does not apply to other geometries).)*

Controls default labeling behavior on multipolygons. The default is `true` and means that only the polygon with largest bbox is labeled.

building

building

Type: `keyword`

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a building symbolizer rule or emitting it with default values.

building-fill

Type: `color`

Default Value: The color gray will be used for fill. *(Gray and fully opaque (alpha = 1), same as `rgb(128,128,128)` or `rgba(128,128,128,1)`.)*

The color of the buildings fill. Note: 0.8 will be used to multiply each color component to auto-generate a darkened wall color.

building-fill-opacity

Type: float

Default Value: 1 (*Color is fully opaque.*)

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0 (*Buildings will not have a visual height and will instead look like flat polygons.*)

The height of the building in pixels.

debug

debug-mode

Type: keyword

Possible values: collision vertex

Default Value: collision (*The otherwise invisible collision boxes will be rendered as squares on the map.*)

The mode for debug rendering.

dot

dot

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a dot symbolizer rule or emitting it with default values.

dot-fill

Type: color

Default Value: gray (*The dot fill color is gray.*)

The color of the area of the dot.

dot-opacity

Type: float

Default Value: 1 (*The opacity of the dot.*)

The overall opacity of the dot.

dot-width

Type: float

Default Value: 1 (*The marker width is 1 pixel.*)

The width of the dot in pixels.

dot-height

Type: float

Default Value: 1 (*The marker height is 1 pixels.*)

The height of the dot in pixels.

dot-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply divide screen overlay darken lighten color-dodge color-burn linear-dodge linear-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*Add the current layer on top of other layers.*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

2.3.0

Style

image-filters

Type: functions

Possible values: agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha color-to-alpha scale-hsla

Default Value: none (*no filters*)

A list of image filters that will be applied to the active rendering canvas for a given style. The presence of one more `image-filters` will trigger a new canvas to be created before starting to render a style and then this canvas will be composited back into the main canvas after rendering all features and after all `image-filters` have been applied. See `direct-image-filters` if you want to apply a filter directly to the main canvas.

image-filters-inflate

Type: `boolean`

Default Value: `false` (*No special handling will be done and image filters that blur data will only blur up to the edge of a tile boundary*)

A property that can be set to `true` to enable using an inflated image internally for seamless blurring across tiles (requires buffered data).

direct-image-filters

Type: `functions`

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha color-to-alpha scale-hsla`

Default Value: none (*no filters*)

A list of image filters to apply to the main canvas (see the `image-filters` doc for how they work on a separate canvas).

comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: `float`

Default Value: `1` (*no separate buffer will be used and no alpha will be applied to the style after rendering*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer).

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color.

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background. Accepted formats: JPG, PNG.

background-image-comp-op

Type: `keyword`

Possible values: `clear` `src` `dst` `src-over` `dst-over` `src-in` `dst-in` `src-out` `dst-out` `src-atop` `dst-atop` `xor` `plus` `minus` `multiply` `screen` `overlay` `darken` `lighten` `color-dodge` `color-burn` `hard-light` `soft-light` `difference` `exclusion` `contrast` `invert` `invert-rgb` `grain-merge` `grain-extract` `hue` `saturation` `color` `value`

Default Value: `src-over` (*The background-image will be blended with the background normally (placed on top of any existing background-color)*)

Set the compositing operation used to blend the image into the background.

background-image-opacity

Type: `float`

Default Value: `1` (*The image opacity will not be changed when applied to the map background*)

Set the opacity of the image.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string).

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: `none` (*No map-specific fonts will be registered*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a polygon symbolizer rule or emitting it with default values.

polygon-fill

Type: color

Default Value: `rgba(128,128,128,1)` (*gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`*)

Fill color to assign to a polygon.

polygon-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the polygon.

polygon-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon edges.

polygon-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*`pow(x,gamma)` is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

polygon-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

Geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-simplify

Type: `float`

Default Value: `0` (*geometry will not be simplified*)

Simplify geometries by the given tolerance.

polygon-simplify-algorithm

Type: `keyword`

Possible values: `radial-distance` `zhao-saalfeld` `visvalingam-whyatt`

Default Value: `radial-distance` (*geometry will be simplified using the radial distance algorithm*)

Simplify geometries by the given algorithm.

polygon-smooth

Type: `float`

Default Value: `0` (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: `functions`

Possible values: `matrix` `translate` `scale` `rotate` `skewX` `skewY`

Default Value: `none` (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a line symbolizer rule or emitting it with default values.

line-color

Type: color

Default Value: rgba(0,0,0,1) (*black and fully opaque (alpha = 1), same as rgb(0,0,0)*)

The color of a drawn line.

line-width

Type: float

Default Value: 1

The width of a line in pixels.

line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line.

line-join

Type: keyword

Possible values: miter round bevel

Default Value: miter

The behavior of lines when joining.

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings.

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line.

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*pow(x,gamma) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

Valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled).

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

Geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

Simplify geometries by the given tolerance

line-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will be simplified using the radial distance algorithm*)

Simplify geometries by the given algorithm.

line-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: float

Default Value: 0 (*no offset*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a marker symbolizer rule or emitting it with default values.

marker-file

Type: uri

Default Value: none (*An ellipse or circle, if width equals height.*)

A file that this marker shows at each placement. If no file is given, the marker will show an ellipse. Accepted formats: SVG, JPG, PNG.

marker-opacity

Type: float

Default Value: 1 (*The stroke-opacity and fill-opacity of the marker.*)

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: float

Default Value: 1 (*opaque*)

The fill opacity of the marker.

marker-line-color

Type: color

Default Value: black

The color of the stroke around the marker.

marker-line-width

Type: float

Default Value: 0.5

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the line.

marker-placement

Type: keyword

Possible values: point line interior

Default Value: point (*Place markers at the center point (centroid) of the geometry*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. 'interior' placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior.

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each*)

A special setting to allow the user to control rendering behavior for 'multi-geometries' (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The 'each' policy is default and means all geometries will get a marker. The 'whole' policy means that the aggregate centroid between all geometries will be used. The 'largest' policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width).

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: boolean

Default Value: false (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

Value to control whether the placement of the feature will prevent the placement of other features.

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated markers in pixels. If the spacing is less than the marker size or larger than the line segment length then no marker will be placed.

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation*)

SVG transformation definition.

marker-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

marker-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield

shield

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a shield symbolizer.

shield-name

Type: `string`

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like `[column_name]`.

shield-file

Type: `uri`

Default Value: `none`

Image file to render behind the shield text. Accepted formats: SVG, JPG, PNG.

shield-face-name

Type: `string`

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: `boolean`

Default Value: `false` (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-size

Type: `float`

Default Value: 10

The size of the shield text in pixels.

shield-fill

Type: `color`

Default Value: `black`

The color of the shield text.

shield-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: `boolean`

Default Value: `false`

Avoid placing shields that intersect with tile boundaries.

shield-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-min-distance

Type: `float`

Default Value: `0`

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: `float`

Default Value: 0

The spacing between repeated occurrences of the same shield on a line.

shield-min-padding

Type: `float`

Default Value: 0

Minimum distance a shield will be placed from the edge of a metatile. This option is similar to `shield-avoid-edges:true` except that the extra margin is used to discard cases where the shield+margin are not fully inside the metatile.

shield-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

shield-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

shield-wrap-character

Type: `string`

Default Value: " "

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: `0` (*no halo*)

Specify the radius of the halo in pixels.

shield-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full`

Exposes an alternate text halo rendering method that sacrifices quality for speed.

shield-character-spacing

Type: `unsigned`

Default Value: `0`

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: `float`

Default Value: `0`

Vertical spacing between lines of multiline labels (in pixels).

shield-text-dx

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the shield right.

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the shield down.

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right.

