

# geo-shell

Jared Erickson

Version 0.7-SNAPSHOT

# Table of Contents

- Introduction..... 1
- Workspace ..... 1
  - Basics..... 1
  - Layers ..... 2
- Layer ..... 2
  - Geoprocessing..... 2
  - Graticule..... 6

# 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

### 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
```

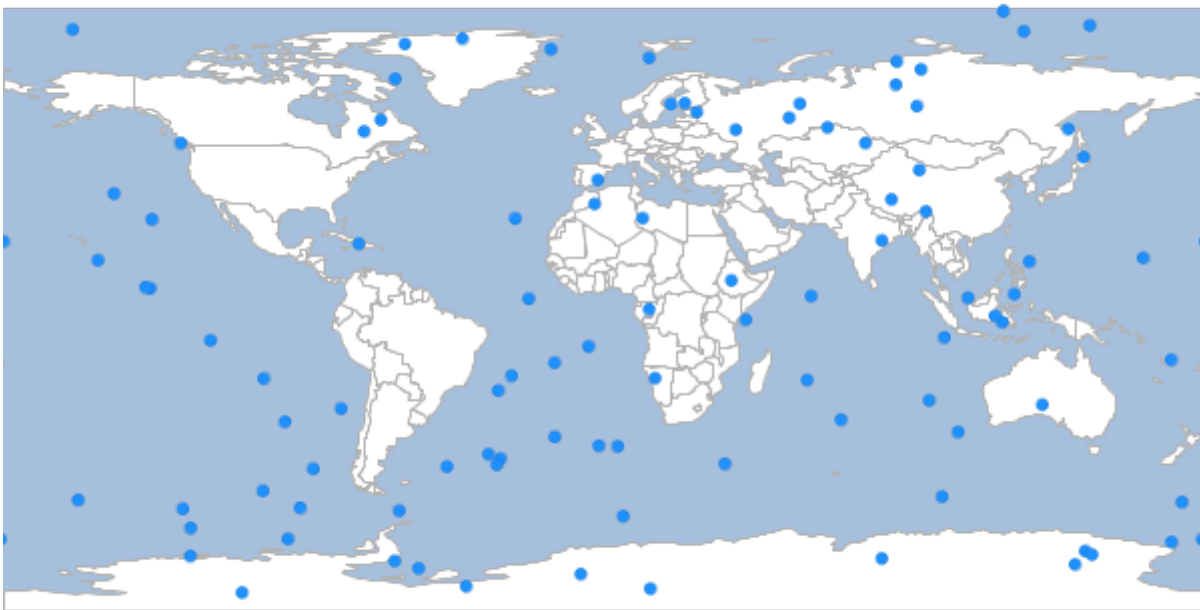
Added points layer to map randomMap

