

# geo-shell

Jared Erickson

Version 0.7-SNAPSHOT

# Table of Contents

Introduction.....	1
Workspace .....	1
Basics.....	1
Layers .....	2
Layer .....	2
Basics.....	2
Geoprocessing.....	3
Graticule.....	14
Format.....	21
Open .....	21
List.....	21
Close .....	22
Rasters .....	22
Raster.....	22
Tile .....	22
Style .....	22
Create .....	22
Vector Default .....	22
Vector Gradient .....	23
Vector Unique Values .....	23
Vector Unique Values From Text File .....	23
Raster Default.....	24
Raster Color Map .....	24
Map .....	24
Other .....	24
Unzip.....	24
Open .....	25

# Introduction

geo-shell is an interactive shell for geospatial analysis.

geo-shell has modules for dealing with **vectors**, **rasters**, **tiles**, **maps**, and **styles**.

For **vector** layers, you can use **workspace** commands access layers of spatial data in datasets like shapefiles, geopackages, or postgis databases. With **layer** commands you can perform geoprocessing functions like calculating centroids or buffer features.

For **raster** layers, you can use **format** commands access individual rasters from geotifs or world images. With **raster** commands you can perform mosaic, raster algebra, or crop functions.

The **tile** commands let you create tile layers, get tiles, and get rasters from tiles.

The **style** commands let you create styles for vector layers and raster.

The **map** commands allow you to visualize vector, raster, and tile layers.

## Workspace

Workspaces hold vector layers. A Workspace can be a GeoPackage database, a directory of Shapefiles, or a PostGIS database.

## Basics

You can open, close, and list Workspaces. The earliest Workspace to open is an in memory Workspace.

*Table 1. Open a Workspace*

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Workspace name	true		
params	The connection parameters	true		

```
geo-shell> workspace open --name mem --params memory
Workspace mem opened!
```

You can open a Workspace with --params or connection parameters. You can give it a name with --name flag.

*List open Workspaces*

```
geo-shell> workspace list
mem = Memory
```

Listing open Workspaces give you the name and the type Workspace.

Table 2. Close a Workspace

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Workspace name	true		

```
geo-shell> workspace close --name mem
Workspace mem closed!
```

Once you close a Workspace by name it will no longer appear with the list command.

## Layers

In this example, we will open a GeoPackage database filled with data from Natural Earth.

### Open a Workspace

```
geo-shell> workspace open --name naturalearth --params src/test/resources/naturalearth.gpkg
Workspace naturalearth opened!
```

### List open Workspaces

```
geo-shell> workspace layers --name naturalearth
countries
ocean
places
states
```

### Close a Workspace

```
geo-shell> workspace close --name naturalearth
Workspace naturalearth closed!
```

## Layer

### Basics

#### Open

Open a Layer from a Workspace

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
layer	The Layer name	true		
name	The name	false		

## Close

Close an Open Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		

## List

List open Layers

# Geoprocessing

## Random Points

Name	Description	Mandatory	Specified Default	Unspecified Default
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
number	The number of points	true		
geometry	The geometry or bounds in which to create the points	true		
projection	The projection	true		
id-field	The id field name	false	id	id
geometry-field	The geometry field name	false	the_geom	the_geom
grid	Whether to create points in a grid	false	false	false
constrained-to-circle	Whether points should be constrained to a circle	false	false	false
gutter-fraction	The size of gutter between cells	false	0	0

```
geo-shell> workspace open --name layers --params memory  
Workspace layers opened!
```

```
geo-shell> layer random --output-workspace layers --output-name points --geometry -180,-90,180,90  
--number 100 --projection EPSG:4326
```

Done!

```
geo-shell> style vector default --layer points --color #1E90FF --file examples/points.sld  
Default Vector Style for points written to /home/travis/build/jericks/geo-shell/examples/points.sld!
```

```
geo-shell> layer style set --name points --style examples/points.sld  
Style /home/travis/build/jericks/geo-shell/examples/points.sld set on points
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg  
Workspace naturalearth opened!
```

```
geo-shell> layer open --workspace naturalearth --layer countries --name countries  
Opened Workspace naturalearth Layer countries as countries
```

```
geo-shell> layer style set --name countries --style examples/countries.sld  
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries
```

```
geo-shell> layer open --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean
```

```
geo-shell> layer style set --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

```
geo-shell> map open --name randomMap  
Map randomMap opened!
```

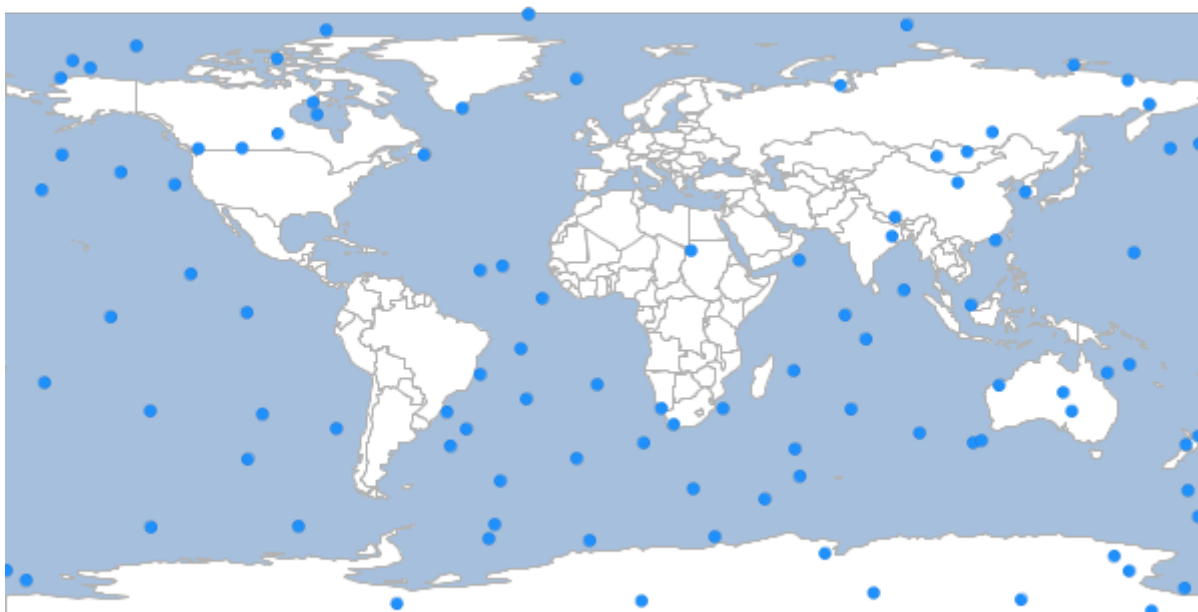
```
geo-shell> map add layer --name randomMap --layer ocean  
Added ocean layer to map randomMap
```