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down.

shield-opacity

Type: float

Default Value: 1

The opacity of the image used for the shield

shield-text-opacity

Type: `float`

Default Value: `1`

The opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: `keyword`

Possible values: `left middle right auto`

Default Value: `auto`

The shield's horizontal alignment from its centerpoint

shield-vertical-alignment

Type: `keyword`

Possible values: `top middle bottom auto`

Default Value: `middle`

The shield's vertical alignment from its centerpoint

shield-placement-type

Type: `keyword`

Possible values: `dummy simple list`

Default Value: `dummy`

Re-position and/or re-size shield to avoid overlaps. "simple" for basic algorithm (using shield-placements string,) "dummy" to turn this feature off.

shield-placements

Type: `string`

Default Value:

If "placement-type" is set to "simple", use this "POSITIONS,[SIZES]" string. An example is `shield-placements: "E,NE,SE,W,NW,SW";`

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters.

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto

Define how text in a shield's label is justified.

shield-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*No transformation*)

SVG transformation definition.

shield-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern

Type: keyword

Status: **unstable**

Possible values: none

Default Value:

Allows omitting a line pattern symbolizer rule or emitting it with default values.

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line. Accepted formats: JPG, PNG.

line-pattern-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

line-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm.

line-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-offset

Type: float

Default Value: 0 (*no offset*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a polygon pattern symbolizer rule or emitting it with default values.

polygon-pattern-file

Type: `uri`

Default Value: `none`

Image to use as a repeated pattern fill within a polygon. Accepted formats: JPG, PNG.

polygon-pattern-alignment

Type: `keyword`

Possible values: `local` `global`

Default Value: `local`

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: `float`

Default Value: `1` (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

polygon-pattern-opacity

Type: `float`

Default Value: `1` (*The image is rendered without modifications*)

Apply an opacity level to the image used for the pattern

polygon-pattern-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

polygon-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm

polygon-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a raster symbolizer rule or emitting it with default values.

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessel mitchell sinc lanczos blackman

Default Value: near

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster-colorizer-default-mode

Type: keyword

Possible values: discrete linear exact

Default Value: linear (*A linear interpolation is used to generate colors between the two nearest stops.*)

This can be either `discrete`, `linear` or `exact`. If it is not specified then the default is `linear`.

raster-colorizer-default-color

Type: color

Default Value: transparent (*Pixels that are not colored by the colorizer stops will be transparent*)

This can be any color. Sets the color that is applied to all values outside of the range of the colorizer-stops. If not supplied pixels will be fully transparent

raster-colorizer-epsilon

Type: float

Default Value: 1.1920928955078125e-07 (*Pixels must very closely match the stop filter otherwise they will not be colored.*)

This can be any positive floating point value and will be used as a tolerance in floating point comparisons. The higher the value the more likely a stop will match and color data.

raster-colorizer-stops

Type: tags

Default Value: (*No colorization will happen without supplying stops.*)

Assigns raster data values to colors. Stops must be listed in ascending order, and contain at a minimum the value and the associated color. You can also include the color-mode as a third argument, like `stop(100, #fff, exact)`.

point

point

Type: keyword

Status: unstable

Possible values: `auto none`

Default Value:

Allows omitting a point symbolizer rule or emitting it with default values.

point-file

Type: uri

Default Value: none

Image file to represent a point. Accepted formats: SVG, PNG, JPG.

point-allow-overlap

Type: boolean

Default Value: false (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

Control whether the placement of the feature will prevent the placement of other features.

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point.

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

Control how this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*No transformation*)

SVG transformation definition.

point-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text

Type: `keyword`

Status: **unstable**

Possible values: `none`

Default Value:

Allows omitting a text symbolizer rule.

text-name

Type: `string`

Default Value:

Value to use for a text label. Data columns are specified using brackets like `[column_name]`

text-face-name

Type: `string`

Default Value:

Font name and style to render a label in

text-size

Type: `float`

Default Value: 10

Text size in pixels

text-ratio

Type: `unsigned`

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in pixels before wrapping text. If set to zero, text doesn't wrap.

text-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

text-wrap-character

Type: `string`

Default Value: `" "`

Use this character instead of a space to wrap long text.

text-spacing

Type: `unsigned`

Default Value: 0

Distance between repeated text labels on a line (aka. label-spacing).

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels.

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels.

text-label-position-tolerance

Type: `float`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`).

text-max-char-angle-delta

Type: `float`

Default Value: 22.5

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 \cdot \pi / 180.0$. The higher the value the fewer labels will be placed around sharp corners.

text-fill

Type: `color`

Default Value: `#000000`

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: `0` (*no halo*)

Specify the radius of the halo in pixels

text-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full`

Exposes an alternate text halo rendering method that sacrifices quality for speed.

text-dx

Type: `float`

Default Value: `0`

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right.

text-dy

Type: `float`

Default Value: `0`

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text up.

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; “top” for dy>0, “bottom” for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false

Avoid placing labels that intersect with tile boundaries.

text-min-distance

Type: float

Default Value: 0

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: float

Default Value: 0

Minimum distance a text label will be placed from the edge of a metatile. This option is similar to shield-avoid-edges:true except that the extra margin is used to discard cases where the shield+margin are not fully inside the metatile.

text-min-path-length

Type: float

Default Value: 0 (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple list`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using `text-placements` string,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements : "E, NE, SE, W, NW, SW"` ;

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters.

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The text's horizontal alignment from its centerpoint.

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto (*Auto alignment means that text will be centered by default except when using the "placement-type" parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the "text-placements" directives.*)

Define how text is justified

text-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

Geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

building

building

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a building symbolizer rule or emitting it with default values.

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.

debug

debug-mode

Type: keyword

Possible values: collision vertex

Default Value: collision

The mode for debug rendering.

2.2.0

Style

image-filters

Type: functions

Possible values: agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha

Default Value: none (*no filters*)

A list of image filters that will be applied to the active rendering canvas for a given style. The presence of one more `image-filters` will trigger a new canvas to be created before starting to render a style and then this canvas will be composited back into the main canvas after rendering all features and after all `image-filters` have been applied. See `direct-image-filters` if you want to apply a filter directly to the main canvas.

direct-image-filters

Type: functions

Possible values: agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen colorize-alpha

Default Value: none (*no filters*)

A list of image filters to apply to the main canvas (see the `image-filters` doc for how they work on a separate canvas)

comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: `src-over` (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: `float`

Default Value: `1` (*no separate buffer will be used and no alpha will be applied to the style after rendering*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer)

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: 0 (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: none (*No map-specific fonts will be registered*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon

Type: `keyword`

Status: `unstable`

Possible values: `auto none`

Default Value:

Allows omitting a polygon symbolizer rule or emitting it with default values.

polygon-fill

Type: color

Default Value: rgba(128,128,128,1) (*gray and fully opaque (alpha = 1), same as rgb(128,128,128)*)

Fill color to assign to a polygon

polygon-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the polygon

polygon-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon edges

polygon-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*pow(x,gamma) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

polygon-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

polygon-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will not be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm

polygon-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line

Type: keyword

Status: **unstable**

Possible values: `auto none`

Default Value:

Allows omitting a line symbolizer rule or emitting it with default values.

line-color

Type: `color`

Default Value: `rgba(0,0,0,1)` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)`*)

The color of a drawn line

line-width

Type: `float`

Default Value: 1

The width of a line in pixels

line-opacity

Type: `float`

Default Value: 1 (*opaque*)

The opacity of a line

line-join

Type: keyword

Possible values: `miter round bevel`

Default Value: `miter`

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \gamma)$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled)

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

line-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will not be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm

line-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: float

Default Value: 0 (*no offset*)

Offsets a line a number of pixels parallel to its actual path. Positive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a marker symbolizer rule or emitting it with default values.

