

Table of Contents

Render Recipes 1

 Creating Maps..... 1

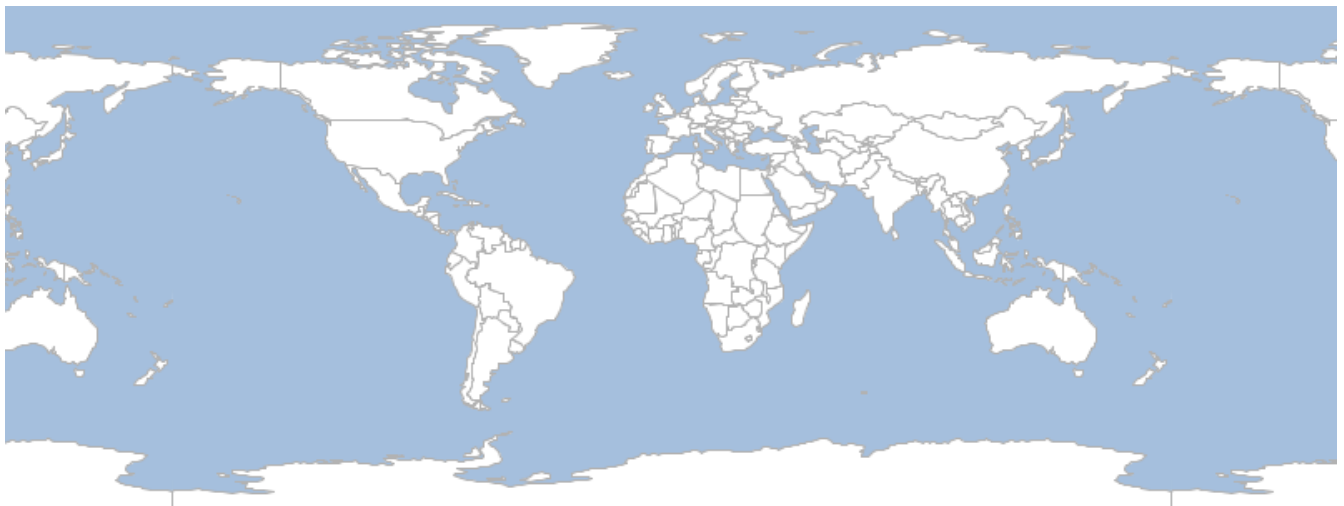
 Rendering Maps 1

Render Recipes

Creating Maps

Create a Map with Layers

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
File file = new File("map.png")
map.render(file)
```



Rendering Maps

Finding Renderers

Get all Renderers

```
List<Renderer> renderers = Renderers.list()
renderers.each { Renderer renderer ->
    println renderer.class.simpleName
}
```

ASCII
Base64
GeoTIFF
GIF
JPEG
Pdf
PNG
Svg

Get a Renderer

```
Renderer renderer = Renderers.find("png")  
println renderer.class.getSimpleName
```

PNG

Image

Render a Map to an image using an Image Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')  
Layer countries = workspace.get("countries")  
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)  
Layer ocean = workspace.get("ocean")  
ocean.style = new Fill("#a5bfdd")  
Map map = new Map(  
    width: 800,  
    height: 300,  
    layers: [ocean, countries]  
)  
Image png = new Image("png")  
BufferedImage image = png.render(map)
```



Render a Map to an OutputStream using the Image Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Image jpeg = new Image("jpeg")
File file = new File("map.jpeg")
jpeg.render(map, new FileOutputStream(file))
```



PNG

Render a Map to an Image using the PNG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
PNG png = new PNG()
BufferedImage image = png.render(map)
```



Render a Map to an OutputStream using the PNG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
PNG png = new PNG()
File file = new File("map.png")
png.render(map, new FileOutputStream(file))
```



JPEG

Render a Map to an Image using the JPEG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
JPEG jpeg = new JPEG()
BufferedImage image = jpeg.render(map)
```



