

# geo-shell

Jared Erickson

Version 0.7-SNAPSHOT

# Table of Contents

Introduction.....	1
Workspace .....	1
Basics.....	1
Layers .....	2
Layer .....	3
Basics.....	3
Geoprocessing.....	11
Graticule.....	44
Format.....	52
Open .....	52
List.....	52
Close .....	53
Rasters .....	53
Raster.....	54
Open .....	54
Close .....	54
List.....	55
Info .....	55
Value .....	55
Envelope.....	56
Get Style .....	56
Set Style .....	56
Add Raster .....	56
Add Constant.....	57
Subtract Raster .....	57
Subtract Constant.....	57
Multiply Raster .....	58
Multiply Constant.....	58
Divide Raster .....	58
Divide Constant .....	58
Contours.....	59
Crop.....	59
Mosaic.....	60
Reclassify .....	60
Reproject .....	60
Scale .....	61
Shaded Relief.....	61
Stylize .....	62

Tile .....	62
Open .....	62
Close .....	62
List .....	63
Info .....	63
Delete .....	64
Generate .....	65
Stitch Raster .....	67
Tiles .....	68
Vector Grid .....	69
Style .....	71
Create .....	71
Vector Default .....	71
Vector Gradient .....	72
Vector Unique Values .....	72
Vector Unique Values From Text File .....	72
Raster Default .....	73
Raster Color Map .....	73
Map .....	73
Open .....	73
Close .....	73
List .....	74
Add Layer .....	74
Add Raster .....	74
Add Tile .....	74
Remove Layer .....	75
Reorder .....	75
Layers .....	75
Draw .....	75
Other .....	76
Unzip .....	76
Open .....	76

# 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.

### Open

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

List open Workspaces. NOTE: No parameters

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

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

## Close

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

List the Layer in a Workspaces.

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

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.

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

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

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

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

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

### Close

Close a Layer.

```
geo-shell> layer close --name countries
```

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

```
geo-shell> workspace open --name naturalearth --params src/test/resources/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 close --name countries  
Layer countries closed!
```

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

## List

List open Layers.

```
geo-shell> layer list
```



No parameters

```
geo-shell> workspace open --name naturalearth --params src/test/resources/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 open --workspace naturalearth --layer ocean --name ocean  
Opened Workspace naturalearth Layer ocean as ocean
```

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

```
geo-shell> layer list  
countries = GeoPackage  
ocean = GeoPackage  
states = GeoPackage
```

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

## Schema

Inspect a Layer's Schema.

```
geo-shell> layer schema --name countries
```

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

```
geo-shell> workspace open --name naturalearth --params src/test/resources/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 schema --name countries  
Name Type
```

```
-----  
the_geom MultiPolygon  
ScaleRank Integer  
FeatureCla String
```

SOVEREIGNT String  
SOVISO String  
SOV\_A3 String  
LEVEL Double  
TYPE String  
NAME String  
SORTNAME String  
ADM0\_A3 String  
NAME\_SM String  
NAME\_LNG String  
TERR\_ String  
PARENTHETI String  
NAME\_ALT String  
LOCAL\_LNG String  
LOCAL\_SM String  
FORMER String  
ABBREV\_ String  
MAP\_COLOR Double  
PEOPLE Double  
GDP\_USDM Double  
FIPS\_10 String  
ISO\_A2 String  
ISO\_A3 String  
ISO\_N3 Double  
ITU String  
IOC String  
FIFA String  
DS String  
WMO String  
GAUL Double  
MARC String  
STANAG1059 String  
GW\_ID Double  
DIAL Double  
INTERNET\_ String  
COG String  
ACTUAL String  
CAPAY String  
CRPAY String  
ANI String  
LIBENR String  
ANCNOM String  
PAYS\_R\_GIO String  
COMMENT String

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



## Count

Count the Feature in a Layer.

```
geo-shell> layer count --name countries
```

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

```
geo-shell> workspace open --name naturalearth --params src/test/resources/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 count --name countries  
177
```

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

## Projection

Get the Projection of a Layer.

```
geo-shell> layer projection --name countries
```

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

```
geo-shell> workspace open --name naturalearth --params src/test/resources/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 projection --name countries  
EPSG:4326
```

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

## Features

Display the Features of a Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default

name	The Layer name	true		
filter	The CQL Filter	false		
sort	A Sort parameter (fld dir)	false		
start	The start index	false		-1
max	The maximum number of records	false		-1
field	A subfield to include	false		

## Get Style

Get the Layer's style.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		
style	The SLD File	false		

## Set Style

Set a Layer's style

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		
style	The SLD or CSS File	true		

## Copy

Copy one Layer to another Workspace.

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		
filter	The CQL Filter	false		
sort	A Sort parameter (fld dir)	false		
start	The start index	false		-1

max	The maximum number of records	false		-1
field	A subfield to include	false		

## Add

Add a new Feature to a Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		
values	The pipe delimited list of values (field=value)	true		

## Remove

Remove a Layer from a Workspace.

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

## Create

Create a new Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
workspace	The Workspace name	true		
name	The new Layer name	true		
fields	The pipe delimited list of fields (name=type)	true		

## Delete

Delete features from the Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
------	-------------	-----------	-------------------	---------------------

name	The Layer name	true		
filter	The CQL Filter	true		

## Update

Calculate the update between a Layer with another Layer

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

## Update Field

Delete features from the Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		
field	The field name	true		
value	The value	true		
filter	The CQL Filter	false	INCLUDE	INCLUDE
script	Whether the value is a script or not	false	false	false

## Add Fields

Add Fields to the input Layer and save the result to the output Layer

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		
fields	The Fields (name=type proj)	true		

## Add Area Field

Add area Field to the input Layer and save the result to the output Layer

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		
area-fieldname	The area field name	true	area	area

## Add ID Field

Add area ID to the input Layer and save the result to the output Layer

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		
id-fieldname	The id field name	true	id	id
start-value	The value to start at	true	1	1

## Add XY Fields

Add x and y coordinate Fields to the input Layer and save the result to the output Layer

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		
x-fieldname	The x field name	true	x	x
y-fieldname	The y field name	true	y	y

## Validity

Check for invalid geometries in the Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Layer name	true		
fields	A comma delimited list of Fields to include	false		

## Geoprocessing

### Clip

Clip the input Layer by the other Layer to produce the output Layer

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

### Convex Hull

Calculate the convexhull of the input Layer and save it to the output Layer.

```
geo-shell> layer convexhull --input-name countries --output-workspace layers --output-name convexhull
```

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 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> **layer convexhull** --input-name countries --output-workspace layers --output-name convexhull  
Done!

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

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

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 convexhull  
Added convexhull layer to map map

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

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



## Convex Hulls

Calculate the convexhull of each Feature in the input Layer and save them to the output Layer.

```
geo-shell> layer convexhulls --input-name countries --output-workspace layers --output-name convexhulls
```

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 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> **layer convexhulls** --input-name countries --output-workspace layers --output-name convexhulls

Done!

