Table of Contents

Layer

Basics

Open

Open a Layer.

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

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

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

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

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

Close

Close a Layer.

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

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

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

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

geo-shell> **layer close** --name countries
Layer countries closed!

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

List

List open Layers.

geo-shell> **layer list**

    🛈   No parameters

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

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

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

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

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

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

Schema

Inspect a Layer's Schema.

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

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

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

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

geo-shell> **layer schema** --name countries
Name Type
------------------ ------------------
the_geom MultiPolygon
ScaleRank Integer
FeatureCla String

SOVEREIGNT String
SOVISO String
SOV_A3 String
LEVEL Double
TYPE String
NAME String
SORTNAME String
ADM0_A3 String
NAME_SM String
NAME_LNG String
TERR_ String
PARENTHETI String
NAME_ALT String
LOCAL_LNG String
LOCAL_SM String
FORMER String
ABBREV_ String
MAP_COLOR Double
PEOPLE Double
GDP_USDM Double
FIPS_10 String
ISO_A2 String
ISO_A3 String
ISO_N3 Double
ITU String
IOC String
FIFA String
DS String
WMO String
GAUL Double
MARC String
STANAG1059 String
GW_ID Double
DIAL Double
INTERNET_ String
COG String
ACTUAL String
CAPAY String
CRPAY String
ANI String
LIBENR String
ANCNOM String
PAYS_R_GIO String
COMMENT String

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

Count

Count the Feature in a Layer.

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

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

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

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

geo-shell> **layer count** --name countries
177

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

Projection

Get the Projection of a Layer.

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

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

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

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

geo-shell> **layer projection** --name countries
EPSG:4326

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

Features

Display the Features of a Layer.

Name	Description	Mandatory	Specified Default	Unspecified Default

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

Get Style

Get the Layer's style.

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

Set Style

Set a Layer's style

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

Copy

Copy one Layer to another Workspace.

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
filter	The CQL Filter	false		
sort	A Sort parameter (fld dir)	false		
start	The start index	false		-1

Name	Description	Mandatory	Specified Default	Unspecified Default
max	The maximum number of records	false		-1
field	A subfield to include	false		

Add

Add a new Feature to a Layer.

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

Remove

Remove a Layer from a Workspace.

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

Create

Create a new Layer.

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

Delete

Delete features from the Layer

Name	Description	Mandatory	Specified Default	Unspecified Default

name	The Layer name	true		
filter	The CQL Filter	true		

Update

Calculate the update between a Layer with another Layer

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

Update Field

Delete features from the Layer

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

Add Fields

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

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
fields	The Fields (name=type proj)	true		

Add Area Field

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

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

Add ID Field

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

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
id-fieldname	The id field name	true	id	id
start-value	The value to start at	true	1	1

Add XY Fields

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

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
x-fieldname	The x field name	true	x	x
y-fieldname	The y field name	true	y	y

Validity

Check for invalid geometries in the Layer.

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

Geoprocessing

Clip

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

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

Convex Hull

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Convex Hulls

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

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

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

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

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

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

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

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

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

geo-shell> **layer convexhulls** --input-name countries --output-workspace layers --output-name convexhulls
Done!