Render a Map to an OutputStream using the JPEG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
JPEG jpeg = new JPEG()
File file = new File("map.jpeg")
jpeg.render(map, new FileOutputStream(file))
```



GIF

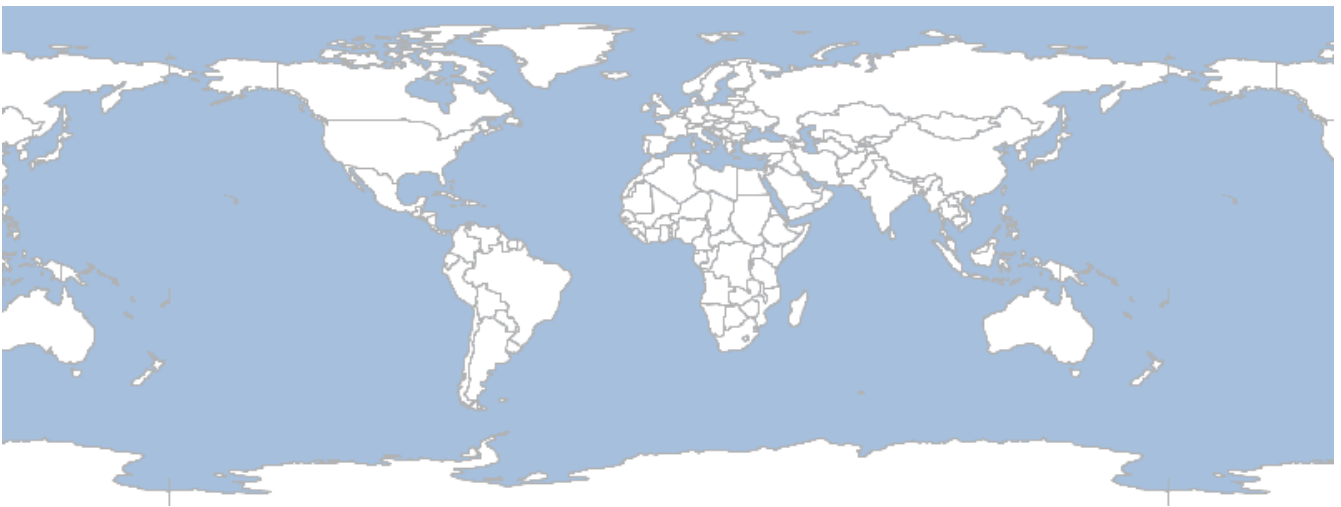
Render a Map to an Image using the GIF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfd9")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
GIF gif = new GIF()
BufferedImage image = gif.render(map)
```



Render a Map to an OutputStream using the GIF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
GIF gif = new GIF()
File file = new File("map.gif")
gif.render(map, new FileOutputStream(file))
```



GeoTIFF

Render a Map to an Image using the GeoTIFF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
GeoTIFF geotiff = new GeoTIFF()
BufferedImage image = geotiff.render(map)
```




Render a Map to an OutputStream using the GeoTIFF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
GeoTIFF geotiff = new GeoTIFF()
File file = new File("map.tif")
geotiff.render(map, new FileOutputStream(file))
```



ASCII

Render a Map to an string using the ASCII Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfd9")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
ASCII ascii = new ASCII(width: 60)
String asciiStr = ascii.render(map)
println asciiStr
```

```
(((((
(((
.....:^(^..??^!^:..!(^...(((^.(^.....:^(^..??^
.....(:((.....((.(((.((((:(.....(:((.....
?..^((((^(^.....((((.(.....?..^((((^(^
.^(((((((^(^.....^(((((:...^..?..^...^(((((((^(^
((.(((((((.....:(((((((^(^(!^^^.....((.(((((((
(((((((.....!(((((((.....(((((((
(((((((^((((.(.(((((((.....:....(.^^((((((((^(
^(((((((.....(((((((^:^^...^(((.(^^(^(((((((
(((((((.....!:(((((((.....(((((((
((.(((((((.....(((((((...(((((((^(.(
^((((((((.....(((((((.^.(
....^((((((((.....(((((((.^^(
....(((((((.....(((((((^:(((((((.....
(^.(
(((((((.....(((((((^(
(((((((.....(((((((
(((((((^(
(((((((.....
....^(((((((?(((((((?....(.....^
....(((
.....
.....
```

Render a Map to an text file using the ASCII Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
ASCII ascii = new ASCII(width: 60)
File file = new File("map.txt")
ascii.render(map, new FileOutputStream(file))
```