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

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

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

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

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 convexhulls

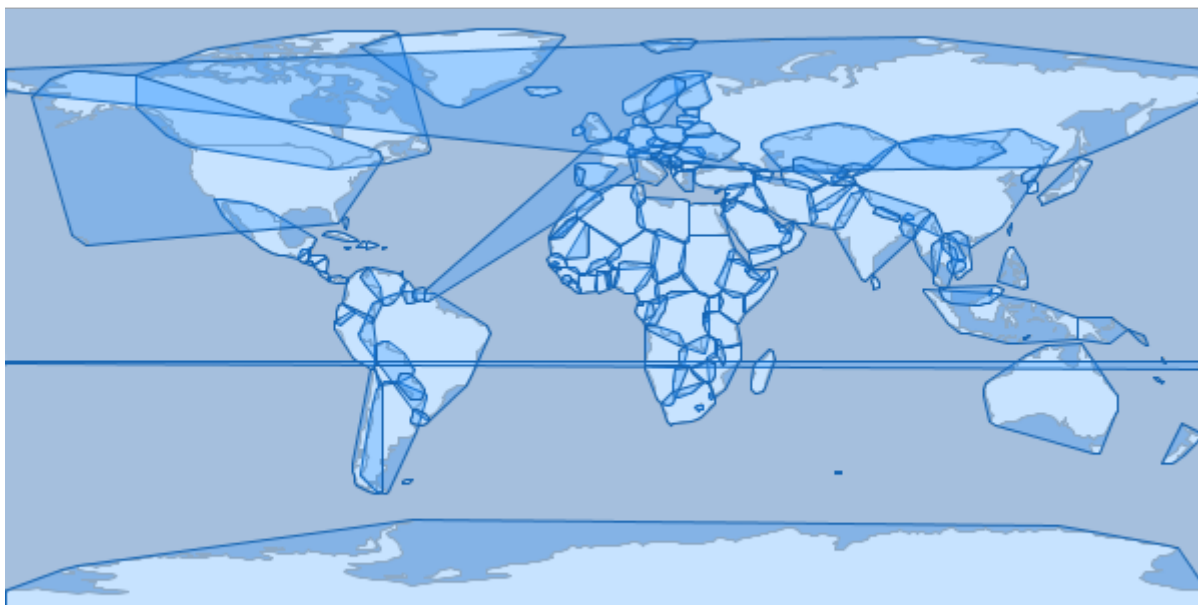
Added convexhulls layer to map map

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

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

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

Map map closed!



## Coordinates

Extract the coordinates each Feature in the input Layer and save them to the output Layer.

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		

## Delaunay

Calculate a delaunay diagram of the input Layer and save it to the output Layer.

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

## Densify

Densify the features of the input Layer and save them to the output Layer

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 distance tolerance	true		

## Dissolve

Dissolve the Features of a Layer by a Field.

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		
field	The field to use to dissolve features	true		
idField	The name of the id field	false	id	id
countField	The name of the count field	false	count	count

## Erase

Erase one Layer from another Layer

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

## Grid Row / Column

Create a grid Layer with rows and columns

Name	Description	Mandatory	Specified Default	Unspecified Default
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
rows	The number of rows	true		
columns	The number of columns	true		
geometry	The constraining geometry	true		
type	The geometry type (point or polygon	false	polygon	polygon
projection	The projection	false	EPSG:4326	EPSG:4326
geometry-field	The geometry field name	false	the_geom	the_geom

## Grid Width / Height

Create a grid Layer with cell width and height

Name	Description	Mandatory	Specified Default	Unspecified Default
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
cell-width	The width of each cell	true		
cell-height	The height of each cell	true		
geometry	The constraining geometry	true		
type	The geometry type (point or polygon	false	polygon	polygon
projection	The projection	false	EPSG:4326	EPSG:4326
geometry-field	The geometry field name	false	the_geom	the_geom

## Identity

Calculate the intersection between a Layer with another Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
other-name	The other Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
postfix-all	Whether to postfix all field names when combining schemas	false	false	false
include-duplicates	Whether to include duplicate field names	false	true	true

## Intersection

Calculate the intersection between a Layer with another Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
other-name	The other Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
postfix-all	Whether to postfix all field names when combining schemas	false	false	false
include-duplicates	Whether to include duplicate field names	false	true	true

## Minimum Circle

Calculate the minimum bounding circle of the input Layer and save it to the output Layer.

geo-shell> **layer mincircle** --input-name countries --output-workspace layers --output-name mincircle

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 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> **layer mincircle** --input-name countries --output-workspace layers --output-name mincircle  
Done!

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

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

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 mincircle
```

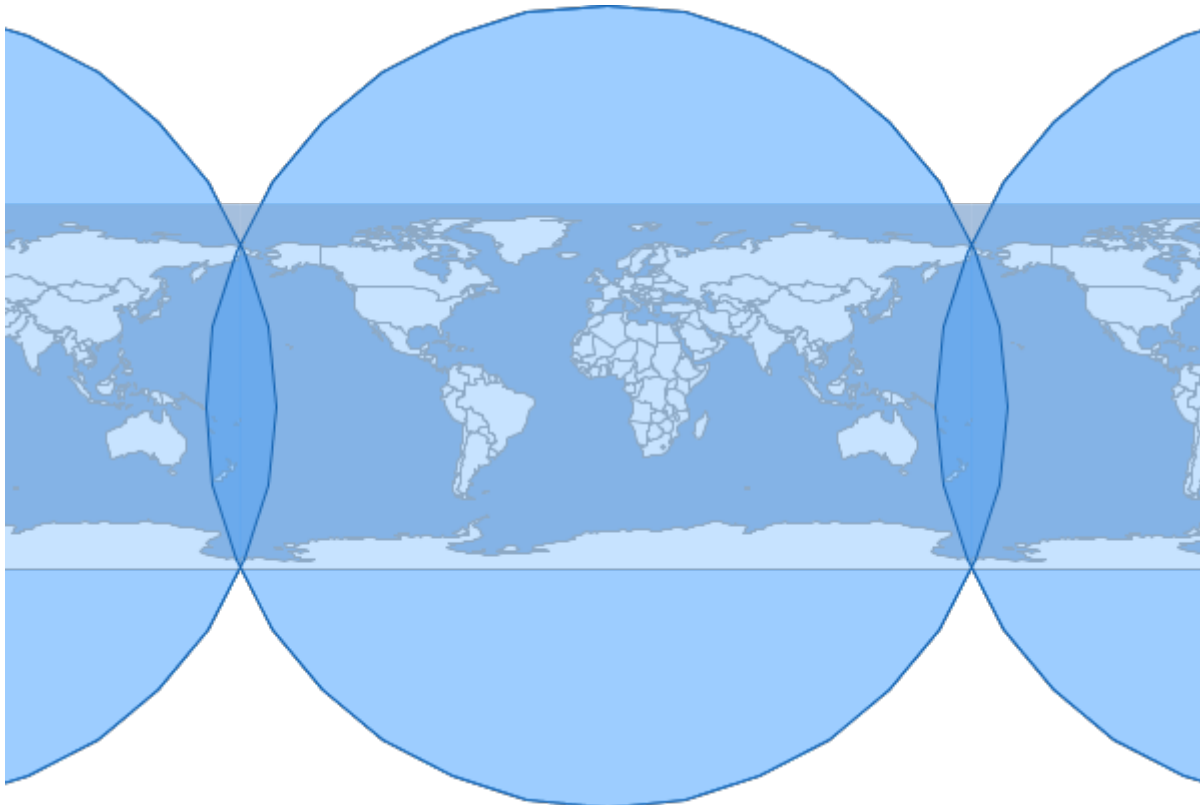
Added mincircle layer to map map

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

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

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

Map map closed!



## Minimum Circles

Calculate the minimum bounding circle of each Feature in the input Layer and save them to the output Layer.