marker-file

Type: uri

Default Value: *(An ellipse or circle, if width equals height)*

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: float

Default Value: 1 *(The stroke-opacity and fill-opacity will be used)*

The overall opacity of the marker, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: float

Default Value: 1 *(opaque)*

The fill opacity of the marker

marker-line-color

Type: color

Default Value: black

The color of the stroke around a marker shape.

marker-line-width

Type: float

Default Value: 0.5

The width of the stroke around a marker shape, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line

marker-placement

Type: keyword

Possible values: point line interior

Default Value: point (*Place markers at the center point (centroid) of the geometry*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. 'interior' placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each*)

A special setting to allow the user to control rendering behavior for 'multi-geometries' (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The 'each' policy is default and means all geometries will get a marker. The 'whole' policy means that the aggregate centroid between all geometries will be used. The 'largest' policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: boolean

Default Value: false (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated markers in pixels. If the spacing is less than the marker size or larger than the line segment length then no marker will be placed

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation*)

SVG transformation definition

marker-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

marker-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield

shield

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a shield symbolizer.

shield-name

Type: `string`

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like `[column_name]`

shield-file

Type: `uri`

Default Value: `none`

Image file to render behind the shield text

shield-face-name

Type: `string`

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: `boolean`

Default Value: `false` (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-size

Type: `float`

Default Value: `10`

The size of the shield text in pixels

shield-fill

Type: `color`

Default Value: `black`

The color of the shield text

shield-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: `boolean`

Default Value: `false`

Avoid placing shields that intersect with tile boundaries.

shield-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-min-distance

Type: `float`

Default Value: `0`

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: `float`

Default Value: 0

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: `float`

Default Value: 0

Minimum distance a shield will be placed from the edge of a metatile.

shield-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in characters before wrapping text

shield-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If `false`, wrapped lines will be a bit longer than wrap-width.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: `0` (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: `unsigned`

Default Value: `0`

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: `float`

Default Value: `0`

Vertical spacing between lines of multiline labels (in pixels)

shield-text-dx

Type: `float`

Default Value: `0`

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

The opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

The opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The shield's horizontal alignment from its centerpoint

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle

The shield's vertical alignment from its centerpoint

shield-placement-type

Type: keyword

Possible values: dummy simple

Default Value: dummy

Re-position and/or re-size shield to avoid overlaps. "simple" for basic algorithm (using shield-placements string,) "dummy" to turn this feature off.

shield-placements

Type: string

Default Value:

If "placement-type" is set to "simple", use this "POSITIONS,[SIZES]" string. An example is shield-placements: "E,NE,SE,W,NW,SW";

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto

Define how text in a shield's label is justified

shield-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: *(No transformation)*

SVG transformation definition

shield-clip

Type: boolean

Default Value: true *(geometry will be clipped to map bounds before rendering)*

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over *(add the current symbolizer on top of other symbolizer)*

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a line pattern symbolizer rule or emitting it with default values.

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line

line-pattern-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

line-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will not be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm

line-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern

Type: keyword

Status: unstable

Possible values: none

Default Value:

Allows omitting a polygon pattern symbolizer rule or emitting it with default values.

polygon-pattern-file

Type: uri

Default Value: none

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: keyword

Possible values: local global

Default Value: local

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

polygon-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications*)

Apply an opacity level to the image used for the pattern

polygon-pattern-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-pattern-simplify

Type: float

Default Value: 0 (*geometry will not be simplified*)

geometries are simplified by the given tolerance

polygon-pattern-simplify-algorithm

Type: keyword

Possible values: radial-distance zhao-saalfeld visvalingam-whyatt

Default Value: radial-distance (*geometry will not be simplified using the radial distance algorithm*)

geometries are simplified by the given algorithm

polygon-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster

Type: keyword

Status: **unstable**

Possible values: auto none

Default Value:

Allows omitting a raster symbolizer rule or emitting it with default values.

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bilinear8 bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessell mitchell sinc lanczos blackman

Default Value: near

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster-colorizer-default-mode

Type: keyword

Possible values: discrete linear exact

Default Value: linear (*A linear interpolation is used to generate colors between the two nearest stops.*)

This can be either discrete, linear or exact. If it is not specified then the default is linear.

raster-colorizer-default-color

Type: color

Default Value: transparent (*Pixels that are not colored by the colorizer stops will be transparent*)

This can be any color. Sets the color that is applied to all values outside of the range of the colorizer-stops. If not supplied pixels will be fully transparent

raster-colorizer-epsilon

Type: float

Default Value: 1.1920928955078125e-07 (*Pixels must very closely match the stop filter otherwise they will not be colored.*)

This can be any positive floating point value and will be used as a tolerance in floating point comparisons. The higher the value the more likely a stop will match and color data.

raster-colorizer-stops

Type: tags

Default Value: (*No colorization will happen without supplying stops.*)

Assigns raster data values to colors. Stops must be listed in ascending order, and contain at a minimum the value and the associated color. You can also include the color-mode as a third argument, like `stop(100, #fff, exact)`.

point

point

Type: keyword

Status: unstable

Possible values: `auto` `none`

Default Value:

Allows omitting a point symbolizer rule or emitting it with default values.

point-file

Type: uri

Default Value: none

Image file to represent a point

point-allow-overlap

Type: boolean

Default Value: false (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation*)

SVG transformation definition

point-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text**text**

Type: `keyword`

Status: unstable

Possible values: `none`

Default Value:

Allows omitting a text symbolizer rule.

text-name

Type: `string`

Default Value:

Value to use for a text label. Data columns are specified using brackets like `[column_name]`

text-face-name

Type: `string`

Default Value:

Font name and style to render a label in

text-size

Type: `float`

Default Value: 10

Text size in pixels

text-ratio

Type: `unsigned`

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

text-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: `unsigned`

Default Value: 0

Distance between repeated text labels on a line (aka. label-spacing)

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: `unsigned`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`)

text-max-char-angle-delta

Type: `float`

Default Value: 22.5

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 \cdot \pi / 180.0$. The higher the value the fewer labels will be placed around sharp corners.

text-fill

Type: `color`

Default Value: `#000000`

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: `0` (*no halo*)

Specify the radius of the halo in pixels

text-halo-rasterizer

Type: `keyword`

Possible values: `full fast`

Default Value: `full`

Exposes an alternate text halo rendering method that sacrifices quality for speed.

text-dx

Type: `float`

Default Value: `0`

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: `float`

Default Value: `0`

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; "bottom" for dy>0, "top" for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false

Avoid placing labels that intersect with tile boundaries.

text-min-distance

Type: float

Default Value: 0

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: float

Default Value: 0

Minimum distance a text label will be placed from the edge of a metatile.

text-min-path-length

Type: float

Default Value: 0 (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using `text-placements string`,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements : "E, NE, SE, W, NW, SW"` ;

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto (*Auto alignment means that text will be centered by default except when using the "placement-type" parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the "text-placements" directives*)

Define how text is justified

text-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

building

building

Type: keyword

Status: unstable

Possible values: auto none

Default Value:

Allows omitting a building symbolizer rule or emitting it with default values.

