

# ESRI Developer Summit

March 22–25, 2010  
Palm Springs, CA

## Python Scripting for Map Automation in ArcGIS 10

*Michael Grossman*

*Jeff Barrette*



# What is map scripting (aka arcpy.mapping)?

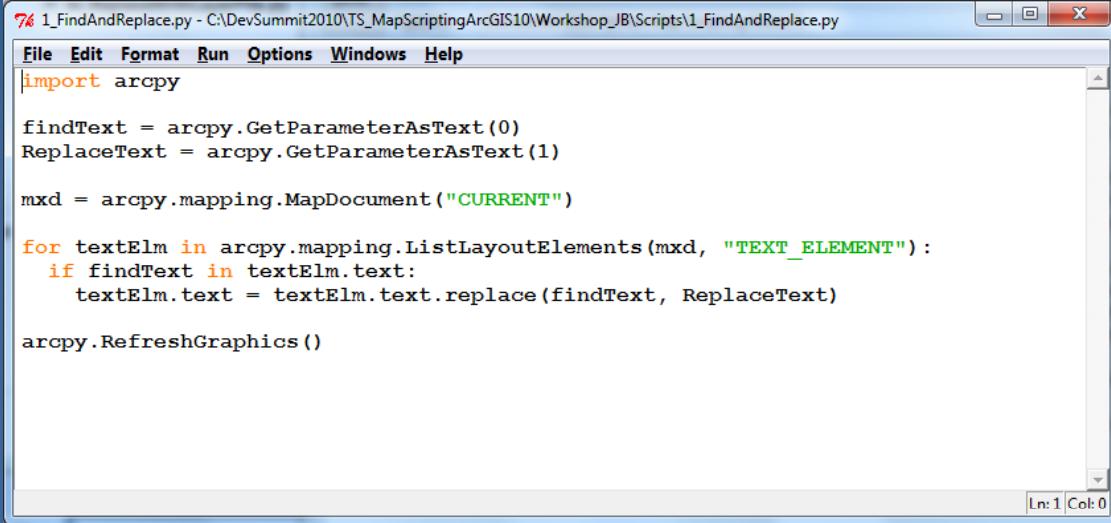
- A new **mapping module** that is part of the geoprocessing ArcPy site-package
- A **python scripting API** that allows our users to:
  - **Manage map documents, layer files, and the data within them**
    - Find a layer with data source X and replace with Y
    - Update a layer's symbology in many MXDs
    - Generate reports that lists document information
      - Data sources, broken layers, spatial reference info, etc.
  - **Automate the exporting and printing of map documents**
  - **Automate map production and create PDF map books**

# Who is arcpy.mapping for? Why was it built?

- **An easy to use, productive scripting environment for the GIS Analyst.**
  - Coarser grained object model
  - Not a complete replacement for ArcObjects
- **An environment to use for basic map/layer management and map automation tasks**
- **A simple way to publish mapping tasks to the server environment.**
  - arcpy.mapping scripts can be easily published as geoprocessing tools