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

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

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

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

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

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

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

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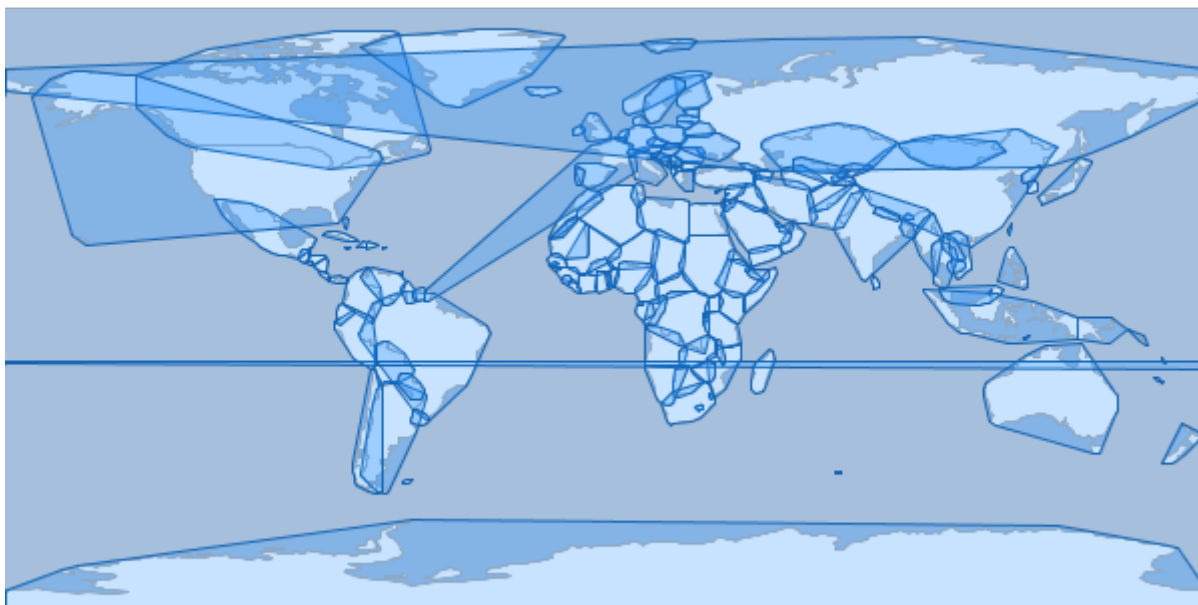
Coordinates

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

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

Delaunay

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

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

Densify

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

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

Dissolve

Dissolve the Features of a Layer by a Field.

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
field	The field to use to dissolve features	true		
idField	The name of the id field	false	id	id
countField	The name of the count field	false	count	count

Erase

Erase one Layer from another Layer

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

Grid Row / Column

Create a grid Layer with rows and columns

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

Grid Width / Height

Create a grid Layer with cell width and height

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

Identity

Calculate the intersection between a Layer with another Layer

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

Intersection

Calculate the intersection between a Layer with another Layer

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

Minimum Circle

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Added countries layer to map map

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

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

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

Minimum Circles

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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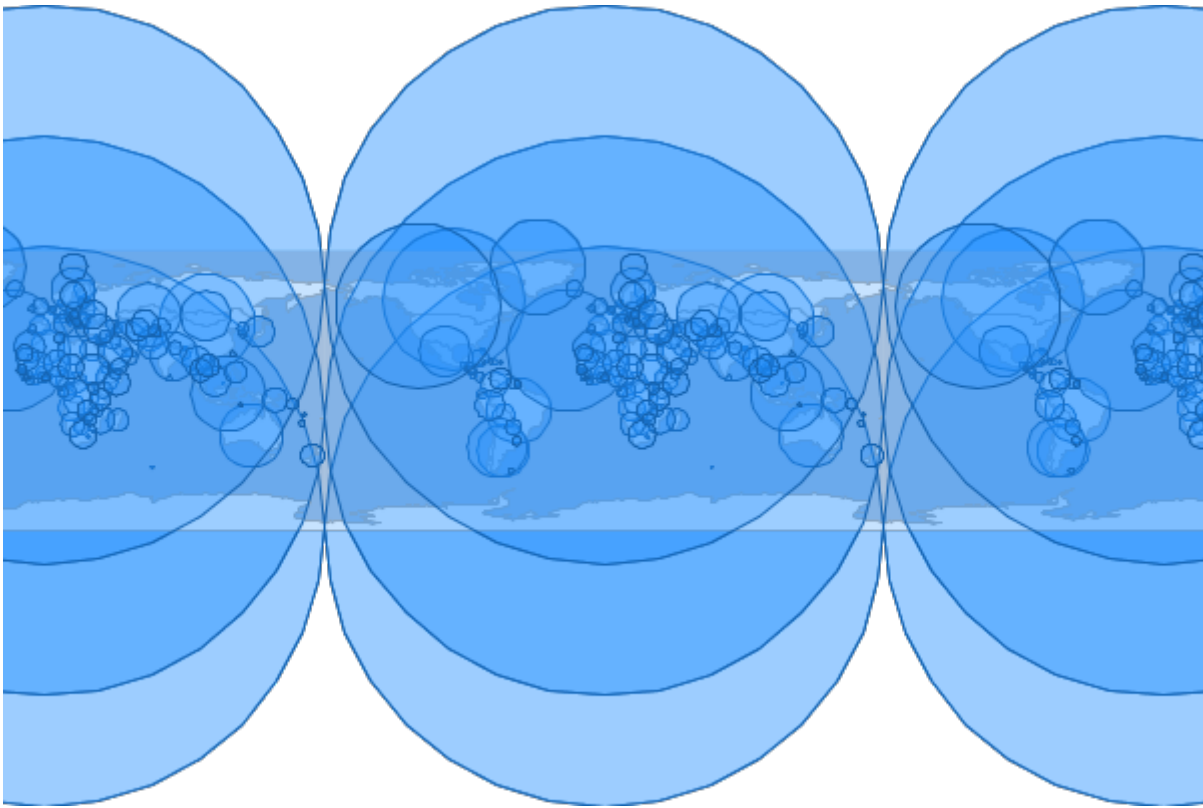
Minimum Rectangle

