Week 12 Lecture:
Advanced Geoprocessing and Python
With ArcObjects

Introduction to Programming for GIS & Remote Sensing
GEO6938-4172
GEO4938-4166
Reminder

The ArcObjects walk-through lab questions are due on Thursday.

Lab this week used to finalize presentations

Presentations on 8th and 10th, NEXT WEEK!

– Present in alphabetical order of first name:
  • Benjamin, Cerian, Jeffrey, Joshua, Lindsay, Mariano, Melissa, Michal, Michelle, Reggie
Re-cap From Past Lectures

We’ve discussed ArcGIS and geoprocessing
  – Model builder
  – Exporting/importing scripts

ArcObjects are components exposing ArcGIS to control by other programming languages:
  – VBA examples
  – In depth walk-through in Lab
Today’s Lecture

Further example of ArcObjects and object relationships

Adding Layers to a Map
Setting Layer Symbology

ArcObjects in Python, it is possible!

Following slides based on lecture from
Sample: Add a Shapefile Programmatically

```vbscript
Set pFeatureLayer.FeatureClass = pFeatureWorkspace.OpenFeatureClass("Country")
```
Sample: Add a Personal Geodatabase FeatureClass Programmatically

Public Sub AddFeatureClass()

    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFeatureLayer As IFeatureLayer
    Dim pMxDocument As IMxDocument
    Dim pMap As IMap

    'Create a new ShapefileWorkspaceFactory object
    'and open a shapefile folder

    Set pWorkspaceFactory = New AccessWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile_  
        ("C:\ArcObjects\data\wilson_NC\wilson.mdb",0)

    'Create a new FeatureLayer and assign a feature class to it

    Set pFeatureLayer = New FeatureLayer
    Set pFeatureLayer.FeatureClass = pFeatureWorkspace.OpenFeatureClass("Streets")
    pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName

    'Add the FeatureLayer to the focus map
    Set pMxDocument = Application.Document
    Set pMap = pMxDocument.FocusMap pMap.AddLayer pFeatureLayer

End Sub
Layer Objects

Dim pFLayer as IFeatureLayer
Set pFLayer = New FeatureLayer

Dim pRLayer as IRasterLayer
Set pRLayer = New RasterLayer
Dim pAWorkspaceFactory as IWorkspaceFactory
Set pAWorkspaceFactory = New AccessWorkspaceFactory
Set pFeatureWorkspace = pAWorkspaceFactory.OpenFromFile
   ("C:\Program Files\ArcGIS\DeveloperKit\Samples\Data\World.mdb", 0)
Dim pFeatureClass as IFeatureClass
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass ("Streets")
Sample: Setting Layer Symbology

'Create a renderer, a symbol and a color to display the building footprint layer

'Color - Light Orange
Dim pLtOrange As IRgbColor
Set pLtOrange = New RgbColor

pLtOrange.Red = 246
pLtOrange.Green = 197
pLtOrange.Blue = 103

'symbol - light orange color and .5 width simple line symbol
Dim pLine As ISimpleLineSymbol
Dim pLine As ISimpleLineSymbol
Set pLine = New SimpleLineSymbol
Set pLinesym = pLine

pLinesym.Color = pLtOrange
pLinesym.Width = 0.5

'renderer
Dim pRenderer As ISimpleRenderer
Set pRenderer = New SimpleRenderer

Set pRenderer.Symbol = pLinesym
pRenderer.Label = "Streets"

'associate the renderer with the feature layer
Dim pGeoFLayer As IGeoFeatureLayer
Set pGeoFLayer = pFeatureLayer

Set pGeoFLayer.Renderer = pRenderer
Setting Layer Symbology: Color

Dim pLtOrange as IRgbColor
Set pLtOrange = New RgbColor
pLtOrange.Red = 246
pLtOrange.Green = 197
pLtOrange.Blue = 103
Display Object Model Diagram
Display Object Model Diagram
Setting Layer Symbology: Symbol

```
Dim pLine as ISimpleLineSymbol
Set pLine = New SimpleLineSymbol

Dim pLineSym as ILineSymbol
Set pLineSym = pLine

pLineSym.Color = pLtOrange
pLineSym.Width = 0.5
```
Dim pRenderer as ISimpleRenderer
Set pRenderer = New SimpleRenderer

Set pRenderer.Symbol = pLinesym
pRenderer.Label = "Streets"
Setting Layer Symbology: FeatureLayer

Dim pGeoFLayer as IGeoFeatureLayer
Set pGeoFLayer = pFeatureLayer
Set pGeoFLayer.Renderer = pRenderer
Examples Using ArcObjects in Python

Must use a .NET go-between to give Python access:


Once installed, all .NET assemblies (modules) on system are available to Python