# Demonstration #1

## Brief introduction to an arcpy.mapping script



The screenshot shows a Windows Notepad window with the title bar '1\_FindAndReplace.py - C:\DevSummit2010\TS\_MapScriptingArcGIS10\Workshop\_JB\Scripts\1\_FindAndReplace.py'. The window contains the following Python code:

```
File Edit Format Run Options Windows Help
import arcpy

findText = arcpy.GetParameterAsText(0)
ReplaceText = arcpy.GetParameterAsText(1)

mx = arcpy.mapping.MapDocument("CURRENT")

for textElm in arcpy.mapping.ListLayoutElements(mx, "TEXT_ELEMENT"):
    if findText in textElm.text:
        textElm.text = textElm.text.replace(findText, ReplaceText)

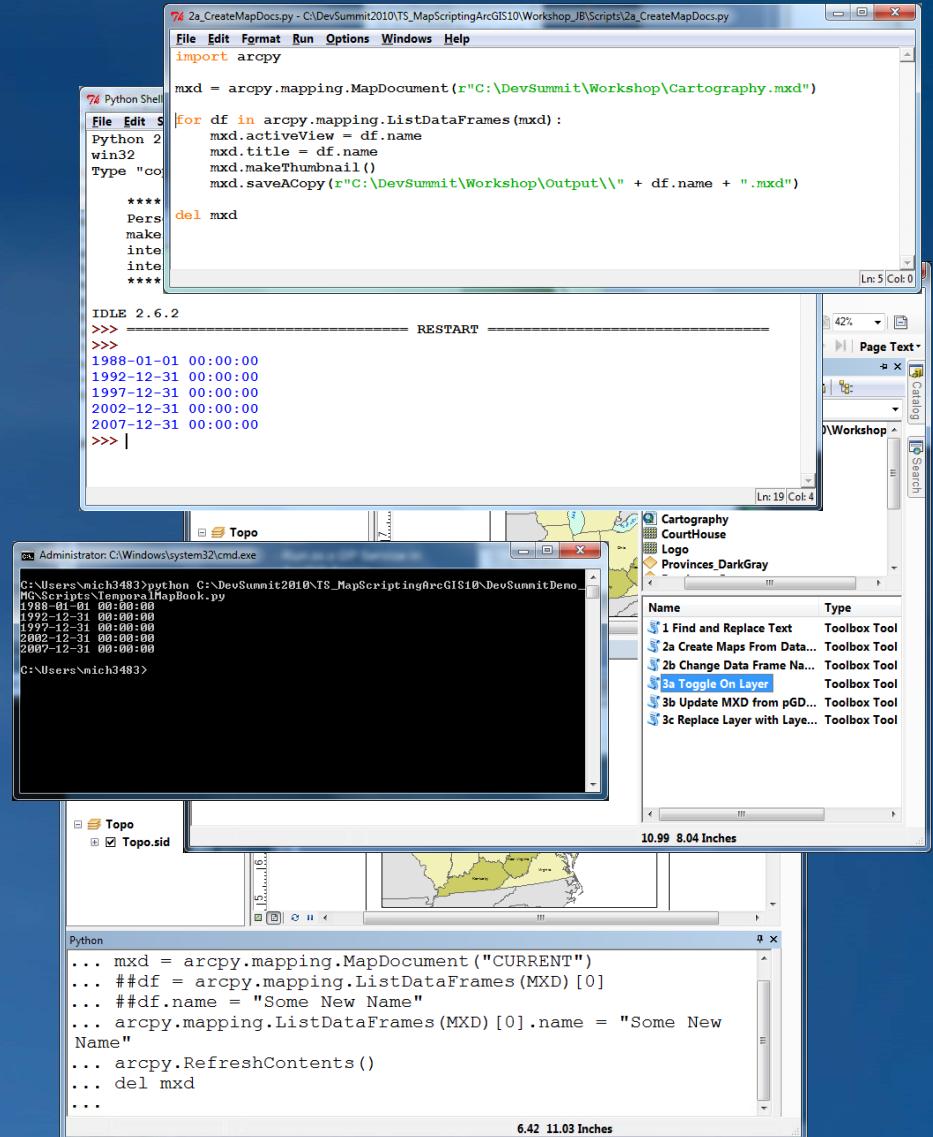
arcpy.RefreshGraphics()
```

The code is a Python script named '1\_FindAndReplace.py' that performs a find and replace operation on text elements in the current ArcMap layout. It uses arcpy.mapping to list layout elements and arcpy.GetParameterAsText to get the find and replace strings from parameters.

