

```
1. geo shell 0.0.1 (java)
Jared-Ericksons-MacBook-Pro:geo-shell jericks$ geo-shell

Welcome to the Geo Shell!
geo-shell>workspace open --name mem --params memory
Workspace mem opened!
geo-shell>layer create --workspace mem --name points --fields "the_geom=Point ES S:4326|fid=Integer|name=String"
Created Layer points!
geo-shell>layer add --name points --values "the_geom=POINT (-122.333056 47.609722)|fid=1|name=Seattle"
Added Feature to points
geo-shell>layer add --name points --values "the_geom=POINT (-122.459444 47.241389)|fid=2|name=Tacoma"
Added Feature to points
geo-shell>layer count --name points
2
geo-shell>[]
```

Geo Shell

Jared Erickson

Version 0.7-SNAPSHOT

Table of Contents

| | |
|-------------------------|----|
| Introduction | 1 |
| Modules | 1 |
| Use | 2 |
| Workspace | 2 |
| Basics..... | 2 |
| Layers | 3 |
| Layer | 3 |
| Basics..... | 3 |
| Geoprocessing | 19 |
| Graticule | 53 |
| Format..... | 61 |
| Open | 61 |
| List | 61 |
| Close | 62 |
| Rasters | 62 |
| Raster..... | 63 |
| Open | 63 |
| Close | 63 |
| List | 64 |
| Info | 64 |
| Value | 65 |
| Envelope..... | 66 |
| Get Style | 66 |
| Set Style | 66 |
| Add Raster | 66 |
| Add Constant..... | 67 |
| Subtract Raster | 67 |
| Subtract Constant..... | 67 |
| Multiply Raster | 68 |
| Multiply Constant | 68 |
| Divide Raster | 68 |
| Divide Constant | 69 |
| Contours | 69 |
| Crop | 69 |
| Mosaic..... | 70 |
| Reclassify | 70 |
| Reproject | 71 |
| Scale | 71 |

| | |
|---|-----|
| Shaded Relief | 71 |
| Stylize | 72 |
| Tile | 72 |
| Open | 72 |
| Close | 73 |
| List | 73 |
| Info | 73 |
| Delete | 74 |
| Generate | 75 |
| Stitch Raster | 77 |
| Tiles | 78 |
| Vector Grid | 79 |
| Style | 81 |
| Create | 81 |
| Vector Default | 84 |
| Vector Gradient | 88 |
| Vector Unique Values | 91 |
| Vector Unique Values From Text File | 93 |
| Raster Default | 96 |
| Raster Color Map | 98 |
| Map | 101 |
| Open | 101 |
| Close | 101 |
| List | 102 |
| Add Layer | 102 |
| Add Raster | 103 |
| Add Tile Layer | 105 |
| Remove Layer | 106 |
| Reorder | 107 |
| Layers | 109 |
| Draw | 110 |
| Display | 111 |
| Built in | 113 |
| Exit / Quit | 113 |
| Help | 113 |
| Run OS Command | 114 |
| Date | 115 |
| Script | 115 |
| System Properties | 118 |
| Version | 125 |
| Download | 126 |

| | |
|------------|-----|
| Unzip..... | 127 |
| Open | 127 |

Introduction

Geo Shell is an interactive shell for geospatial analysis.



```
1. geo shell 0.0.1 (java)
Jared-Ericksons-MacBook-Pro:geo-shell jericks$ geo-shell

Welcome to the Geo Shell!
geo-shell>workspace open --name mem --params memory
Workspace mem opened!
geo-shell>layer create --workspace mem --name points --fields "the_geom=Point EPSG:4326|fid=Integer|name=String"
Created Layer points!
geo-shell>layer add --name points --values "the_geom=POINT (-122.333056 47.609722)|fid=1|name=Seattle"
Added Feature to points
geo-shell>layer add --name points --values "the_geom=POINT (-122.459444 47.241389)|fid=2|name=Tacoma"
Added Feature to points
geo-shell>layer count --name points
2
geo-shell>[]
```

Modules

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 geotiffs 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.

Use

You can use geo-shell interactively by typing **geo-shell** at the command line.

Or you can write scripts and then execute them from the command line by typing **geo-shell -cmdfile script.txt**

Or by using the **script --file script.txt** command within a geo-shell session.

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.

```
geo-shell> layer features --name states --filter "NAME_1='North Dakota'"
```

| 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 | | |
|-------|-----------------------|-------|--|--|

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

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

geo-shell> **layer features** --name states --filter "NAME_1='North Dakota'"

Feature (states.3)

the_geom = MULTIPOLYGON
 FID_1 = 31
 ScaleRank = 2
 FeatureCla = 1st Order Admin Polys
 OBJECTID = 22
 VertexCou = 223.0
 ISO = USA
 NAME_0 = United States
 NAME_1 = North Dakota
 VARNAME_1 = ND | N.D.
 NL_NAME_1 =
 HASC_1 = US.ND
 TYPE_1 = State
 ENGTTYPE_1 = State
 VALIDFR_1 = 18891102
 VALIDTO_1 = Present
 REMARKS_1 =
 Region =
 RegionVar =
 ProvNumber = 23
 NEV_Countr = United States
 FIRST_FIPS =
 FIRST_HASC =
 FIPS_1 = US38
 gadm_level = 1.0
 CheckMe = 0
 Region_Cod =
 Region_C_1 =
 ScaleRan_1 = 1
 Region_C_2 =
 Region_C_3 =
 Country_Pr =

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

Get Style

Get the Layer's style.

```
geo-shell> layer style get --name states --style target/states.sld
```

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

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

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

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

```
geo-shell> layer style get --name states --style target/states.sld  
states style written to /home/travis/build/jericks/geo-shell/target/states.sld
```

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

```

<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <sld:PolygonSymbolizer>
            <sld:Fill>
              <sld:CssParameter name="fill">#f2f2f2</sld:CssParameter>
            </sld:Fill>
          </sld:PolygonSymbolizer>
          <sld:LineSymbolizer>
            <sld:Stroke>
              <sld:CssParameter name="stroke">#a9a9a9</sld:CssParameter>
              <sld:CssParameter name="stroke-width">0.5</sld:CssParameter>
            </sld:Stroke>
          </sld:LineSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>

```

Set Style

Set a Layer's style

```
geo-shell> layer style get --name states --style target/states_simple.sld
```

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

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

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

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

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

```
geo-shell> layer style get --name states --style target/states_simple.sld  
states style written to /home/travis/build/jericks/geo-shell/target/states_simple.sld
```

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

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

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

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

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



Copy

Copy one Layer to another Workspace.

```
geo-shell> layer copy --input-name states_gpkg --output-workspace shapefiles --output-name states
```

| 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 | | |

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

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

```
geo-shell> workspace open --name shapefiles --params target/
Workspace shapefiles opened!
```

```
geo-shell> layer copy --input-name states_gpkg --output-workspace shapefiles --output-name states
Done!
```

```
geo-shell> layer count --name states
52
```

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

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

Create

Create a new Layer.

```
geo-shell> layer create --workspace mem --name points --fields "the_geom=Point
EPSG:4326|fid=Int|name=String"
```

| 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 | | |

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

```
geo-shell> layer create --workspace mem --name points --fields "the_geom=Point
EPSG:4326|fid=Int|name=String"
Created Layer points!
```

```
geo-shell> layer schema --name points
Name Type
-----
```

```
the_geom Point
fid Integer
name String
```

Add

Add a new Feature to a Layer.

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.333056
47.609722)|fid=1|name=Seattle"
```

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

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

```
geo-shell> layer create --workspace mem --name points --fields "the_geom=Point
EPSG:4326|fid=Int|name=String"
Created Layer points!
```

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.333056
47.609722)|fid=1|name=Seattle"
Added Feature to points
```

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.459444
47.241389)|fid=2|name=Tacoma"
Added Feature to points
```

```
geo-shell> layer count --name points  
2
```

Delete

Delete features from the Layer

```
geo-shell> layer delete --name points --filter "fid=2"
```

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|--------|----------------|-----------|-------------------|---------------------|
| name | The Layer name | true | | |
| filter | The CQL Filter | true | | |

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

```
geo-shell> layer create --workspace mem --name points --fields "the_geom=Point  
EPSG:4326|fid=Int|name=String"  
Created Layer points!
```

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.333056  
47.609722)|fid=1|name=Seattle"  
Added Feature to points
```

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.459444  
47.241389)|fid=2|name=Tacoma"  
Added Feature to points
```

```
geo-shell> layer count --name points  
2
```

```
geo-shell> layer delete --name points --filter "fid=2"  
Deleted fid=2 Features from points
```

```
geo-shell> layer count --name points  
1
```

Remove

Remove a Layer from a Workspace.

```
geo-shell> layer remove --layer polygons --workspace mem
```

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

```

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

geo-shell> layer create --workspace mem --name points --fields "the_geom=Point
EPSG:4326|fid=Int|name=String"
Created Layer points!

geo-shell> layer create --workspace mem --name lines --fields "the_geom=LineString
EPSG:4326|fid=Int|name=String"
Created Layer lines!

geo-shell> layer create --workspace mem --name polygons --fields "the_geom=Polygon
EPSG:4326|fid=Int|name=String"
Created Layer polygons!

```

geo-shell> **workspace layers** --name mem
lines
points
polygons

geo-shell> **layer remove** --layer polygons --workspace mem
Layer polygons removed from Workspace mem

geo-shell> **workspace layers** --name mem
lines
points

Update Field

Update the values of a field

```
geo-shell> layer updatefield --name points --field state --value WA
```

| 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 |

```

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

```

```

geo-shell> layer create --workspace mem --name points --fields "the_geom=Point
EPSG:4326|fid=Int|name=String|state=String"
Created Layer points!

```

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.333056  
47.609722)|fid=1|name=Seattle"
```

Added Feature to points

```
geo-shell> layer add --name points --values "the_geom=POINT (-122.459444  
47.241389)|fid=2|name=Tacoma"
```

Added Feature to points

```
geo-shell> layer updatefield --name points --field state --value WA  
Done updating state with WA!
```

```
geo-shell> layer features --name points
```

```
Feature (fid—7be85bd1_16fc5c446bc_-79e6)
```

```
-----  
the_geom = POINT (-122.333056 47.609722)  
fid = 1  
name = Seattle  
state = WA
```

```
Feature (fid—7be85bd1_16fc5c446bc_-79e4)
```

```
-----  
the_geom = POINT (-122.459444 47.241389)  
fid = 2  
name = Tacoma  
state = WA
```

Add Fields

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

```
geo-shell> layer addfields --input-name points --output-workspace mem --output-name points2  
--fields "name=String,state=String"
```

| 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 | | |

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

```
geo-shell> layer create --workspace mem --name points --fields "the_geom=Point EPSG:4326"  
Created Layer points!
```

```
geo-shell> layer addfields --input-name points --output-workspace mem --output-name points2  
--fields "name=String,state=String"  
Done!
```

```
geo-shell> layer schema --name points2  
Name Type
```

```
-----  
the_geom Point  
name String  
state String
```

Add Area Field

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

```
geo-shell> layer addareafield --input-name states --output-workspace mem --output-name  
states_area --area-fieldname AREA
```

| 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 |

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

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

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

```
geo-shell> layer addareafield --input-name states --output-workspace mem --output-name  
states_area --area-fieldname AREA  
Done!
```

```
geo-shell> layer schema --name states_area  
Name Type
```

```
-----  
the_geom MultiPolygon  
FID_1 Integer  
ScaleRank Integer  
FeatureCla String  
OBJECTID Integer
```

```

VertexCou Double
ISO String
NAME_0 String
NAME_1 String
VARNAME_1 String
NL_NAME_1 String
HASC_1 String
TYPE_1 String
ENGTYPY_1 String
VALIDFR_1 String
VALIDTO_1 String
REMARKS_1 String
Region String
RegionVar String
ProvNumber Integer
NEV_Countr String
FIRST_FIPS String
FIRST_HASC String
FIPS_1 String
gadm_level Double
CheckMe Integer
Region_Cod String
Region_C_1 String
ScaleRan_1 Integer
Region_C_2 String
Region_C_3 String
Country_Pr String
AREA Double

```

```
geo-shell> layer features --name states_area --filter "NAME_1='North Dakota'" --field "NAME_0,AREA"
```

Feature (fid—7be85bd1_16fc5c446bc_-792a)

```

NAME_0 = United States
AREA = 21.804544852979944

```

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
```

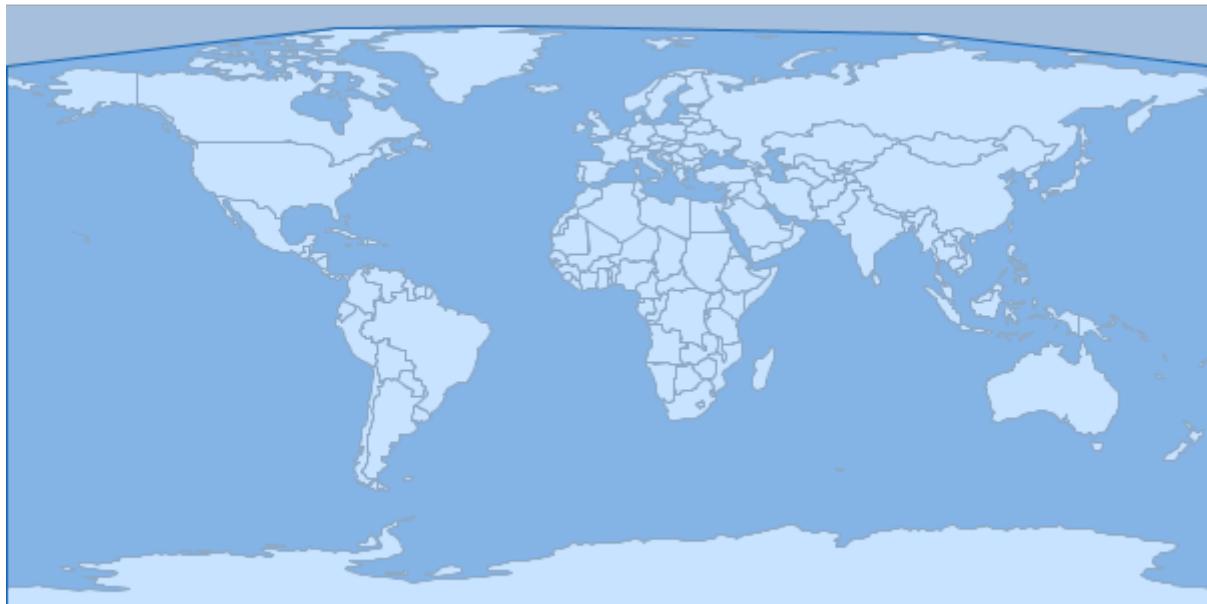
```
Added convexhull layer to map map
```

```
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
```

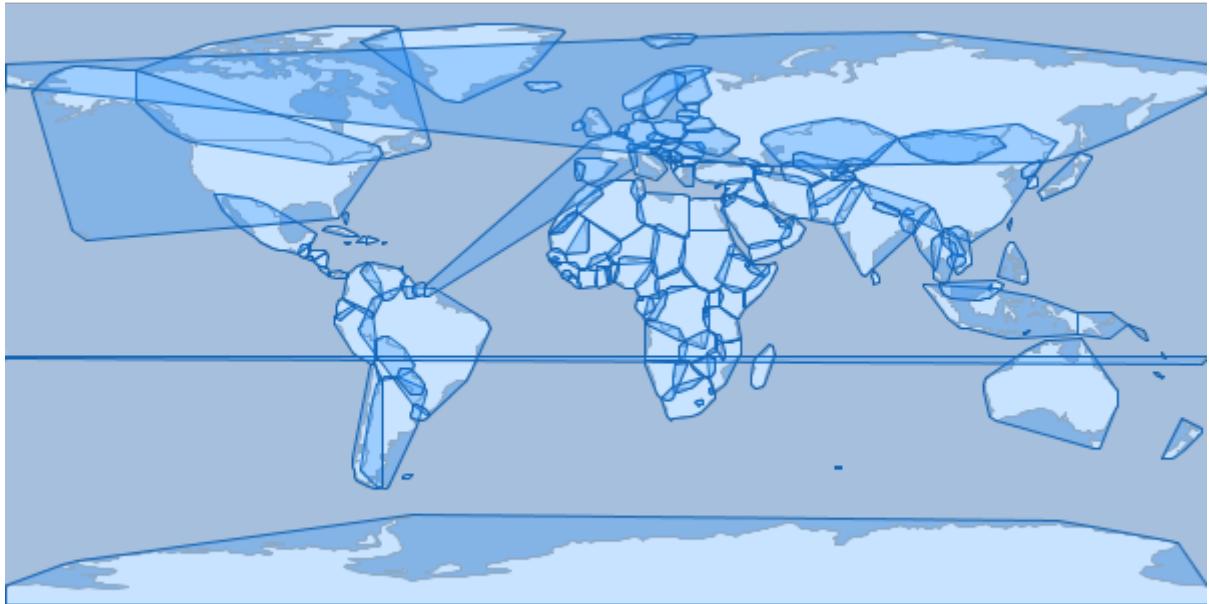
```
Added convexhulls layer to map map
```