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Minimum Rectangles

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

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

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

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

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

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

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

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

Opened Workspace naturalearth Layer ocean as ocean

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

geo-shell> **layer minrects** --input-name countries --output-workspace layers --output-name minrects
Done!

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

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

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

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

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

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

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

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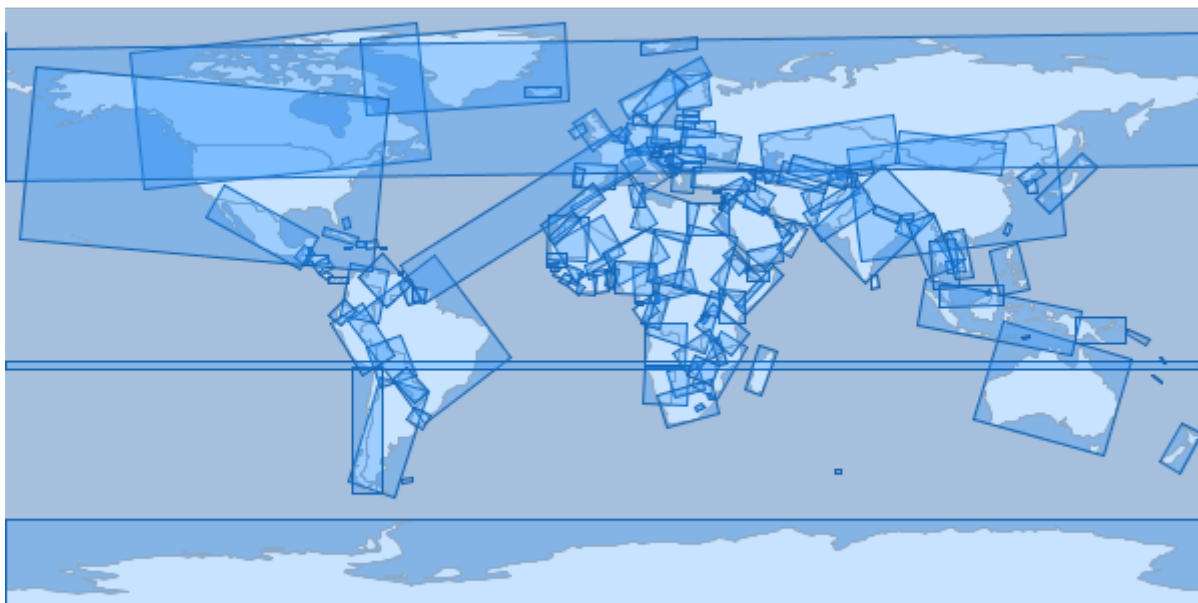
Octangle Envelope

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Octangle Envelopes

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

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

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

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

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

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

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

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

Opened Workspace naturalearth Layer ocean as ocean

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

geo-shell> **layer octagonalenvelopes** --input-name countries --output-workspace layers --output
-name octagonalenvelopes
Done!