```
geo-shell> map add layer --name randomMap --layer countries  
Added countries layer to map randomMap
```

```
geo-shell> map add layer --name randomMap --layer points  
Added points layer to map randomMap
```

```
geo-shell> map draw --name randomMap --file examples/random_points.png  
Done drawing /home/travis/build/jericks/geo-shell/examples/random_points.png!
```

```
geo-shell> map close --name randomMap  
Map randomMap closed!
```



## Buffer

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
distance	The buffer distance	true		

```
geo-shell> workspace open --name layers --params memory
```

Workspace layers opened!

```
geo-shell> layer random --output-workspace layers --output-name points --geometry -180,-90,180,90  
--number 100 --projection EPSG:4326
```

Done!

```
geo-shell> layer buffer --input-name points --output-workspace layers --output-name buffers  
--distance 10
```

Done!

```
geo-shell> style vector default --layer points --color #1E90FF --file examples/points.sld
```

Default Vector Style for points written to /home/travis/build/jericks/geo-shell/examples/points.sld!

```
geo-shell> style vector default --layer buffers --color #1E90FF --opacity 0.25 --file  
examples/buffers.sld
```

Default Vector Style for buffers written to /home/travis/build/jericks/geo-shell/examples/buffers.sld!

```
geo-shell> layer style set --name points --style examples/points.sld
```

Style /home/travis/build/jericks/geo-shell/examples/points.sld set on points

```
geo-shell> layer style set --name buffers --style examples/buffers.sld
```

Style /home/travis/build/jericks/geo-shell/examples/buffers.sld set on buffers

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg
```

Workspace naturalearth opened!

```
geo-shell> layer open --workspace naturalearth --layer countries --name countries
```

Opened Workspace naturalearth Layer countries as countries

```
geo-shell> layer style set --name countries --style examples/countries.sld
```

Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

```
geo-shell> layer open --workspace naturalearth --layer ocean --name ocean
```

Opened Workspace naturalearth Layer ocean as ocean

```
geo-shell> layer style set --name ocean --style examples/ocean.sld
```

Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

```
geo-shell> map open --name map
```

Map map opened!

```
geo-shell> map add layer --name map --layer ocean
```

Added ocean layer to map map

```
geo-shell> map add layer --name map --layer countries
```

Added countries layer to map map

```
geo-shell> map add layer --name map --layer buffers
```

Added buffers layer to map map

```
geo-shell> map add layer --name map --layer points
```

Added points layer to map map

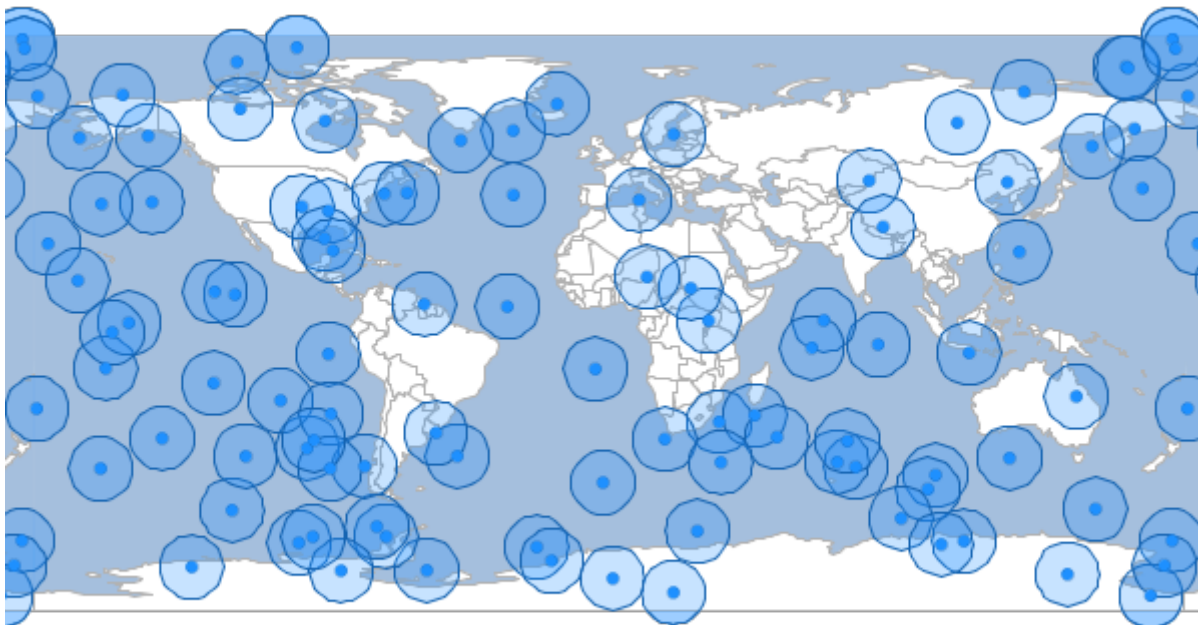
```
geo-shell> map draw --name map --file examples/layer_buffer.png
```

Done drawing /home/travis/build/jericks/geo-shell/examples/layer\_buffer.png!

```
geo-shell> map close --name map
```

Map map closed!





