

# Table of Contents

Carto Recipes ..... 1

    Items ..... 1

# Carto Recipes

The Carto classes are in the [geoscript.carto](#) package.

The Carto package contains classes for creating cartographic documents. All items are added to the document with x and y coordinates whose origin is the upper left and width and a height.

## Items

### Adding a Map

*Add a map*

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180, -85, 180, 85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .map(new MapItem(20, 20, pageSize.width - 40, pageSize.height - 40).map
(map))
        .build(outputStream)

}
```



## **Adding a Text**

```

Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .text(new TextItem(20,20, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 42))
            .verticalAlign(VerticalAlign.MIDDLE)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .map(new MapItem(20, 80, pageSize.width - 40, pageSize.height - 100).map
(map))
        .build(outputStream)
}

```

# World Map



**Adding a Rectangle**

## Add a rectangle

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .rectangle(new RectangleItem(10,10, pageSize.width - 20, pageSize.height -
20))
        .rectangle(new RectangleItem(20,20, pageSize.width - 40, 60))
        .rectangle(new RectangleItem(20,90, pageSize.width - 40, pageSize.height -
110))
        .text(new TextItem(20,20, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 32))
            .verticalAlign(VerticalAlign.MIDDLE)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .map(new MapItem(30, 100, pageSize.width - 60, pageSize.height - 120).map
(map))
        .build(outputStream)
}
```

## World Map



**Adding a North Arrow**

### Add a north arrow

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height - 1)
            .fillColor(Color.WHITE)
        )
        .map(new MapItem(20, 20, pageSize.width - 40, pageSize.height - 40).map(map))
        .northArrow(new NorthArrowItem(pageSize.width - 60, pageSize.height - 80, 40,
60))
        .build(outputStream)
}
```





## Adding a Date

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height - 1)
            .fillColor(Color.WHITE)
        )
        .text(new TextItem(20,15, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 42))
            .verticalAlign(VerticalAlign.TOP)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .dateText(new DateTextItem(20,58, pageSize.width - 40, 20)
            .font(new Font("Arial", Font.ITALIC, 18))
            .verticalAlign(VerticalAlign.BOTTOM)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .map(new MapItem(20, 80, pageSize.width - 40, pageSize.height - 100).map(map))
        .build(outputStream)
}
```

# World Map

03/15/2020



**Adding Scale Text**

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height - 1)
            .fillColor(Color.WHITE)
        )
        .text(new TextItem(20,15, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 42))
            .verticalAlign(VerticalAlign.TOP)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .scaleText(new ScaleTextItem(20,58, pageSize.width - 40, 20)
            .map(map)
            .format("#")
            .prefixText("Scale: ")
            .font(new Font("Arial", Font.ITALIC, 18))
            .verticalAlign(VerticalAlign.BOTTOM)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .map(new MapItem(20, 80, pageSize.width - 40, pageSize.height - 100).map(map))
        .build(outputStream)
}
```

# World Map

*Scale: 1:238541766*



**Adding a Line**

```

Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .text(new TextItem(20,20, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 42))
            .verticalAlign(VerticalAlign.MIDDLE)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .line(new LineItem(20, 70, pageSize.width - 40, 1)
            .strokeWidth(2)
            .strokeColor(Color.DARK_GRAY)
        )
        .map(new MapItem(20, 80, pageSize.width - 40, pageSize.height - 100).map
(map))
        .build(outputStream)
}

```

# World Map



**Adding a Grid**

*Adding a grid is a nice way to placing items.*

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .grid(new GridItem(0,0,pageSize.width, pageSize.height)
            .size(20)
            .strokeColor(Color.GRAY)
            .strokeWidth(1.0)
        )
        .text(new TextItem(20,20, pageSize.width - 40, 60)
            .text("World Map")
            .font(new Font("Arial", Font.BOLD, 42))
            .verticalAlign(VerticalAlign.MIDDLE)
            .horizontalAlign(HorizontalAlign.CENTER)
        )
        .map(new MapItem(20, 80, pageSize.width - 40, pageSize.height - 100).map
(map))
        .build(outputStream)

}
```



# World Map



**Adding a Paragraph**

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File
('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject(
"EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height
- 1)
            .fillColor(Color.WHITE)
        )
        .map(new MapItem(20, 20, pageSize.width - 40, pageSize.height - 100
).map(map))
        .paragraph(new ParagraphItem(20, pageSize.height - 60, pageSize.width
- 40, 60)
            .font(new Font("Arial", Font.PLAIN, 12))
            .color(Color.BLACK)
            .text("""Natural Earth is a public domain map dataset available at
1:10m, 1:50m, and 1:110 million scales.
Featuring tightly integrated vector and raster data, with Natural Earth you can make a
variety of visually pleasing,
well-crafted maps with cartography or GIS software.
"""))
        )
        .build(outputStream)
}
```



Natural Earth is a public domain map dataset available at 1:10m, 1:50m, and 1:110 million scales. Featuring tightly integrated vector and raster data, with Natural Earth you can make a variety of visually pleasing, well-crafted maps with cartography or GIS software.

## Adding an Image

## Adding an image

```
Workspace workspace = new GeoPackage('src/main/resources/data.gpkg')
Layer countries = workspace.get("countries")
countries.style = new SLDReader().read(new File('src/main/resources/countries.sld'))
Layer ocean = workspace.get("ocean")
ocean.style = new SLDReader().read(new File('src/main/resources/ocean.sld'))
Map map = new Map(
    layers: [ocean, countries],
    bounds: new Bounds(-180,-85,180,85, "EPSG:4326").reproject("EPSG:3857"),
    projection: new Projection("EPSG:3857")
)

File file = new File("map.png")
file.withOutputStream { OutputStream outputStream ->

    PageSize pageSize = PageSize.LETTER_LANDSCAPE

    CartoFactories.findByName("png")
        .create(pageSize)
        .rectangle(new RectangleItem(0, 0, pageSize.width - 1, pageSize.height -
1)
            .fillColor(Color.WHITE)
        )
        .map(new MapItem(20, 20, pageSize.width - 40, pageSize.height - 40).map
(map))
        .image(new ImageItem(pageSize.width - 100, pageSize.height - 100, 80, 80)
            .path(new File("src/main/resources/image.png")))
        )
        .build(outputStream)
}
```