```
geo-shell> layer mincircles --input-name countries --output-workspace layers --output-name mincircles
```

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 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> layer mincircles --input-name countries --output-workspace layers --output-name mincircles
Done!

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

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

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

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

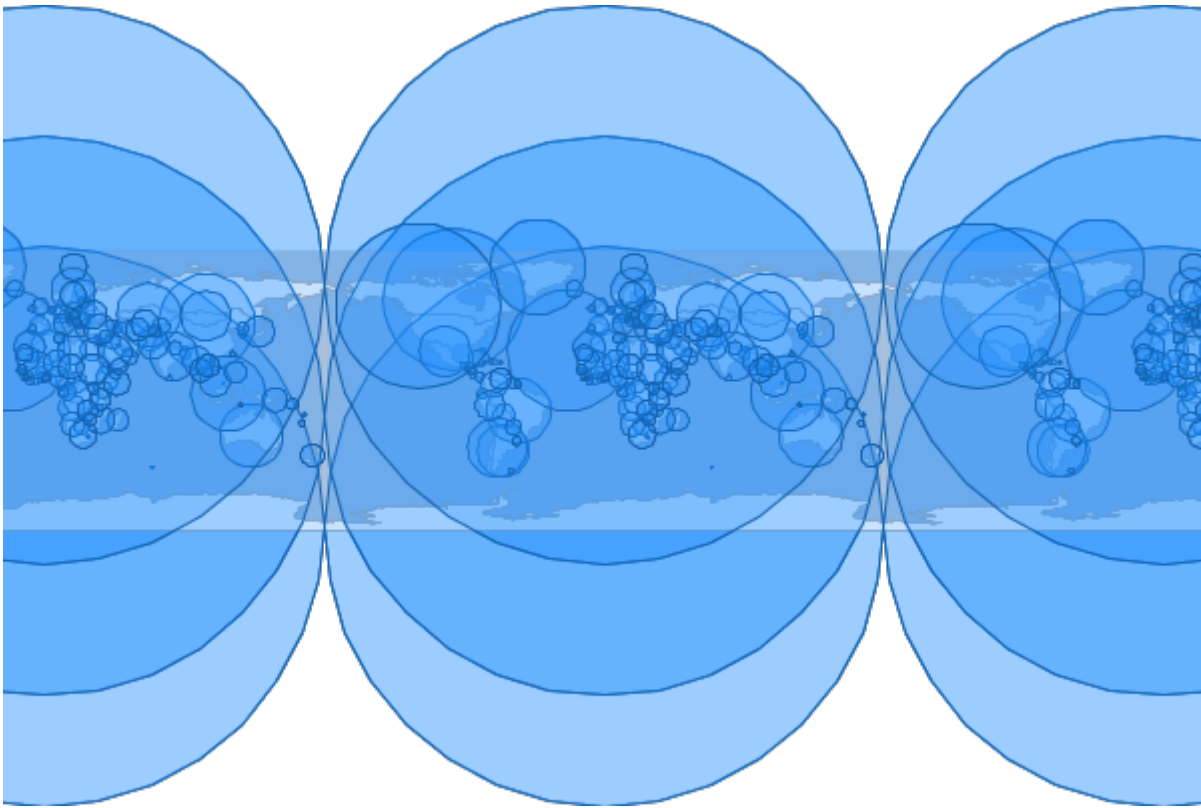
geo-shell> map add layer --name map --layer countries
Added countries layer to map map

geo-shell> map add layer --name map --layer mincircles
Added mincircles layer to map map

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

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





## Minimum Rectangle

Calculate the minimum rectangle of the input Layer and save it to the output Layer.

```
geo-shell> layer minrect --input-name countries --output-workspace layers --output-name minrect
```

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 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> **layer minrect** --input-name countries --output-workspace layers --output-name minrect  
Done!

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

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

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 minrect  
Added minrect layer to map map

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

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



## Minimum Rectangles

Calculate the minimum rectangle of each Feature in the input Layer and save them to the output Layer.

```
geo-shell> layer minrects --input-name countries --output-workspace layers --output-name minrects
```

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 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> layer minrects --input-name countries --output-workspace layers --output-name minrects
```

Done!

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

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

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

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

```
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 minrects
```

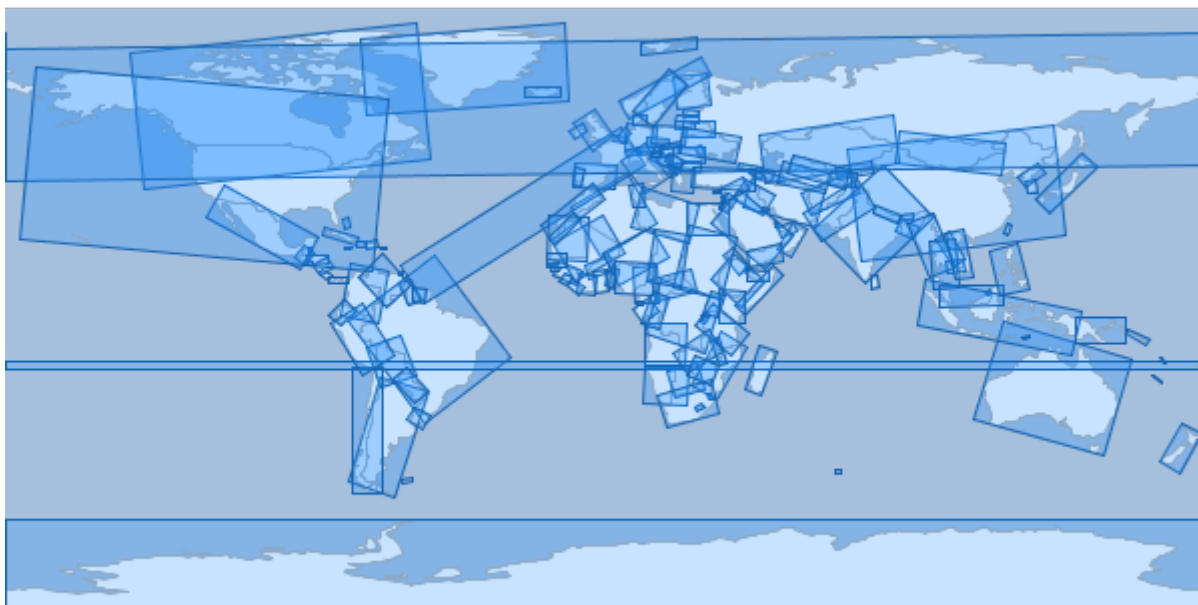
Added minrects layer to map map

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

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

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

Map map closed!



## Octangle Envelope

Calculate the octagonal envelope of the input Layer and save it to the output Layer.

```
geo-shell> layer octagonalenvelope --input-name countries --output-workspace layers --output
-name octagonalenvelope
```

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 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> **layer octagonalenvelope** --input-name countries --output-workspace layers --output  
-name octagonalenvelope  
Done!

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

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

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 octagonalenvelope  
Added octagonalenvelope layer to map map

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

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



## Octangle Envelopes

Calculate the octagonal envelope of each Feature in the input Layer and save them to the output Layer.

```
geo-shell> layer octagonalenvelopes --input-name countries --output-workspace layers --output
-name octagonalenvelopes
```

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 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> layer octagonalenvelopes --input-name countries --output-workspace layers --output  
-name octagonalenvelopes
```

Done!

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

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

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

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

```
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 octagonalenvelopes
```

Added octagonalenvelopes layer to map map

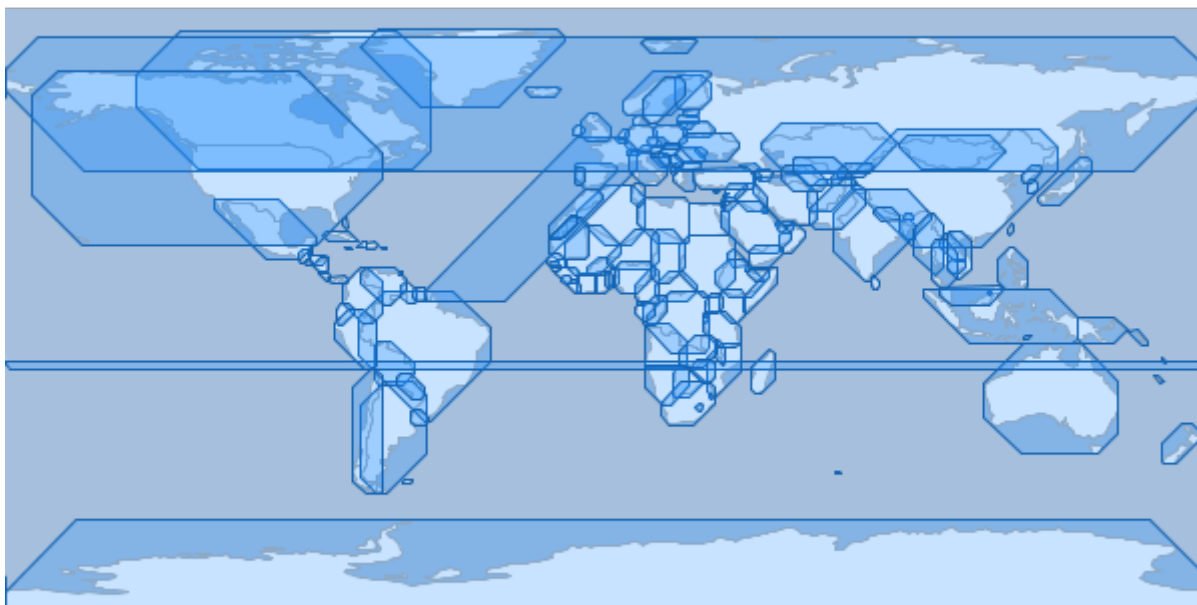
```
geo-shell> map draw --name map --file examples/layer_octagonalenvelopes.png
```

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

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

Map map closed!





## Points Along Lines

Create points along lines

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 distance between points	true		

## Simplify

Simplify the features of the input Layer and save them to the output Layer

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		

algorithm	The simplify algorithm (DouglasPeucker - dp or TopologyPreserving - tp)	false	tp	tp
distance	The distance tolerance	true		

## Symmetric Difference

Calculate the symmetric difference between a Layer and another Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
other-name	The other Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
postfix-all	Whether to postfix all field names when combining schemas	false	false	false
include-duplicates	Whether to include duplicate field names	false	true	true

## Transform

Transform the features of the input Layer and save them to the output Layer

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		
transforms	The pipe delimited list of transforms (field=expression or function)	true		

## Union

Union a Layer with another Layer

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
other-name	The other Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
postfix-all	Whether to postfix all field names when combining schemas	false	false	false
include-duplicates	Whether to include duplicate field names	false	true	true

## Voronoi

Calculate a voronoi diagram of the input Layer and save it to the output Layer.

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

## Random Points

Create a Layer with a number of randomly located points

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

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
```