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.

debug

debug-mode

Type: keyword

Possible values: collision vertex

Default Value: collision

The mode for debug rendering

2.1.1

Style

image-filters

Type: functions

Possible values: agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen

Default Value: none (*no filters*)

A list of image filters.

comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: float

Default Value: 1 (*no separate buffer will be used and no alpha will be applied to the style after rendering*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer)

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: *(This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.)*

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: *none (No map-specific fonts will be registered)*

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon-fill

Type: `color`

Default Value: `rgba(128,128,128,1)` *(gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`)*

Fill color to assign to a polygon

polygon-opacity

Type: `float`

Default Value: `1` *(opaque)*

The opacity of the polygon

polygon-gamma

Type: `float`

Default Value: `1` *(fully antialiased)* Range: 0-1 Level of antialiasing of polygon edges

polygon-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power* ($pow(x, \text{gamma})$) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

polygon-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line-color

Type: `color`

Default Value: `rgba(0,0,0,1)` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)`*)

The color of a drawn line

line-width

Type: `float`

Default Value: `1`

The width of a line in pixels

line-opacity

Type: `float`

Default Value: `1` (*opaque*)

The opacity of a line

line-join

Type: `keyword`

Possible values: `miter` `round` `bevel`

Default Value: `miter`

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*power(x,gamma) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled)

line-miterlimit

Type: float

Default Value: 4 (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: float

Default Value: 0 (*no offset*)

Offsets a line a number of pixels parallel to its actual path. Postive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker-file

Type: `uri`

Default Value: (*An ellipse or circle, if width equals height*)

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: `float`

Default Value: `1` (*The stroke-opacity and fill-opacity will be used*)

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: `float`

Default Value: `1` (*opaque*)

The fill opacity of the marker

marker-line-color

Type: color

Default Value: black

The color of the stroke around a marker shape.

marker-line-width

Type: float

Default Value: 0.5

The width of the stroke around a marker shape, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line

marker-placement

Type: keyword

Possible values: point line interior

Default Value: point (*Place markers at the center point (centroid) of the geometry*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. ‘interior’ placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior

marker-multi-policy

Type: keyword

Possible values: each whole largest

Default Value: each (*If a feature contains multiple geometries and the placement type is either point or interior then a marker will be rendered for each*)

A special setting to allow the user to control rendering behavior for ‘multi-geometries’ (when a feature contains multiple geometries). This setting does not apply to markers placed along lines. The ‘each’ policy is default and means all geometries will get a marker. The ‘whole’ policy means that the aggregate centroid between all geometries

will be used. The ‘largest’ policy means that only the largest (by bounding box areas) feature will get a rendered marker (this is how text labeling behaves by default).

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: expression

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: expression

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: boolean

Default Value: false (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated labels

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation*)

SVG transformation definition

marker-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

marker-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

marker-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield

shield-name

Type: expression

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like [column_name]

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text

shield-face-name

Type: `string`

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: `boolean`

Default Value: `false` (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to `true` if you are trying to position text beside rather than on top of the shield image

shield-size

Type: `float`

Default Value: `10`

The size of the shield text in pixels

shield-fill

Type: `color`

Default Value: `black`

The color of the shield text

shield-placement

Type: `keyword`

Possible values: `point` `line` `vertex` `interior`

Default Value: `point`

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

shield-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-min-distance

Type: `float`

Default Value: `0`

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: `float`

Default Value: `0`

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: `float`

Default Value: `0`

Determines the minimum amount of padding that a shield gets relative to other shields

shield-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in characters before wrapping text

shield-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: unsigned

Default Value: 0

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0

Vertical spacing between lines of multiline labels (in pixels)

shield-text-dx

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The shield's horizontal alignment from its centerpoint

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle

The shield's vertical alignment from its centerpoint

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto

Define how text in a shield's label is justified

shield-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line

line-pattern-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern-file

Type: `uri`

Default Value: `none`

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: `keyword`

Possible values: `local` `global`

Default Value: `local`

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: `float`

Default Value: `1` (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

polygon-pattern-opacity

Type: `float`

Default Value: `1` (*The image is rendered without modifications*)

(Default 1.0) - Apply an opacity level to the image used for the pattern

polygon-pattern-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bilinear8 bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessell mitchell sinc lanczos blackman

Default Value: near

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

point

point-file

Type: uri

Default Value: none

Image file to represent a point

point-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: `float`

Default Value: `1` (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: (*No transformation*)

SVG transformation definition

point-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text-name

Type: expression

Default Value:

Value to use for a text label. Data columns are specified using brackets like [column_name]

text-face-name

Type: string

Default Value:

Font name and style to render a label in

text-size

Type: float

Default Value: 10

Text size in pixels

text-ratio

Type: unsigned

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

text-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: `unsigned`

Default Value: 0

Distance between repeated text labels on a line (aka. label-spacing)

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: `unsigned`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels (only works with `placement:line`)

text-max-char-angle-delta

Type: `float`

Default Value: 22.5

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 * \pi / 180.0$. The higher the value the fewer labels will be placed around sharp corners.

text-fill

Type: `color`

Default Value: `#000000`

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: color

Default Value: #FFFFFF (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: float

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

text-dx

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; "bottom" for dy>0, "top" for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

text-min-distance

Type: `float`

Default Value: `0`

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: `float`

Default Value: `0`

Determines the minimum amount of padding that a text symbolizer gets relative to other text

text-min-path-length

Type: `float`

Default Value: `0` (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `expression`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple list`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using `text-placements string`.) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements : "E, NE, SE, W, NW, SW"`;

text-transform

Type: `keyword`

Possible values: `none uppercase lowercase capitalize`

Default Value: `none`

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto (*Auto alignment means that text will be centered by default except when using the “placement-type” parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the “text-placements” directives*)

Define how text is justified

text-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

building

building-fill

Type: `color`

Default Value: `#FFFFFF`

The color of the buildings walls.

building-fill-opacity

Type: `float`

Default Value: `1`

The opacity of the building as a whole, including all walls.

building-height

Type: `expression`

Default Value: `0`

The height of the building in pixels.

2.1.0

Style

image-filters

Type: `functions`

Possible values: `agg-stack-blur emboss blur gray sobel edge-detect x-gradient y-gradient invert sharpen`

Default Value: `none` (*no filters*)

A list of image filters.

comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current layer on top of other layers*)

Composite operation. This defines how this layer should behave relative to layers atop or below it.

opacity

Type: float

Default Value: 1 (*no separate buffer will be used and no alpha will be applied to the style after rendering*)

An alpha value for the style (which means an alpha applied to all features in separate buffer and then composited back to main buffer)

Symbolizers

map

background-color

Type: color

Default Value: none (*transparent*)

Map Background color

background-image

Type: uri

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: (*This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.*)

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: `none` (*No map-specific fonts will be registered*)

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon-fill

Type: color

Default Value: `rgba(128,128,128,1)` (*gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`*)

Fill color to assign to a polygon

polygon-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the polygon

polygon-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon edges

polygon-gamma-method

Type: keyword

Possible values: `power linear none threshold multiply`

Default Value: `power` (*`pow(x,gamma)` is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

polygon-clip

Type: boolean

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line

line-color

Type: color

Default Value: rgba(0,0,0,1) (*black and fully opaque (alpha = 1), same as rgb(0,0,0)*)

The color of a drawn line

line-width

Type: float

Default Value: 1

The width of a line in pixels

line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line

line-join

Type: keyword

Possible values: miter round bevel

Default Value: miter

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-gamma-method

Type: keyword

Possible values: power linear none threshold multiply

Default Value: power (*pow(x,gamma) is used to calculate pixel gamma, which produces slightly smoother line and polygon antialiasing than the 'linear' method, while other methods are usually only used to disable AA*)

An Antigrain Geometry specific rendering hint to control the quality of antialiasing. Under the hood in Mapnik this method is used in combination with the 'gamma' value (which defaults to 1). The methods are in the AGG source at https://github.com/mapnik/mapnik/blob/master/deps/agg/include/agg_gamma_functions.h

line-dasharray

Type: `numbers`

Default Value: `none` (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: `numbers`

Default Value: `none` (*solid line*)

valid parameter but not currently used in renderers (only exists for experimental svg support in Mapnik which is not yet enabled)

line-miterlimit

Type: `float`

Default Value: `4` (*Will auto-convert miters to bevel line joins when theta is less than 29 degrees as per the SVG spec: 'miterLength / stroke-width = 1 / sin (theta / 2)'*)

The limit on the ratio of the miter length to the stroke-width. Used to automatically convert miter joins to bevel joins for sharp angles to avoid the miter extending beyond the thickness of the stroking path. Normally will not need to be set, but a larger value can sometimes help avoid jaggy artifacts.

line-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-smooth

Type: `float`

Default Value: `0` (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-offset

Type: float

Default Value: 0 (*no offset*)

Offsets a line a number of pixels parallel to its actual path. Postive values move the line left, negative values move it right (relative to the directionality of the line).