## Centroid

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg
Workspace naturalearth opened!
```

```
geo-shell> layer open --workspace naturalearth --layer countries --name countries
Opened Workspace naturalearth Layer countries as countries
```

```
geo-shell> layer style set --name countries --style examples/countries.sld
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries
```

```
geo-shell> layer centroid --input-name countries --output-name centroids --output-workspace
layers
Done!
```

```
geo-shell> style vector default --layer centroids --color #1E90FF --file examples/centroids.sld
Default Vector Style for centroids written to /home/travis/build/jericks/geo-
```

shell/examples/centroids.sld!

```
geo-shell> layer style set --name centroids --style examples/centroids.sld  
Style /home/travis/build/jericks/geo-shell/examples/centroids.sld set on centroids
```

```
geo-shell> layer open --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean
```

```
geo-shell> layer style set --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

```
geo-shell> map open --name map  
Map map opened!
```

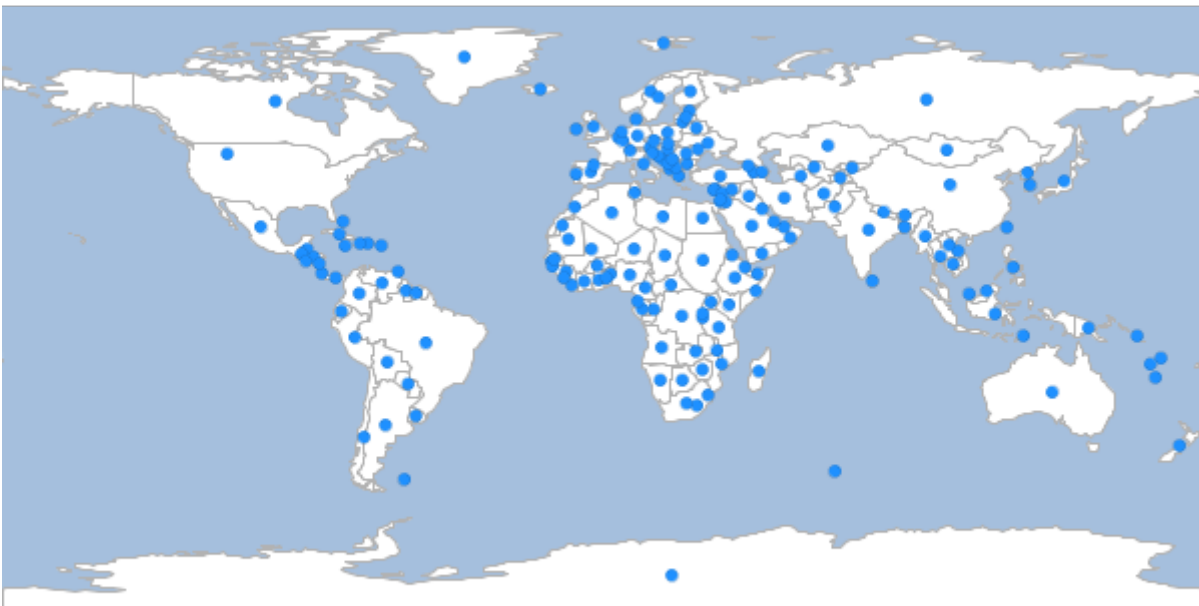
```
geo-shell> map add layer --name map --layer ocean  
Added ocean layer to map map
```

```
geo-shell> map add layer --name map --layer countries  
Added countries layer to map map
```

```
geo-shell> map add layer --name map --layer centroids  
Added centroids layer to map map
```

```
geo-shell> map draw --name map --file examples/layer_centroid.png  
Done drawing /home/travis/build/jericks/geo-shell/examples/layer_centroid.png!
```

```
geo-shell> map close --name map  
Map map closed!
```



## Interior Point

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg
Workspace naturalearth opened!
```

```
geo-shell> layer open --workspace naturalearth --layer countries --name countries
Opened Workspace naturalearth Layer countries as countries
```

```
geo-shell> layer style set --name countries --style examples/countries.sld
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries
```

```
geo-shell> layer interiorpoint --input-name countries --output-name interiorpoints --output
-workspace layers
Done!
```

```
geo-shell> style vector default --layer interiorpoints --color #1E90FF --file
examples/interiorpoints.sld
Default Vector Style for interiorpoints written to /home/travis/build/jericks/geo-
shell/examples/interiorpoints.sld!
```

```
geo-shell> layer style set --name interiorpoints --style examples/interiorpoints.sld
Style /home/travis/build/jericks/geo-shell/examples/interiorpoints.sld set on interiorpoints
```

```
geo-shell> layer open --workspace naturalearth --layer ocean --name ocean
Opened Workspace naturalearth Layer ocean as ocean
```

```
geo-shell> layer style set --name ocean --style examples/ocean.sld
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

```
geo-shell> map open --name map
Map map opened!
```

```
geo-shell> map add layer --name map --layer ocean
Added ocean layer to map map
```