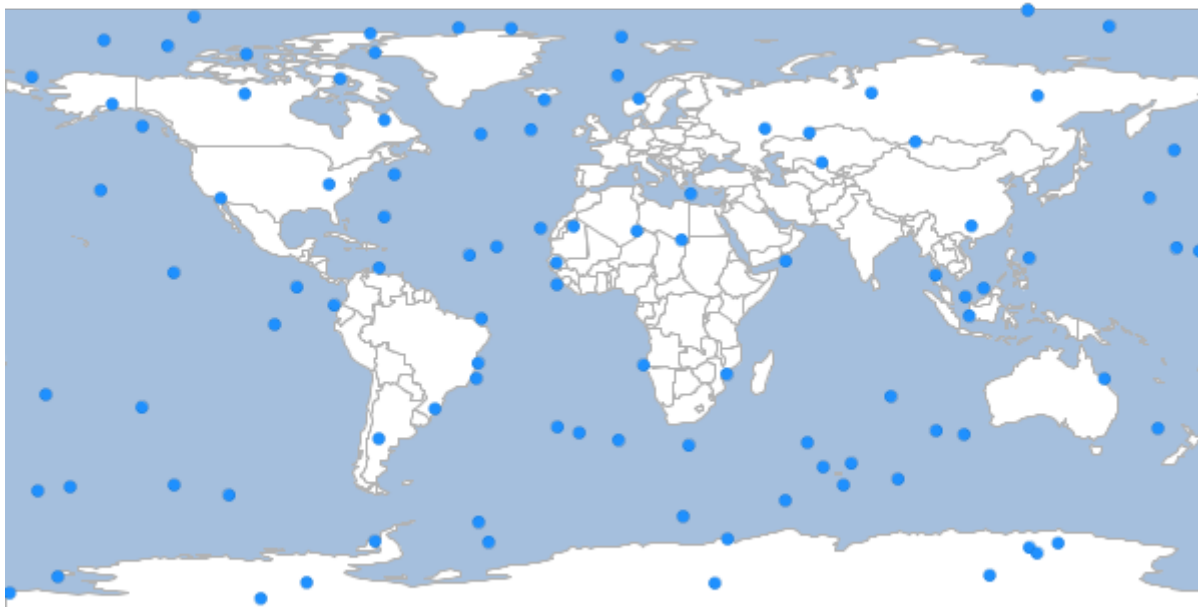
Added points layer to map randomMap

```
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

Buffer the input Layer to the output Layer.

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

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
```

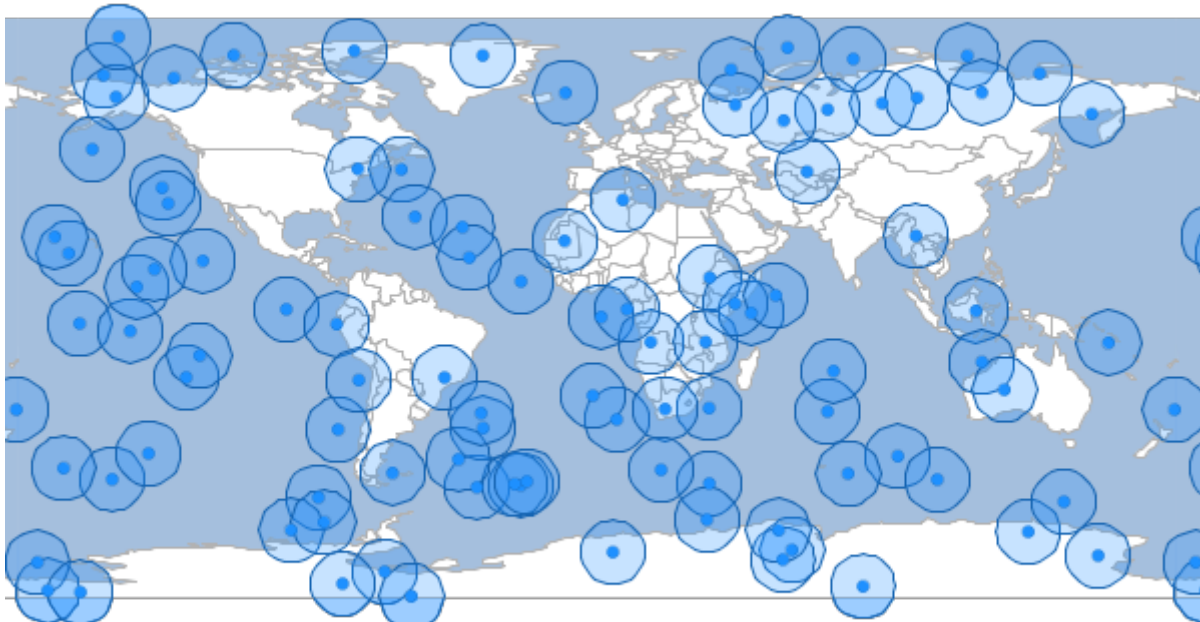
Added points layer to map map

```
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

Calculate the centroids of the input Layer to the output Layer.

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

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
```

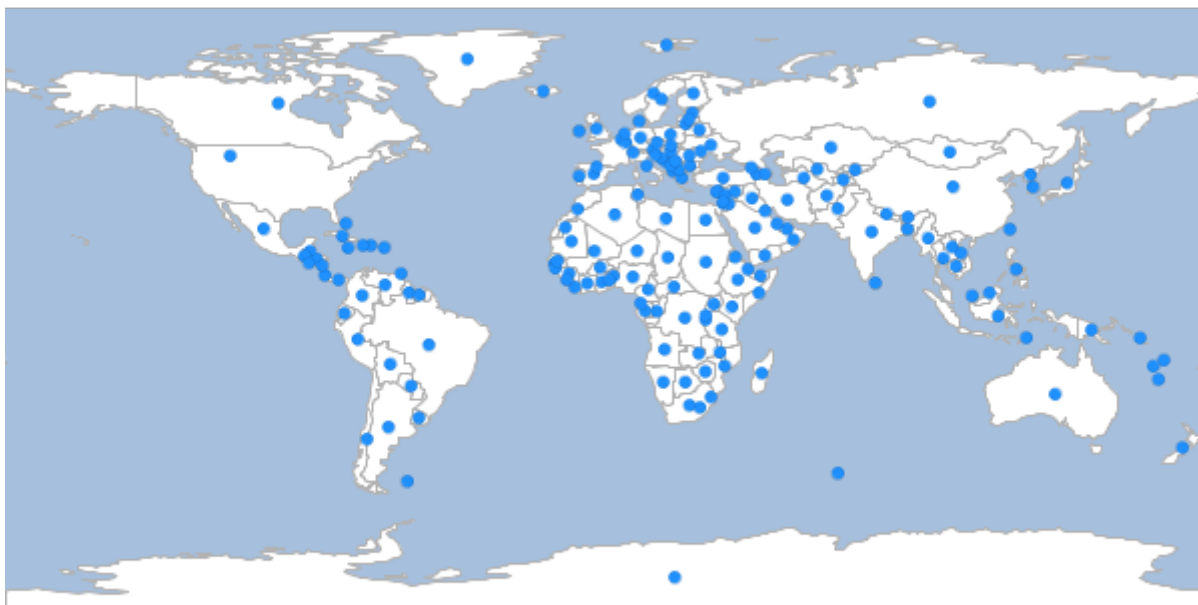
```
geo-shell> map add layer --name map --layer countries
Added countries layer to map map
```

```
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

Calculate the interior points of the input Layer to the output Layer.

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

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!
```