Find and replace text in an ArcMap layout

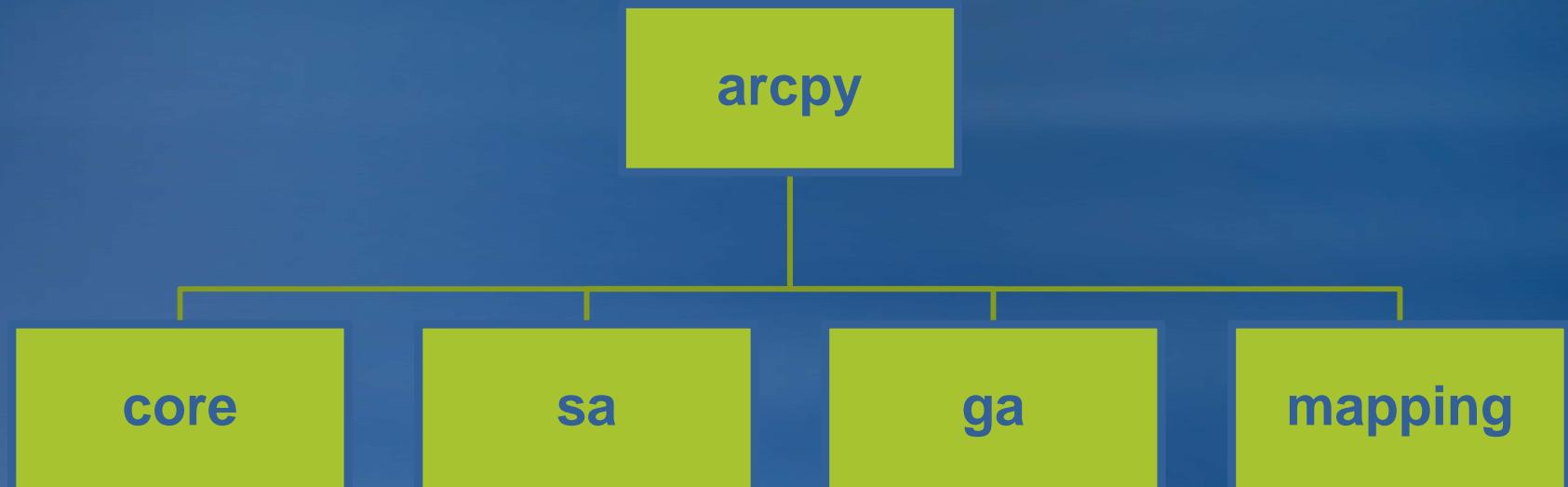
# Running arcpy.mapping scripts

- **arcpy.mapping is part of the Geoprocessing (GP) framework**
- **Run in wide variety of places**
  - Script tool in ArcMap or ArcCatalog
  - Python Window in ArcMap
  - Standalone python - from IDE, from the command line, or as a scheduled task
  - Run as a GP Service in ArcGIS Server



# Tour of arcpy.mapping

## Overview



# Tour of arcpy.mapping

## Managing Documents and Layers

### CLASSES

MapDocument  
Layer  
TableView  
LabelClass  
DataFrame  
DataFrameTime  
GraphicElement  
LegendElement  
PictureElement  
TextElement  
MapSurroundElement  
PictureElement

### FUNCTIONS

MapDocument  
Layer  
ListBrokenDataSources  
ListDataFrames  
ListLayers  
ListLayoutElements  
ListPrinterNames  
ListTableViews  
AddLayer  
AddLayerToGroup  
InsertLayer  
MoveLayer  
RemoveLayer  
UpdateLayer

# Tour of arcpy.mapping

## Printing, Exporting, Server Publishing, Map Books

### CLASSES

`DataDrivenPages`  
`PDFDocument`

### FUNCTIONS

`ExportToAI`  
`ExportToBMP`  
`ExportToEMF`  
`ExportToEPS`  
`ExportToGIF`  
`ExportToJPEG`  
`ExportToPDF`  
`ExportToPNG`  
`ExportToSVG`  
`ExportToTIFF`  
`PDFDocumentCreate`  
`PDFDocumentOpen`  
`PrintMap`  
`PublishMSDToServer`  
`AnalyzeForMSD`  
`ConvertToMSD`