```
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

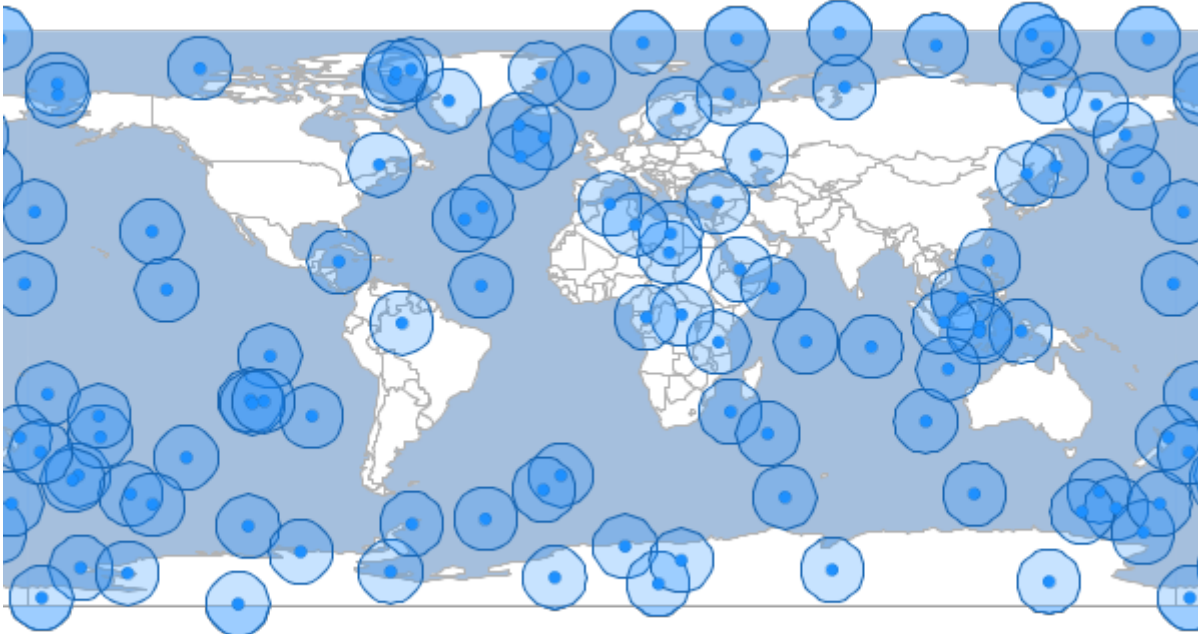
geo-shell> **map add layer** --name map --layer countries  
Added countries layer to map map

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!
```



## 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!
```

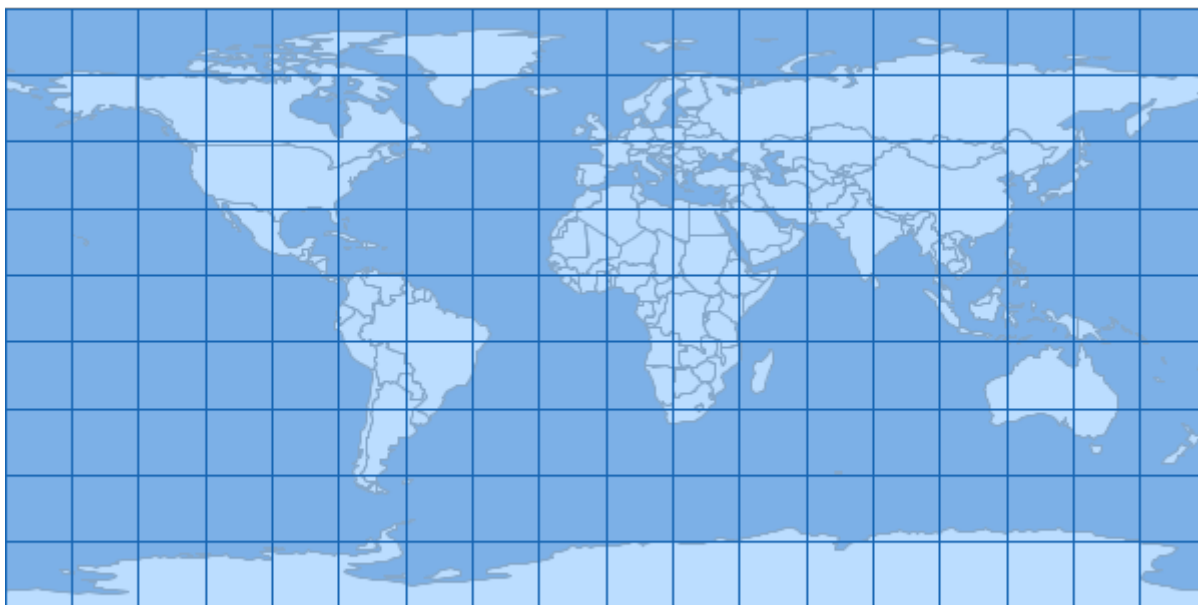
```
geo-shell> map add layer --name graticule --layer ocean
Added ocean layer to map graticule
```

```
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!
```



## 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!
```