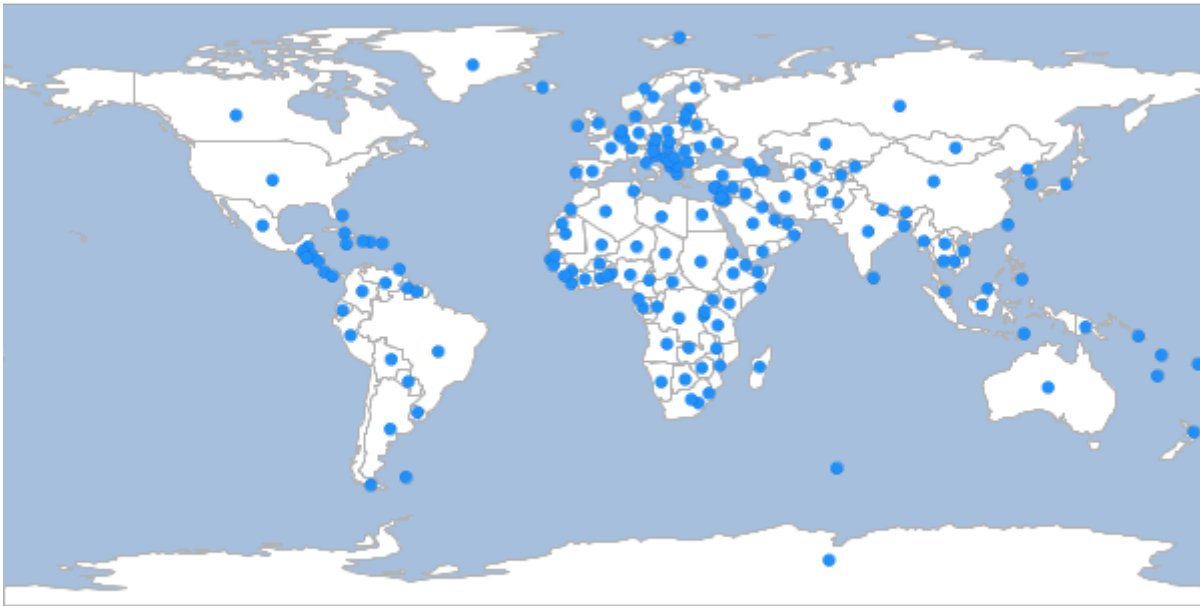
```
geo-shell> map add layer --name map --layer ocean
Added ocean layer to map map
```

```
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

Calculate the extent of the input Layer and save it to the output Layer.

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

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

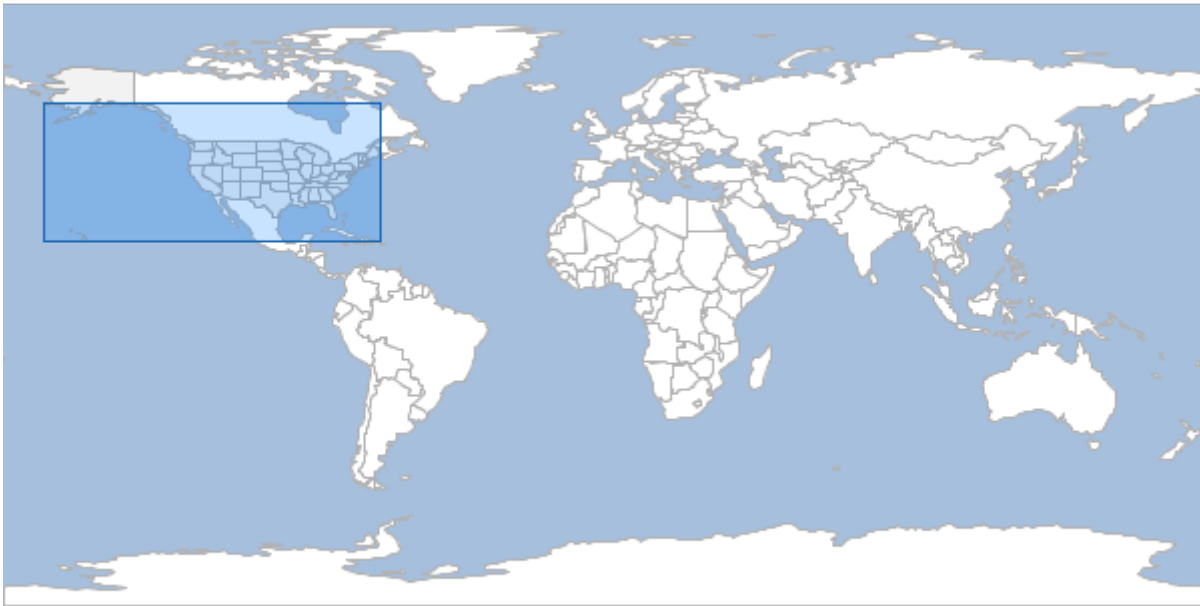
geo-shell> **map add layer** --name map --layer countries  
Added countries layer to map map

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

Calculate the extents of each Feature in the input Layer and save them to the output Layer.

```
geo-shell> layer extents --input-name states --output-workspace layers --output-name state_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

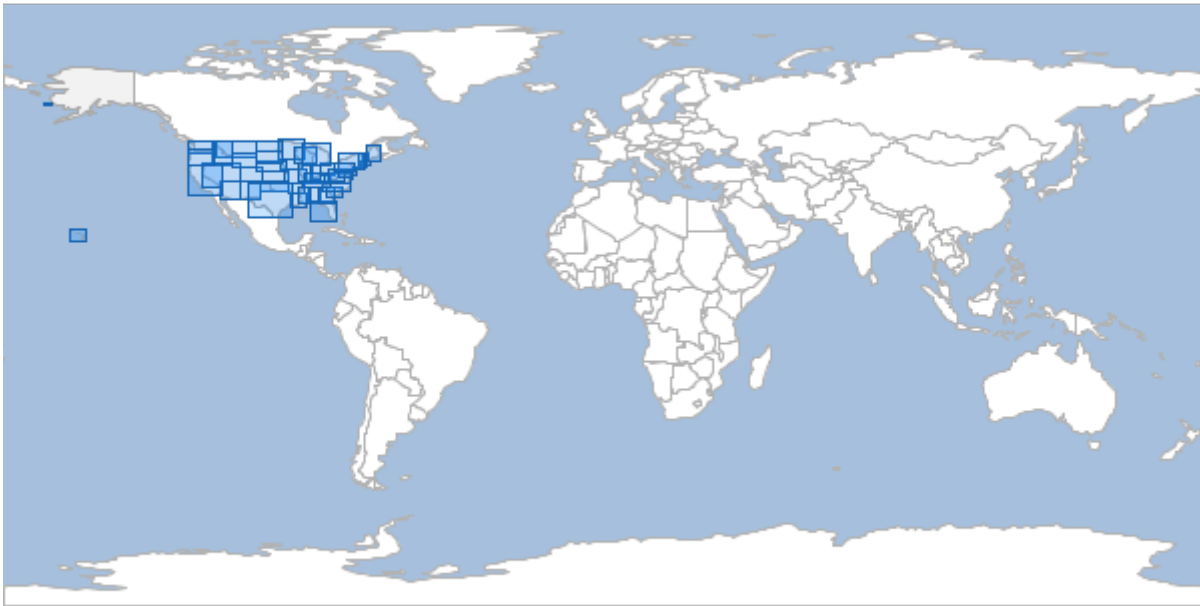
geo-shell> map add layer --name map --layer countries
Added countries layer to map map