```
(((((
(((
.....:^(^..??^^!^:..!(^...(((^..^((^.....:^(^..??^
.....(:((.....((.(.(((:(.....(:((.....
?..^((((^(^.....((((.(.....?..^((((^(^
.^(((((((^(^.....^(((((:...^..?..^...^(((((((^(^
((.(((((((.....:(((((((^(^(!^^^.....((.(((((((
(((((((((((((((((!((((((((.....((((((((((((
((((((((^((((.(.(((((((.....:....(.^..^((((((((^(
^((((((((((((((((.(.(((((((^:~^...^(((.(^(^(^(
(((((((((((((((((!:(((((((.....((((((((((((((((
((.((((((((((((((((.....(((((((...(((((((^(.(.((((
^(((((((((((((((((.....(((((((.^..((((((((^(
....^(((((((((((((((((.....(((((((.^^(
....((((((((((((((((.....(((((((^:((((((((.....
(^..((((((((((((((((.....((((((((((((((((^(
((((((((((((((((.....((((((((((((((((((((
((((((((((((((((^(((((((((((((((((((((
((((((((((((((((((((((((((((((((((((
....^(((((((((((((((((?((((((((((((?....^
....(((.....(((.....(((.....
.....
```

Base64

Render a Map to an string using the Base64 Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Base64 base64 = new Base64()
String base64Str = base64.render(map)
println base64Str
```

```
image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAYAAAEsC...
```

Render a Map to an text file using the Base64 Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Base64 base64 = new Base64()
File file = new File("map.txt")
base64.render(map, new FileOutputStream(file))
```

```
iVBORw0KGgoAAAANSUhEUgAAAYAAAEsCAYAAAA7Ldc6AACAAE...
```

PDF

Render a Map to a PDF Document using the PDF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Pdf pdf = new Pdf()
com.lowagie.text.Document document = pdf.render(map)
```



Render a Map to a PDF file using the PDF Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Pdf pdf = new Pdf()
File file = new File("map.pdf")
pdf.render(map, new FileOutputStream(file))
```



SVG

Render a Map to a SVG Document using the SVG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Svg svg = new Svg()
org.w3c.dom.Document document = svg.render(map)
```




Render a Map to a SVG file using the SVG Renderer

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new Fill("#ffffff") + new Stroke("#b2b2b2", 0.5)
Layer ocean = workspace.get("ocean")
ocean.style = new Fill("#a5bfdd")
Map map = new Map(
    width: 800,
    height: 300,
    layers: [ocean, countries]
)
Svg svg = new Svg()
File file = new File("map.svg")
svg.render(map, new FileOutputStream(file))
```

render_svg_file.svg

file:///Users/jericks/Projects/geoscript-cookbook/src/docs/asciidoc/output/render_svg_file.svg
Reader



×

📄

📁

🔄

⬇️

📄 1

🔔

🕒

💬 0

🕒 0

⚠️ 0

🔍 Search

📁 Elements

🌐 Network

📄 Resour...

🕒 Timeli...

🔧 Debug...

📁 Storage

📄 Console

+

svg

g

g

path

rule="evenodd" stroke="none"></path>

<path fill="none" d="M438.1356 80.0012 L438.254 81.1033 L436.759 81.4169

L435.0334 81.9288 L434.1053 80.8082 L434.6036 79.9136 L435.5878 79.6553

L436.5285 79.4939 L438.1356 80.0012 Z" clip-path="url(#clipPath1)" fill-

rule="evenodd"></path>

<path fill="white" d="M405.2444 117.1774 L405.2635 118.2377 L407.1124

118.0746 L407.117 121.913 L406.2057 123.0262 L406.0638 124.0531 L404.5833

124.3175 L402.3092 124.4607 L400.6248 125.1185 L399.5562 125.1262 L398.2227

125.0436 L396.6649 125.735 L395.0538 127.0031 L394.1287 127.7706 L393.3227

127.5459 L392.6214 129.0956 L391.2984 130.4769 L390.8824 131.7479 L390.9928

132.7154 L389.9159 133.1727 L388.8892 132.6153 L387.2954 133.0879 L386.1077

Node

Styles

Layers

path — Style Attribute

No Properties — Click to Edit

Media: all

path — HTML Attributes

fill: #ffffff;

clip-path: url(#clipPath1);

fill-rule: evenodd;

+

Filter Styles