```
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
```

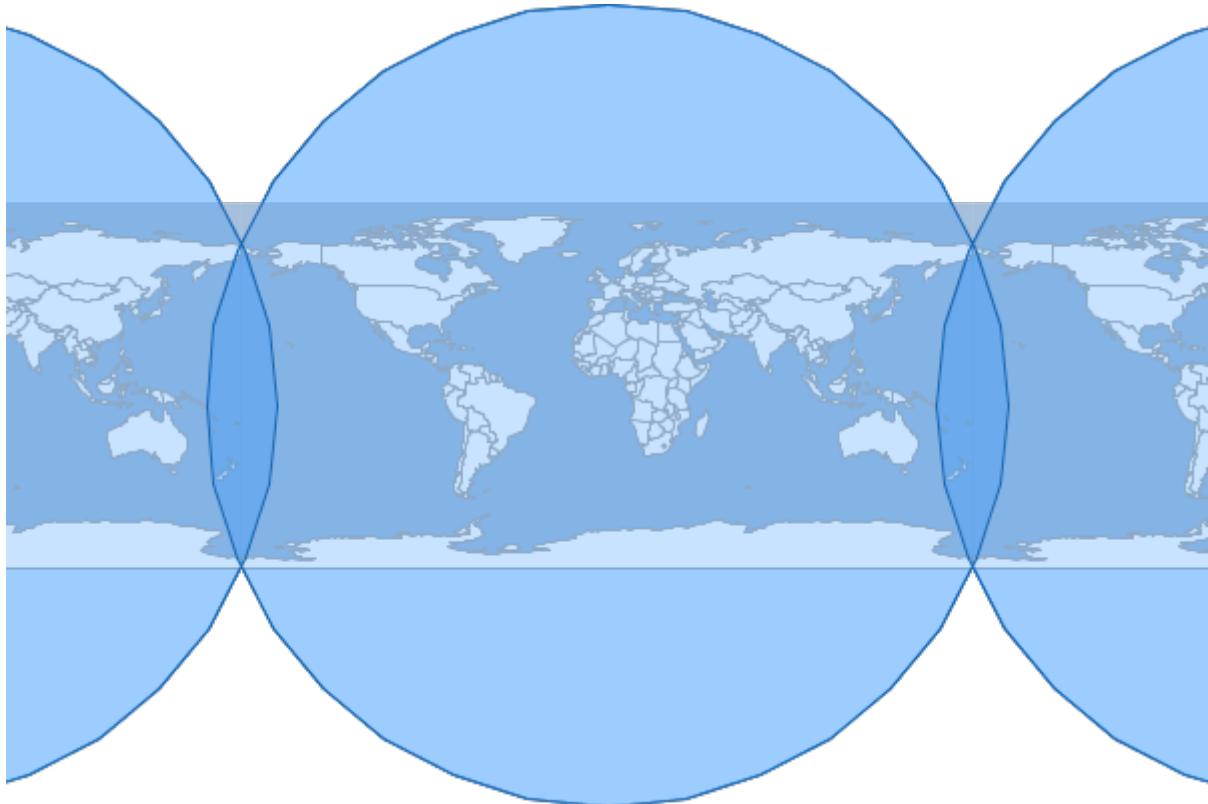
Added mincircle layer to map map

```
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/naturelearth.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!
```

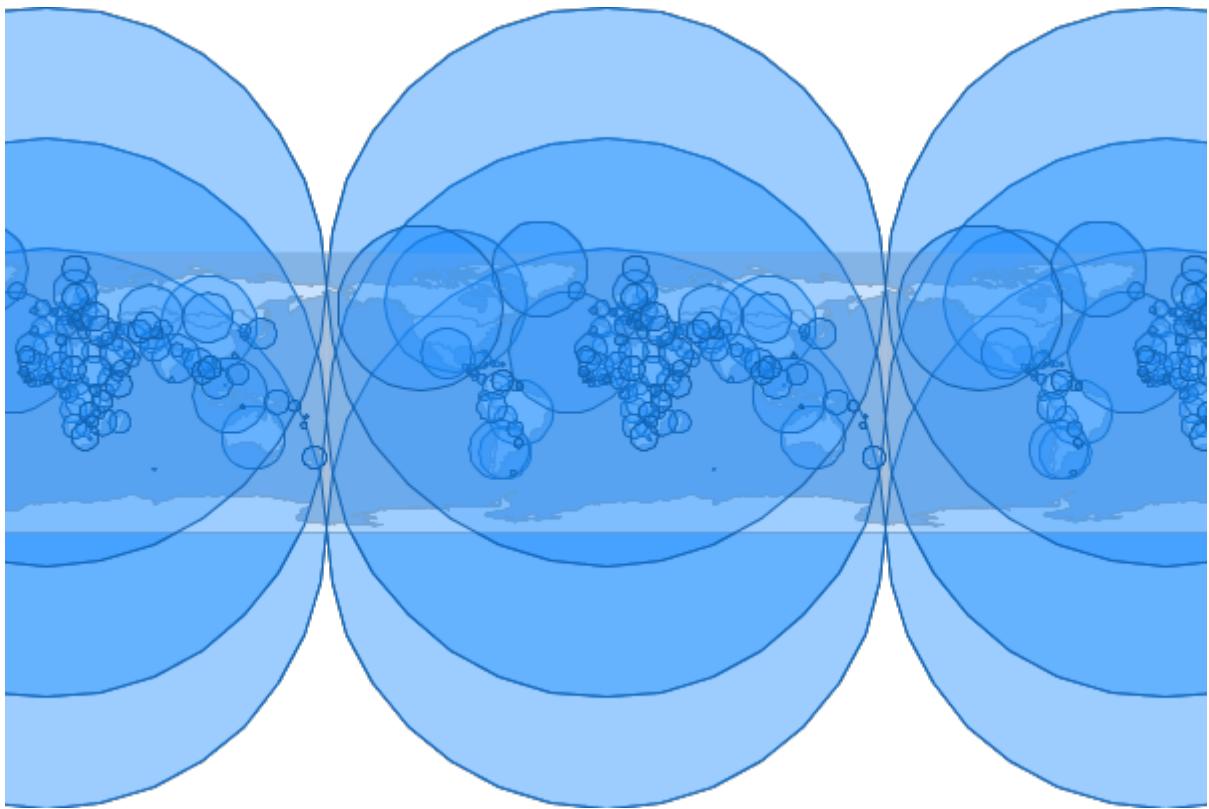
```
geo-shell> map add layer --name map --layer ocean
Added ocean layer to map map
```

```
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
```

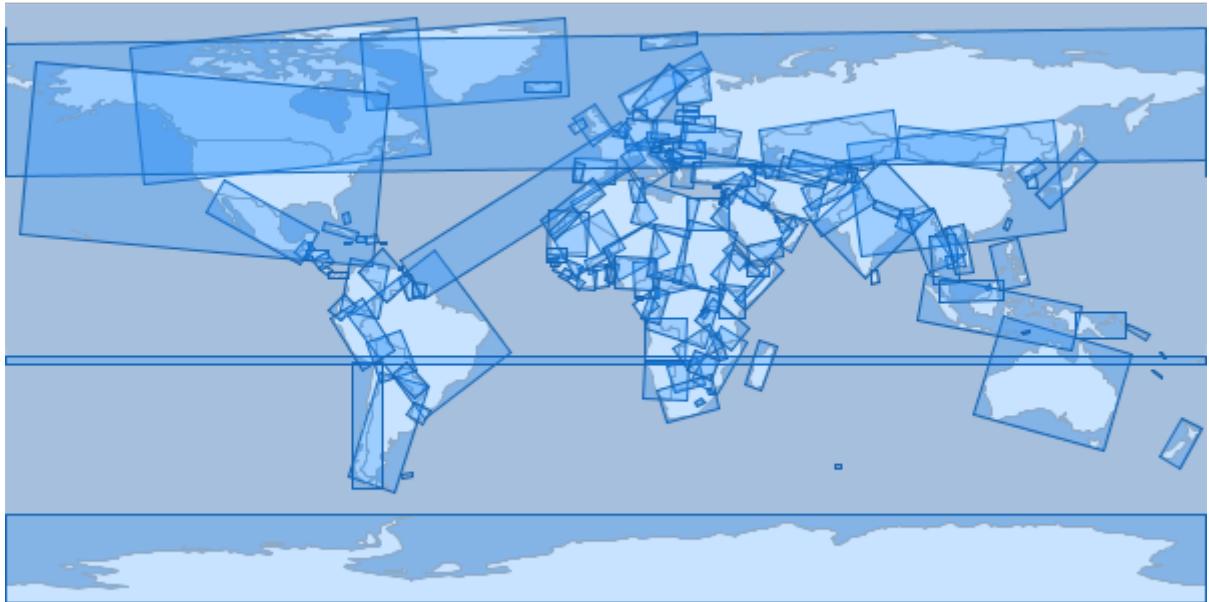
```
Added minrects layer to map map
```

```
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
```

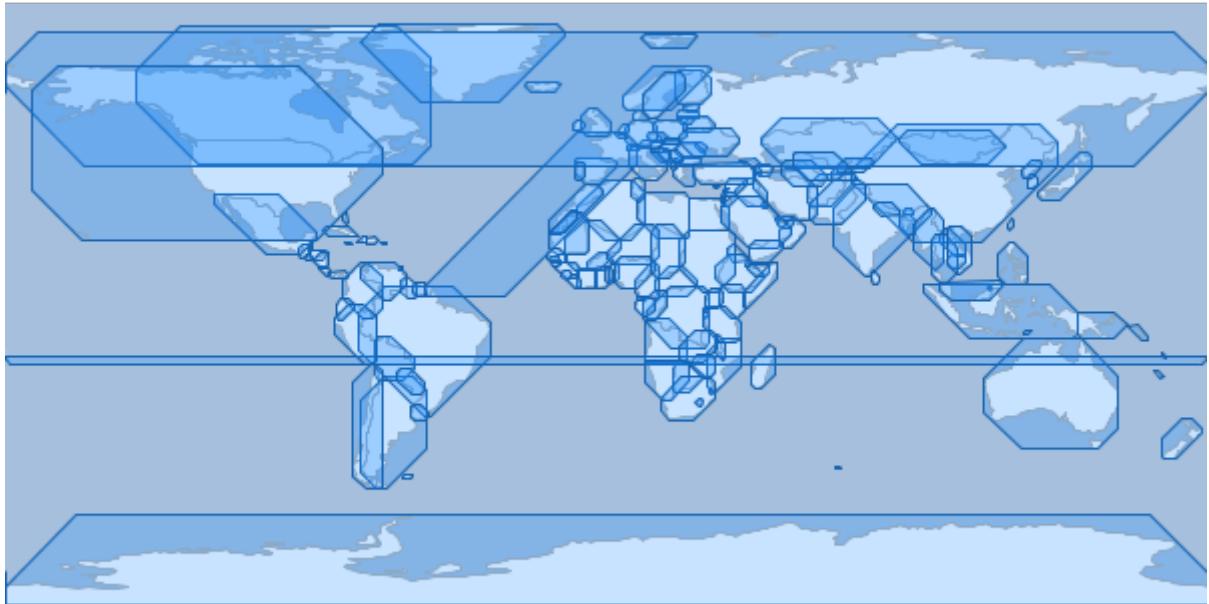
Added octagonalenvelopes layer to map map

```
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 |

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 | | |

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/naturelearth.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!
```

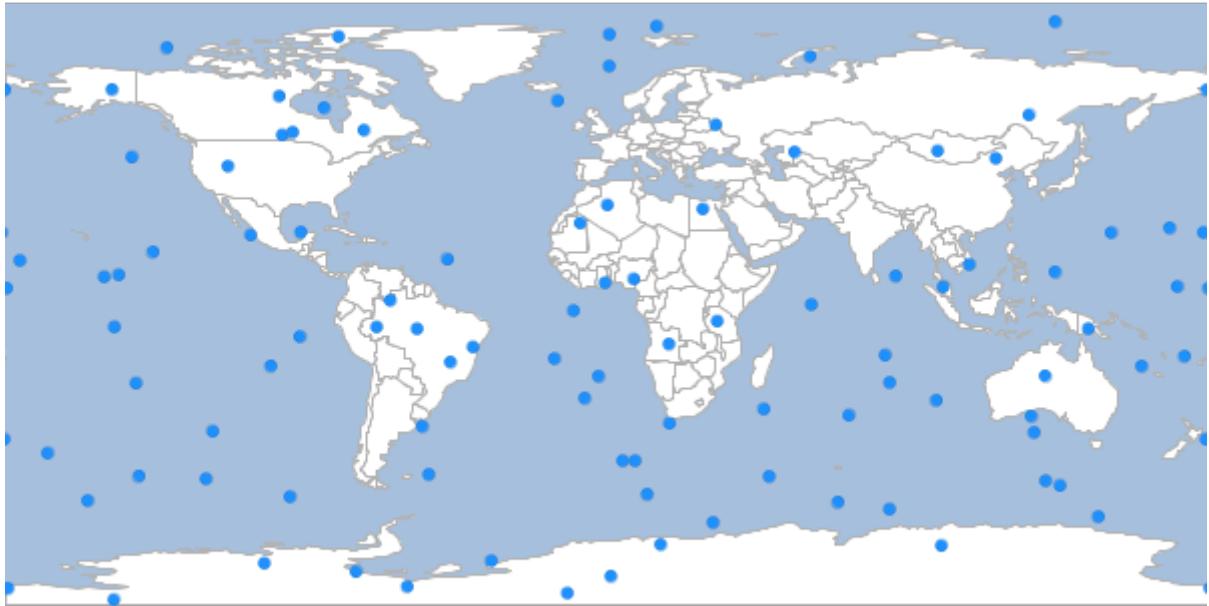
```
geo-shell> map add layer --name randomMap --layer ocean  
Added ocean layer to map randomMap
```

```
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

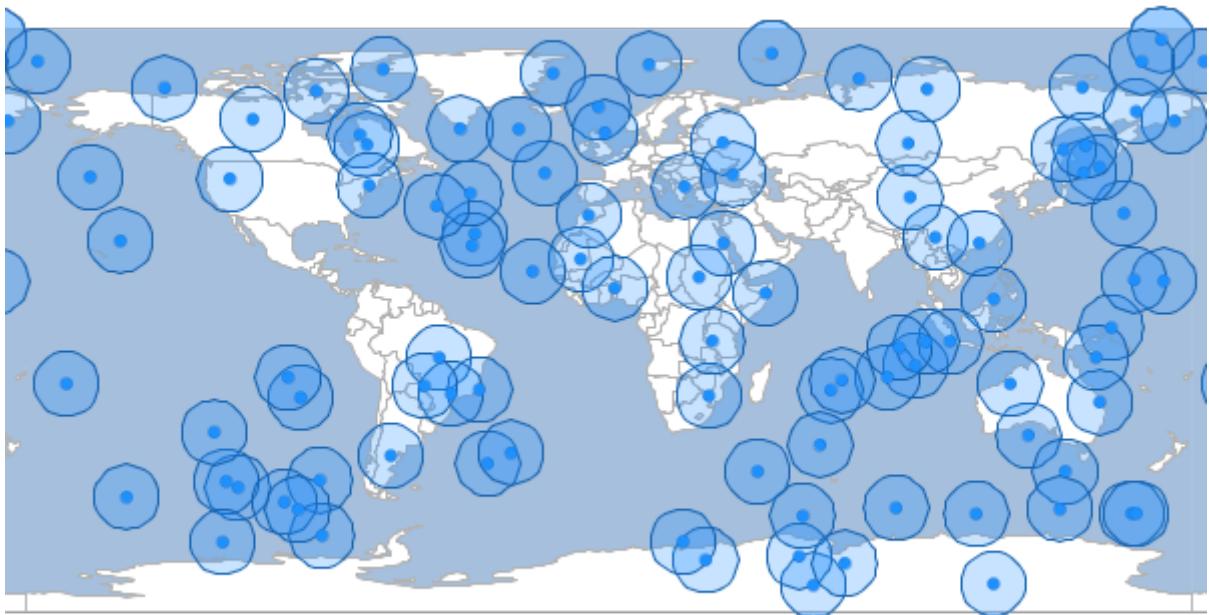
geo-shell> map add layer --name map --layer countries
Added countries layer to map map

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!
```

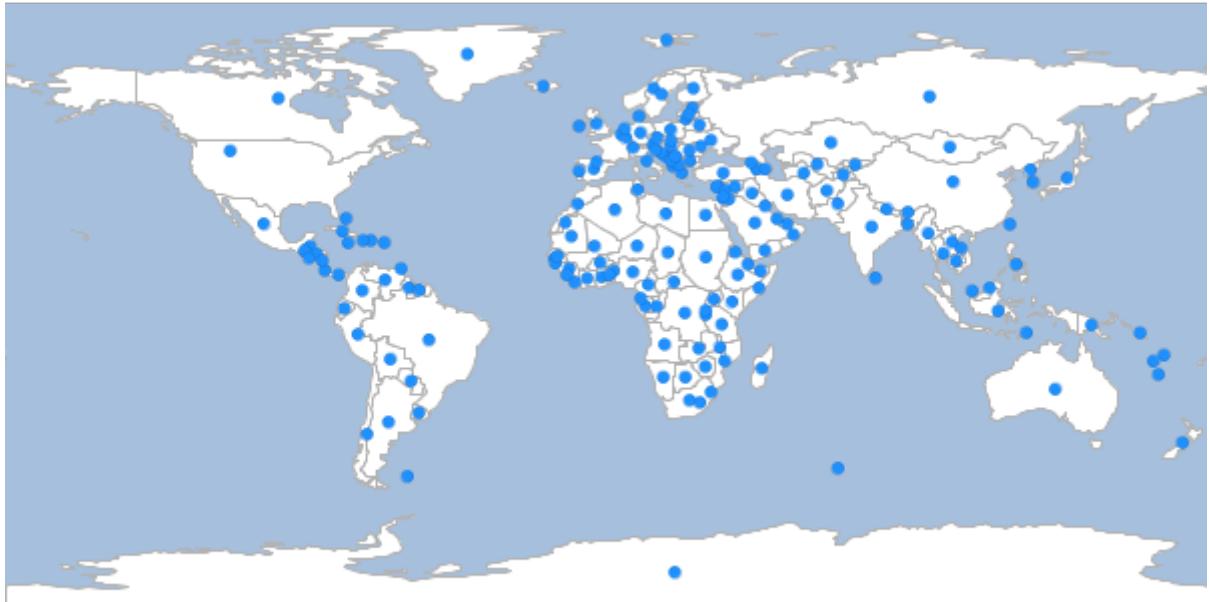
```
geo-shell> map add layer --name map --layer ocean
Added ocean layer to map map
```