## arcpy.mapping for Map Documents (MXDs)

- Opening Map Documents (MXD) with arcpy.mapping
- Use the arcpy.mapping.MapDocument function
- Takes a path to MXD file or special keyword "CURRENT"

# arcpy.mapping for Map Documents (MXDs)

- The CURRENT keyword versus MXDs on disk

- Get map from disk

```
mapDoc = arcpy.mapping.MapDocument(r"C:\Maps\LocalGovernmentInfrastructure.mxd")
```

- Get map from current ArcMap session

```
mapDoc = arcpy.mapping.MapDocument("CURRENT")
```

- When using CURRENT

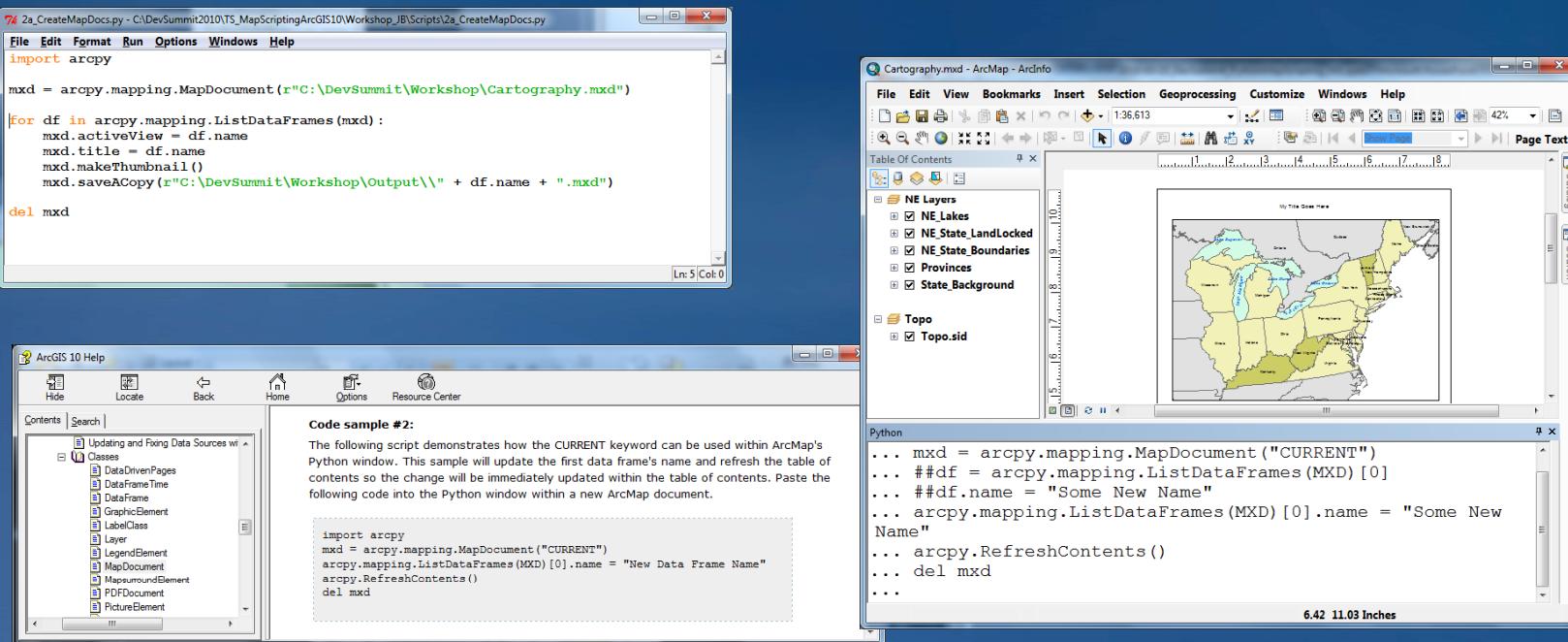
- Always run in foreground (checkbox in script tool props)
  - Be wary of open conflicts, file contention