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

line-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

markers

marker-file

Type: `uri`

Default Value: *(An ellipse or circle, if width equals height)*

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: `float`

Default Value: 1 *(The stroke-opacity and fill-opacity will be used)*

The overall opacity of the marker, if set, overrides both the opacity of the fill and the opacity of the stroke.

marker-fill-opacity

Type: `float`

Default Value: 1 *(opaque)*

The fill opacity of the marker

marker-line-color

Type: `color`

Default Value: `black`

The color of the stroke around a marker shape.

marker-line-width

Type: `float`

Default Value: 0.5

The width of the stroke around a marker shape, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line

marker-placement

Type: keyword

Possible values: point line interior

Default Value: point (*Place markers at the center point (centroid) of the geometry*)

Attempt to place markers on a point, in the center of a polygon, or if markers-placement:line, then multiple times along a line. 'interior' placement can be used to ensure that points placed on polygons are forced to be inside the polygon interior

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: `color`

Default Value: `blue`

The color of the area of the marker.

marker-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated labels

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: *(No transformation)*

SVG transformation definition

marker-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

marker-smooth

Type: `float`

Default Value: `0` (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

marker-geometry-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

marker-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

shield**shield-name**Type: `string`

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like `[column_name]`

shield-fileType: `uri`Default Value: `none`Image file to render behind the shield text

shield-face-nameType: `string`

Default Value:

Font name and style to use for the shield text

shield-unlock-imageType: `boolean`Default Value: `false` (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-sizeType: `float`Default Value: `10`The size of the shield text in pixels

shield-fill

Type: `color`

Default Value: `black`

The color of the shield text

shield-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

shield-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow shields to overlap with other map elements already placed.*)

Control whether overlapping shields are shown or hidden.

shield-min-distance

Type: `float`

Default Value: `0`

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: `float`

Default Value: 0

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: `float`

Default Value: 0

Determines the minimum amount of padding that a shield gets relative to other shields

shield-wrap-width

Type: `unsigned`

Default Value: 0

Length of a chunk of text in characters before wrapping text

shield-wrap-before

Type: `boolean`

Default Value: `false`

Wrap text before wrap-width is reached. If `false`, wrapped lines will be a bit longer than wrap-width.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: `0` (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: `unsigned`

Default Value: `0`

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: `float`

Default Value: `0`

Vertical spacing between lines of multiline labels (in pixels)

shield-text-dx

Type: `float`

Default Value: `0`

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The shield's horizontal alignment from its centerpoint

shield-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: middle

The shield's vertical alignment from its centerpoint

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-justify-alignment

Type: keyword

Possible values: left center right auto

Default Value: auto

Define how text in a shield's label is justified

shield-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

shield-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

line-pattern

line-pattern-file

Type: uri

Default Value: none

An image file to be repeated and warped along a line

line-pattern-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

line-pattern-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

line-pattern-geometry-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: none (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

line-pattern-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

polygon-pattern

polygon-pattern-file

Type: uri

Default Value: none

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: keyword

Possible values: local global

Default Value: local

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

polygon-pattern-opacity

Type: float

Default Value: 1 (*The image is rendered without modifications*)

(Default 1.0) - Apply an opacity level to the image used for the pattern

polygon-pattern-clip

Type: `boolean`

Default Value: `true` (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

polygon-pattern-smooth

Type: `float`

Default Value: `0` (*no smoothing*) Range: 0-1 Smooths out geometry angles. 0 is no smoothing, 1 is fully smoothed. Values greater than 1 will produce wild, looping geometries.

polygon-pattern-geometry-transform

Type: `functions`

Possible values: `matrix translate scale rotate skewX skewY`

Default Value: `none` (*geometry will not be transformed*)

Allows transformation functions to be applied to the geometry.

polygon-pattern-comp-op

Type: `keyword`

Possible values: `clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value`

Default Value: `src-over` (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

raster

raster-opacity

Type: `float`

Default Value: `1` (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: near fast bilinear bilinear8 bicubic spline16 spline36 hanning hamming hermite kaiser quadric catrom gaussian bessel mitchell sinc lanczos blackman

Default Value: near

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: unsigned

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

raster-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

point

point-file

Type: uri

Default Value: none

Image file to represent a point

point-allow-overlap

Type: boolean

Default Value: false (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: boolean

Default Value: false (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: float

Default Value: 1 (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: keyword

Possible values: centroid interior

Default Value: centroid

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: functions

Possible values: matrix translate scale rotate skewX skewY

Default Value: *(No transformation)*

SVG transformation definition

point-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over *(add the current symbolizer on top of other symbolizer)*

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

text

text-name

Type: string

Default Value:

Value to use for a text label. Data columns are specified using brackets like [column_name]

text-face-name

Type: string

Default Value:

Font name and style to render a label in

text-size

Type: float

Default Value: 10

Text size in pixels

text-ratio

Type: unsigned

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: unsigned

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-before

Type: boolean

Default Value: false

Wrap text before wrap-width is reached. If false, wrapped lines will be a bit longer than wrap-width.

text-wrap-character

Type: string

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: unsigned

Default Value: 0

Distance between repeated text labels on a line (aka. label-spacing)

text-character-spacing

Type: float

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: float

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: unsigned

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels (only works with placement:line)

text-max-char-angle-delta

Type: float

Default Value: 22.5

The maximum angle change, in degrees, allowed between adjacent characters in a label. This value internally is converted to radians to the default is $22.5 * \pi / 180.0$. The higher the value the fewer labels will be placed around sharp corners.

text-fill

Type: color

Default Value: #000000

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: `color`

Default Value: #FFFFFF (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: `float`

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

text-dx

Type: `float`

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: `float`

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom auto

Default Value: auto (*Default affected by value of dy; “bottom” for dy>0, “top” for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: boolean

Default Value: false

Tell positioning algorithm to avoid labeling near intersection edges.

text-min-distance

Type: float

Default Value: 0

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: float

Default Value: 0

Determines the minimum amount of padding that a text symbolizer gets relative to other text

text-min-path-length

Type: float

Default Value: 0 (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple list`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using `text-placements` string,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. An example is `text-placements : "E, NE, SE, W, NW, SW"` ;

text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: auto

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center auto

Default Value: auto (*Auto alignment means that text will be centered by default except when using the “placement-type” parameter - in that case either right or left justification will be used automatically depending on where the text could be fit given the “text-placements” directives*)

Define how text is justified

text-clip

Type: boolean

Default Value: true (*geometry will be clipped to map bounds before rendering*)

geometries are clipped to map bounds by default for best rendering performance. In some cases users may wish to disable this to avoid rendering artifacts.

text-comp-op

Type: keyword

Possible values: clear src dst src-over dst-over src-in dst-in src-out dst-out src-atop dst-atop xor plus minus multiply screen overlay darken lighten color-dodge color-burn hard-light soft-light difference exclusion contrast invert invert-rgb grain-merge grain-extract hue saturation color value

Default Value: src-over (*add the current symbolizer on top of other symbolizer*)

Composite operation. This defines how this symbolizer should behave relative to symbolizers atop or below it.

building

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.