```
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!
```

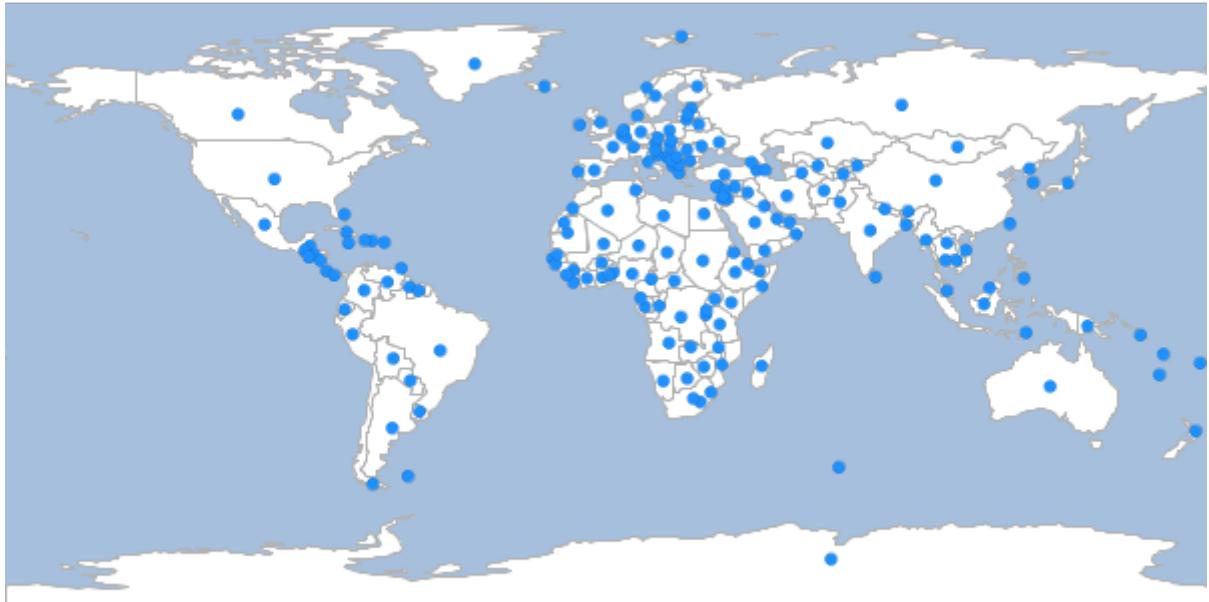
```
geo-shell> map add layer --name map --layer ocean  
Added ocean layer to map map
```

```
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
```

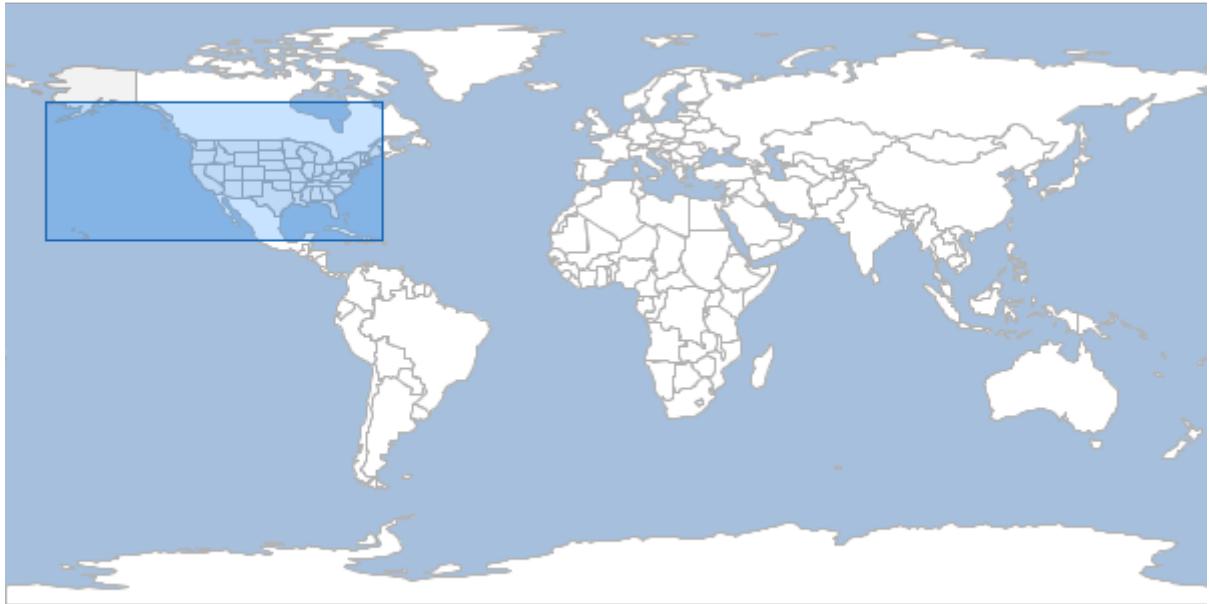
```
geo-shell> map add layer --name map --layer countries  
Added countries layer to map map
```

```
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

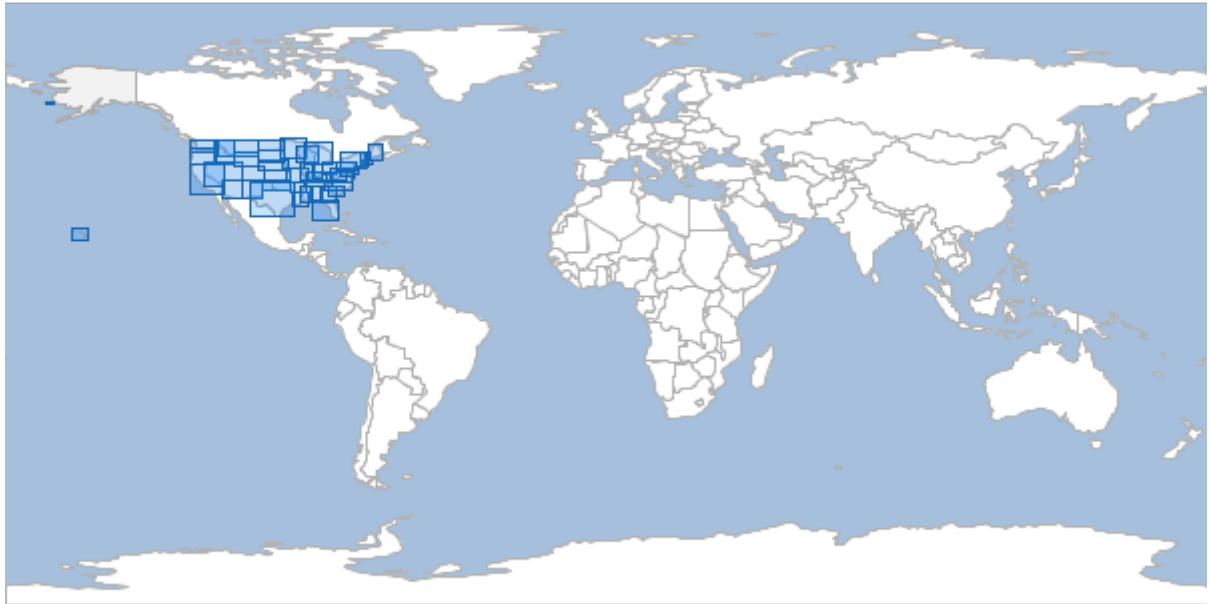
geo-shell> map add layer --name map --layer countries
Added countries layer to map map

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!
```

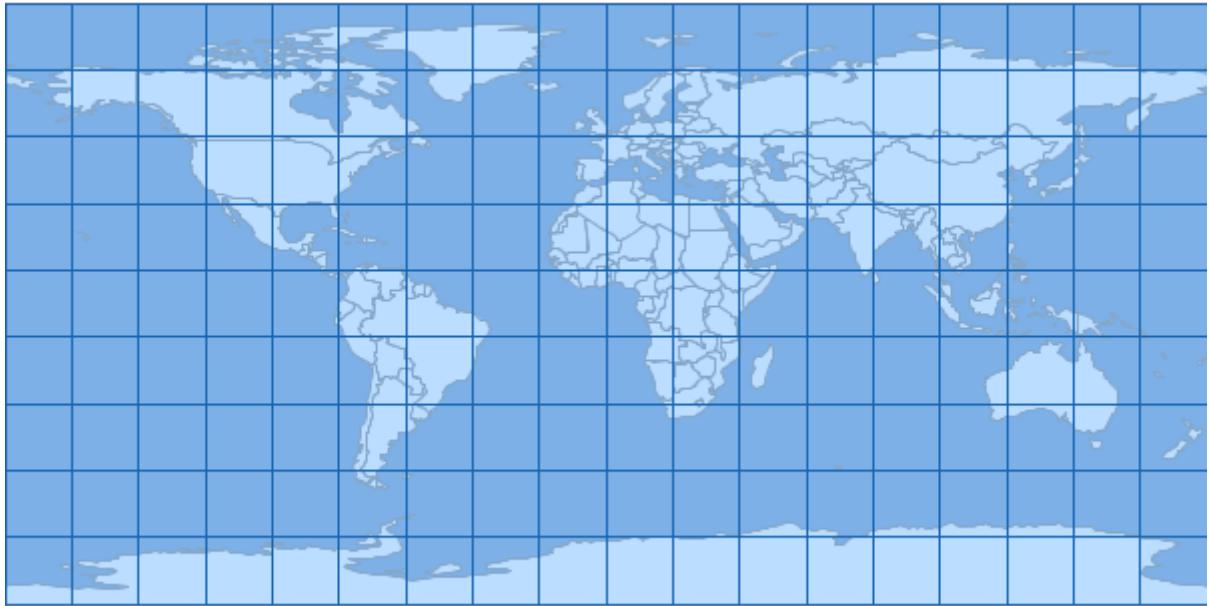
```
geo-shell> map add layer --name graticule --layer ocean
Added ocean layer to map graticule
```

```
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!
```

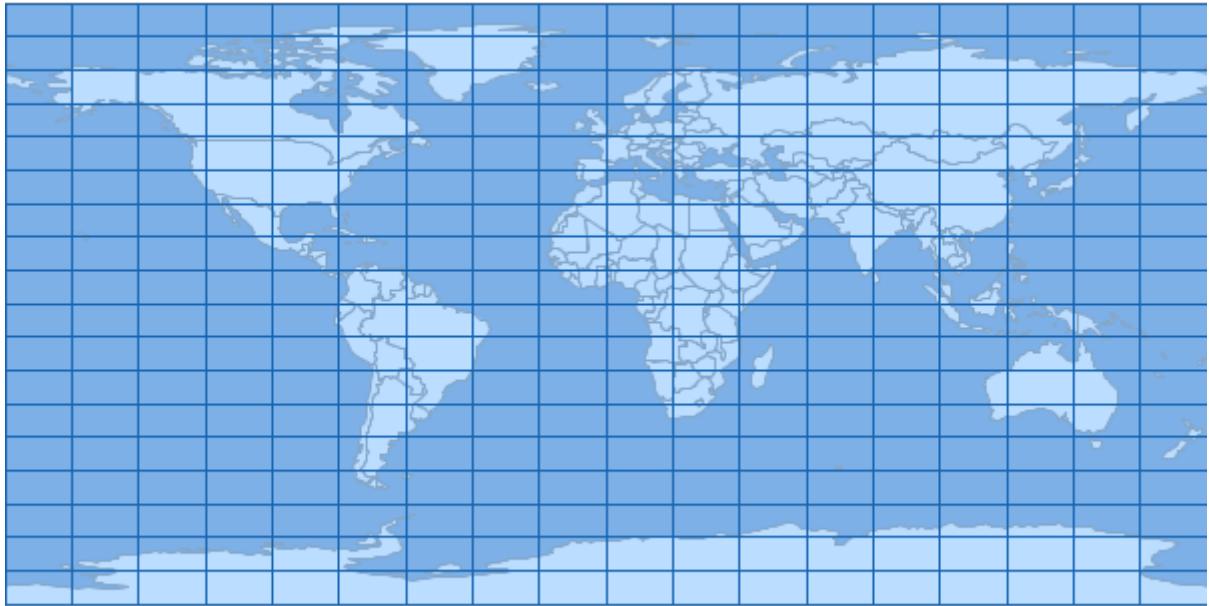
```
geo-shell> map add layer --name graticule --layer ocean  
Added ocean layer to map graticule
```

```
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!
```

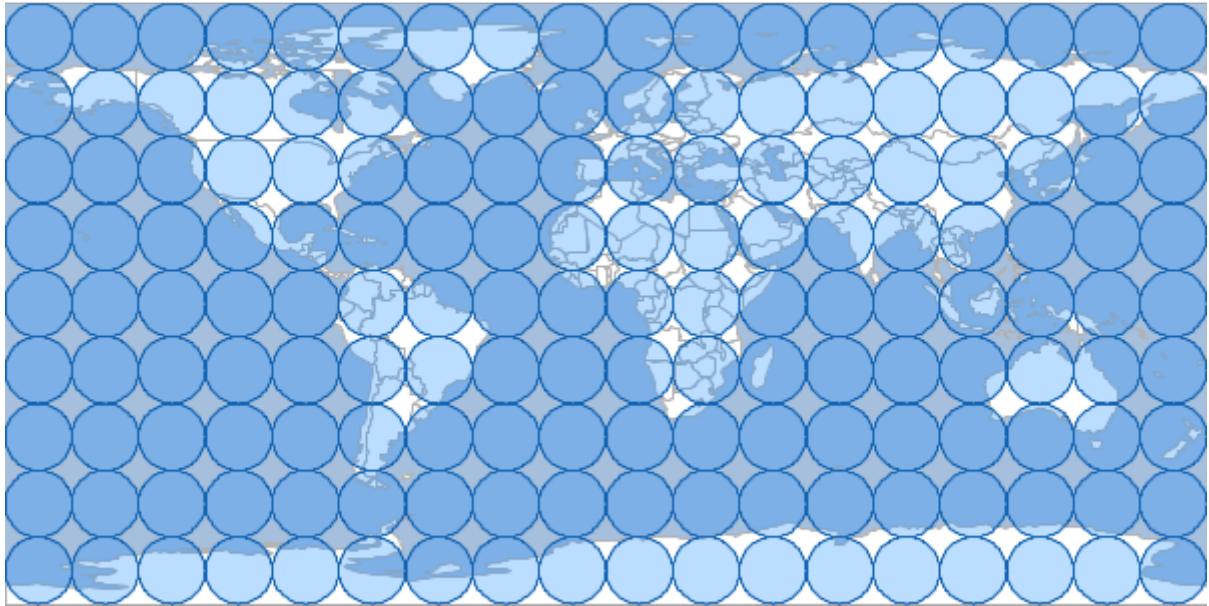
```
geo-shell> map add layer --name graticule --layer ocean  
Added ocean layer to map graticule
```

```
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!
```

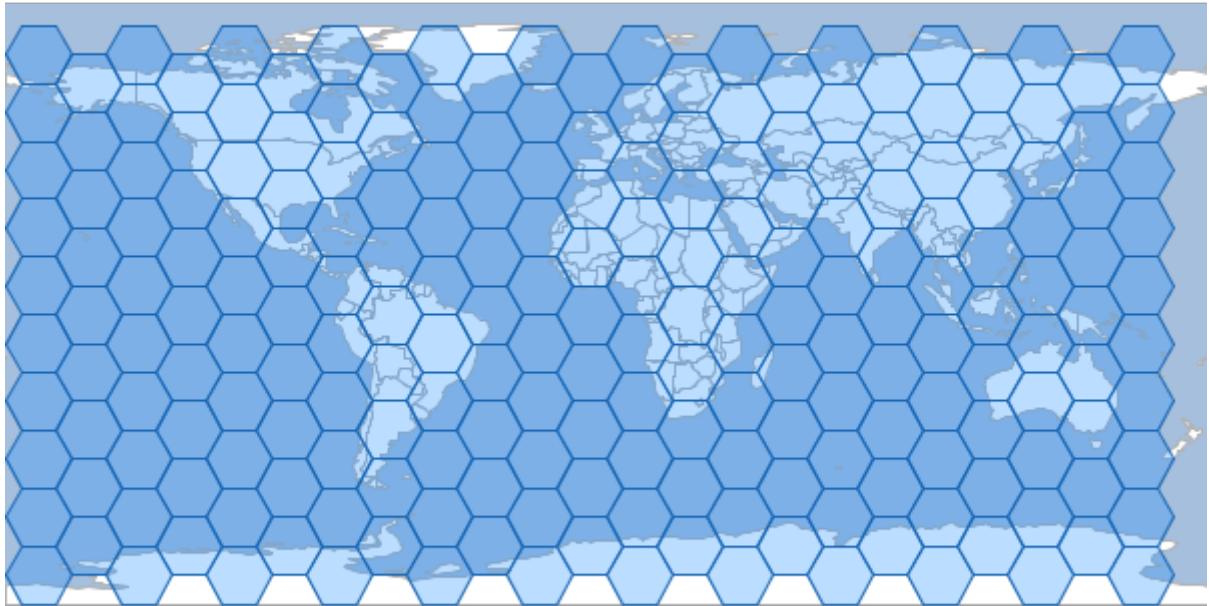
```
geo-shell> map add layer --name graticule --layer ocean  
Added ocean layer to map graticule
```