- Limitations and pre-authoring

- No "New Map" function, so keep an empty MXD available

# Demonstration #2

## Working with Map Documents (MXDs)



- A standalone script that creates a new MXD for each data frame
- Use the Python Window to update a data frame name

# arcpy.mapping for Map Layers and Data Frames

- The “List” functions
  - `ListLayers`
  - `ListDataFrames`
  - Watch the list indexes (you may often forget to use [0])
    - `df = arcpy.mapping.ListDataFrames(MXD)[0]`
- **DataFrame properties and methods**
  - `Map Navigation`
  - `DataFrameTime`

# arcpy.mapping for Map Layers and Data Frames

- Layer functions

## FUNCTIONS

`Layer`  
`ListLayers`  
`ListTableViews`

`AddLayer`  
`AddLayerToGroup`  
`InsertLayer`  
`MoveLayer`  
`RemoveLayer`  
`UpdateLayer`

# arcpy.mapping for Map Layers and Data Frames

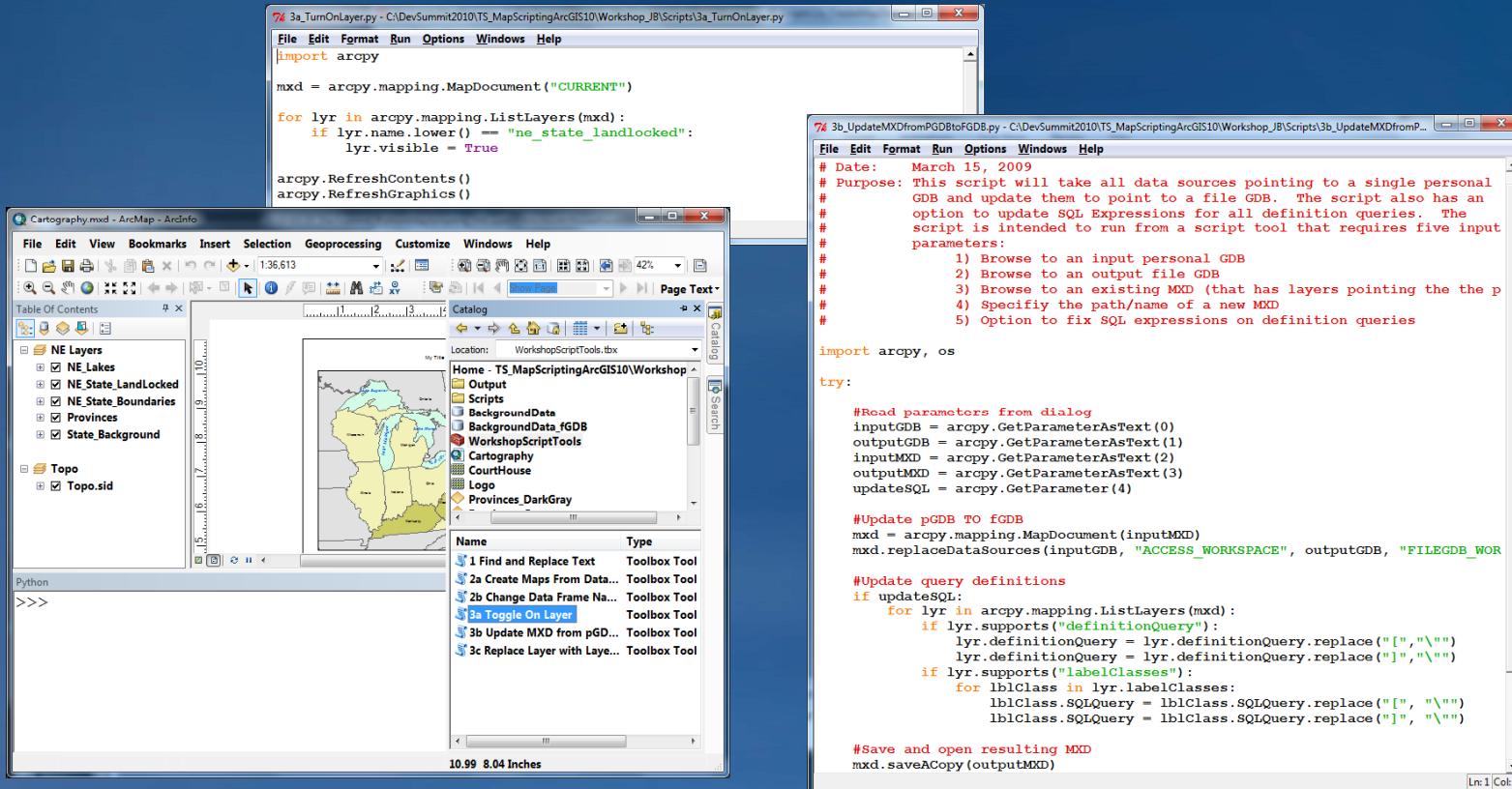
- When and what to pre-author for Map Document scenarios
  - Layer Symbology (aka Renderer) properties are not accessible
  - Pre-Author Layer files and use UpdateLayer or data source methods to connect it to data
- Scenarios for updating layers
  - changing layer symbology
  - changing layer type
    - eg. batch replace a traditional featureclass "basemap" layer with a basemap layer from a server