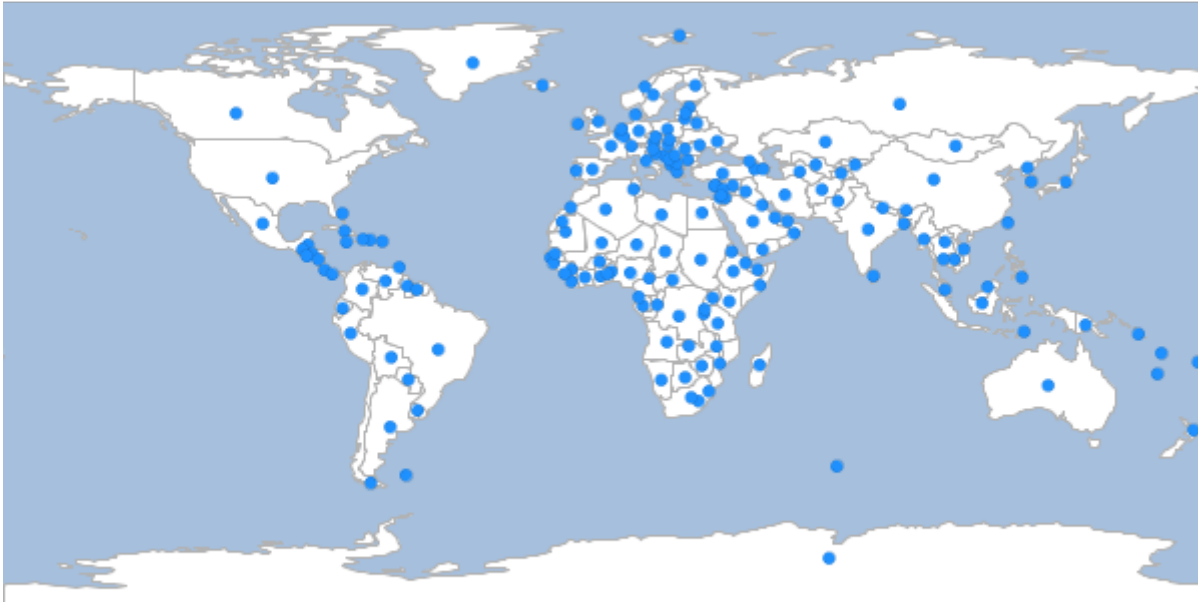
```
geo-shell> map add layer --name map --layer countries
Added countries layer to map map
```

```
geo-shell> map add layer --name map --layer interiorpoints
```

Added interiorpoints layer to map map

```
geo-shell> map draw --name map --file examples/layer_interiorpoint.png  
Done drawing /home/travis/build/jericks/geo-shell/examples/layer_interiorpoint.png!
```

```
geo-shell> map close --name map  
Map map closed!
```



## Extent

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
geometry-field	The geometry field name	false	the_geom	the_geom

```
geo-shell> workspace open --name layers --params memory  
Workspace layers opened!
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg  
Workspace naturalearth opened!
```

```
geo-shell> layer style set --name states --style examples/states.sld  
Unable to find Layer states
```

geo-shell> **layer open** --workspace naturalearth --layer states --name states

Opened Workspace naturalearth Layer states as states

geo-shell> **layer extent** --input-name states --output-workspace layers --output-name usa

Done!

geo-shell> **style vector default** --layer usa --color #1E90FF --opacity 0.25 --file examples/extent.sld

Default Vector Style for usa written to /home/travis/build/jericks/geo-shell/examples/extent.sld!

geo-shell> **layer style set** --name usa --style examples/extent.sld

Style /home/travis/build/jericks/geo-shell/examples/extent.sld set on usa

geo-shell> **layer open** --workspace naturalearth --layer countries --name countries

Opened Workspace naturalearth Layer countries as countries

geo-shell> **layer style set** --name countries --style examples/countries.sld

Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

geo-shell> **layer open** --workspace naturalearth --layer ocean --name ocean

Opened Workspace naturalearth Layer ocean as ocean

geo-shell> **layer style set** --name ocean --style examples/ocean.sld

Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

geo-shell> **map open** --name map

Map map opened!

geo-shell> **map add layer** --name map --layer ocean

Added ocean layer to map map

geo-shell> **map add layer** --name map --layer countries

Added countries layer to map map

geo-shell> **map add layer** --name map --layer states

Added states layer to map map

geo-shell> **map add layer** --name map --layer usa

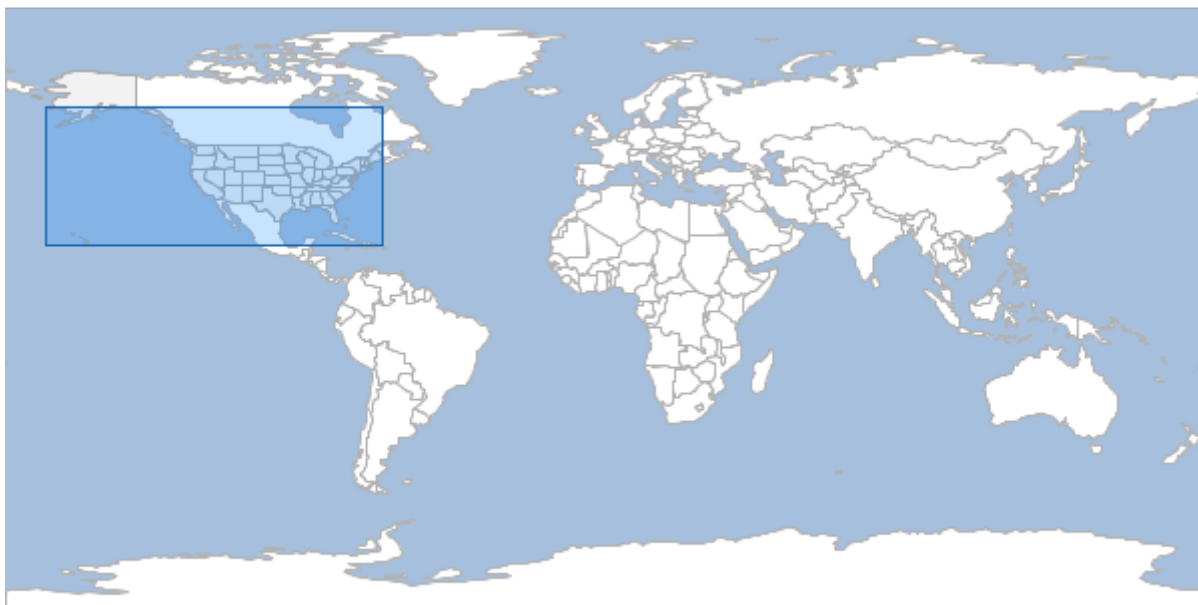
Added usa layer to map map

geo-shell> **map draw** --name map --file examples/layer\_extent.png

Done drawing /home/travis/build/jericks/geo-shell/examples/layer\_extent.png!

geo-shell> **map close** --name map

Map map closed!