```
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.

```
geo-shell> raster info --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 info --name earth  
Format: GeoTIFF  
Size: 800, 400  
Projection ID: EPSG:4326  
Projection WKT: GEOGCS["WGS 84",  
DATUM["World Geodetic System 1984",  
SPHEROID["WGS 84", 6378137.0, 298.257223563, AUTHORITY["EPSG", "7030"]],
```

```

AUTHORITY["EPSG","6326"],
PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG","8901"]],
UNIT["degree", 0.017453292519943295],
AXIS["Geodetic longitude", EAST],
AXIS["Geodetic latitude", NORTH],
AUTHORITY["EPSG","4326"]

Extent: -179.9999999999997, -89.9999999998205, 179.99999999996405, 90.0
Pixel Size: 0.4499999999995505, 0.449999999999551
Block Size: 800, 8
Bands:
RED_BAND
Min Value: 56.0 Max Value: 255.0
GREEN_BAND
Min Value: 84.0 Max Value: 255.0
BLUE_BAND
Min Value: 91.0 Max Value: 255.0

```

```

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

```

```

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

```

Value

Get a value from the Raster.

```

geo-shell> raster value --name earth --x 60 --y 45

```

```

geo-shell> raster value --name earth --x 10 --y 15 --type pixel

```

| 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 |

```

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 value --name earth --x 60 --y 45
235.0

```

```
geo-shell> raster value --name earth --x 10 --y 15 --type pixel  
109.0
```

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

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

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 versa | 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,
 -2.003747120513706E7,2.0036395147881314E7,2.003747120513706E7,EPSC: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.

```
geo-shell> style create --params "stroke=black stroke-width=0.25 fill=wheat" --file examples/style_create.sld
```

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

```
geo-shell> style create --params "stroke=black stroke-width=0.25 fill=wheat" --file examples/style_create.sld
```

```
Style stroke=black stroke-width=0.25 fill=wheat written to /home/travis/build/jericks/geo-shell/examples/style_create.sld!
```

```
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 style set --name countries --style examples/style_create.sld
```

Style /home/travis/build/jericks/geo-shell/examples/style_create.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!
```

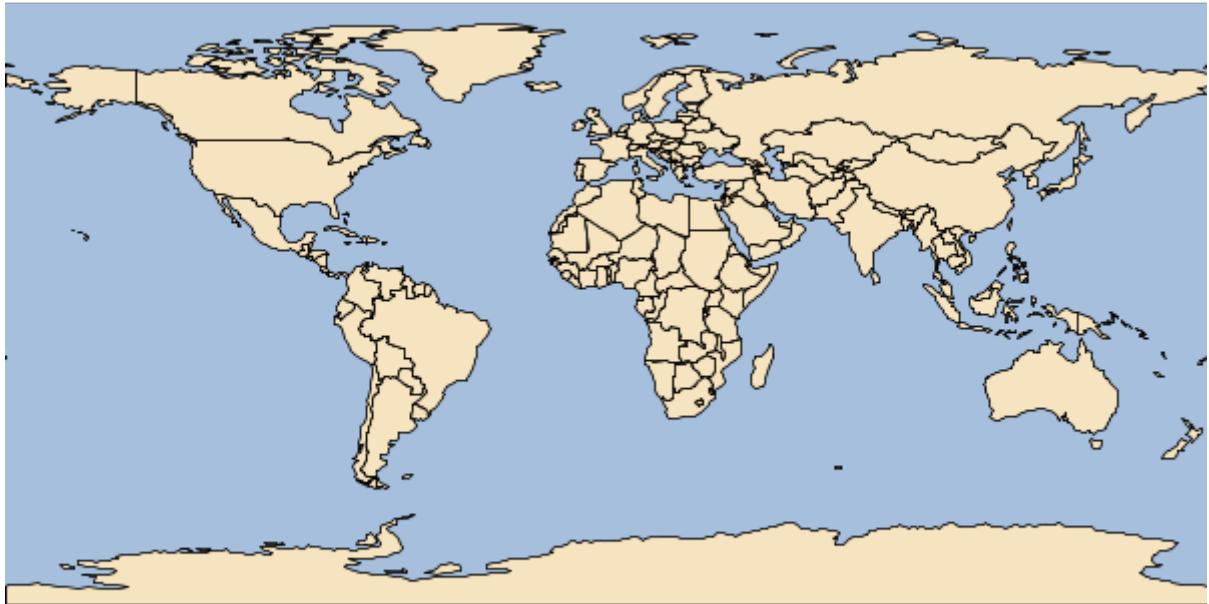
```
geo-shell> map add layer --name map --layer ocean
Added ocean layer to map map
```

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

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

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

```
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <sld:PolygonSymbolizer>
            <sld:Fill>
              <sld:CssParameter name="fill">#f5deb3</sld:CssParameter>
              <sld:CssParameter name="fill-opacity">0.6</sld:CssParameter>
            </sld:Fill>
          </sld:PolygonSymbolizer>
          <sld:LineSymbolizer>
            <sld:Stroke>
              <sld:CssParameter name="stroke-width">0.25</sld:CssParameter>
            </sld:Stroke>
          </sld:LineSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>
```



Vector Default

Create a default vector style.

```
geo-shell> style vector default --layer countries --color #F5F5DC --file examples/countries_default.sld
```

| 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 | | |

```
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> style vector default --layer countries --color #F5F5DC --file examples/countries_default.sld
Default Vector Style for countries written to /home/travis/build/jericks/geo-shell/examples/countries_default.sld!
```

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

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

```
geo-shell> style vector default --layer ocean --color DeepSkyBlue --file examples/ocean_default.sld
Default Vector Style for ocean written to /home/travis/build/jericks/geo-shell/examples/ocean_default.sld!
```

```
geo-shell> layer style set --name ocean --style examples/ocean_default.sld
Style /home/travis/build/jericks/geo-shell/examples/ocean_default.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 draw --name map --file examples/style_vector_default.png
Done drawing /home/travis/build/jericks/geo-shell/examples/style_vector_default.png!
```

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

Country Style

```

<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <sld:PolygonSymbolizer>
            <sld:Fill>
              <sld:CssParameter name="fill">#f5f5dc</sld:CssParameter>
            </sld:Fill>
          </sld:PolygonSymbolizer>
          <sld:LineSymbolizer>
            <sld:Stroke>
              <sld:CssParameter name="stroke">#abab9a</sld:CssParameter>
              <sld:CssParameter name="stroke-width">0.5</sld:CssParameter>
            </sld:Stroke>
          </sld:LineSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>

```

Ocean Style

```
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <sld:PolygonSymbolizer>
            <sld:Fill>
              <sld:CssParameter name="fill">#00bfff</sld:CssParameter>
            </sld:Fill>
          </sld:PolygonSymbolizer>
          <sld:LineSymbolizer>
            <sld:Stroke>
              <sld:CssParameter name="stroke">#0085b2</sld:CssParameter>
              <sld:CssParameter name="stroke-width">0.5</sld:CssParameter>
            </sld:Stroke>
          </sld:LineSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>
```



Vector Gradient

Create a gradient vector style.

```
geo-shell> style vector gradient --layer countries --field PEOPLE --colors greens --number 8  
--method quantile --file examples/style_vector_gradient.sld
```

| 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 | | |

```
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> style vector gradient --layer countries --field PEOPLE --colors greens --number 8
--method quantile --file examples/style_vector_gradient.sld
Gradient Vector Style for countries's PEOPLE Field written to /home/travis/build/jericks/geo-
shell/examples/style_vector_gradient.sld!

geo-shell> layer style set --name countries --style examples/style_vector_gradient.sld
Style /home/travis/build/jericks/geo-shell/examples/style_vector_gradient.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 draw --name map --file examples/style_vector_gradient.png
Done drawing /home/travis/build/jericks/geo-shell/examples/style_vector_gradient.png!

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

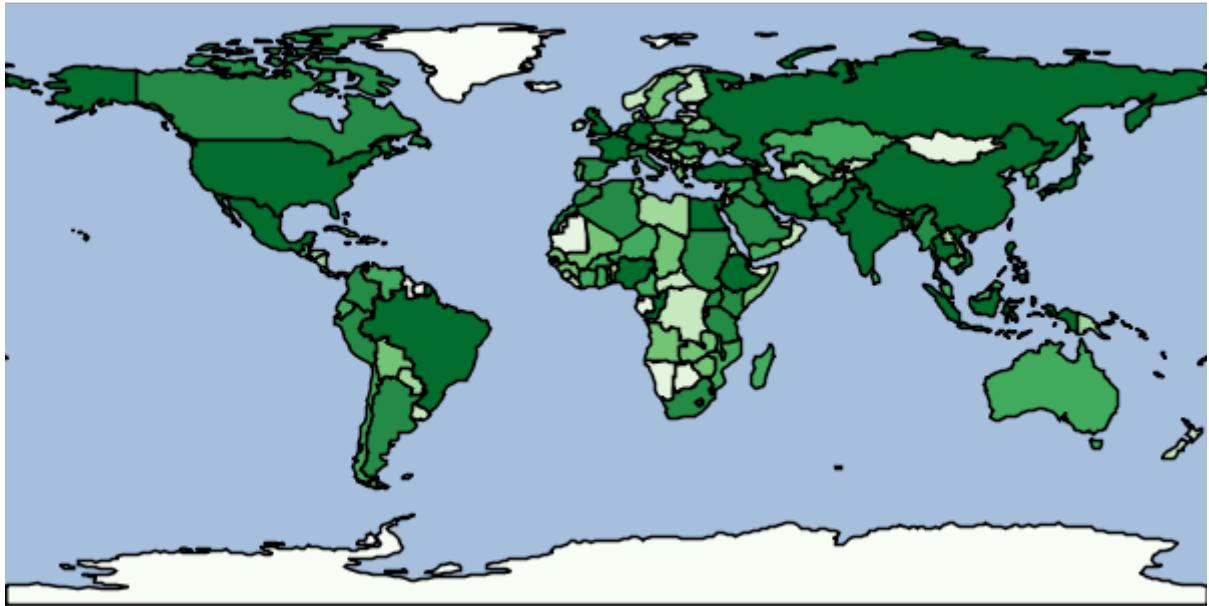
```

```

<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
 xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <ogc:Filter>
            <ogc:And>
              <ogc:PropertyIsGreaterThanOrEqualTo>
                <ogc:PropertyName>PEOPLE</ogc:PropertyName>
                <ogc:Literal>0</ogc:Literal>
              </ogc:PropertyIsGreaterThanOrEqualTo>
              <ogc:PropertyIsLessThan>
                <ogc:PropertyName>PEOPLE</ogc:PropertyName>
                <ogc:Literal>833285</ogc:Literal>
              </ogc:PropertyIsLessThan>
            </ogc:And>
          </ogc:Filter>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>

```

```
</ogc:PropertyIsLessThan>
</ogc:And>
</ogc:Filter>
<sld:PolygonSymbolizer>
  <sld:Fill>
    <sld:CssParameter name="fill">#F7FCF5</sld:CssParameter>
  </sld:Fill>
</sld:PolygonSymbolizer>
<sld:LineSymbolizer>
  <sld:Stroke/>
</sld:LineSymbolizer>
</sld:Rule>
<sld:Rule>
  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsGreaterThanOrEqualTo>
        <ogc:PropertyName>PEOPLE</ogc:PropertyName>
        <ogc:Literal>833285</ogc:Literal>
      </ogc:PropertyIsGreaterThanOrEqualTo>
      <ogc:PropertyIsLessThan>
        <ogc:PropertyName>PEOPLE</ogc:PropertyName>
        <ogc:Literal>3360474</ogc:Literal>
      </ogc:PropertyIsLessThan>
    </ogc:And>
  </ogc:Filter>
  <sld:PolygonSymbolizer>
    <sld:Fill>
      <sld:CssParameter name="fill">#E5F5E0</sld:CssParameter>
    </sld:Fill>
  </sld:PolygonSymbolizer>
  <sld:LineSymbolizer>
```



Vector Unique Values

Create a unique values vector style.

```
geo-shell> style vector uniquevalues --layer countries --field NAME --colors random --file examples/style_vector_uniquevalues.sld
```

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|--------|-----------------|-----------|-------------------|---------------------|
| layer | The Layer | true | | |
| field | The field | true | | |
| colors | The colors | true | | |
| file | The output file | 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> style vector uniquevalues --layer countries --field NAME --colors random --file examples/style_vector_uniquevalues.sld
Unique Values Vector Style for countries's NAME Field written to /home/travis/build/jericks/geo-shell/examples/style_vector_uniquevalues.sld!
```

```
geo-shell> layer style set --name countries --style examples/style_vector_uniquevalues.sld
Style /home/travis/build/jericks/geo-shell/examples/style_vector_uniquevalues.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 draw --name map --file examples/style_vector_uniquevalues.png
Done drawing /home/travis/build/jericks/geo-shell/examples/style_vector_uniquevalues.png!

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

```

```

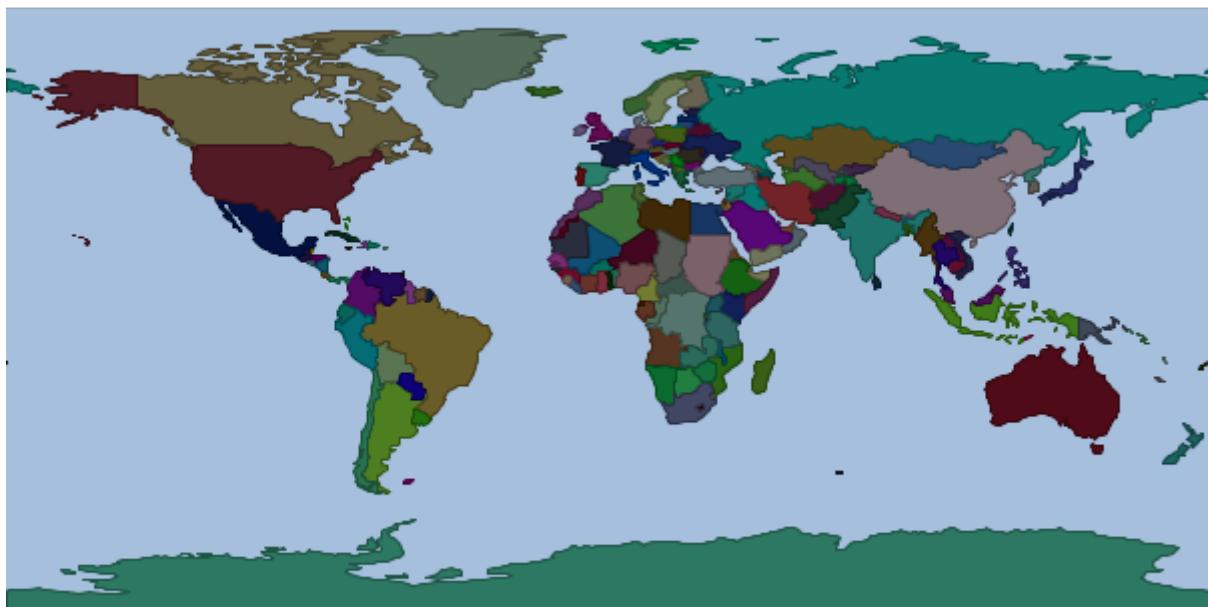
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
  xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
  xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
  version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>NAME</ogc:PropertyName>
              <ogc:Literal>Afghanistan</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <sld:PolygonSymbolizer>
            <sld:Fill>
              <sld:CssParameter name="fill">#4b0f30</sld:CssParameter>
            </sld:Fill>
          </sld:PolygonSymbolizer>
          <sld:LineSymbolizer>
            <sld:Stroke>
              <sld:CssParameter name="stroke">#340a21</sld:CssParameter>
              <sld:CssParameter name="stroke-width">0.5</sld:CssParameter>
            </sld:Stroke>
          </sld:LineSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>

```

```

</sld:Rule>
<sld:Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>NAME</ogc:PropertyName>
      <ogc:Literal>Albania</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <sld:PolygonSymbolizer>
    <sld:Fill>
      <sld:CssParameter name="fill">#19572f</sld:CssParameter>
    </sld:Fill>
  </sld:PolygonSymbolizer>
  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">#113c20</sld:CssParameter>
      <sld:CssParameter name="stroke-width">0.5</sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>
</sld:Rule>
<sld:Rule>
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>

```



Vector Unique Values From Text File

Create a unique values vector style from a text file

geo-shell> style vector uniquevaluesfromtext --field UnitSymbol --textFile

```
src/test/resources/mars/I1802ABC_geo_units_RGBlut.txt --geometryType polygon --styleFile examples/style_vector_uniquevaluesfromtext.sld
```

| 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 ysld file | true | | |

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

```
geo-shell> layer open --workspace mars --layer geo_units_oc_dd --name mars  
Opened Workspace mars Layer geo_units_oc_dd as mars
```

```
geo-shell> style vector uniquevaluesfromtext --field UnitSymbol --textFile  
src/test/resources/mars/I1802ABC_geo_units_RGBlut.txt --geometryType polygon --styleFile  
examples/style_vector_uniquevaluesfromtext.sld  
Create a unique values style from /home/travis/build/jericks/geo-  
shell/src/test/resources/mars/I1802ABC_geo_units_RGBlut.txt for UnitSymbol and polygon to  
/home/travis/build/jericks/geo-shell/examples/style_vector_uniquevaluesfromtext.sld
```