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

Create a square graticule.

```
geo-shell> layer graticule square --workspace layers --name squares --bounds -180,-90,180,90
--length 20
```

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

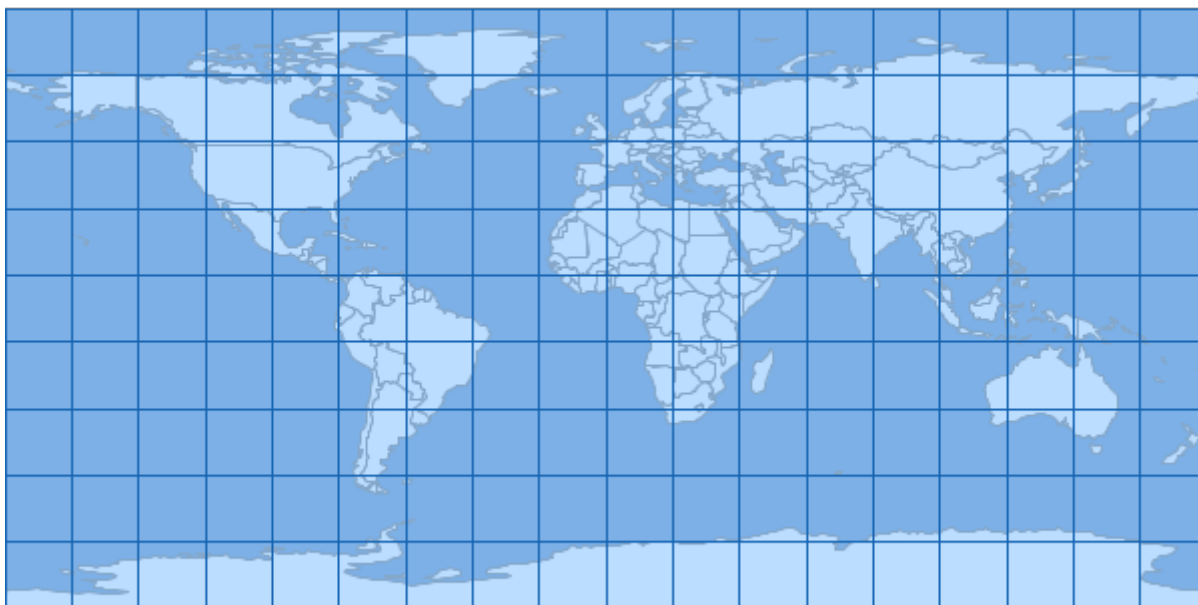
geo-shell> **map draw** --name graticule --file examples/square\_graticules.png

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

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

Map graticule closed!





## Rectangle

Create a rectangle graticule.

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

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!
```

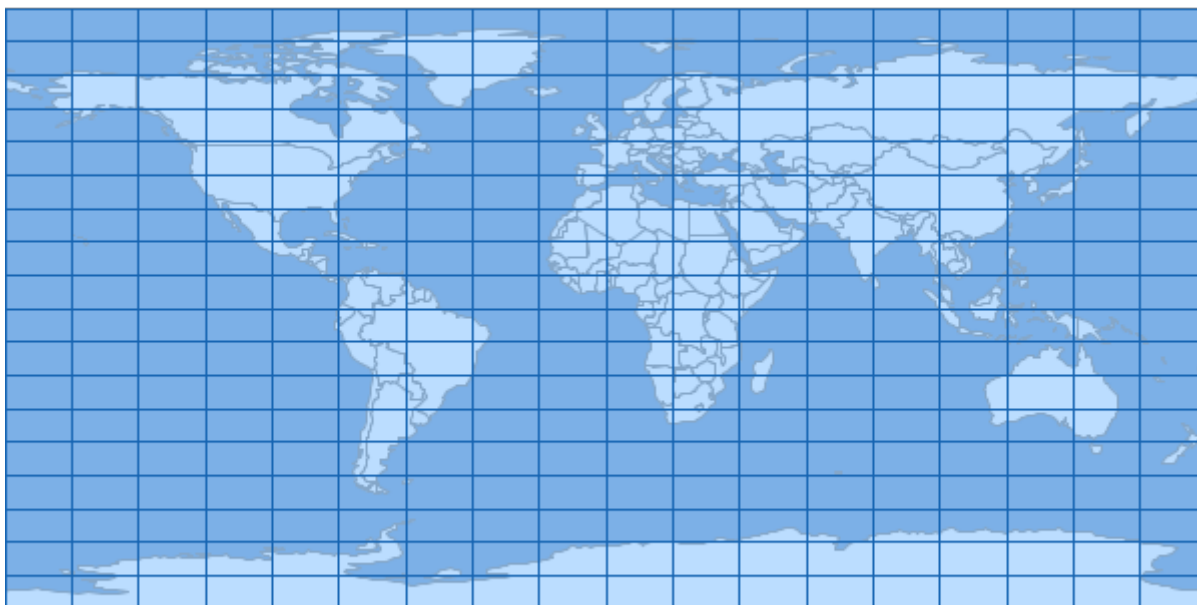
```
geo-shell> map add layer --name graticule --layer ocean  
Added ocean layer to map graticule
```

```
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

Create a oval graticule.

```
geo-shell> layer graticule oval --workspace layers --name ovals --bounds -180,-90,180,90 --size 20
```

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!
```

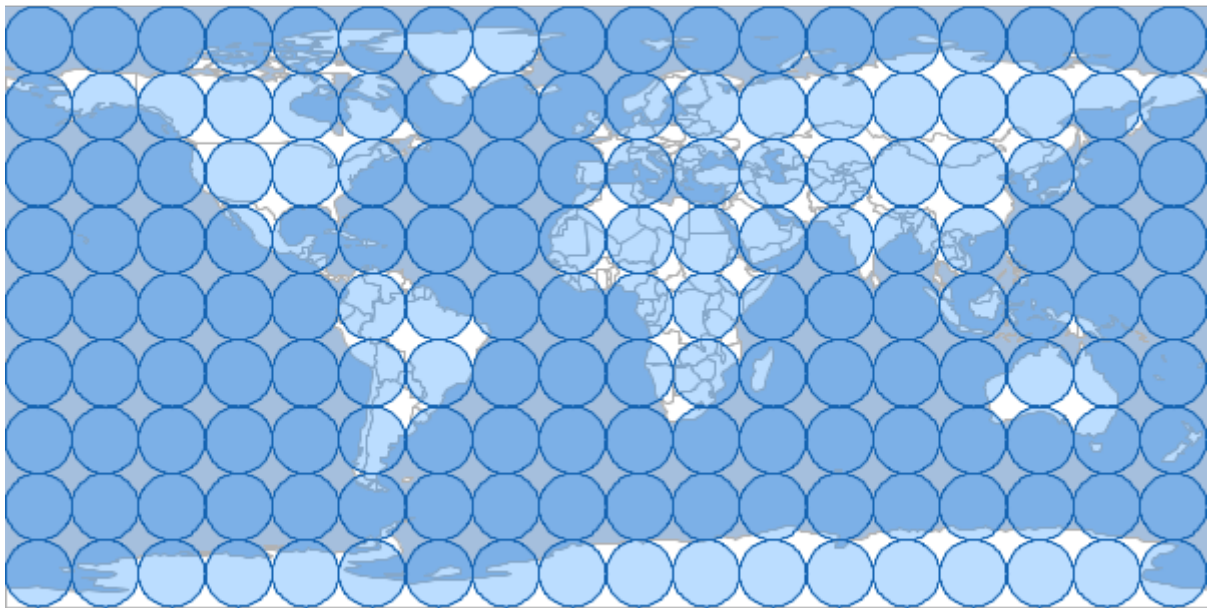
```
geo-shell> map add layer --name graticule --layer ocean  
Added ocean layer to map graticule
```