2.0.2

Style

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: *(This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.)*

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: *none (No map-specific fonts will be registered)*

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon**polygon-fill**

Type: `color`

Default Value: `rgba(128,128,128,1)` *(gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`)*

Fill color to assign to a polygon

polygon-gamma

Type: `float`

Default Value: `1` *(fully antialiased)* Range: 0-1 Level of antialiasing of polygon edges

polygon-opacity

Type: `float`

Default Value: `1` *(opaque)*

The opacity of the polygon

polygon-smooth

Type: float

Default Value: 0 (*no smoothing*) Range: 0-1 Amount of smoothing applied

line

line-color

Type: color

Default Value: rgba(0,0,0,1) (*black and fully opaque (alpha = 1), same as rgb(0,0,0)*)

The color of a drawn line

line-width

Type: float

Default Value: 1

The width of a line in pixels

line-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of a line

line-join

Type: keyword

Possible values: miter round bevel

Default Value: miter

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

valid parameter but not currently used in renderers

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

markers

marker-file

Type: `uri`

Default Value: *(An ellipse or circle, if width equals height)*

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: `float`

Default Value: 1 *(opaque)*

The overall opacity of the marker

marker-line-color

Type: `color`

Default Value: `black`

The color of the stroke around a marker shape.

marker-line-width

Type: `float`

Default Value: 0.5

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: `float`

Default Value: 1 *(opaque)*

The opacity of a line

marker-placement

Type: keyword

Possible values: point line

Default Value: line

Attempt to place markers on a point once or on a line repeatedly

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated labels

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `string`

Default Value: (*no transformation*)

An SVG transformation definition

shield

shield-name

Type: string

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like [column_name]

shield-face-name

Type: string

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: boolean

Default Value: false (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-size

Type: float

Default Value: 10

The size of the shield text in pixels

shield-fill

Type: color

Default Value: black

The color of the shield text

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: boolean

Default Value: false

Tell positioning algorithm to avoid labeling near intersection edges.

shield-min-distance

Type: float

Default Value: 0

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: float

Default Value: 0

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: float

Default Value: 0

Determines the minimum amount of padding that a shield gets relative to other shields

shield-wrap-width

Type: `float`

Default Value: 0

Length before wrapping long names.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0

Vertical spacing between lines of multiline labels (in pixels)

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text

shield-text-dx

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The shield's horizontal alignment from its centerpoint

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-no-text

Type: `boolean`

Default Value: `false`

Whether the shield should make room for a text label.

shield-justify-alignment

Type: `string`

Default Value: `middle`

Define how text in a shield's label is justified

line-pattern

line-pattern-file

Type: `uri`

Default Value: `none`

An image file to be repeated and warped along a line

polygon-pattern

polygon-pattern-file

Type: `uri`

Default Value: `none`

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: `keyword`

Possible values: `local` `global`

Default Value: `local`

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

raster

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-mode

Type: keyword

Possible values: normal grain_merge grain_merge2 multiply multiply2 divide divide2 screen hard_light

Default Value: normal

The blending technique used to overlay this raster image on the layer below. Normal simply covers the layer. Grain merge adds the two layers together and subtracts 128 from the value, making the resulting area sometimes high-contrast. Screen often gives a lighter, washed-out appearance. Multiply multiplies the pixels, giving a high-contrast result. Divide divides the upper layer by the lower layer, making a lighter version.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: fast bilinear bilinear8 bicubic spline16 gaussian lanczos

Default Value: fast

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: `integer`

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

point

point-file

Type: `uri`

Default Value: none

Image file to represent a point

point-allow-overlap

Type: `boolean`

Default Value: false (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: false (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `string`

Default Value: *(No transformation)*

SVG transformation definition

text

text-name

Type: `string`

Default Value:

Value to use for a text label. Data columns are specified using brackets like `[column_name]`

text-face-name

Type: `string`

Default Value:

Font name and style to render a label in

text-size

Type: `float`

Default Value: `10`

Text size in pixels

text-ratio

Type: `float`

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `float`

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: `float`

Default Value: 0

Distance between repeated text labels on a line

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: `float`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels

text-max-char-angle-delta

Type: `float`

Default Value: none

If present, the maximum angle change, in degrees, allowed between adjacent characters in a label. This will stop label placement around sharp corners.

text-fill

Type: `color`

Default Value: #000000

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: color

Default Value: #FFFFFF (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: float

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

text-dx

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom

Default Value: middle (*Default affected by value of dy; "bottom" for dy>0, "top" for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

text-min-distance

Type: `float`

Default Value: `0`

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: `float`

Default Value: `0`

Determines the minimum amount of padding that a text symbolizer gets relative to other text

text-min-path-length

Type: `float`

Default Value: `0` (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. See TextSymbolizer docs for format.

text-transform

Type: `keyword`

Possible values: `none uppercase lowercase capitalize`

Default Value: `none`

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center

Default Value: center

Set the text alignment.

building

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.

2.0.1

Style

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: *(This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.)*

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: *none (No map-specific fonts will be registered)*

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon**polygon-fill**

Type: `color`

Default Value: `rgba(128,128,128,1)` *(gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`)*

Fill color to assign to a polygon

polygon-gamma

Type: `float`

Default Value: `1` *(fully antialiased)* Range: 0-1 Level of antialiasing of polygon edges

polygon-opacity

Type: `float`

Default Value: `1` *(opaque)*

The opacity of the polygon

polygon-smooth

Type: `float`

Default Value: 0 (*no smoothing*) Range: 0-1 Amount of smoothing applied

line

line-color

Type: `color`

Default Value: `rgba(0,0,0,1)` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)`*)

The color of a drawn line

line-width

Type: `float`

Default Value: 1

The width of a line in pixels

line-opacity

Type: `float`

Default Value: 1 (*opaque*)

The opacity of a line

line-join

Type: `keyword`

Possible values: `miter` `round` `bevel`

Default Value: `miter`

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

valid parameter but not currently used in renderers

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

markers

marker-file

Type: `uri`

Default Value: *(An ellipse or circle, if width equals height)*

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: `float`

Default Value: 1 *(opaque)*

The overall opacity of the marker

marker-line-color

Type: `color`

Default Value: `black`

The color of the stroke around a marker shape.

marker-line-width

Type: `float`

Default Value: 0.5

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: `float`

Default Value: 1 *(opaque)*

The opacity of a line

marker-placement

Type: keyword

Possible values: point line

Default Value: line

Attempt to place markers on a point once or on a line repeatedly

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated labels

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `string`

Default Value: (*no transformation*)

An SVG transformation definition

shield

shield-name

Type: string

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like [column_name]

shield-face-name

Type: string

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: boolean

Default Value: false (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-size

Type: float

Default Value: 10

The size of the shield text in pixels

shield-fill

Type: color

Default Value: black

The color of the shield text

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: boolean

Default Value: false

Tell positioning algorithm to avoid labeling near intersection edges.

shield-min-distance

Type: float

Default Value: 0

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: float

Default Value: 0

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: float

Default Value: 0

Determines the minimum amount of padding that a shield gets relative to other shields

shield-wrap-width

Type: `float`

Default Value: 0

Length before wrapping long names.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0

Vertical spacing between lines of multiline labels (in pixels)

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text

shield-text-dx

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The shield's horizontal alignment from its centerpoint

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-no-text

Type: `boolean`

Default Value: `false`

Whether the shield should make room for a text label.

shield-justify-alignment

Type: `string`

Default Value: `middle`

Define how text in a shield's label is justified

line-pattern

line-pattern-file

Type: `uri`

Default Value: `none`

An image file to be repeated and warped along a line

polygon-pattern

polygon-pattern-file

Type: `uri`

Default Value: `none`

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: `keyword`

Possible values: `local` `global`

Default Value: `local`

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

raster

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-mode

Type: keyword

Possible values: normal grain_merge grain_merge2 multiply multiply2 divide divide2 screen hard_light

Default Value: normal

The blending technique used to overlay this raster image on the layer below. Normal simply covers the layer. Grain merge adds the two layers together and subtracts 128 from the value, making the resulting area sometimes high-contrast. Screen often gives a lighter, washed-out appearance. Multiply multiplies the pixels, giving a high-contrast result. Divide divides the upper layer by the lower layer, making a lighter version.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: fast bilinear bilinear8 bicubic spline16 gaussian lanczos