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

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

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

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

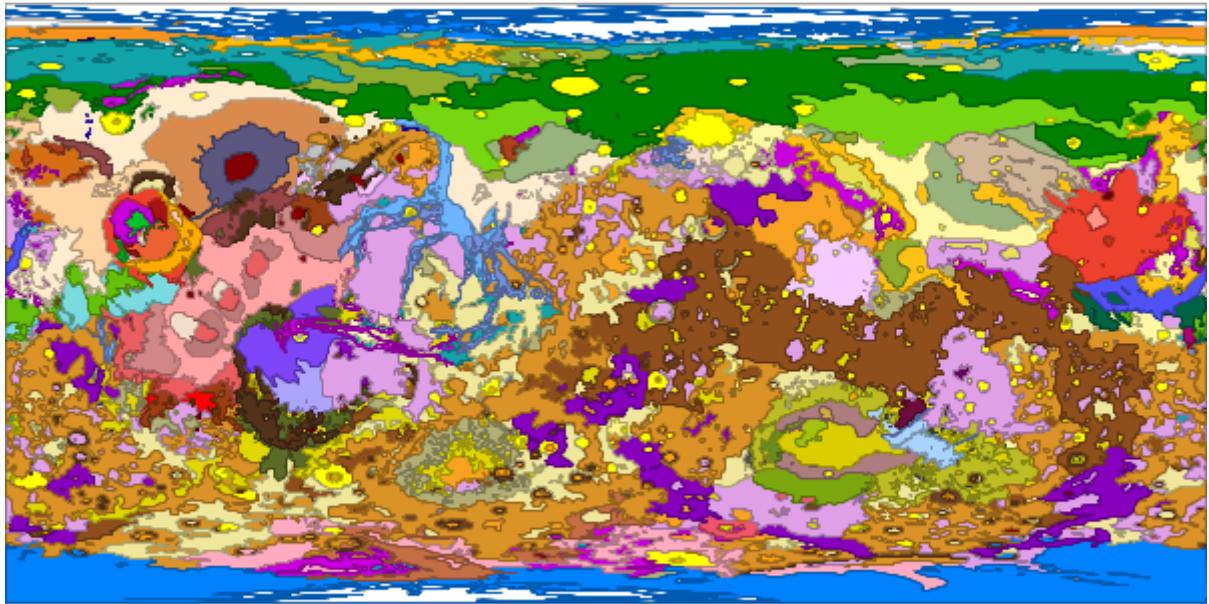
```
geo-shell> map close --name map  
Map map closed!
```

```
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor  
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"  
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"  
version="1.0.0">  
  <sld:UserLayer>  
    <sld:LayerFeatureConstraints>  
      <sld:FeatureTypeConstraint/>  
    </sld:LayerFeatureConstraints>
```

```

<sld:UserStyle>
  <sld:Name>Default Styler</sld:Name>
  <sld:FeatureTypeStyle>
    <sld:Name>name</sld:Name>
    <sld:Rule>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>UnitSymbol</ogc:PropertyName>
          <ogc:Literal>AHa</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <sld:PolygonSymbolizer>
        <sld:Fill>
          <sld.CssParameter name="fill">#af006f</sld.CssParameter>
        </sld:Fill>
      </sld:PolygonSymbolizer>
      <sld:LineSymbolizer>
        <sld:Stroke>
          <sld.CssParameter name="stroke">#7a004d</sld.CssParameter>
          <sld.CssParameter name="stroke-width">0.5</sld.CssParameter>
        </sld:Stroke>
      </sld:LineSymbolizer>
    </sld:Rule>
    <sld:Rule>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>UnitSymbol</ogc:PropertyName>
          <ogc:Literal>AHat</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <sld:PolygonSymbolizer>
        <sld:Fill>
          <sld.CssParameter name="fill">#c03616</sld.CssParameter>
        </sld:Fill>
      </sld:PolygonSymbolizer>
      <sld:LineSymbolizer>
        <sld:Stroke>
          <sld.CssParameter name="stroke">#86250f</sld.CssParameter>
          <sld.CssParameter name="stroke-width">0.5</sld.CssParameter>
        </sld:Stroke>
      </sld:LineSymbolizer>
    </sld:Rule>
    <sld:Rule>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>

```



Raster Default

Create a default raster style.

```
geo-shell> style raster default --raster pc --opacity 0.75 --file examples/style_raster_default.sld
```

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

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

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

```
geo-shell> style raster default --raster pc --opacity 0.75 --file examples/style_raster_default.sld  
Default Raster Style for pc written to /home/travis/build/jericks/geo-  
shell/examples/style_raster_default.sld!
```

```
geo-shell> raster style set --name pc --style examples/style_raster_default.sld  
Style /home/travis/build/jericks/geo-shell/examples/style_raster_default.sld set on pc
```

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

```
geo-shell> map add raster --name map --raster pc
```

```
Added pc layer to map map
```

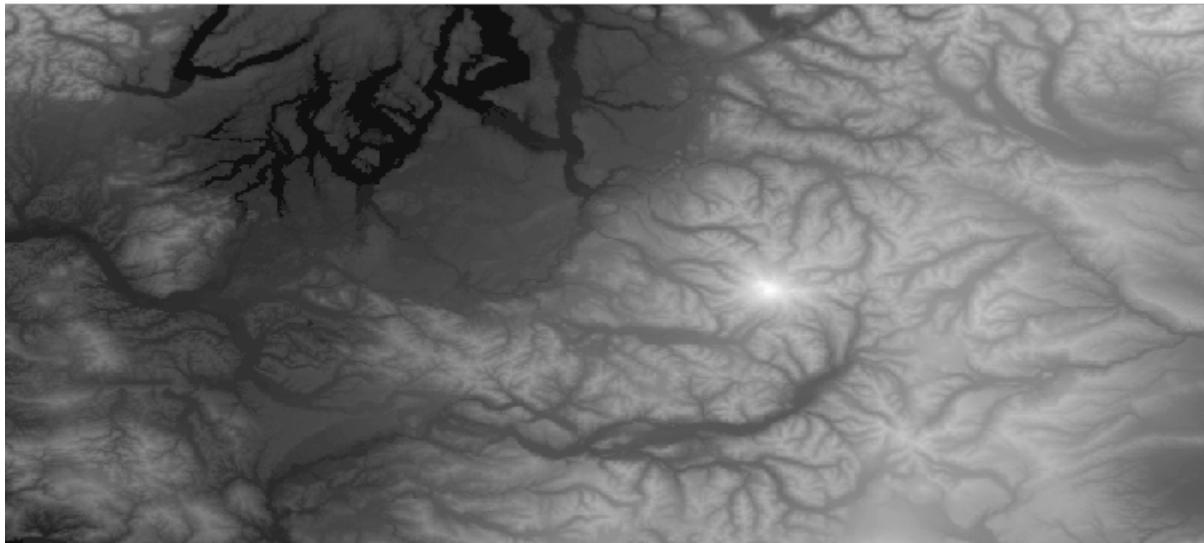
```
geo-shell> map draw --name map --file examples/style_raster_default.png
```

```
Done drawing /home/travis/build/jericks/geo-shell/examples/style_raster_default.png!
```

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

```
Map map closed!
```

```
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
 xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 version="1.0.0">
  <sld:UserLayer>
   <sld:LayerFeatureConstraints>
    <sld:FeatureTypeConstraint/>
   </sld:LayerFeatureConstraints>
   <sld:UserStyle>
    <sld:Name>Default Styler</sld:Name>
    <sld:FeatureTypeStyle>
     <sld:Name>name</sld:Name>
     <sld:Rule>
      <sld:RasterSymbolizer>
       <sld:Geometry>
        <ogc:Literal>grid</ogc:Literal>
       </sld:Geometry>
       <sld:Opacity>0.75</sld:Opacity>
       <sld:ContrastEnhancement/>
      </sld:RasterSymbolizer>
     </sld:Rule>
    </sld:FeatureTypeStyle>
   </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>
```



Raster Color Map

Create a color map raster style.

```
geo-shell> style raster colormap --raster pc --values
"25=#9fd182,470=#3e7f3c,920=#133912,1370=#08306b,1820=#fffff5"
examples/style_raster_colormap.sld
```

| 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 | | |

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

```
geo-shell> raster open --format pierce_county --raster pc --name pc
```

Opened Format pierce_county Raster pc as pc

```
geo-shell> style raster colormap --raster pc --values
"25=#9fd182,470=#3e7f3c,920=#133912,1370=#08306b,1820=#fffff5"
--file examples/style_raster_colormap.sld
Colormap Raster Style for pc written to /home/travis/build/jericks/geo-
shell/examples/style_raster_colormap.sld!
```

```
geo-shell> raster style set --name pc --style examples/style_raster_colormap.sld
Style /home/travis/build/jericks/geo-shell/examples/style_raster_colormap.sld set on pc
```

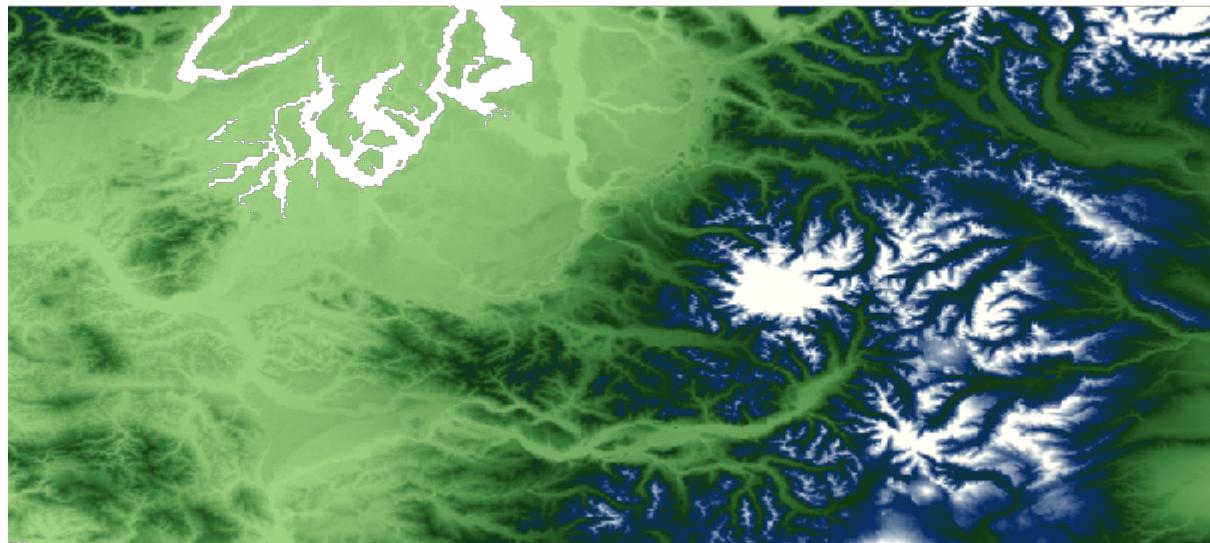
```
geo-shell> map open --name map
Map map opened!
```

```
geo-shell> map add raster --name map --raster pc
Added pc layer to map map
```

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

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

```
<?xml version="1.0" encoding="UTF-8"?><sld:StyledLayerDescriptor
xmlns="http://www.opengis.net/sld" xmlns:sld="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0">
  <sld:UserLayer>
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint/>
    </sld:LayerFeatureConstraints>
    <sld:UserStyle>
      <sld:Name>Default Styler</sld:Name>
      <sld:FeatureTypeStyle>
        <sld:Name>name</sld:Name>
        <sld:Rule>
          <sld:RasterSymbolizer>
            <sld:Geometry>
              <ogc:Literal>grid</ogc:Literal>
            </sld:Geometry>
            <sld:ColorMap>
              <sld:ColorMapEntry color="#9fd182" opacity="1.0" quantity="25"/>
              <sld:ColorMapEntry color="#3e7f3c" opacity="1.0" quantity="470"/>
              <sld:ColorMapEntry color="#133912" opacity="1.0" quantity="920"/>
              <sld:ColorMapEntry color="#08306b" opacity="1.0" quantity="1370"/>
              <sld:ColorMapEntry color="#fffff5" opacity="1.0" quantity="1820"/>
            </sld:ColorMap>
            <sld:ContrastEnhancement/>
          </sld:RasterSymbolizer>
        </sld:Rule>
      </sld:FeatureTypeStyle>
    </sld:UserStyle>
  </sld:UserLayer>
</sld:StyledLayerDescriptor>
```



Map

Open

Open a new Map.

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

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

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

Map earth opened!

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

Map earth closed!

Close

Close a Map.

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

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

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

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

List

List open Maps.

```
geo-shell> map list
```



No parameters

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

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

```
geo-shell> map list
```

earth

us

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

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

Add Layer

Add a Vector Layer.

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

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

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

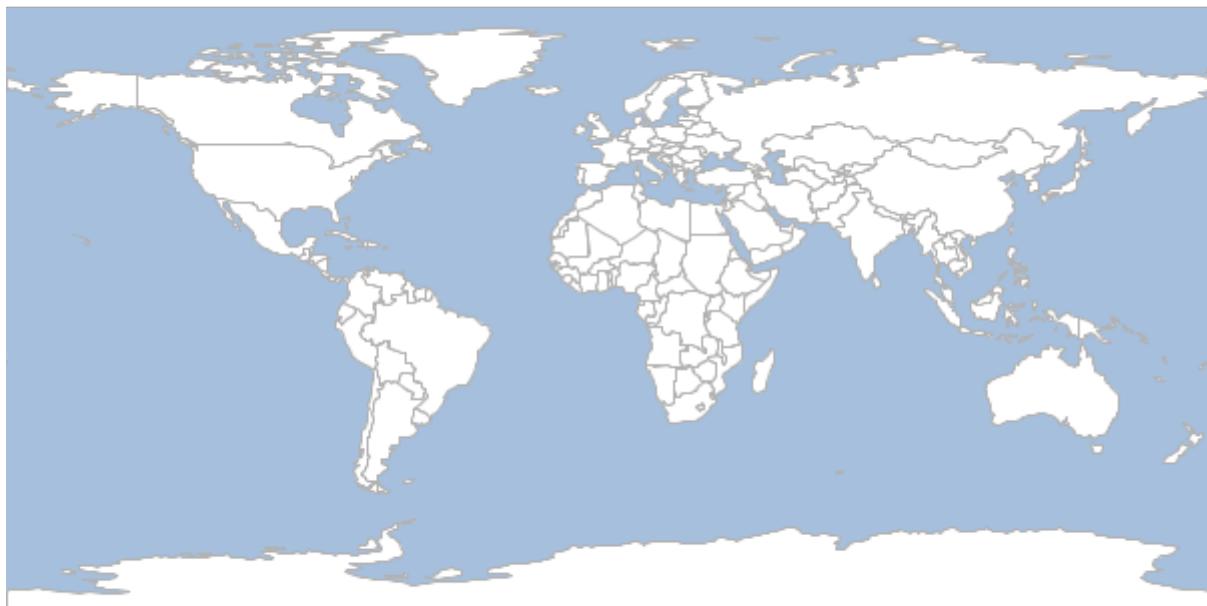
```
geo-shell> layer style set --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

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

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



Add Raster

Add a Raster Layer.

```
geo-shell> map add raster --name world --raster earth
```

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

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

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> **map add raster** --name world --raster earth
Added earth layer to map world

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

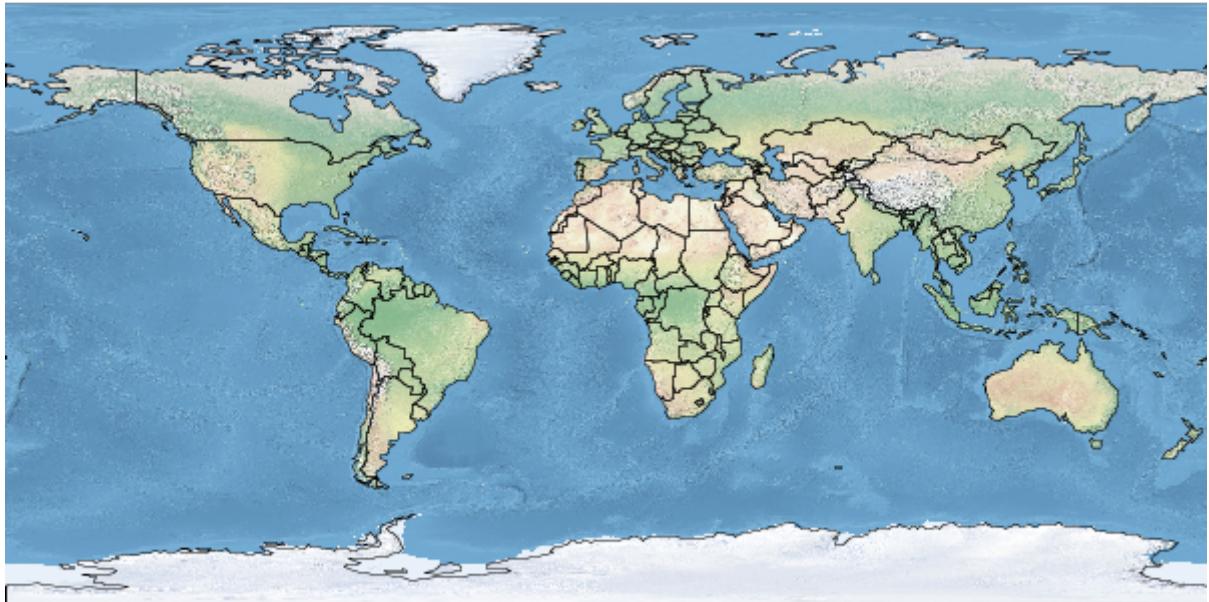
geo-shell> **style create** --params "stroke=black stroke-width=0.1" --file examples/outline.sld
Style stroke=black stroke-width=0.1 written to /home/travis/build/jericks/geo-shell/examples/outline.sld!

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

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

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

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



Add Tile Layer

Add a Tile Layer.

```
geo-shell> map add tile --name world --tile tiles
```

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

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

Map world opened!

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

Tile Layer tiles opened!