```
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

Create a hexagon graticule.

```
geo-shell> layer graticule hexagon --workspace layers --name hexagons --bounds -180,-90,180,90
--length 10
```

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

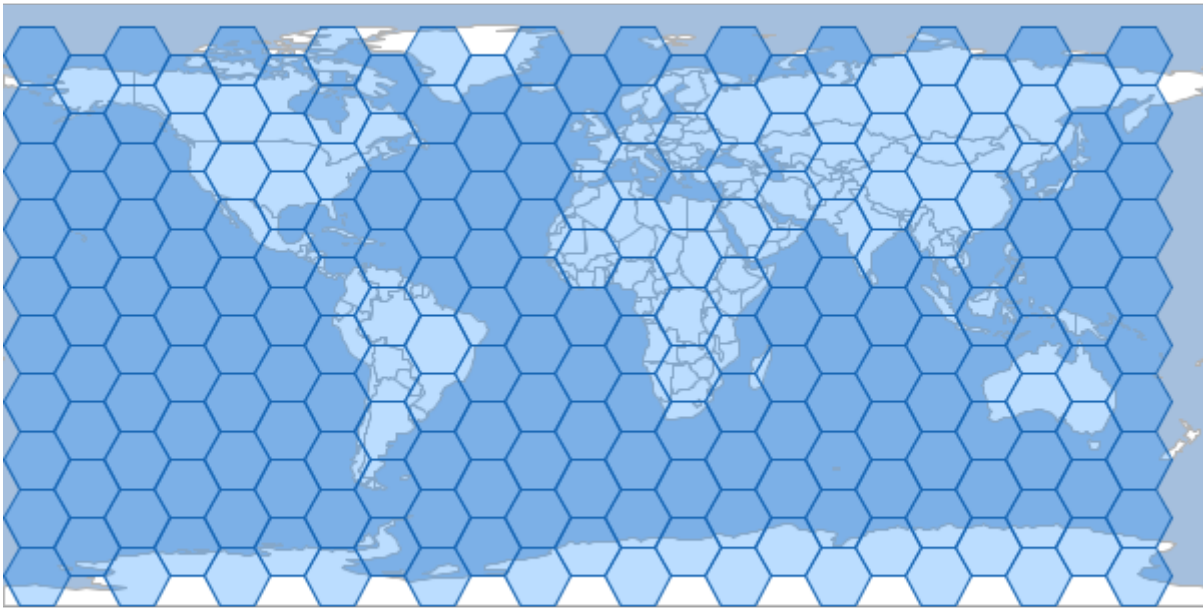
Added hexagons layer to map graticule

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 Raster Format.

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

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

```
geo-shell> format open --name earth --input src/test/resources/earth.tif  
Format earth opened!
```

```
geo-shell> format close --name earth  
Format earth closed!
```

## List

List open Raster Formats.

```
geo-shell> format list
```



No parameters

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

```
Format earth opened!
```

```
geo-shell> format open --name raster --input src/test/resources/raster.tif
```

```
Format raster opened!
```

```
geo-shell> format list
```

```
earth = GeoTIFF
```

```
raster = GeoTIFF
```

```
geo-shell> format close --name earth
```

```
Format earth closed!
```

```
geo-shell> format close --name raster
```

```
Format raster closed!
```

## Close

Close a Raster Format.

```
geo-shell> format close --name earth
```

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

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

```
Format earth opened!
```

```
geo-shell> format close --name earth
```

```
Format earth closed!
```

## Rasters

List the Rasters in a Format.

```
geo-shell> format rasters --name earth
```

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

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

```
Format earth opened!
```

```
geo-shell> format rasters --name earth
```

```
earth
```

```
geo-shell> format close --name earth
```



Format earth closed!

# Raster

## Open

Open a Raster.

```
geo-shell> raster open --format earth --raster earth --name earth
```

Name	Description	Mandatory	Specified Default	Unspecified Default
format	The Format name	true		
raster	The Raster name	true		
name	The name	false		

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

Format earth opened!

```
geo-shell> raster open --format earth --raster earth --name earth
```

Opened Format earth Raster earth as earth

```
geo-shell> raster close --name earth
```

Raster earth closed!

```
geo-shell> format close --name earth
```

Format earth closed!

## Close

Close a Raster.

```
geo-shell> raster close --name earth
```

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

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
```

Format earth opened!

```
geo-shell> raster open --format earth --raster earth --name earth
```

Opened Format earth Raster earth as earth

```
geo-shell> raster close --name earth
```

Raster earth closed!

```
geo-shell> format close --name earth
Format earth closed!
```

## List

List open Rasters.

```
geo-shell> raster list
```



No parameters

```
geo-shell> format open --name earth --input src/test/resources/earth.tif
Format earth opened!
```

```
geo-shell> raster open --format earth --raster earth --name earth
Opened Format earth Raster earth as earth
```

```
geo-shell> raster list
earth = GeoTIFF
```

```
geo-shell> raster close --name earth
Raster earth closed!
```

```
geo-shell> format close --name earth
Format earth closed!
```

## Info

Get information about a Raster.

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

## Value

Get a value from the Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
band	The x coordinate	false	0	0
x	The x coordinate	true		
y	The y coordinate	true		
type	The y coordinate	false	geometry	geometry

# Envelope

Create a Vector Layer from the envelope of a Raster.

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

# Get Style

Get the Raster's style.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
style	The SLD File	false		

# Set Style

Set a Raster's style

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
style	The SLD or CSS File	true		

# Add Raster

Add two Rasters together

Name	Description	Mandatory	Specified Default	Unspecified Default
name1	The Raster name	true		
name2	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

## Add Constant

Add constant values to a Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
values	The values	true		

## Subtract Raster

Subtract one Raster from another

Name	Description	Mandatory	Specified Default	Unspecified Default
name1	The Raster name	true		
name2	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

## Subtract Constant

Subtract constant values from a Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
values	The values	true		
from	Whether to subtract the Raster from the constant or vice verse	false	false	false

# Multiply Raster

Multiply two Raster together

Name	Description	Mandatory	Specified Default	Unspecified Default
name1	The Raster name	true		
name2	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

# Multiply Constant

Multiply constant values to a Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
values	The values	true		

# Divide Raster

Divide one Raster by another Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name1	The Raster name	true		
name2	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

# Divide Constant

Divide constant values against a Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
values	The values	true		

## Contours

Create contours.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
band	The Raster band to contour	false	0	0
levels	The contour level or interval	true		
simplify	Whether to simplify	false	false	false
smooth	Whether to smooth	false	false	false
bounds	The Bounds	false		

## Crop

Crop a Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
geometry	The geometry	true		

# Mosaic

Mosaic two Rasters together

Name	Description	Mandatory	Specified Default	Unspecified Default
name1	The Raster name	true		
name2	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

## Reclassify

Reclassify a Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
ranges	The comma delimited reclassification ranges (from-to=value)	true		
band	The Raster band to contour	false	0	0
nodata	The NODATA value	false	0	0

## Reproject

Project a Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		

output-name	The output Raster name	false		
projection	The projection	true		

## Scale

Scale a Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
x	The scale factor along the x axis	true		
y	The scale factor along the y axis	true		
x-trans	The x translation	false	0	0
y-trans	The y translation	false	0	0
interpolation	The interpolation method (bicubic, bicubic2, bilinear, nearest)	false	nearest	nearest

## Shaded Relief

Create a shaded relief raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		
scale	The scale	true		
altitude	The altitude	true		
azimuth	The azimuth	true		
resx	The x resolution	false	0.5	0.5
resy	The y resolution	false	0.5	0.5
zetafactory	The zeta factory	false	1.0	1.0



algorithm	The x resolution	false	DEFAULT	DEFAULT
-----------	------------------	-------	---------	---------

## Stylize

Create a new Raster by baking the style into an existing Raster

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The Raster name	true		
output-format	The output Format Workspace	true		
output-name	The output Raster name	false		

## Tile

### Open

Open a Tile Layer.

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
```

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

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
Tile Layer countries opened!
```

```
geo-shell> tile close --name countries
Tile Layer countries closed!
```

### Close

Close a Tile Layer.