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

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

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

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

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

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

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

geo-shell> **map close** --name map
Map map closed!
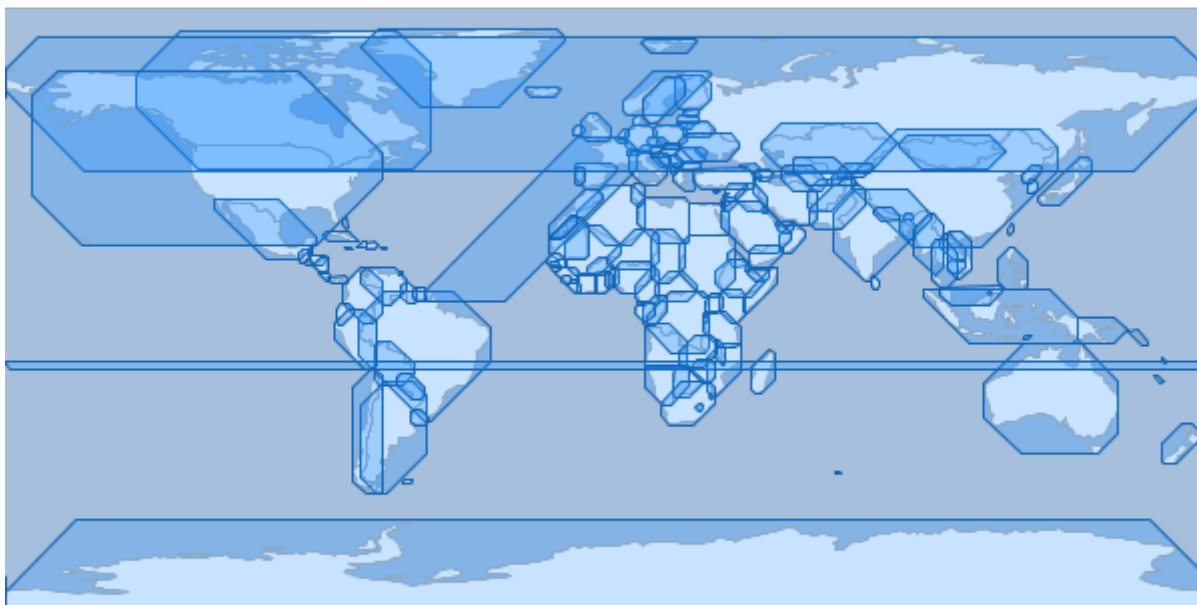
Points Along Lines

Create points along lines

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

Simplify

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

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

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

Symmetric Difference

Calculate the symmetric difference between a Layer and another Layer.

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

Transform

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

Name	Description	Mandatory	Specified Default	Unspecified Default
input-name	The Layer name	true		
output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
transforms	The pipe delimited list of transforms (field=expression or function)	true		

Union

Union a Layer with another Layer

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

Voronoi

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

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

Random Points

Create a Layer with a number of randomly located points

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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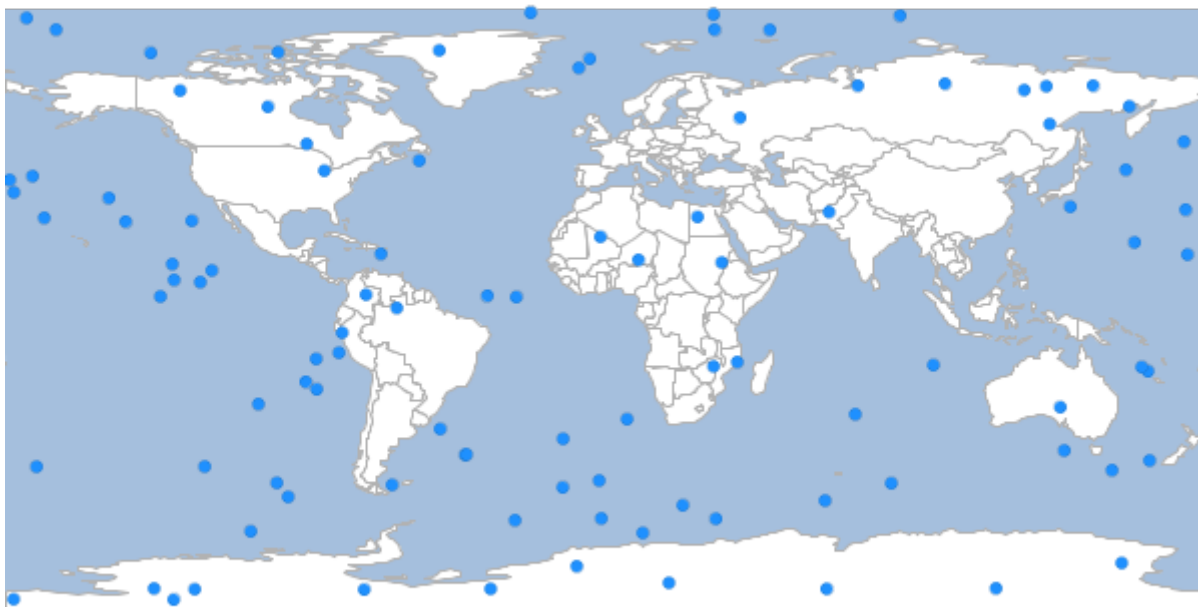
geo-shell> **map add layer** --name randomMap --layer points
Added points layer to map randomMap