```
geo-shell> map add tile --name world --tile tiles
```

Added tiles layer to map world

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

Workspace naturalearth opened!

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

Opened Workspace naturalearth Layer places as places

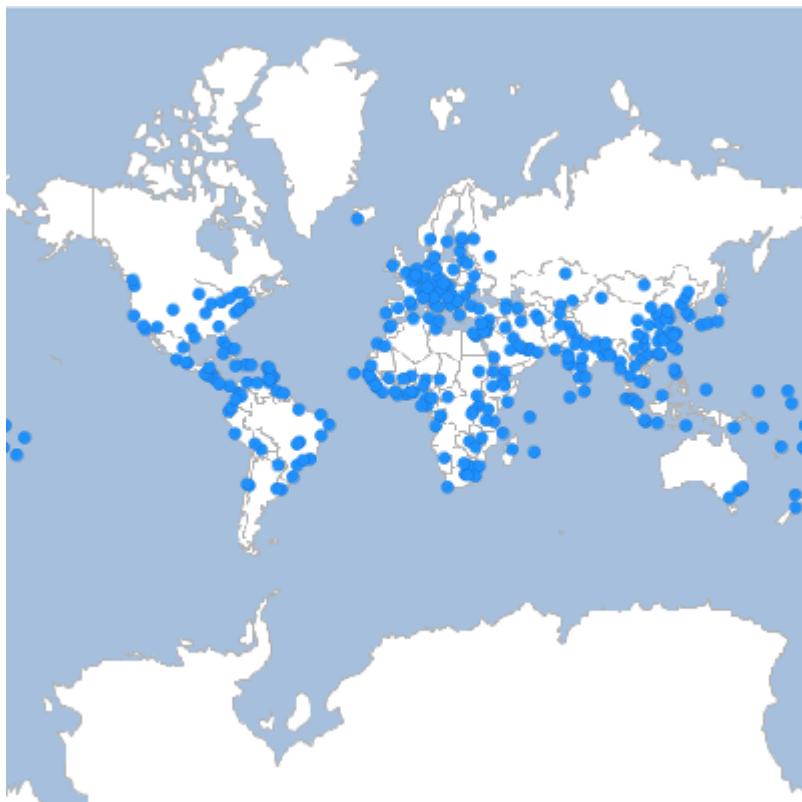
```
geo-shell> style vector default --layer places --color #1E90FF --file examples/places.sld
Default Vector Style for places written to /home/travis/build/jericks/geo-shell/examples/places.sld!
```

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

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

```
geo-shell> map draw --name world --width 400 --height 400 --file examples/map_add_tile.png
Done drawing /home/travis/build/jericks/geo-shell/examples/map_add_tile.png!
```

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



Remove Layer

Remove a Layer.

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

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

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

```
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> map add raster --name world --raster earth
Added earth layer to map world
```

```
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> style create --params "stroke=black stroke-width=0.1" --file examples/outline.sld
Style      stroke=black      stroke-width=0.1      written      to      /home/travis/build/jericks/geo-
shell/examples/outline.sld!
```

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

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

```
geo-shell> map layers --name world
earth
countries
```

```
geo-shell> map remove layer --name world --layer countries
Removed countries layer from map world
```

```
geo-shell> map layers --name world
earth
```

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

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 | | |

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

```
Map world 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> style create --params "stroke=black stroke-width=0.1" --file examples/outline.sld
```

```
Style      stroke=black      stroke-width=0.1      written      to      /home/travis/build/jericks/geo-  
shell/examples/outline.sld!
```

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

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

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

```
Added countries layer to map world
```

```
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> map add raster --name world --raster earth
```

```
Added earth layer to map world
```

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

```
countries
```

```
earth
```

```
geo-shell> map reorder --name world --layer countries --order 1
```

```
Moved countries from 0 to 1
```

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

```
earth
```

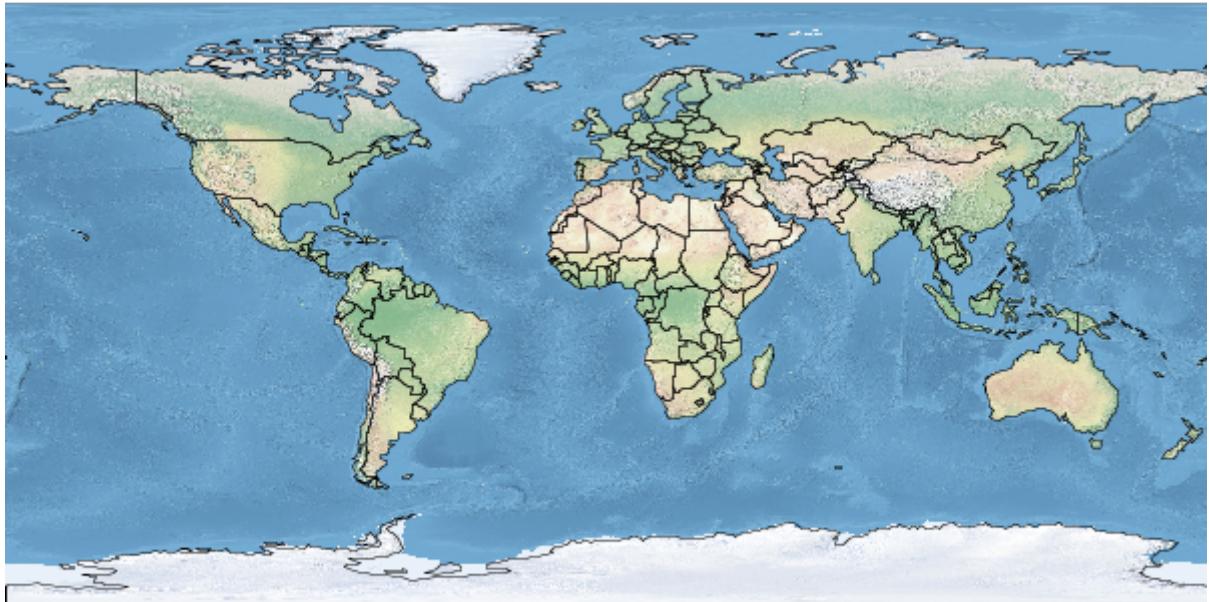
```
countries
```

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

```
Done drawing /home/travis/build/jericks/geo-shell/examples/map_reordered.png!
```

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

```
Map world closed!
```



Layers

List the Map's Layers.

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

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

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

Map world opened!

```
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> map add raster --name world --raster earth
```

Added earth layer to map world

```
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> style create --params "stroke=black stroke-width=0.1" --file examples/outline.sld
```

```
Style stroke=black stroke-width=0.1 written to /home/travis/build/jericks/geo-shell/examples/outline.sld!
```

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

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

```
geo-shell> map layers --name world  
earth  
countries
```

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

Draw

Draw a map.

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

| 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 | | |

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

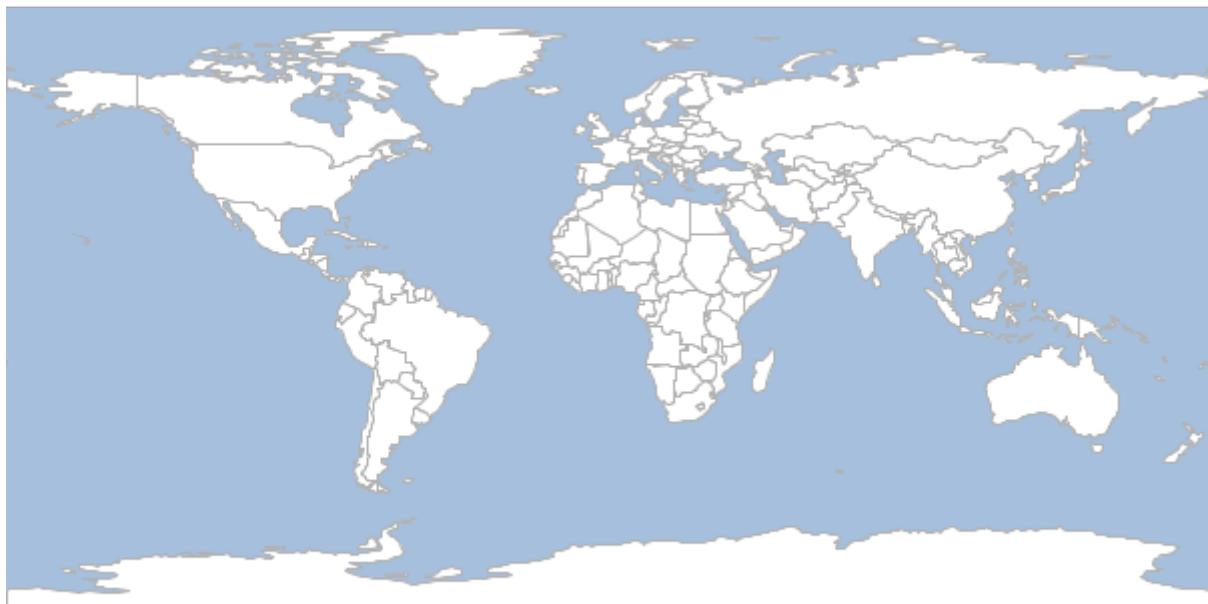
```
geo-shell> layer style set --name ocean --style examples/ocean.sld  
Style /home/travis/build/jericks/geo-shell/examples/ocean.sld set on ocean
```

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

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



Display

Display a map in a GUI.

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

| 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 |
| background-color | The background color | false | | |

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

Map world 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 /Users/jericks/Projects/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 /Users/jericks/Projects/geo-shell/examples/ocean.sld set on ocean

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

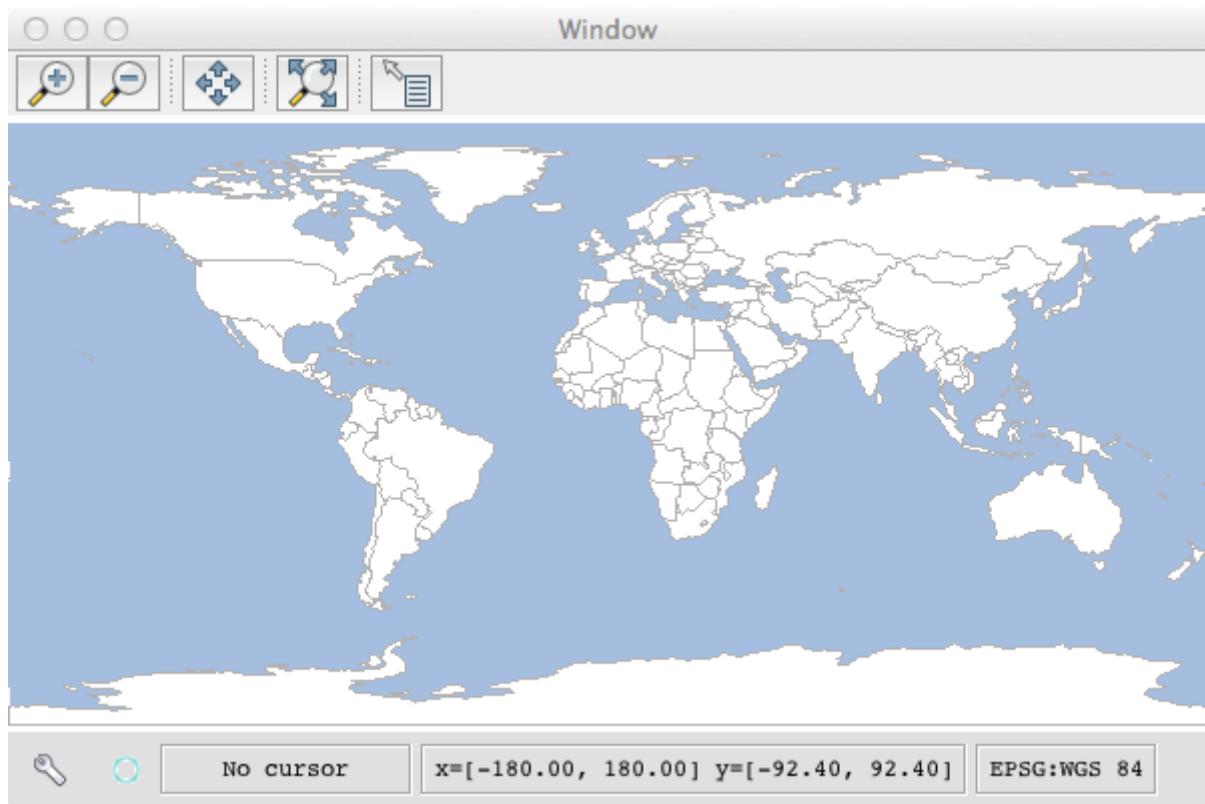
Added ocean layer to map world

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

Added countries layer to map world

geo-shell> **map display** --name world

Displaying...



Built in

Exit / Quit

Exits the shell

```
geo-shell> exit
```



No parameters

Help

List all commands usage

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|------|----------------------------------|-----------|-------------------|---------------------|
| | Command name to provide help for | false | | |

View all commands

```
geo-shell> help
```

- * ! - Allows execution of operating system (OS) commands
- * // - Inline comment markers (start of line only)
- * ; - Inline comment markers (start of line only)
- * clear - Clears the console

- * cls - Clears the console
- * date - Displays the local date and time
- * download - Download a URL to a file.
- * exit - Exits the shell
- * format close - Close a Raster Format.
- * format list - List open Raster Formats.

Get help for a command

```
geo-shell> help layer open
```

Keyword: layer open

Description: Open a Layer.

Keyword: workspace

Help: The Workspace name

Mandatory: true

Default if specified: 'NULL'

Default if unspecified: 'NULL'

Keyword: layer

Help: The Layer name

Mandatory: true

Default if specified: 'NULL'

Default if unspecified: 'NULL'

Keyword: name

Help: The name

Mandatory: false

Default if specified: 'NULL'

Default if unspecified: 'NULL'

* layer open - Open a Layer.

Run OS Command

Allows execution of operating system (OS) commands

```
geo-shell> ! ls src/test/resources/mars
```

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|------|------------------------|-----------|-------------------|---------------------|
| | The command to execute | false | | |

```
geo-shell> ! ls src/test/resources/mars
```

```
geo_units_oc_dd.dbf  
geo_units_oc_dd.prj  
geo_units_oc_dd.qix
```

```
geo_units_oc_dd.sbn  
geo_units_oc_dd.sbx  
geo_units_oc_dd.shp  
geo_units_oc_dd.shp.xml  
geo_units_oc_dd.shx  
I1802ABC_geo_units_RGBlut.txt
```

Date

Displays the local date and time

```
geo-shell> date
```



No parameters

```
geo-shell> date
```

Tuesday, January 21, 2020 1:42:46 AM UTC

Script

Parses the specified resource file and executes its commands

```
geo-shell> script src/test/resources/layer_count.txt
```

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|-------------|--|-----------|-------------------|---------------------|
| | The file to locate and execute | true | | |
| lineNumbers | Display line numbers when executing the script | false | true | false |

```
workspace open --name naturalearth --params src/test/resources/naturalearth.gpkg  
layer open --workspace naturalearth --layer countries --name countries  
layer count --name countries  
workspace close --name naturalearth
```

```
geo-shell> script src/test/resources/layer_count.txt
```

```
Jan 21, 2020 1:42:45 AM org.geotools.data.DataAccessFinder getDataStore  
WARNING: MBTiles with vector tiles should be used, but could not connect  
java.io.IOException: org.sqlite.SQLiteException: [SQLITE_ERROR] SQL error or missing database (no
```

such table: metadata)

```
at org.geotools.mbtiles.MBTilesFile.loadMetaData(MBTilesFile.java:433)
at org.geotools.mbtiles.MBTilesFile.loadMetaData(MBTilesFile.java:412)
at org.geotools.mbtiles.MBTilesDataStore.<init>(MBTilesDataStore.java:75)
at
org.geotools.mbtiles.MBTilesDataStoreFactory.createDataStore(MBTilesDataStoreFactory.java:121)
at org.geotools.mbtiles.MBTilesDataStoreFactory.createDataStore(MBTilesDataStoreFactory.java:39)
at org.geotools.data.DataAccessFinder.getDataStore(DataAccessFinder.java:119)
at org.geotools.data.DataStoreFinder.getDataStore(DataStoreFinder.java:69)
at org.geotools.data.DataStoreFinder$getDataStore.call(Unknown Source)
at geoscript.workspace.WorkspaceFactory.create(WorkspaceFactory.groovy:32)
at geoscript.workspace.WorkspaceFactory$create$0.callCurrent(Unknown Source)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallCurrent(CallSiteArray.java:51)
at geoscript.workspace.WorkspaceFactory$create$0.callCurrent(Unknown Source)
at geoscript.workspace.WorkspaceFactory.create(WorkspaceFactory.groovy:22)
at geoscript.workspace.WorkspaceFactory$create.call(Unknown Source)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47)
at geoscript.workspace.WorkspaceFactory$create.call(Unknown Source)
at geoscript.workspace.Workspace.getWorkspace(Workspace.groovy:283)
at geoscript.workspace.Workspace$getWorkspace.call(Unknown Source)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:115)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:127)
at org.geoshell.vector.WorkspaceCommands.open(WorkspaceCommands.groovy:24)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.springframework.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:216)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.springframework.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:216)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:115)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:127)
at org.geoshell.docs.AbstractDocTest.run(AbstractDocTest.groovy:52)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:101)
at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:323)
at
org.codehaus.groovy.runtime.metaclass.ClosureMetaClass.invokeMethod(ClosureMetaClass.java:35
1)
at org.codehaus.groovy.runtime.callsite.PogoMetaClassSite.callCurrent(PogoMetaClassSite.java:64)
```

```
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallCurrent(CallSiteArray.java:51)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:156)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:184)
at org.geoshell.docs.AbstractDocTest$_run_closure1.doCall(AbstractDocTest.groovy:32)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:101)
at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:323)
at
org.codehaus.groovy.runtime.metaclass.ClosureMetaClass.invokeMethod(ClosureMetaClass.java:26
3)
at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1041)
at groovy.lang.Closure.call(Closure.java:405)
at
org.codehaus.groovy.runtime.DefaultGroovyMethods.eachWithIndex(DefaultGroovyMethods.java:2
246)
at
org.codehaus.groovy.runtime.DefaultGroovyMethods.eachWithIndex(DefaultGroovyMethods.java:2
226)
at
org.codehaus.groovy.runtime.DefaultGroovyMethods.eachWithIndex(DefaultGroovyMethods.java:2
276)
at org.codehaus.groovy.runtime.dgm$199.invoke(Unknown Source)
at
org.codehaus.groovy.runtime.callsite.PojoMetaMethodSite$PojoMetaMethodSiteNoUnwrapNoCoerc
e.invoke(PojoMetaMethodSite.java:244)
at org.codehaus.groovy.runtime.callsite.PojoMetaMethodSite.call(PojoMetaMethodSite.java:53)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:115)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:127)
at org.geoshell.docs.AbstractDocTest.run(AbstractDocTest.groovy:31)
at org.geoshell.docs.AbstractDocTest$run.callCurrent(Unknown Source)
at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallCurrent(CallSiteArray.java:51)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:156)
at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:184)
at org.geoshell.docs.BuiltinDocTest.script(BuiltinDocTest.groovy:66)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)
at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)
at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:26)
at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:27)
```