```
geo-shell> tile close --name countries
```

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

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
```

Tile Layer countries opened!

```
geo-shell> tile close --name countries
Tile Layer countries closed!
```

# List

List open Tile Layers.

```
geo-shell> tile list
```



No parameters

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
Tile Layer countries opened!
```

```
geo-shell> tile list
countries = MBTiles
```

```
geo-shell> tile close --name countries
Tile Layer countries closed!
```

# Info

Get information about a Tile Layer.

```
geo-shell> tile info --name countries
```

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

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
Tile Layer countries opened!
```

```
geo-shell> tile info --name countries
countries
EPSG:3857
-2.0036395147881314E7,
-2.0037471205137067E7,2.0036395147881314E7,2.003747120513706E7,EPSG:3857
BOTTOM_LEFT
256,256
0,1,1,156412.0,156412.0
1,2,2,78206.0,78206.0
2,4,4,39103.0,39103.0
3,8,8,19551.5,19551.5
4,16,16,9775.75,9775.75
5,32,32,4887.875,4887.875
```

6,64,64,2443.9375,2443.9375  
7,128,128,1221.96875,1221.96875  
8,256,256,610.984375,610.984375  
9,512,512,305.4921875,305.4921875  
10,1024,1024,152.74609375,152.74609375  
11,2048,2048,76.373046875,76.373046875  
12,4096,4096,38.1865234375,38.1865234375  
13,8192,8192,19.09326171875,19.09326171875  
14,16384,16384,9.546630859375,9.546630859375  
15,32768,32768,4.7733154296875,4.7733154296875  
16,65536,65536,2.38665771484375,2.38665771484375  
17,131072,131072,1.193328857421875,1.193328857421875  
18,262144,262144,0.5966644287109375,0.5966644287109375  
19,524288,524288,0.29833221435546875,0.29833221435546875

geo-shell> **tile close** --name countries  
Tile Layer countries closed!

## Delete

Delete tiles from a Tile Layer.

geo-shell> **tile delete** --name tiles --z 3

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The tile name	true		
tile	The tile z/x/y	false		
bounds	The bounds	false		
width	The width	false	400	400
height	The height	false	400	400
z	The zoom level	false	0	-1
minx	The min x or column	false		-1
miny	The min y or row	false		-1
maxx	The max x or column	false		-1
maxy	The max y or row	false		-1

geo-shell> **tile open** --name tiles --params target/tiles.mbtiles  
Tile Layer tiles 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 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 world
```

Map world opened!

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

Added ocean layer to map world

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

Added countries layer to map world

```
geo-shell> tile generate --name tiles --map world --start 0 --end 3
```

Tiles generated!

```
geo-shell> tile delete --name tiles --z 3
```

Deleting tiles at z level 3

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

Map world closed!

## Generate

Generate tiles for a Tile Layer.

```
geo-shell> tile generate --name tiles --map world --start 0 --end 3
```

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The tile name	true		
map	The map name	true		
start	The map name	true		
end	The map name	true		
bounds	The map name	false		
metatile	The metatile width,height	false		
missingOnly	The map name	false	false	false
verbose	The map name	false	false	false

```
geo-shell> tile open --name tiles --params target/tiles.mbtiles
```

Tile Layer tiles 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 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 world  
Map world opened!
```

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

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

```
geo-shell> tile generate --name tiles --map world --start 0 --end 3  
Tiles generated!
```

```
geo-shell> format open --name world_level2 --input examples/tile_generate.png  
Format world_level2 opened!
```

```
geo-shell> tile stitch raster --name tiles --format world_level2 --raster world_level2 --z 2  
Done stitching Raster world_level2 from tiles!
```

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



## Stitch Raster

Create a Raster from a Tile Layer.

```
geo-shell> tile stitch raster --name countries --format states --raster states --bounds  
-18217695.5734,1222992.4526,-4207094.0368,7924991.0926
```

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The tile name	true		
format	The raster format name	true		
raster	The raster name	true		
bounds	The bounds	false		

width	The raster width	false	400	400
height	The raster height	false	400	400
z	The zoom level	false	0	-1
minx	The min x or column	false		-1
miny	The min y or row	false		-1
maxx	The max x or column	false		-1
maxy	The max y or row	false		-1

Create a Raster from a Tile Layer with a geographic bounds.

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
Tile Layer countries opened!
```

```
geo-shell> format open --name states --input examples/tile_stitch_bounds.png
Format states opened!
```

```
geo-shell> tile stitch raster --name countries --format states --raster states --bounds
-18217695.5734,1222992.4526,-4207094.0368,7924991.0926
Done stitching Raster states from countries!
```



## Tiles

List tiles within a given bounds.

```
geo-shell> tile tiles --name countries --z 8 --bounds -13787405.4140,5872198.2610,
-13349574.1159,6081635.7185
```

Name	Description	Mandatory	Specified Default	Unspecified Default
------	-------------	-----------	-------------------	---------------------

name	The tile name	true		
bounds	The bounds	true		
z	The zoom level	true		

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
Tile Layer countries opened!
```

```
geo-shell> tile tiles --name countries --z 8 --bounds -13787405.4140,5872198.2610,
-13349574.1159,6081635.7185
8/39/165
8/40/165
8/41/165
8/42/165
8/39/166
8/40/166
8/41/166
8/42/166
```

```
geo-shell> tile close --name countries
Tile Layer countries closed!
```

## Vector Grid

Create a Vector Grid Layer from the pyramid of a Tile Layer.

```
geo-shell> tile vector grid --name countries --workspace layers --layer level3 --z 3
```

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The tile name	true		
workspace	The workspace name	true		
layer	The layer name	true		
bounds	The bounds	false		
width	The raster width	false	400	400
height	The raster height	false	400	400
z	The zoom level	false	0	-1
minx	The min x or column	false		-1
miny	The min y or row	false		-1
maxx	The max x or column	false		-1
maxy	The max y or row	false		-1

```
geo-shell> tile open --name countries --params src/test/resources/countries.mbtiles
```



Tile Layer countries opened!

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

```
geo-shell> tile vector grid --name countries --workspace layers --layer level3 --z 3  
Done generating the vector grid level3 from countries!
```

```
geo-shell> style vector default --layer level3 --color #ffffff --opacity 0.25 --file examples/level3.sld  
Default Vector Style for level3 written to /home/travis/build/jericks/geo-shell/examples/level3.sld!
```

```
geo-shell> layer style set --name level3 --style examples/level3.sld  
Style /home/travis/build/jericks/geo-shell/examples/level3.sld set on level3
```

```
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 vectorGridMap  
Map vectorGridMap opened!
```

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

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

```
geo-shell> map add layer --name vectorGridMap --layer level3  
Added level3 layer to map vectorGridMap
```

```
geo-shell> map draw --name vectorGridMap --file examples/tile_vector_grid.png --projection  
EPSG:3857 --width 400 --height 400 --bounds -20026376.39,-20048966.10,20026376.39,20048966.10  
Done drawing /home/travis/build/jericks/geo-shell/examples/tile_vector_grid.png!
```

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



## 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

### Open

Open a new Map.

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

### Close

Close a Tile Layer.

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

## List

List open Maps. NOTE: No parameters

## Add Layer

Add a Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
layer	The layer	true		
mapLayerName	The map layer name	false		

## Add Raster

Add a Raster.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
raster	The raster	true		
mapLayerName	The map layer name	false		

## Add Tle

Add a Tile.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
tile	The tile	true		
mapLayerName	The map layer name	false		

## Remove Layer

Remove a Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
layer	The layer name	true		

## Reorder

Reorder a Layer in the Map.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
layer	The layer name	true		
order	The order parameters	true		

## Layers

List the Map's Layers.

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

## Draw

Draw a map.

Name	Description	Mandatory	Specified Default	Unspecified Default
name	The map name	true		
bounds	The Bounds	false		
projection	The Projection	false		
width	The width	false	600	600
height	The height	false	400	400
type	The type	false	png	png
file	The file	false		
background-color	The background color	false		

# Other

## Unzip

Unzip a 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		