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

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

Buffer

Buffer the input Layer to the output Layer.

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

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

output-workspace	The output Layer Workspace	true		
output-name	The output Layer name	true		
distance	The buffer distance	true		

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

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

geo-shell> **layer buffer** --input-name points --output-workspace layers --output-name buffers --distance 10
Done!

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

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

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

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

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

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

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

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

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

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

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

Added ocean layer to map map

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

geo-shell> **map add layer** --name map --layer buffers
Added buffers layer to map map
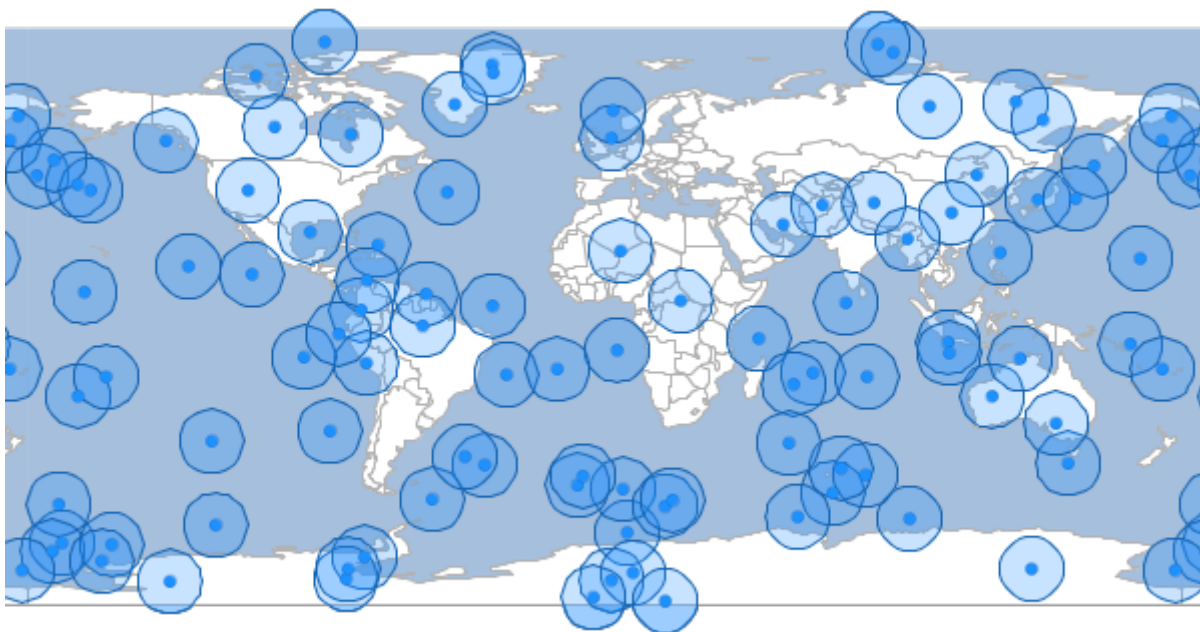
geo-shell> **map add layer** --name map --layer points
Added points layer to map map

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

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

Centroid

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

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

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

output-name	The output Layer name	true			

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Interior Point

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

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

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

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

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

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

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

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

Done!

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

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

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

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

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

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

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

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

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

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

Extent

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Extents

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

geo-shell> **layer extents** --input-name states --output-workspace layers --output-name state_extents

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

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

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

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

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

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

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

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

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

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

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

Opened Workspace naturalearth Layer countries as countries

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

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

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

Opened Workspace naturalearth Layer ocean as ocean

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

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

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

Map map opened!

