Even more OGR

Even more OGR

Overview

1. Opening Data
2. Getting Features
3. Attribute Filtering
4. Spatial Filtering
5. Filtering: An Example

Opening Data

Opening Data

```
from osgeo import ogr

indata = r"C:\data\points.shp"

# new: do not need to specify driver with Open function
data = ogr.Open(indata)

# still need to get layer per OGR data abstraction
layer = ogr.GetLayer()
```

Getting Features

Getting Features

```
# get features by index
feature = layer.GetFeature(index)

# iterate through features
for index in xrange(layer.GetFeatureCount()):
    feature = layer.GetFeature(index)

# simplified iteration
for feature in layer:
    # do something with feature
```

Getting Features

```python
# get features sequentially
feature = layer.GetNextFeature()

# need to start over?
layer.ResetReading()

# want to start with a specific feature index?
# note: not an efficient function
layer.SetNextByIndex()

# want to use this strategy to loop through features with a while?
while True:
    feature = layer.GetNextFeature()

    # when we get to the end GetNextFeature will return None
    if not feature:
        break

    # do something with feature

# another popular way to do this:
feature = layer.GetNextFeature()
while feature:
    # do something with feature

    feature = layer.GetNextFeature
```

Attribute Filtering

Attribute Filtering

- Simple attribute queries can be done using the `SetAttributeFilter()` method
- Query syntax is very similar to ArcGIS (as it is based on SQL)

```
# how many features total?
layer.GetFeatureCount()
# 1045

# how many are type SN2?
layer.SetAttributeFilter("typeCode = 'SN2'")
layer.GetFeatureCount()
# 236

# how many are type SN2 and bigger than 10?
layer.SetAttributeFilter("typeCode = 'SN2' AND size > 10")
layer.GetFeatureCount()
# 35

# reset the filter
layer.SetAttributeFilter(None)
layer.GetFeatureCount()
# 1045
```

Attribute Filtering

- Advanced queries can use the `ExecuteSQL()` method of a data source
    - Returns a "virtual" layer called a results set
    - See the docs for in-depth info
- Use `ReleaseResultSet(result_layer)` when finished with a results set
- `SetAttributeFilter()` is essentially just the `WHERE` of an executed SQL expression

```
result = datasrc.ExecuteSQL("SELECT * FROM sites WHERE cat = 5 ORDER BY id desc")

for feature in result:
    # do something with feature

datasrc.ReleaseResultSet(result)
```

Spatial Filtering

Spatial Filtering

- Spatial filtering available via two methods:
  - `SetSpatialFilter(polygon_geom)`
  - `SetSpatialFilterRect(<minx>, <miny>, <maxx>, <maxy>)`
- Works exactly like attribute filtering
  - Use `SetSpatialFilter(None)` to clear filter

Filtering: An Example

Filtering: An Example

```python
from osgeo import ogr
import os

indata = r"C:\GEOG410\Labs\Lab4\data\water.shp"
outdata = r"C:\GEOG410\Labs\Lab4\data\filtered.shp"

driver = ogr.GetDriverByName("ESRI Shapefile")

# open in data source; create out data source
inds = driver.Open(indata)
outds = driver.CreateDataSource(outdata)

# open inlayer; create outlayer using inlayer parameters
inlayer = inds.GetLayer()
outlayer = outds.CreateLayer(os.path.splitext(os.path.basename(outdata))[0],
                             geom_type=inlayer.GetGeomType(),
                             srs=inlayer.GetSpatialRef())

# copy field defns from inlayer to outlayer
inlayerdefn = inlayer.GetLayerDefn()
for fieldindex in xrange(inlayerdefn.GetFieldCount()):
    fielddefn = inlayerdefn.GetFieldDefn(fieldindex)
    outlayer.CreateField(fielddefn)

# continued on next slide
```

Filtering: An Example

```python
# set attribute and spatial filters
inlayer.SetAttributeFilter("AreaSqKm > 0.001")
inlayer.SetSpatialFilterRect(496046, 53778, 503754, 58054)

# loop through selected features, copying to new shp
for feature in inlayer:
    outlayer.CreateFeature(feature)
    feature = None

# close out layers and data sources
inlayer = None
outlayer = None
inds = None
outds = None
```