## Extents

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg
Workspace naturalearth opened!
```

```
geo-shell> layer style set --name states --style examples/states.sld
Unable to find Layer states
```

```
geo-shell> layer open --workspace naturalearth --layer states --name states
Opened Workspace naturalearth Layer states as states
```

```
geo-shell> layer extents --input-name states --output-workspace layers --output-name state_extents
Done!
```

```
geo-shell> style vector default --layer state_extents --color #1E90FF --opacity 0.25 --file
examples/extent.sld
Default Vector Style for state_extents written to /home/travis/build/jericks/geo-
```

shell/examples/extent.sld!

geo-shell> **layer style set** --name state\_extents --style examples/extent.sld  
Style /home/travis/build/jericks/geo-shell/examples/extent.sld set on state\_extents

geo-shell> **layer open** --workspace naturalearth --layer countries --name countries  
Opened Workspace naturalearth Layer countries as countries

geo-shell> **layer style set** --name countries --style examples/countries.sld  
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

geo-shell> **layer open** --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean

geo-shell> **layer style set** --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

geo-shell> **map open** --name map  
Map map opened!

geo-shell> **map add layer** --name map --layer ocean  
Added ocean layer to map map

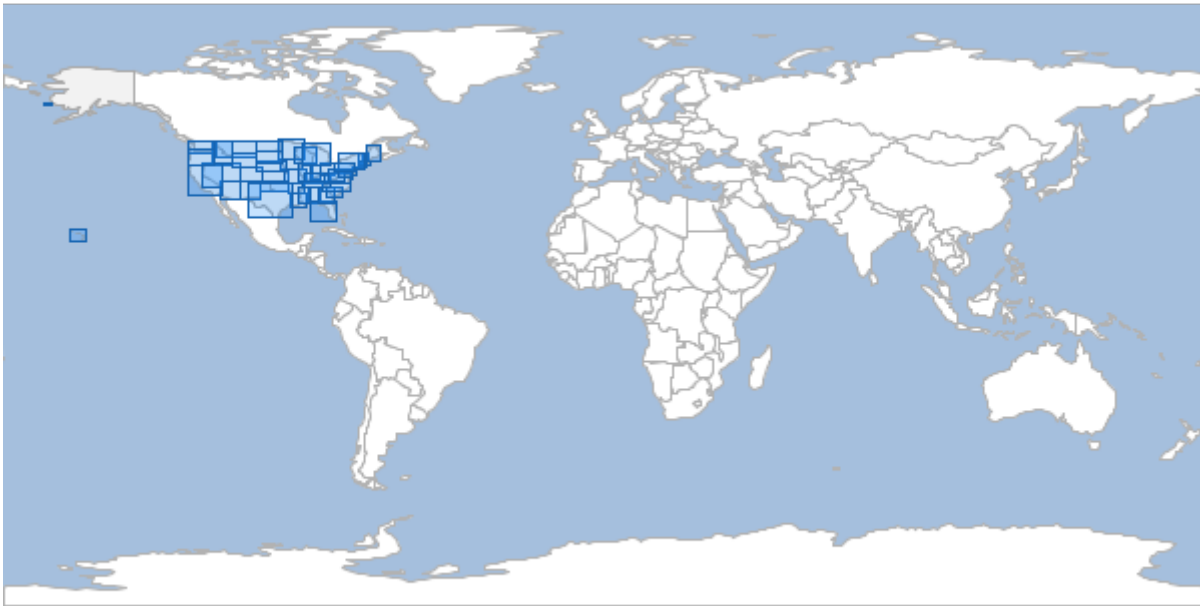
geo-shell> **map add layer** --name map --layer countries  
Added countries layer to map map

geo-shell> **map add layer** --name map --layer states  
Added states layer to map map

geo-shell> **map add layer** --name map --layer state\_extents  
Added state\_extents layer to map map

geo-shell> **map draw** --name map --file examples/layer\_extents.png  
Done drawing /home/travis/build/jericks/geo-shell/examples/layer\_extents.png!

geo-shell> **map close** --name map  
Map map closed!



## Graticule

### Square

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
name	The new Layer name	true		
bounds	The bounds	true		
length	The length	true		
spacing	The spacing	false	-1	-1

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> layer graticule square --workspace layers --name squares --bounds -180,-90,180,90
--length 20
Created Square Graticule Layer squares!
```

```
geo-shell> style vector default --layer squares --color #1E90FF --opacity 0.30 --file
examples/squares.sld
Default Vector Style for squares written to /home/travis/build/jericks/geo-shell/examples/squares.sld!
```

```
geo-shell> layer style set --name squares --style examples/squares.sld
```



Style /home/travis/build/jericks/geo-shell/examples/squares.sld set on squares

geo-shell> **workspace open** --name naturalearth --params examples/naturalearth.gpkg  
Workspace naturalearth opened!

geo-shell> **layer open** --workspace naturalearth --layer countries --name countries  
Opened Workspace naturalearth Layer countries as countries

geo-shell> **layer style set** --name countries --style examples/countries.sld  
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

geo-shell> **layer open** --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean

geo-shell> **layer style set** --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

geo-shell> **map open** --name graticule  
Map graticule opened!