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

Added ocean layer to map map

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

Added countries layer to map map

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

Added states layer to map map

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

Added state_extents layer to map map

geo-shell> **map draw** --name map --file examples/layer_extents.png

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

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

Map map closed!
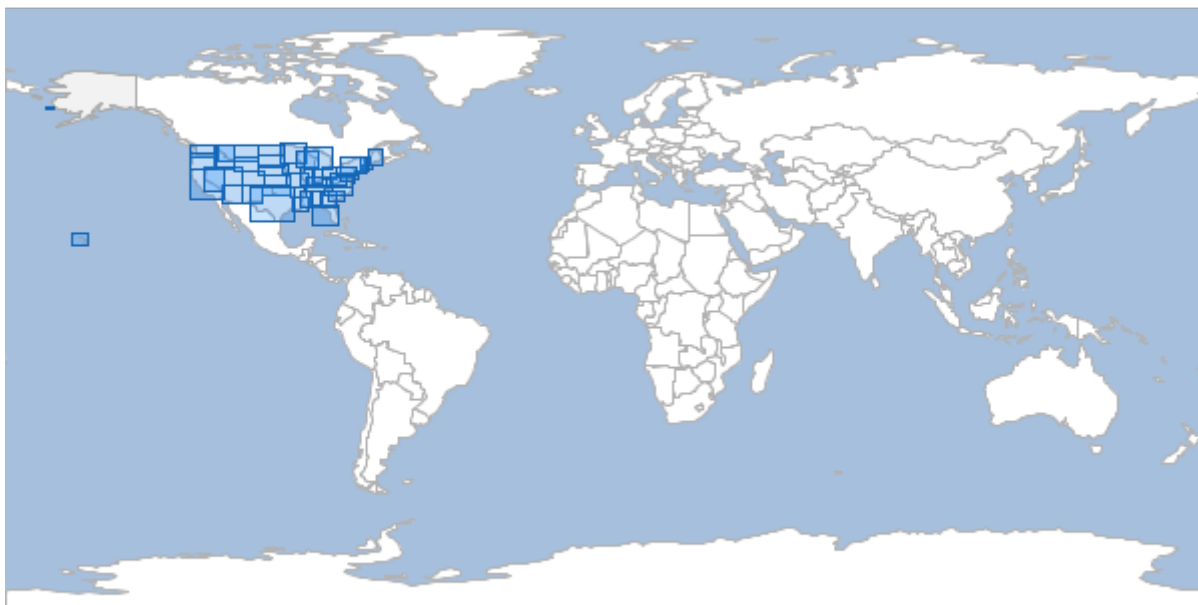
Graticule

Square

Create a square graticule.

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

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

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

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

geo-shell> **style vector default** --layer squares --color #1E90FF --opacity 0.30 --file

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

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

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

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

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

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

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

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

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

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

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

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

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

Rectangle

Create a rectangle graticule.

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

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

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

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

geo-shell> **style vector default** --layer rectangles --color #1E90FF --opacity 0.30 --file examples/rectangles.sld