```
at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)
at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)
at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)
at org.junit.runners.ParentRunner.run(ParentRunner.java:363)
at org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:365)
at org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:273)
at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:238)
at org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:159)
at
org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:383)
at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:344)
at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:125)
at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:417)
Caused by: org.sqlite.SQLiteException: [SQLITE_ERROR] SQL error or missing database (no such table: metadata)
at org.sqlite.core.DB.newSQLException(DB.java:941)
at org.sqlite.core.DB.newSQLException(DB.java:953)
at org.sqlite.core.DB.throwex(DB.java:918)
at org.sqlite.core.NativeDB.prepare_utf8(Native Method)
at org.sqlite.core.NativeDB.prepare(NativeDB.java:134)
at org.sqlite.core.DB.prepare(DB.java:257)
at org.sqlite.core.CorePreparedStatement.<init>(CorePreparedStatement.java:47)
at org.sqlite.jdbc3.JDBC3PreparedStatement.<init>(JDBC3PreparedStatement.java:30)
at org.sqlite.jdbc4.JDBC4PreparedStatement.<init>(JDBC4PreparedStatement.java:19)
at org.sqlite.jdbc4.JDBC4Connection.prepareStatement(JDBC4Connection.java:35)
at org.sqlite.jdbc3.JDBC3Connection.prepareStatement(JDBC3Connection.java:241)
at org.sqlite.jdbc3.JDBC3Connection.prepareStatement(JDBC3Connection.java:205)
at org.geotools.jdbc.util.SqlUtil$PreparedStatementBuilder.<init>(SqlUtil.java:57)
at org.geotools.jdbc.util.SqlUtil.prepare(SqlUtil.java:240)
at org.geotools.mbtiles.MBTilesFile.loadMetaEntry(MBTilesFile.java:831)
at org.geotools.mbtiles.MBTilesFile.loadMetaData(MBTilesFile.java:419)
... 107 more
```

Workspace naturalearth opened!

Opened Workspace naturalearth Layer countries as countries

177

Workspace naturalearth closed!

System Properties

Shows the shell's properties

```
geo-shell> system properties
```



No parameters

```
geo-shell> system properties
```

```
awt.toolkit = sun.awt.X11.XToolkit
basedir = /home/travis/build/jericks/geo-shell
file.encoding = UTF-8
file.encoding.pkg = sun.io
file.separator = /
java.awt.graphicsenv = sun.awt.X11GraphicsEnvironment
java.awt.printerjob = sun.print.PSPrinterJob
java.class.path = /home/travis/build/jericks/geo-shell/target/test-classes:/home/travis/build/jericks/geo-shell/target/classes:/home/travis/.m2/repository/org/geoscript/geoscript-groovy/1.15-SNAPSHOT/geoscript-groovy-1.15-SNAPSHOT.jar:/home/travis/.m2/repository/com/wdtinc/mapbox-vector-tile/3.0.0/mapbox-vector-tile-3.0.0.jar:/home/travis/.m2/repository/org/slf4j/slf4j-api/1.7.25/slf4j-api-1.7.25.jar:/home/travis/.m2/repository/org/geotools/gt-main/23-SNAPSHOT/gt-main-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-referencing/23-SNAPSHOT/gt-referencing-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/ejml/ejml-ddense/0.34/ejml-ddense-0.34.jar:/home/travis/.m2/repository/org/ejml/ejml-core/0.34/ejml-core-0.34.jar:/home/travis/.m2/repository/commons-pool/commons-pool/1.5.4/commons-pool-1.5.4.jar:/home/travis/.m2/repository/jgridshift/jgridshift-core/1.2/jgridshift-core-1.2.jar:/home/travis/.m2/repository/javax/javaee-api/7.0/javaee-api-7.0.jar:/home/travis/.m2/repository/com/sun/mail/javax.mail/1.5.0/javax.mail-1.5.0.jar:/home/travis/.m2/repository/javax/activation/activation/1.1/activation-1.1.jar:/home/travis/.m2/repository/net/sf/geographiclib/GeographicLib-Java/1.49/GeographicLib-Java-1.49.jar:/home/travis/.m2/repository/org/locationtech/jts/jts-core/1.16.1/jts-core-1.16.1.jar:/home/travis/.m2/repository/org/apache/commons/commons-text/1.6/commons-text-1.6.jar:/home/travis/.m2/repository/com/fasterxml/jackson/core/jackson-core/2.10.1/jackson-core-2.10.1.jar:/home/travis/.m2/repository/javax/media/jai_core/1.1.3/jai_core-1.1.3.jar:/home/travis/.m2/repository/org/geotools/gt-epsg-hsql/23-SNAPSHOT/gt-epsg-hsql-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/hsqldb/hsqldb/2.4.1/hsqldb-2.4.1.jar:/home/travis/.m2/repository/org/geotools/gt-epsg-extension/23-SNAPSHOT/gt-epsg-extension-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-render/23-SNAPSHOT/gt-render-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-coverage/23-SNAPSHOT/gt-coverage-23-SNAPSHOT.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/affine/jt-affine/1.1.12/jt-affine-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/algebra/jt-algebra/1.1.12/jt-algebra-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/bandmerge/jt-bandmerge/1.1.12/jt-bandmerge-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/bandselect/jt-bandselect/1.1.12/jt-bandselect-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/bandcombine/jt-bandcombine/1.1.12/jt-bandcombine-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/border/jt-border/1.1.12/jt-border-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/buffer/jt-buffer/1.1.12/jt-buffer-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/crop/jt-crop/1.1.12/jt-crop-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/iterators/jt-iterators/1.1.12/jt-iterators-1.1.12.jar
```

1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/lookup/jt-lookup/1.1.12/jt-lookup-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/mosaic/jt-mosaic/1.1.12/jt-mosaic-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/nullop/jt-nullop/1.1.12/jt-nullop-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/rescale/jt-rescale/1.1.12/jt-rescale-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/scale/jt-scale/1.1.12/jt-scale-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/scale2/jt-scale2/1.1.12/jt-scale2-
1.1.12.jar:/home/travis/.m2/repository/org/huldra/math/bigint/0.7.1/bigint-
0.7.1.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/stats/jt-stats/1.1.12/jt-stats-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/translate/jt-translate/1.1.12/jt-translate-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/utilities/jt-utilities/1.1.12/jt-utilities-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/warp/jt-warp/1.1.12/jt-warp-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/zonal/jt-zonal/1.1.12/jt-zonal-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/binarize/jt-binarize/1.1.12/jt-binarize-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/format/jt-format/1.1.12/jt-format-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/colorconvert/jt-colorconvert/1.1.12/jt-
colorconvert-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/errordiffusion/jt-
errordiffusion/1.1.12/jt-errordiffusion-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/orderdither/jt-orderdither/1.1.12/jt-
orderdither-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/colorindexer/jt-
colorindexer/1.1.12/jt-colorindexer-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/imagefunction/jt-
imagefunction/1.1.12/jt-imagefunction-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/piecewise/jt-piecewise/1.1.12/jt-
piecewise-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/classifier/jt-classifier/1.1.12/jt-
classifier-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/rlookup/jt-rlookup/1.1.12/jt-
rlookup-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/vectorbin/jt-vectorbin/1.1.12/jt-
vectorbin-1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/shadedrelief/jt-
shadedrelief/1.1.12/jt-shadedrelief-1.1.12.jar:/home/travis/.m2/repository/org/geotools/gt-cql/23-
SNAPSHOT/gt-cql-23-
SNAPSHOT.jar:/home/travis/.m2/repository/com/conversantmedia/disruptor/1.2.13/disruptor-
1.2.13.jar:/home/travis/.m2/repository/org/geotools/gt-shapefile/23-SNAPSHOT/gt-shapefile-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/jdbc/gt-jdbc-postgis/23-SNAPSHOT/gt-jdbc-
postgis-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/postgresql/postgresql/42.2.5/postgresql-
42.2.5.jar:/home/travis/.m2/repository/org/geotools/gt-jdbc/23-SNAPSHOT/gt-jdbc-23-
SNAPSHOT.jar:/home/travis/.m2/repository/commons-dbcp/commons-dbcp/1.4/commons-dbcp-
1.4.jar:/home/travis/.m2/repository/org/geotools/jdbc/gt-jdbc-h2/23-SNAPSHOT/gt-jdbc-h2-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/opengeo/geodb/0.9/geodb-
0.9.jar:/home/travis/.m2/repository/com/h2database/h2/1.1.119/h2-
1.1.119.jar:/home/travis/.m2/repository/org/geotools/jdbc/gt-jdbc-mysql/23-SNAPSHOT/gt-jdbc-mysql-
23-SNAPSHOT.jar:/home/travis/.m2/repository/mysql/mysql-connector-java/5.1.46/mysql-connector-
java-5.1.46.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-wfs/23-SNAPSHOT/gt-xsd-wfs-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/ogc/net.opengis.wfs/23-
SNAPSHOT/net.opengis.wfs-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/ogc/org.w3.xlink/23-
SNAPSHOT/org.w3.xlink-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/ogc/net.opengis.fes/23-
SNAPSHOT/net.opengis.fes-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/eclipse/emf/org.eclipse.emf.common/2.15.0/org.eclip

se.emf.common-
2.15.0.jar:/home/travis/.m2/repository/org/eclipse/emf/org.eclipse.emf.ecore/2.15.0/org.eclipse.emf.ecore-
2.15.0.jar:/home/travis/.m2/repository/org/eclipse/emf/org.eclipse.emf.ecore.xmi/2.15.0/org.eclipse.emf.ecore.xmi-2.15.0.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-filter/23-SNAPSHOT/gt-xsd-filter-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-gml2/23-SNAPSHOT/gt-xsd-gml2-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-fes/23-SNAPSHOT/gt-xsd-fes-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-ows/23-SNAPSHOT/gt-xsd-ows-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/ogc/net.opengis.ows/23-SNAPSHOT/net.opengis.ows-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-wfs-ng/23-SNAPSHOT/gt-wfs-ng-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-complex/23-SNAPSHOT/gt-complex-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-app-schema-resolver/23-SNAPSHOT/gt-app-schema-resolver-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-gml3/23-SNAPSHOT/gt-xsd-gml3-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-xml/23-SNAPSHOT/gt-xml-23-SNAPSHOT.jar:/home/travis/.m2/repository/commons-collections/commons-collections/3.2.2/commons-collections-3.2.2.jar:/home/travis/.m2/repository/org/apache/xml/xml-commons-resolver/1.2/xml-commons-resolver-1.2.jar:/home/travis/.m2/repository/xpp3/xpp3_min/1.1.4c/xpp3_min-1.1.4c.jar:/home/travis/.m2/repository/commons-httpclient/commons-httpclient/3.1/commons-httpclient-3.1.jar:/home/travis/.m2/repository/commons-codec/commons-codec/1.2/commons-codec-1.2.jar:/home/travis/.m2/repository/org/geotools/gt-charts/23-SNAPSHOT/gt-charts-23-SNAPSHOT.jar:/home/travis/.m2/repository/jfree/eastwood/1.1.1-20090908/eastwood-1.1.1-20090908.jar:/home/travis/.m2/repository/jfree/jfreechart/1.0.10/jfreechart-1.0.10.jar:/home/travis/.m2/repository/jfree/jcommon/1.0.13/jcommon-1.0.13.jar:/home/travis/.m2/repository/org/geotools/gt-brewer/23-SNAPSHOT/gt-brewer-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-svg/23-SNAPSHOT/gt-svg-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-transcoder/1.10/batik-transcoder-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-anim/1.10/batik-anim-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-css/1.10/batik-css-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-ext/1.10/batik-ext-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-parser/1.10/batik-parser-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-svg-dom/1.10/batik-svg-dom-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik.awt-util/1.10/batik.awt-util-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/xmlgraphics-commons/2.2/xmlgraphics-commons-2.2.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-bridge/1.10/batik-bridge-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-script/1.10/batik-script-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-dom/1.10/batik-dom-1.10.jar:/home/travis/.m2/repository/xalan/xalan/2.7.2/xalan-2.7.2.jar:/home/travis/.m2/repository/xalan/serializer/2.7.2/serializer-2.7.2.jar:/home/travis/.m2/repository/xml-apis/xml-apis/1.3.04/xml-apis-1.3.04.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-gvt/1.10/batik-gvt-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-svggen/1.10/batik-svggen-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-util/1.10/batik-util-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-constants/1.10/batik-constants-1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-i18n/1.10/batik-i18n-

1.10.jar:/home/travis/.m2/repository/org/apache/xmlgraphics/batik-xml/1.10/batik-xml-
1.10.jar:/home/travis/.m2/repository/xml-apis/xml-apis-ext/1.3.04/xml-apis-ext-
1.3.04.jar:/home/travis/.m2/repository/org/geotools/gt-property/23-SNAPSHOT/gt-property-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-geojson/23-SNAPSHOT/gt-geojson-23-
SNAPSHOT.jar:/home/travis/.m2/repository/com/googlecode/json-simple/json-simple/1.1/json-simple-
1.1.jar:/home/travis/.m2/repository/org/apache/commons/commons-lang3/3.8.1/commons-lang3-
3.8.1.jar:/home/travis/.m2/repository/org/geotools/gt-swing/23-SNAPSHOT/gt-swing-23-
SNAPSHOT.jar:/home/travis/.m2/repository/com/miglayout/miglayout/3.7/miglayout-3.7-
swing.jar:/home/travis/.m2/repository/org/geotools/gt-process/23-SNAPSHOT/gt-process-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/jaitools/jt-utils/1.5.0/jt-utils-
1.5.0.jar:/home/travis/.m2/repository/javax/media/jai_imageio/1.1/jai_imageio-
1.1.jar:/home/travis/.m2/repository/org/geotools/gt-process-feature/23-SNAPSHOT/gt-process-
feature-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-process-geometry/23-
SNAPSHOT/gt-process-geometry-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-
xsd-kml/23-SNAPSHOT/gt-xsd-kml-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/xsd/gt-xsd-core/23-SNAPSHOT/gt-xsd-core-
23-SNAPSHOT.jar:/home/travis/.m2/repository/picocontainer/picocontainer/1.2/picocontainer-
1.2.jar:/home/travis/.m2/repository/commons-jxpath/commons-jxpath/1.3/commons-jxpath-
1.3.jar:/home/travis/.m2/repository/org/eclipse/xsd/org.eclipse.xsd/2.12.0/org.eclipse.xsd-
2.12.0.jar:/home/travis/.m2/repository/org/geotools/gt-css/23-SNAPSHOT/gt-css-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/parboiled/parboiled-java/1.2.0/parboiled-java-
1.2.0.jar:/home/travis/.m2/repository/org/parboiled/parboiled-core/1.2.0/parboiled-core-
1.2.0.jar:/home/travis/.m2/repository/org/ow2/asm/asm/6.2.1/asm-
6.2.1.jar:/home/travis/.m2/repository/org/ow2/asm/asm-tree/6.2.1/asm-tree-
6.2.1.jar:/home/travis/.m2/repository/org/ow2/asm/asm-analysis/6.2.1/asm-analysis-
6.2.1.jar:/home/travis/.m2/repository/org/ow2/asm/asm-util/6.2.1/asm-util-
6.2.1.jar:/home/travis/.m2/repository/org/geotools/gt-yld/23-SNAPSHOT/gt-yld-23-
SNAPSHOT.jar:/home/travis/.m2/repository/org/yaml/snakeyaml/1.23/snakeyaml-
1.23.jar:/home/travis/.m2/repository/com/google/code/findbugs/jsr305/3.0.0/jsr305-
3.0.0.jar:/home/travis/.m2/repository/org/geotools/gt-geotiff/23-SNAPSHOT/gt-geotiff-23-
SNAPSHOT.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-
tiff/1.3.2/imageio-ext-tiff-1.3.2.jar:/home/travis/.m2/repository/org/geotools/gt-image/23-
SNAPSHOT/gt-image-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-imagemosaic/23-
SNAPSHOT/gt-imagemosaic-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-
metadata/23-SNAPSHOT/gt-metadata-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-
opengis/23-SNAPSHOT/gt-opengis-23-
SNAPSHOT.jar:/home/travis/.m2/repository/systems/uom/systems-common-java8/0.7.2/systems-
common-java8-0.7.2.jar:/home/travis/.m2/repository/tec/uom/uom-se/1.0.8/uom-se-
1.0.8.jar:/home/travis/.m2/repository/javax/measure/unit-api/1.0/unit-api-
1.0.jar:/home/travis/.m2/repository/tec/uom/lib/uom-lib-common/1.0.2/uom-lib-common-
1.0.2.jar:/home/travis/.m2/repository/si/uom/si-quantity/0.7.1/si-quantity-
0.7.1.jar:/home/travis/.m2/repository/si/uom/si-units-java8/0.7.1/si-units-java8-
0.7.1.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-streams/1.3.2/imageio-
ext-streams-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-
geocore/1.3.2/imageio-ext-geocore-1.3.2.jar:/home/travis/.m2/repository/commons-
beanutils/commons-beanutils/1.9.2/commons-beanutils-1.9.2-
noclassprop.jar:/home/travis/.m2/repository/org/jaitools/jt-vectorbinarize/1.5.0/jt-vectorbinarize-
1.5.0.jar:/home/travis/.m2/repository/net/sf/ehcache/ehcache/2.10.3/ehcache-

2.10.3.jar:/home/travis/.m2/repository/javax/xml/bind/jaxb-api/2.4.0-b180830.0359/jaxb-api-2.4.0-b180830.0359.jar:/home/travis/.m2/repository/org/glassfish/jaxb/jaxb-runtime/2.4.0-b180830.0438/jaxb-runtime-2.4.0-b180830.0438.jar:/home/travis/.m2/repository/org/glassfish/jaxb/txw2/2.4.0-b180830.0438/txw2-2.4.0-b180830.0438.jar:/home/travis/.m2/repository/com/sun/istack/istack-commons-runtime/3.0.7/istack-commons-runtime-3.0.7.jar:/home/travis/.m2/repository/org/jvnet/staxex/stax-ex/1.8/stax-ex-1.8.jar:/home/travis/.m2/repository/com/sun/xml/fastinfoset/FastInfoset/1.2.15/FastInfoset-1.2.15.jar:/home/travis/.m2/repository/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar:/home/travis/.m2/repository/javax/media/jai_codec/1.1.3/jai_codec-1.1.3.jar:/home/travis/.m2/repository/org/geotools/gt-arcgrid/23-SNAPSHOT/gt-arcgrid-23-SNAPSHOT.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-arcgrid/1.3.2/imageio-ext-arcgrid-1.3.2.jar:/home/travis/.m2/repository/org/geotools/gt-grassraster/23-SNAPSHOT/gt-grassraster-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-gtopo30/23-SNAPSHOT/gt-gtopo30-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-imagepyramid/23-SNAPSHOT/gt-imagepyramid-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-imageio-ext-gdal/23-SNAPSHOT/gt-imageio-ext-gdal-23-SNAPSHOT.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalarcbinarygrid/1.3.2/imageio-ext-gdalarcbinarygrid-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalframework/1.3.2/imageio-ext-gdalframework-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalmrsid/1.3.2/imageio-ext-gdalmrsid-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalecw/1.3.2/imageio-ext-gdalecw-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdaldted/1.3.2/imageio-ext-gdaldted-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalkakadujp2/1.3.2/imageio-ext-gdalkakadujp2-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalmrsidjp2/1.3.2/imageio-ext-gdalmrsidjp2-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalecwp2/1.3.2/imageio-ext-gdalecwp2-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalehdr/1.3.2/imageio-ext-gdalehdr-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalenvihdr/1.3.2/imageio-ext-gdalenvihdr-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/gdalerdasimg/1.3.2/imageio-ext-gdalerdasimg-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalnif/1.3.2/imageio-ext-gdalnif-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalrpftoc/1.3.2/imageio-ext-gdalrpftoc-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalidrisi/1.3.2/imageio-ext-gdalidrisi-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-gdalvrt/1.3.2/imageio-ext-gdalvrt-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/gdalsrp/1.3.2/imageio-ext-gdalsrp-1.3.2.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-imagereadmt/1.3.2/imageio-ext-imagereadmt-1.3.2.jar:/home/travis/.m2/repository/org/geotools/gt-netcdf/23-SNAPSHOT/gt-netcdf-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-coverage-api/23-SNAPSHOT/gt-coverage-api-23-SNAPSHOT.jar:/home/travis/.m2/repository/com/github/spotbugs/spotbugs-annotations/4.0.0-0

RC1/spotbugs-annotations-4.0.0-RC1.jar:/home/travis/.m2/repository/it/geosolutions/imageio-ext/imageio-ext-utilities/1.3.2/imageio-ext-utilities-
1.3.2.jar:/home/travis/.m2/repository/edu/ucar/cdm/4.6.11/cdm-
4.6.11.jar:/home/travis/.m2/repository/edu/ucar/udunits/4.6.11/udunits-
4.6.11.jar:/home/travis/.m2/repository/edu/ucar/httpservices/4.6.11/httpservices-
4.6.11.jar:/home/travis/.m2/repository/org/apache/httpcomponents/httpclient/4.5.1/httpclient-
4.5.1.jar:/home/travis/.m2/repository/org/apache/httpcomponents/httpmime/4.5.1/httpmime-
4.5.1.jar:/home/travis/.m2/repository/org/apache/httpcomponents/httpcore/4.4.4/httpcore-
4.4.4.jar:/home/travis/.m2/repository/joda-time/joda-time/2.8.1/joda-time-
2.8.1.jar:/home/travis/.m2/repository/net/jcip/jcip-annotations/1.0/jcip-annotations-
1.0.jar:/home/travis/.m2/repository/org/quartz-scheduler/quartz/2.2.0/quartz-
2.2.0.jar:/home/travis/.m2/repository/c3p0/c3p0/0.9.1.1/c3p0-
0.9.1.1.jar:/home/travis/.m2/repository/com/beust/jcommander/1.35/jcommander-
1.35.jar:/home/travis/.m2/repository/com/amazonaws/aws-java-sdk-s3/1.11.236/aws-java-sdk-s3-
1.11.236.jar:/home/travis/.m2/repository/com/amazonaws/aws-java-sdk-kms/1.11.236/aws-java-sdk-
kms-1.11.236.jar:/home/travis/.m2/repository/com/amazonaws/aws-java-sdk-core/1.11.236/aws-java-
sdk-core-1.11.236.jar:/home/travis/.m2/repository/software/amazon/ion/ion-java/1.0.2/ion-java-
1.0.2.jar:/home/travis/.m2/repository/com/fasterxml/jackson/dataformat/jackson-dataformat-
cbor/2.6.7/jackson-dataformat-cbor-2.6.7.jar:/home/travis/.m2/repository/com/amazonaws/jmespath-
java/1.11.236/jmespath-java-
1.11.236.jar:/home/travis/.m2/repository/edu/ucar/netcdf4/4.6.11/netcdf4-
4.6.11.jar:/home/travis/.m2/repository/net/java/dev/jna/jna/4.2.2/jna-
4.2.2.jar:/home/travis/.m2/repository/edu/ucar/bufr/4.6.11/bufr-
4.6.11.jar:/home/travis/.m2/repository/opendap/opendap/2.1/opendap-
2.1.jar:/home/travis/.m2/repository/commons-cli/commons-cli/1.4/commons-cli-
1.4.jar:/home/travis/.m2/repository/org/slf4j/slf4j-log4j12/1.6.4/slf4j-log4j12-
1.6.4.jar:/home/travis/.m2/repository/log4j/log4j/1.2.16/log4j-
1.2.16.jar:/home/travis/.m2/repository/org/jdom/jdom2/2.0.6/jdom2-
2.0.6.jar:/home/travis/.m2/repository/jaxen/jaxen/1.1.6/jaxen-
1.1.6.jar:/home/travis/.m2/repository/org/geotools/gt-process-raster/23-SNAPSHOT/gt-process-raster-
23-SNAPSHOT.jar:/home/travis/.m2/repository/org/jaitools/jt-zonalstats/1.5.0/jt-zonalstats-
1.5.0.jar:/home/travis/.m2/repository/org/jaitools/jt-rangefield/1.5.0/jt-rangefield-
1.5.0.jar:/home/travis/.m2/repository/org/jaitools/jt-contour/1.5.0/jt-contour-
1.5.0.jar:/home/travis/.m2/repository/org/jaitools/jt-attributeop/1.5.0/jt-attributeop-
1.5.0.jar:/home/travis/.m2/repository/org/jaitools/jt-vectorize/1.5.0/jt-vectorize-
1.5.0.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/jiffle/jt-jiffle-op/1.1.12/jt-jiffle-op-
1.1.12.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/jiffle/jt-jiffle-language/1.1.12/jt-jiffle-
language-1.1.12.jar:/home/travis/.m2/repository/org/antlr/antlr4-runtime/4.7.1/antlr4-runtime-
4.7.1.jar:/home/travis/.m2/repository/org/codehaus/janino/janino/3.0.8/janino-
3.0.8.jar:/home/travis/.m2/repository/org/codehaus/janino/commons-compiler/3.0.8/commons-
compiler-3.0.8.jar:/home/travis/.m2/repository/it/geosolutions/jaiext/classbreaks/jt-
classbreaks/1.1.12/jt-classbreaks-1.1.12.jar:/home/travis/.m2/repository/org/geotools/gt-wms/23-
SNAPSHOT/gt-wms-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-transform/23-
SNAPSHOT/gt-transform-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-mbtiles/23-
SNAPSHOT/gt-mbtiles-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/xerial/sqlite-
jdbc/3.30.1/sqlite-jdbc-3.30.1.jar:/home/travis/.m2/repository/com/fasterxml/jackson/core/jackson-
databind/2.10.1/jackson-databind-
2.10.1.jar:/home/travis/.m2/repository/com/fasterxml/jackson/core/jackson-