# arcpy.mapping for Map Layers and Data Frames

- **Updating Data Sources**
  - use `arcpy.mapping` for migrating Map Documents and Layer files to new data sources. Batch migration.
  - Fancier scripts can help mitigate migration pain: SQL syntax changes, etc.

# Demonstration #3

## Working with Map Layers and Data Frames



The image shows a screenshot of ArcMap and two Python script windows.

**ArcMap Window:** The map view shows a state map of Wisconsin with several layers selected, including "NE\_Lakes", "NE\_State\_LandLocked", "NE\_State\_Boundaries", "Provinces", and "State\_Background". The "Topo" folder contains "Topo.sid". The Python window shows the following code:

```
3a_TurnOnLayer.py - C:\DevSummit2010\TS_MapScriptingArcGIS10\Workshop_JB\Scripts\3a_TurnOnLayer.py
File Edit Format Run Options Windows Help
import arcpy

mxd = arcpy.mapping.MapDocument("CURRENT")

for lyr in arcpy.mapping.ListLayers(mxd):
    if lyr.name.lower() == "ne_state_landlocked":
        lyr.visible = True

arcpy.RefreshContents()
arcpy.RefreshGraphics()
```

**Script Windows:** There are two code editors side-by-side.

**Left Script (3a\_TurnOnLayer.py):**

```
3a_TurnOnLayer.py - C:\DevSummit2010\TS_MapScriptingArcGIS10\Workshop_JB\Scripts\3a_TurnOnLayer.py
File Edit Format Run Options Windows Help
import arcpy

mxd = arcpy.mapping.MapDocument("CURRENT")

for lyr in arcpy.mapping.ListLayers(mxd):
    if lyr.name.lower() == "ne_state_landlocked":
        lyr.visible = True

arcpy.RefreshContents()
arcpy.RefreshGraphics()
```