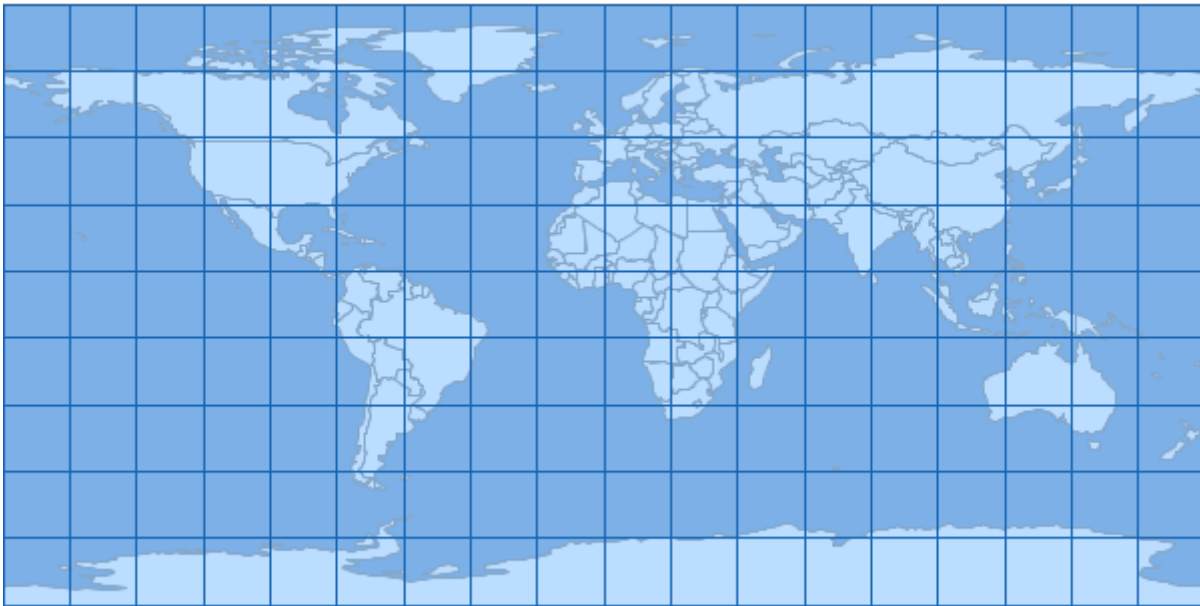
geo-shell> **map add layer** --name graticule --layer ocean  
Added ocean layer to map graticule

geo-shell> **map add layer** --name graticule --layer countries  
Added countries layer to map graticule

geo-shell> **map add layer** --name graticule --layer squares  
Added squares layer to map graticule

geo-shell> **map draw** --name graticule --file examples/square\_graticules.png  
Done drawing /home/travis/build/jericks/geo-shell/examples/square\_graticules.png!

geo-shell> **map close** --name graticule  
Map graticule closed!



## Rectangle

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
name	The new Layer name	true		
bounds	The bounds	true		
width	The width	true		
height	The height	true		
spacing	The spacing	false	-1	-1

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> layer graticule rectangle --workspace layers --name rectangles --bounds -180,-90,180,90
--width 20 --height 10
Created Rectangle Graticule Layer rectangles!
```

```
geo-shell> style vector default --layer rectangles --color #1E90FF --opacity 0.30 --file
examples/rectangles.sld
Default Vector Style for rectangles written to /home/travis/build/jericks/geo-shell/examples/rectangles.sld!
```

```
geo-shell> layer style set --name rectangles --style examples/rectangles.sld
Style /home/travis/build/jericks/geo-shell/examples/rectangles.sld set on rectangles
```

geo-shell> **workspace open** --name naturalearth --params examples/naturalearth.gpkg

Workspace naturalearth opened!

geo-shell> **layer open** --workspace naturalearth --layer countries --name countries

Opened Workspace naturalearth Layer countries as countries

geo-shell> **layer style set** --name countries --style examples/countries.sld

Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

geo-shell> **layer open** --workspace naturalearth --layer ocean --name ocean

Opened Workspace naturalearth Layer ocean as ocean

geo-shell> **layer style set** --name ocean --style examples/ocean.sld

Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

geo-shell> **map open** --name graticule

Map graticule opened!

geo-shell> **map add layer** --name graticule --layer ocean

Added ocean layer to map graticule

geo-shell> **map add layer** --name graticule --layer countries

Added countries layer to map graticule

geo-shell> **map add layer** --name graticule --layer rectangles

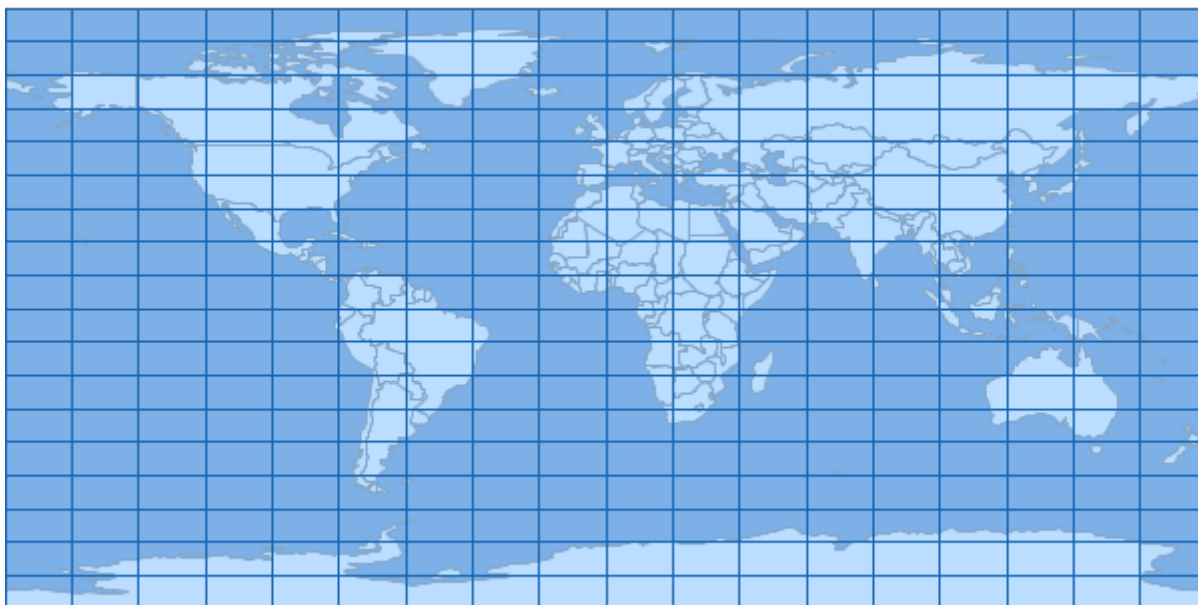
Added rectangles layer to map graticule

geo-shell> **map draw** --name graticule --file examples/rectangle\_graticules.png

Done drawing /home/travis/build/jericks/geo-shell/examples/rectangle\_graticules.png!

geo-shell> **map close** --name graticule

Map graticule closed!



## Oval

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
name	The new Layer name	true		
bounds	The bounds	true		
size	The size	true		