```
annotations/2.10.1/jackson-annotations-2.10.1.jar:/home/travis/.m2/repository/no/ecc/vectortile/java-vector-tile/1.3.7/java-vector-tile-1.3.7.jar:/home/travis/.m2/repository/org/geotools/gt-geopkg/23-SNAPSHOT/gt-geopkg-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-grid/23-SNAPSHOT/gt-grid-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-geobuf/23-SNAPSHOT/gt-geobuf-23-SNAPSHOT.jar:/home/travis/.m2/repository/com/google/protobuf/protobuf-java/3.4.0/protobuf-java-3.4.0.jar:/home/travis/.m2/repository/org/geotools/gt-flatgeobuf/23-SNAPSHOT/gt-flatgeobuf-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/wololo/flatgeobuf/2.0.2/flatgeobuf-2.0.2.jar:/home/travis/.m2/repository/com/google.flatbuffers/flatbuffers-java/1.11.0/flatbuffers-java-1.11.0.jar:/home/travis/.m2/repository/org/geotools/gt-ogr-jni/23-SNAPSHOT/gt-ogr-jni-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/geotools/gt-ogr-core/23-SNAPSHOT/gt-ogr-core-23-SNAPSHOT.jar:/home/travis/.m2/repository/org/gdal/gdal/2.2.0/gdal-2.2.0.jar:/home/travis/.m2/repository/org/codehaus/groovy/groovy/2.5.9/groovy-2.5.9.jar:/home/travis/.m2/repository/org/codehaus/groovy/groovy-sql/2.5.9/groovy-sql-2.5.9.jar:/home/travis/.m2/repository/org/codehaus/groovy/groovy-xml/2.5.9/groovy-xml-2.5.9.jar:/home/travis/.m2/repository/org/codehaus/groovy/groovy-json/2.5.9/groovy-json-2.5.9.jar:/home/travis/.m2/repository/org/codehaus/groovy/groovy-swing/2.5.9/groovy-swing-2.5.9.jar:/home/travis/.m2/repository/com/opencsv/opencsv/3.7/opencsv-3.7.jar:/home/travis/.m2/repository/com/lowagie/itext/2.1.7/itext-2.1.7.jar:/home/travis/.m2/repository/org/springframework/shell/spring-shell/1.2.0.RELEASE/spring-shell-1.2.0.RELEASE.jar:/home/travis/.m2/repository/com/google/guava/guava/17.0/guava-17.0.jar:/home/travis/.m2/repository/jline/jline/2.12/jline-2.12.jar:/home/travis/.m2/repository/org/springframework/spring-context-support/4.2.4.RELEASE/spring-context-support-4.2.4.RELEASE.jar:/home/travis/.m2/repository/org/springframework/spring-beans/4.2.4.RELEASE/spring-beans-4.2.4.RELEASE.jar:/home/travis/.m2/repository/org/springframework/spring-context/4.2.4.RELEASE/spring-context-4.2.4.RELEASE.jar:/home/travis/.m2/repository/org/springframework/spring-aop/4.2.4.RELEASE/spring-aop-4.2.4.RELEASE.jar:/home/travis/.m2/repository/aopalliance/aopalliance/1.0/aopalliance-1.0.jar:/home/travis/.m2/repository/org/springframework/spring-expression/4.2.4.RELEASE/spring-expression-4.2.4.RELEASE.jar:/home/travis/.m2/repository/commons-io/commons-io/2.4/commons-io-2.4.jar:/home/travis/.m2/repository/org/springframework/spring-core/4.2.4.RELEASE/spring-core-4.2.4.RELEASE.jar:/home/travis/.m2/repository/commons-logging/commons-logging/1.2/commons-logging-1.2.jar:/home/travis/.m2/repository/junit/junit/4.12/junit-4.12.jar:/home/travis/.m2/repository/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar:  
java.class.version = 52.0  
java.endorsed.dirs = /usr/lib/jvm/java-8-oracle/jre/lib/endorsed
```

Version

Displays shell version



No parameters

geo-shell> **version**

Download

Download a URL to a file.

```
geo-shell> download --url https://astropedia.astrogeology.usgs.gov/download/Mars/Geology/Mars15MGeologicGISRenovation.zip --file mars.zip --overwrite false
```

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|-----------|--------------------------------------|-----------|-------------------|---------------------|
| url | The url | true | | |
| file | The file | true | | |
| overwrite | Whether to overwrite the file or not | false | true | true |

```
geo-shell> download --url https://astropedia.astrogeology.usgs.gov/download/Mars/Geology/Mars15MGeologicGISRenovation.zip --file mars.zip --overwrite false
Downloading https://astropedia.astrogeology.usgs.gov/download/Mars/Geology/Mars15MGeologicGISRenovation.zip to /Users/jericks/Projects/geo-shell/mars.zip...
```

```
geo-shell> unzip --file mars.zip --directory mars
Unzipping /Users/jericks/Projects/geo-shell/mars.zip to /Users/jericks/Projects/geo-shell/mars
```

```
geo-shell> style vector uniquevaluesfromtext --field UnitSymbol --geometryType Polygon
--styleFile mars/units.sld --textFile mars/I1802ABC_Mars_global_geology/I1802ABC_geo_units_RGBlut.txt
Create a unique values style from /Users/jericks/Projects/geo-shell/mars/I1802ABC_Mars_global_geology/I1802ABC_geo_units_RGBlut.txt for UnitSymbol and Polygon to /Users/jericks/Projects/geo-shell/mars/units.sld
```

```
geo-shell> workspace open --name mars --params
mars/I1802ABC_Mars_global_geology/Shapefiles/I1802ABC_Mars2000_Sphere/geo_units_oc_dd.shp
Workspace mars opened!
```

```
geo-shell> layer open --workspace mars --layer geo_units_oc_dd
Opened Workspace mars Layer geo_units_oc_dd as mars:geo_units_oc_dd
```

```
geo-shell> layer style set --name mars:geo_units_oc_dd --style mars/units.sld
Style /Users/jericks/Projects/geo-shell/mars/units.sld set on mars:geo_units_oc_dd
```

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

```
geo-shell> map add layer --name mars --layer mars:geo_units_oc_dd
Added mars:geo_units_oc_dd layer to map mars
```

```
geo-shell> map draw --name mars
```

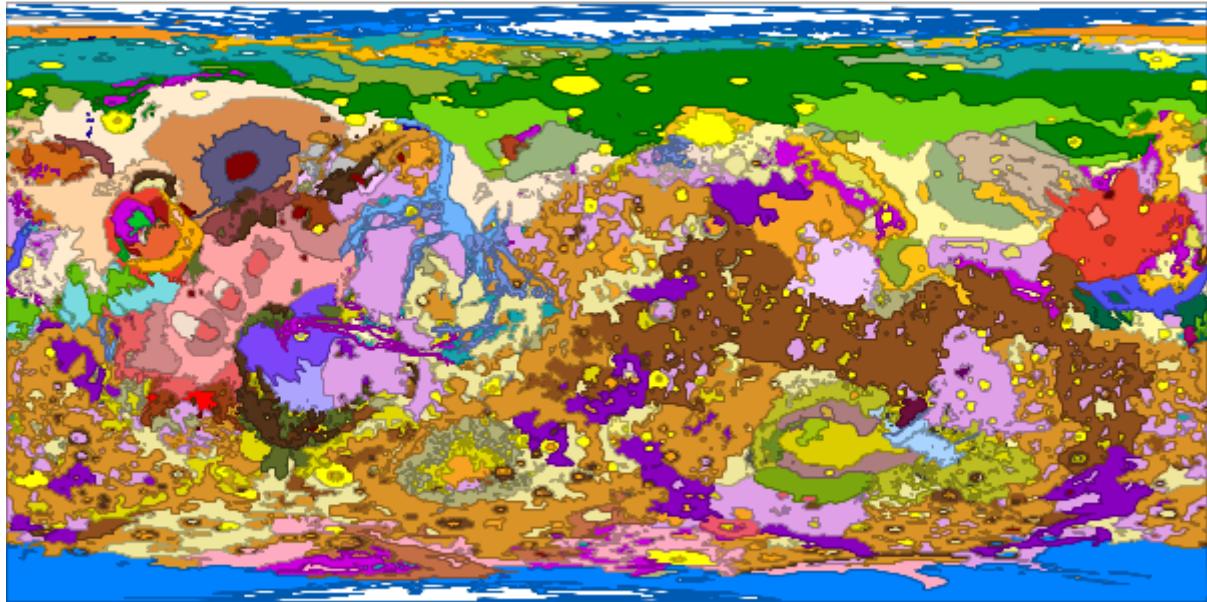
Done drawing /Users/jericks/Projects/geo-shell/image.png!

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

Map mars closed!

geo-shell> **open** --file image.png

Opening /Users/jericks/Projects/geo-shell/image.png...



Unzip

Unzip a file

geo-shell> **unzip** --file mars.zip --directory mars

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

geo-shell> **unzip** --file mars.zip --directory mars

Unzipping /Users/jericks/Projects/geo-shell/mars.zip to /Users/jericks/Projects/geo-shell/mars

Open

Open a File.

geo-shell> **open** --file image.png

| Name | Description | Mandatory | Specified Default | Unspecified Default |
|------|-------------|-----------|-------------------|---------------------|
| file | The File | true | | |

```
geo-shell> open --file image.png
Opening /Users/jericks/Projects/geo-shell/image.png...
```