**Right Script (3b\_UpdateMXDfromPGDBtoFGDB.py):**

```
3b_UpdateMXDfromPGDBtoFGDB.py - C:\DevSummit2010\TS_MapScriptingArcGIS10\Workshop_JB\Scripts\3b_UpdateMXDfromP...
File Edit Format Run Options Windows Help
# Date: March 15, 2009
# Purpose: This script will take all data sources pointing to a single personal
# GDB and update them to point to a file GDB. The script also has an
# option to update SQL Expressions for all definition queries. The
# script is intended to run from a script tool that requires five input
# parameters:
# 1) Browse to an input personal GDB
# 2) Browse to an output file GDB
# 3) Browse to an existing MXD (that has layers pointing the the p
# 4) Specifiy the path/name of a new MXD
# 5) Option to fix SQL expressions on definition queries

import arcpy, os
try:
    #Read parameters from dialog
    inputGDB = arcpy.GetParameterAsText(0)
    outputGDB = arcpy.GetParameterAsText(1)
    inputMXD = arcpy.GetParameterAsText(2)
    outputMXD = arcpy.GetParameterAsText(3)
    updateSQL = arcpy.GetParameter(4)

    #Update pGDB TO fGDB
    mxd = arcpy.mapping.MapDocument(inputMXD)
    mxd.replaceDataSources(inputGDB, "ACCESS_WORKSPACE", outputGDB, "FILEGDB_WOR

    #Update query definitions
    if updateSQL:
        for lyr in arcpy.mapping.ListLayers(mxd):
            if lyr.supports("definitionQuery"):
                lyr.definitionQuery = lyr.definitionQuery.replace("[", "\\""]
                lyr.definitionQuery = lyr.definitionQuery.replace("]", "\\"")
            if lyr.supports("labelClasses"):
                for lblClass in lyr.labelClasses:
                    lblClass.SQLQuery = lblClass.SQLQuery.replace("[", "\\""]
                    lblClass.SQLQuery = lblClass.SQLQuery.replace("]", "\\"")

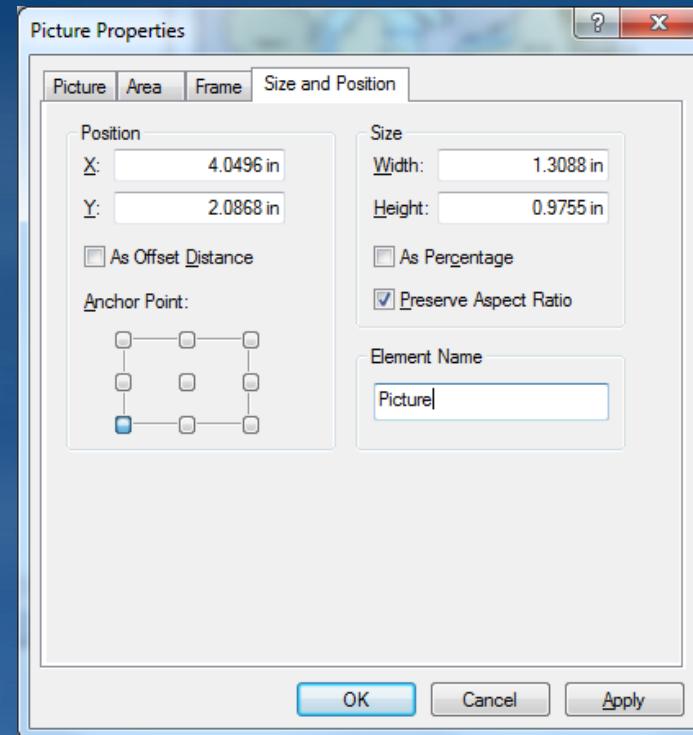
    #Save and open resulting MXD
    mxd.saveACopy(outputMXD)

```

- A script that finds a layer and turns it on
- A script that updates layer data sources
- A script that replaces a layer