```
geo-shell> workspace open --name layers --params memory
Workspace layers opened!
```

```
geo-shell> layer graticule oval --workspace layers --name ovals --bounds -180,-90,180,90 --size 20
Created Oval Graticule Layer ovals!
```

```
geo-shell> style vector default --layer ovals --color #1E90FF --opacity 0.30 --file examples/ovals.sld
Default Vector Style for ovals written to /home/travis/build/jericks/geo-shell/examples/ovals.sld!
```

```
geo-shell> layer style set --name ovals --style examples/ovals.sld
Style /home/travis/build/jericks/geo-shell/examples/ovals.sld set on ovals
```

```
geo-shell> workspace open --name naturalearth --params examples/naturalearth.gpkg
Workspace naturalearth opened!
```

```
geo-shell> layer open --workspace naturalearth --layer countries --name countries
Opened Workspace naturalearth Layer countries as countries
```

```
geo-shell> layer style set --name countries --style examples/countries.sld
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries
```

```
geo-shell> layer open --workspace naturalearth --layer ocean --name ocean
Opened Workspace naturalearth Layer ocean as ocean
```

```
geo-shell> layer style set --name ocean --style examples/ocean.sld
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

```
geo-shell> map open --name graticule
Map graticule opened!
```

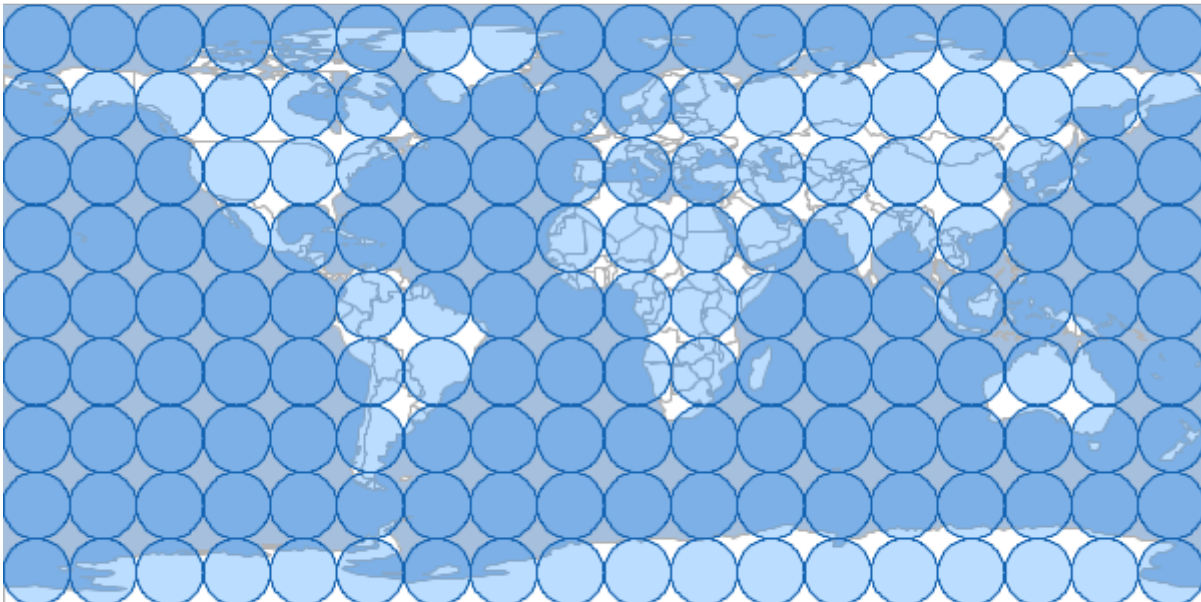
```
geo-shell> map add layer --name graticule --layer ocean
Added ocean layer to map graticule
```

```
geo-shell> map add layer --name graticule --layer countries
Added countries layer to map graticule
```

```
geo-shell> map add layer --name graticule --layer ovals
Added ovals layer to map graticule
```

```
geo-shell> map draw --name graticule --file examples/oval_graticules.png
Done drawing /home/travis/build/jericks/geo-shell/examples/oval_graticules.png!
```

```
geo-shell> map close --name graticule
Map graticule closed!
```



## Hexagon

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
name	The new Layer name	true		
bounds	The bounds	true		
length	The length	true		
spacing	The spacing	false	5	5
orientation	The orientation (flat or angled)	false	flat	flat

geo-shell> **workspace open** --name layers --params memory  
Workspace layers opened!

geo-shell> **layer graticule hexagon** --workspace layers --name hexagons --bounds -180,-90,180,90 --length 10  
Created Hexagon Graticule Layer hexagons!

geo-shell> **style vector default** --layer hexagons --color #1E90FF --opacity 0.30 --file examples/hexagons.sld  
Default Vector Style for hexagons written to /home/travis/build/jericks/geo-shell/examples/hexagons.sld!