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

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

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

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

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

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

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

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

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

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

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

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

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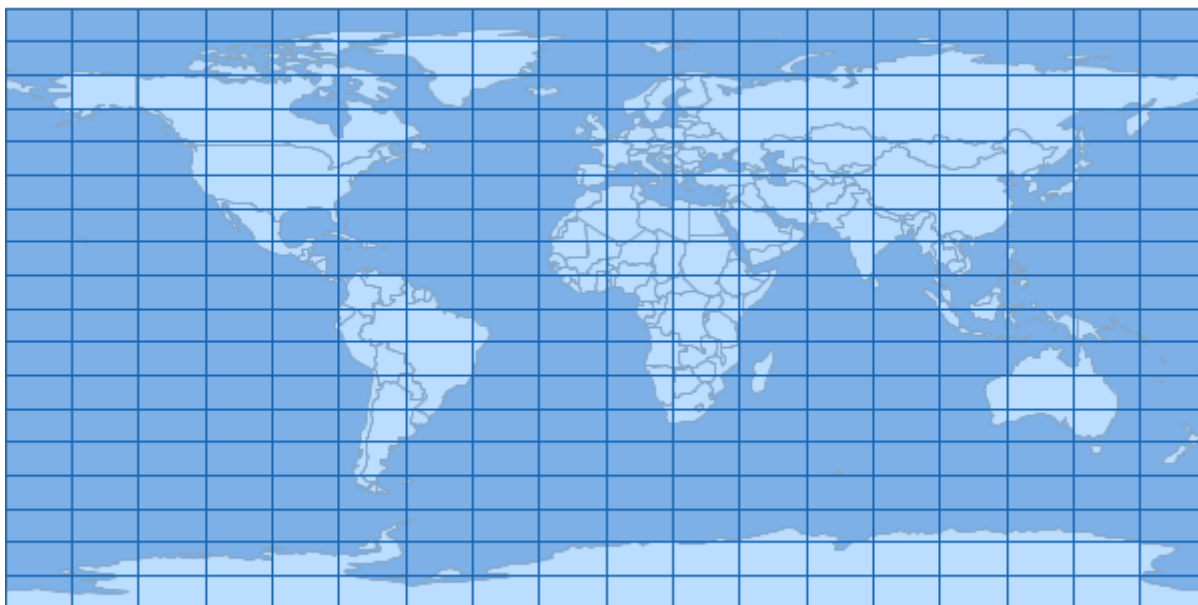
Oval

Create a oval graticule.

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

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

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

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

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

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

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

Workspace naturalearth opened!

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

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

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

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

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

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

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

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

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

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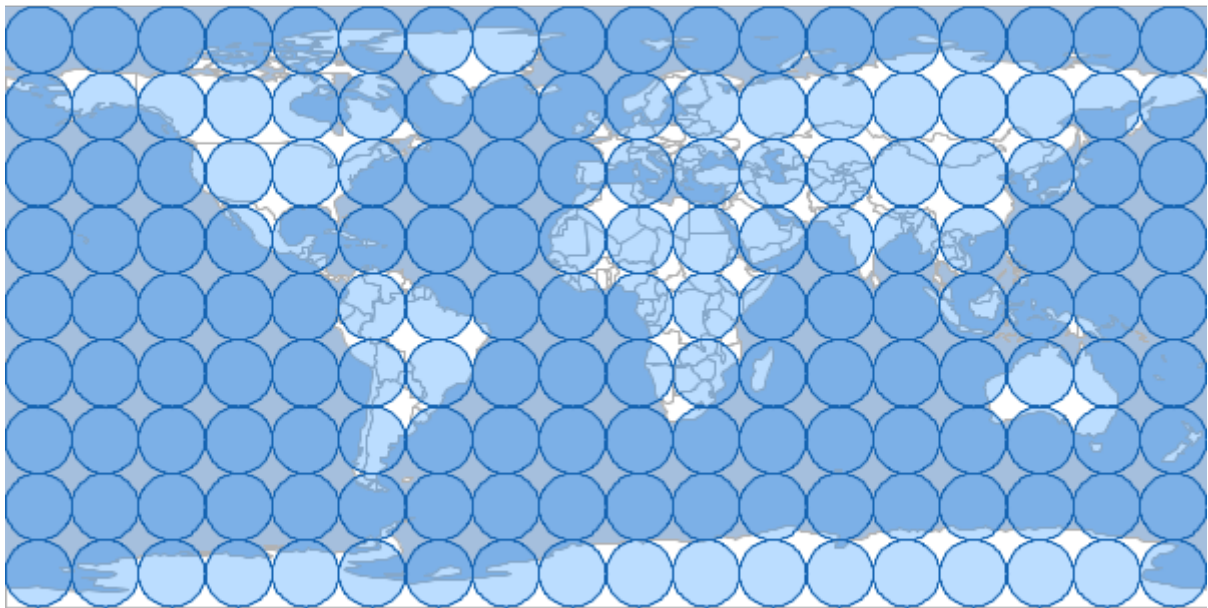
Hexagon

Create a hexagon graticule.

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

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

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

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

geo-shell> **style vector default** --layer hexagons --color #1E90FF --opacity 0.30 --file

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

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

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

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

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

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

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

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

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

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

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

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

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