Default Value: fast

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: `integer`

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

point

point-file

Type: `uri`

Default Value: `none`

Image file to represent a point

point-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `string`

Default Value: *(No transformation)*

SVG transformation definition

text

text-name

Type: `string`

Default Value:

Value to use for a text label. Data columns are specified using brackets like `[column_name]`

text-face-name

Type: `string`

Default Value:

Font name and style to render a label in

text-size

Type: `float`

Default Value: `10`

Text size in pixels

text-ratio

Type: `float`

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `float`

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: `float`

Default Value: 0

Distance between repeated text labels on a line

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: `float`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels

text-max-char-angle-delta

Type: `float`

Default Value: none

If present, the maximum angle change, in degrees, allowed between adjacent characters in a label. This will stop label placement around sharp corners.

text-fill

Type: `color`

Default Value: #000000

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: color

Default Value: #FFFFFF (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: float

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

text-dx

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom

Default Value: middle (*Default affected by value of dy; "bottom" for dy>0, "top" for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

text-min-distance

Type: `float`

Default Value: `0`

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: `float`

Default Value: `0`

Determines the minimum amount of padding that a text symbolizer gets relative to other text

text-min-path-length

Type: `float`

Default Value: `0` (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. See TextSymbolizer docs for format.

text-transform

Type: `keyword`

Possible values: `none uppercase lowercase capitalize`

Default Value: `none`

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center

Default Value: center

Set the text alignment.

building

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.

2.0.0

Style

Symbolizers

map

background-color

Type: `color`

Default Value: `none` (*transparent*)

Map Background color

background-image

Type: `uri`

Default Value: (*transparent*)

An image that is repeated below all features on a map as a background.

srs

Type: `string`

Default Value: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` (*The proj4 literal of EPSG:4326 is assumed to be the Map's spatial reference and all data from layers within this map will be plotted using this coordinate system. If any layers do not declare an srs value then they will be assumed to be in the same srs as the Map and not transformations will be needed to plot them in the Map's coordinate space*)

Map spatial reference (proj4 string)

buffer-size

Type: `float`

Default Value: `0` (*No buffer will be used*)

Extra tolerance around the map (in pixels) used to ensure labels crossing tile boundaries are equally rendered in each tile (e.g. cut in each tile). Not intended to be used in combination with “avoid-edges”.

base

Type: `string`

Default Value: *(This base path defaults to an empty string meaning that any relative paths to files referenced in styles or layers will be interpreted relative to the application process.)*

Any relative paths used to reference files will be understood as relative to this directory path if the map is loaded from an in memory object rather than from the filesystem. If the map is loaded from the filesystem and this option is not provided it will be set to the directory of the stylesheet.

font-directory

Type: `uri`

Default Value: *none (No map-specific fonts will be registered)*

Path to a directory which holds fonts which should be registered when the Map is loaded (in addition to any fonts that may be automatically registered).

polygon

polygon-fill

Type: `color`

Default Value: `rgba(128,128,128,1)` *(gray and fully opaque (alpha = 1), same as `rgb(128,128,128)`)*

Fill color to assign to a polygon

polygon-gamma

Type: `float`

Default Value: `1` *(fully antialiased)* Range: 0-1 Level of antialiasing of polygon edges

polygon-opacity

Type: `float`

Default Value: `1` *(opaque)*

The opacity of the polygon

polygon-smooth

Type: `float`

Default Value: 0 (*no smoothing*) Range: 0-1 Amount of smoothing applied

line

line-color

Type: `color`

Default Value: `rgba(0,0,0,1)` (*black and fully opaque (alpha = 1), same as `rgb(0,0,0)`*)

The color of a drawn line

line-width

Type: `float`

Default Value: 1

The width of a line in pixels

line-opacity

Type: `float`

Default Value: 1 (*opaque*)

The opacity of a line

line-join

Type: `keyword`

Possible values: `miter` `round` `bevel`

Default Value: `miter`

The behavior of lines when joining

line-cap

Type: keyword

Possible values: butt round square

Default Value: butt

The display of line endings

line-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of stroke line

line-dasharray

Type: numbers

Default Value: none (*solid line*)

A pair of length values [a,b], where (a) is the dash length and (b) is the gap length respectively. More than two values are supported for more complex patterns.

line-dash-offset

Type: numbers

Default Value: none (*solid line*)

valid parameter but not currently used in renderers

line-rasterizer

Type: keyword

Possible values: full fast

Default Value: full

Exposes an alternate AGG rendering method that sacrifices some accuracy for speed.

markers

marker-file

Type: `uri`

Default Value: *(An ellipse or circle, if width equals height)*

An SVG file that this marker shows at each placement. If no file is given, the marker will show an ellipse.

marker-opacity

Type: `float`

Default Value: 1 *(opaque)*

The overall opacity of the marker

marker-line-color

Type: `color`

Default Value: `black`

The color of the stroke around a marker shape.

marker-line-width

Type: `float`

Default Value: 0.5

The width of the stroke around the marker, in pixels. This is positioned on the boundary, so high values can cover the area itself.

marker-line-opacity

Type: `float`

Default Value: 1 *(opaque)*

The opacity of a line

marker-placement

Type: keyword

Possible values: point line

Default Value: line

Attempt to place markers on a point once or on a line repeatedly

marker-type

Type: keyword

Possible values: arrow ellipse

Default Value: ellipse

The default marker-type. If a SVG file is not given as the marker-file parameter, the renderer provides either an arrow or an ellipse (a circle if height is equal to width)

marker-width

Type: float

Default Value: 10

The width of the marker, if using one of the default types.

marker-height

Type: float

Default Value: 10

The height of the marker, if using one of the default types.

marker-fill

Type: color

Default Value: blue

The color of the area of the marker.

marker-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow makers to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping markers are shown or hidden.

marker-ignore-placement

Type: `boolean`

Default Value: `false` (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

marker-spacing

Type: `float`

Default Value: `100`

Space between repeated labels

marker-max-error

Type: `float`

Default Value: `0.2`

The maximum difference between actual marker placement and the marker-spacing parameter. Setting a high value can allow the renderer to try to resolve placement conflicts with other symbolizers.

marker-transform

Type: `string`

Default Value: (*no transformation*)

An SVG transformation definition

shield

shield-name

Type: string

Default Value:

Value to use for a shield's text label. Data columns are specified using brackets like [column_name]

shield-face-name

Type: string

Default Value:

Font name and style to use for the shield text

shield-unlock-image

Type: boolean

Default Value: false (*text alignment relative to the shield image uses the center of the image as the anchor for text positioning.*)

This parameter should be set to true if you are trying to position text beside rather than on top of the shield image

shield-size

Type: float

Default Value: 10

The size of the shield text in pixels

shield-fill

Type: color

Default Value: black

The color of the shield text

shield-placement

Type: keyword

Possible values: point line vertex interior

Default Value: point

How this shield should be placed. Point placement attempts to place it on top of points, line places along lines multiple times per feature, vertex places on the vertexes of polygons, and interior attempts to place inside of polygons.

shield-avoid-edges

Type: boolean

Default Value: false

Tell positioning algorithm to avoid labeling near intersection edges.

shield-min-distance

Type: float

Default Value: 0

Minimum distance to the next shield symbol, not necessarily the same shield.

shield-spacing

Type: float

Default Value: 0

The spacing between repeated occurrences of the same shield on a line

shield-min-padding

Type: float

Default Value: 0

Determines the minimum amount of padding that a shield gets relative to other shields

shield-wrap-width

Type: `float`

Default Value: 0

Length before wrapping long names.

shield-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long names.

shield-halo-fill

Type: `color`

Default Value: `#FFFFFF` (*white*)