geo-shell> **layer style set** --name hexagons --style examples/hexagons.sld  
Style /home/travis/build/jericks/geo-shell/examples/hexagons.sld set on hexagons

geo-shell> **workspace open** --name naturalearth --params examples/naturalearth.gpkg  
Workspace naturalearth opened!

geo-shell> **layer open** --workspace naturalearth --layer countries --name countries  
Opened Workspace naturalearth Layer countries as countries

geo-shell> **layer style set** --name countries --style examples/countries.sld  
Style /home/travis/build/jericks/geo-shell/examples/countries.sld set on countries

geo-shell> **layer open** --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean

geo-shell> **layer style set** --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean

geo-shell> **map open** --name graticule  
Map graticule opened!

geo-shell> **map add layer** --name graticule --layer ocean  
Added ocean layer to map graticule

geo-shell> **map add layer** --name graticule --layer countries

Added countries layer to map graticule

```
geo-shell> map add layer --name graticule --layer hexagons
```

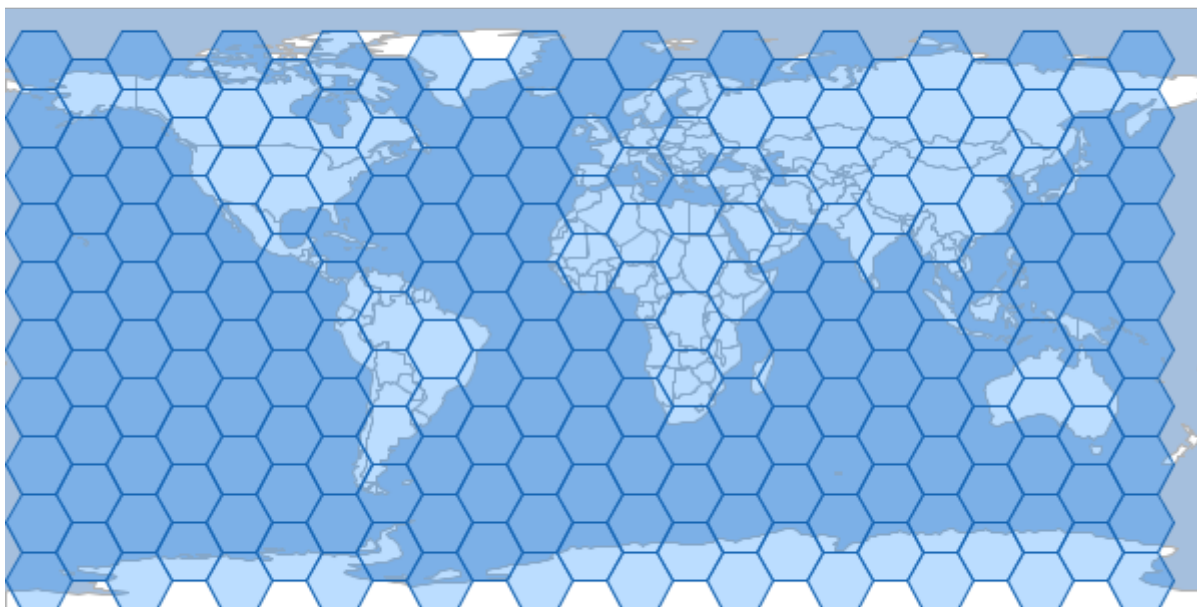
Added hexagons layer to map graticule

```
geo-shell> map draw --name graticule --file examples/hexagon_graticules.png
```

Done drawing /home/travis/build/jericks/geo-shell/examples/hexagon\_graticules.png!

```
geo-shell> map close --name graticule
```

Map graticule closed!



## Format

### Open

Open a Format

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Format name	false		
input	The input string	true		

### List

List open Formats

# Close

Close a Format

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Format name	true		

# Rasters

List the Rasters in a Format

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Format name	true		

# Raster

# Tile

# Style

# Create

Create a simple style.

Name	Description	Mandatory	Specified Default	Unspecified Default
params	The style parameters	true		
file	The output file	true		

# Vector Default

Create a default vector style.

Name	Description	Mandatory	Specified Default	Unspecified Default
layer	The Layer	true		
color	The color	false	#f2f2f2	#f2f2f2
opacity	The opacity	false	1.0	1.0
file	The output file	true		



## Vector Gradient

Create a gradient vector style.

Name	Description	Mandatory	Specified Default	Unspecified Default
layer	The Layer	true		
field	The field	true		
number	The number of categories	true		
colors	The colors	true		
method	The classification method (Quantile or EqualInterval)	false	Quantile	Quantile
elsemode	The else mode (ignore, min, max)	false	ignore	ignore
file	The output file	true		

## Vector Unique Values

Create a unique values vector style.

Name	Description	Mandatory	Specified Default	Unspecified Default
layer	The Layer	true		
field	The field	true		
colors	The colors	true		
file	The output file	true		

## Vector Unique Values From Text File

Create a unique values vector style from a text file

Name	Description	Mandatory	Specified Default	Unspecified Default
field	The field name	true		
geometryType	The geometry type	true		
textFile	The input text file	true		
styleFile	The output sld or ysls file	true		

## Raster Default

Create a default raster style.

Name	Description	Mandatory	Specified Default	Unspecified Default
raster	The Raster	true		
opacity	The opacity	false	1.0	1.0
file	The output file	true		

## Raster Color Map

Create a color map raster style.

Name	Description	Mandatory	Specified Default	Unspecified Default
raster	The Raster	true		
opacity	The opacity	false	1.0	1.0
values	The comma delimited list of values (key=value)	true		
type	The type (intervals, values, ramp)	false	ramp	ramp
extended	Whether to use extended colors or not	false	false	false
file	The output file	true		

## Map

## Other

## Unzip

Unzip a zip file

Name	Description	Mandatory	Specified Default	Unspecified Default
file	The zip file	true		
directory	The directory	true		

# Open

Open a File

Name	Description	Mandatory	Specified Default	Unspecified Default
file	The File	true		