Specifies the color of the halo around the text.

shield-halo-radius

Type: `float`

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

shield-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing between characters (in pixels). Currently works for point placement only, not line placement.

shield-line-spacing

Type: float

Default Value: 0

Vertical spacing between lines of multiline labels (in pixels)

shield-file

Type: uri

Default Value: none

Image file to render behind the shield text

shield-text-dx

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-text-dy

Type: float

Default Value: 0

Displace text within shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-dx

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

shield-dy

Type: float

Default Value: 0

Displace shield by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

shield-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the image used for the shield

shield-text-opacity

Type: float

Default Value: 1

(Default 1.0) - opacity of the text placed on top of the shield

shield-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The shield's horizontal alignment from its centerpoint

shield-text-transform

Type: keyword

Possible values: none uppercase lowercase capitalize

Default Value: none

Transform the case of the characters

shield-no-text

Type: `boolean`

Default Value: `false`

Whether the shield should make room for a text label.

shield-justify-alignment

Type: `string`

Default Value: `middle`

Define how text in a shield's label is justified

line-pattern

line-pattern-file

Type: `uri`

Default Value: `none`

An image file to be repeated and warped along a line

polygon-pattern

polygon-pattern-file

Type: `uri`

Default Value: `none`

Image to use as a repeated pattern fill within a polygon

polygon-pattern-alignment

Type: `keyword`

Possible values: `local` `global`

Default Value: `local`

Specify whether to align pattern fills to the layer or to the map.

polygon-pattern-gamma

Type: float

Default Value: 1 (*fully antialiased*) Range: 0-1 Level of antialiasing of polygon pattern edges

raster

raster-opacity

Type: float

Default Value: 1 (*opaque*)

The opacity of the raster symbolizer on top of other symbolizers.

raster-mode

Type: keyword

Possible values: normal grain_merge grain_merge2 multiply multiply2 divide divide2 screen hard_light

Default Value: normal

The blending technique used to overlay this raster image on the layer below. Normal simply covers the layer. Grain merge adds the two layers together and subtracts 128 from the value, making the resulting area sometimes high-contrast. Screen often gives a lighter, washed-out appearance. Multiply multiplies the pixels, giving a high-contrast result. Divide divides the upper layer by the lower layer, making a lighter version.

raster-filter-factor

Type: float

Default Value: -1 (*Allow the datasource to choose appropriate downscaling.*)

This is used by the Raster or Gdal datasources to pre-downscale images using overviews. Higher numbers can sometimes cause much better scaled image output, at the cost of speed.

raster-scaling

Type: keyword

Possible values: fast bilinear bilinear8 bicubic spline16 gaussian lanczos

Default Value: fast

The scaling algorithm used to making different resolution versions of this raster layer. Bilinear is a good compromise between speed and accuracy, while lanczos gives the highest quality.

raster-mesh-size

Type: `integer`

Default Value: 16 (*Reprojection mesh will be 1/16 of the resolution of the source image*)

A reduced resolution mesh is used for raster reprojection, and the total image size is divided by the mesh-size to determine the quality of that mesh. Values for mesh-size larger than the default will result in faster reprojection but might lead to distortion.

point

point-file

Type: `uri`

Default Value: none

Image file to represent a point

point-allow-overlap

Type: `boolean`

Default Value: false (*Do not allow points to overlap with each other - overlapping markers will not be shown.*)

Control whether overlapping points are shown or hidden.

point-ignore-placement

Type: `boolean`

Default Value: false (*do not store the bbox of this geometry in the collision detector cache*)

value to control whether the placement of the feature will prevent the placement of other features

point-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A value from 0 to 1 to control the opacity of the point

point-placement

Type: `keyword`

Possible values: `centroid interior`

Default Value: `centroid`

How this point should be placed. Centroid calculates the geometric center of a polygon, which can be outside of it, while interior always places inside of a polygon.

point-transform

Type: `string`

Default Value: *(No transformation)*

SVG transformation definition

text

text-name

Type: `string`

Default Value:

Value to use for a text label. Data columns are specified using brackets like `[column_name]`

text-face-name

Type: `string`

Default Value:

Font name and style to render a label in

text-size

Type: `float`

Default Value: 10

Text size in pixels

text-ratio

Type: `float`

Default Value: 0

Define the amount of text (of the total) present on successive lines when wrapping occurs

text-wrap-width

Type: `float`

Default Value: 0

Length of a chunk of text in characters before wrapping text

text-wrap-character

Type: `string`

Default Value:

Use this character instead of a space to wrap long text.

text-spacing

Type: `float`

Default Value: 0

Distance between repeated text labels on a line

text-character-spacing

Type: `float`

Default Value: 0

Horizontal spacing adjustment between characters in pixels

text-line-spacing

Type: `float`

Default Value: 0

Vertical spacing adjustment between lines in pixels

text-label-position-tolerance

Type: `float`

Default Value: 0

Allows the label to be displaced from its ideal position by a number of pixels

text-max-char-angle-delta

Type: `float`

Default Value: none

If present, the maximum angle change, in degrees, allowed between adjacent characters in a label. This will stop label placement around sharp corners.

text-fill

Type: `color`

Default Value: #000000

Specifies the color for the text

text-opacity

Type: `float`

Default Value: 1 (*Fully opaque*)

A number from 0 to 1 specifying the opacity for the text

text-halo-fill

Type: color

Default Value: #FFFFFF (*white*)

Specifies the color of the halo around the text.

text-halo-radius

Type: float

Default Value: 0 (*no halo*)

Specify the radius of the halo in pixels

text-dx

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the X axis. A positive value will shift the text right

text-dy

Type: float

Default Value: 0

Displace text by fixed amount, in pixels, +/- along the Y axis. A positive value will shift the text down

text-vertical-alignment

Type: keyword

Possible values: top middle bottom

Default Value: middle (*Default affected by value of dy; "bottom" for dy>0, "top" for dy<0.*)

Position of label relative to point position.

text-avoid-edges

Type: `boolean`

Default Value: `false`

Tell positioning algorithm to avoid labeling near intersection edges.

text-min-distance

Type: `float`

Default Value: `0`

Minimum permitted distance to the next text symbolizer.

text-min-padding

Type: `float`

Default Value: `0`

Determines the minimum amount of padding that a text symbolizer gets relative to other text

text-min-path-length

Type: `float`

Default Value: `0` (*place labels on all paths*)

Place labels only on paths longer than this value.

text-allow-overlap

Type: `boolean`

Default Value: `false` (*Do not allow text to overlap with other text - overlapping markers will not be shown.*)

Control whether overlapping text is shown or hidden.

text-orientation

Type: `float`

Default Value: `0`

Rotate the text.

text-placement

Type: `keyword`

Possible values: `point line vertex interior`

Default Value: `point`

Control the style of placement of a point versus the geometry it is attached to.

text-placement-type

Type: `keyword`

Possible values: `dummy simple`

Default Value: `dummy`

Re-position and/or re-size text to avoid overlaps. “simple” for basic algorithm (using text-placements string,) “dummy” to turn this feature off.

text-placements

Type: `string`

Default Value:

If “placement-type” is set to “simple”, use this “POSITIONS,[SIZES]” string. See TextSymbolizer docs for format.

text-transform

Type: `keyword`

Possible values: `none uppercase lowercase capitalize`

Default Value: `none`

Transform the case of the characters

text-horizontal-alignment

Type: keyword

Possible values: left middle right auto

Default Value: middle

The text's horizontal alignment from its centerpoint

text-align

Type: keyword

Possible values: left right center

Default Value: center

Set the text alignment.

building

building-fill

Type: color

Default Value: #FFFFFF

The color of the buildings walls.

building-fill-opacity

Type: float

Default Value: 1

The opacity of the building as a whole, including all walls.

building-height

Type: float

Default Value: 0

The height of the building in pixels.