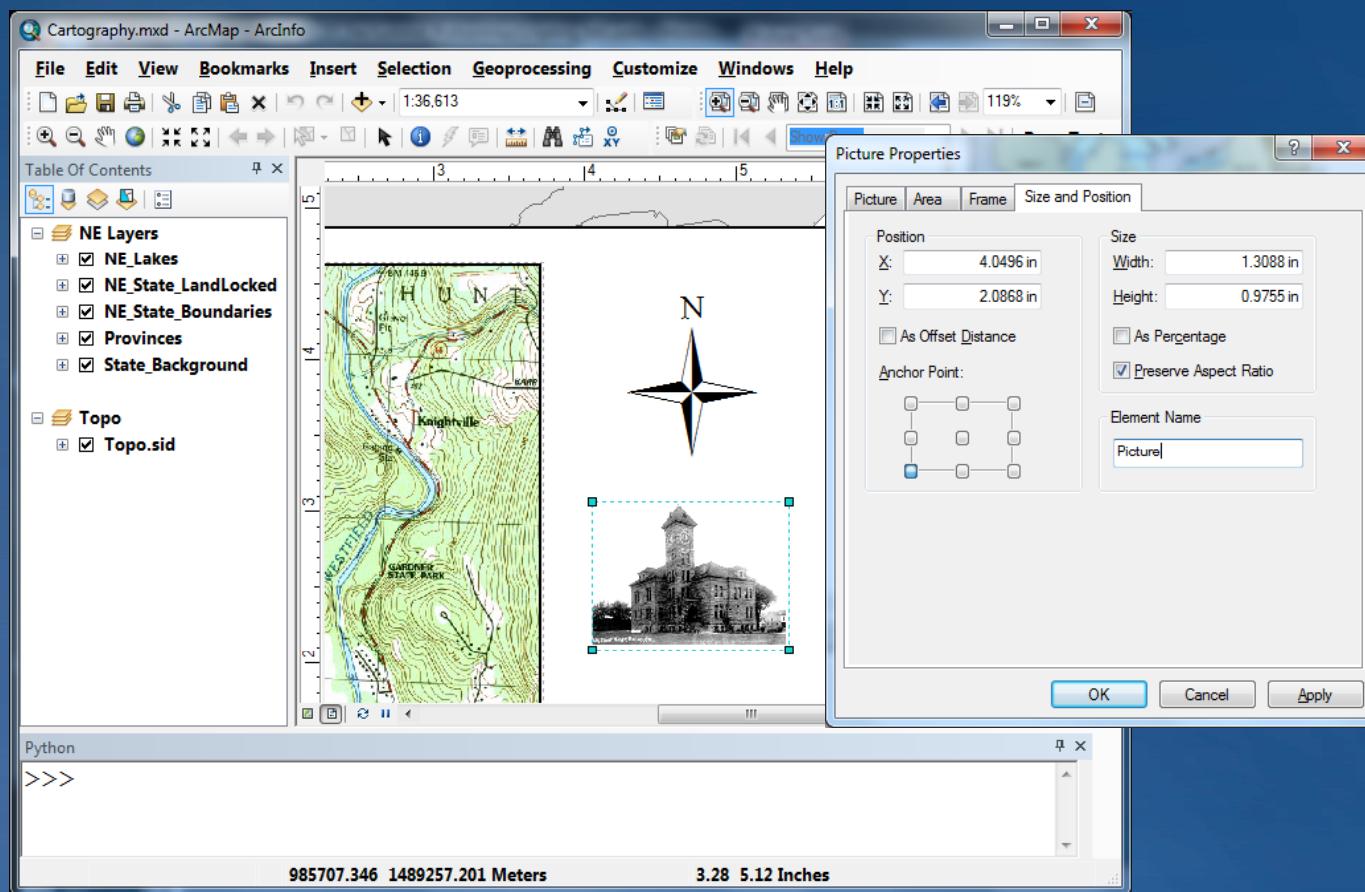
# arcpy.mapping for the Page Layout

- When and what to pre-author for layout manipulation scenarios
  - Name your layout elements
  - Cannot add new elements, so pre-author and hide off page if necessary



# Demonstration #4

## Working with layout elements



- A script tool to find a picture element and change its data source

# arcpy.mapping for Printing and Exporting

- Export functions
- Print Functions
- Map Server Publishing
- Map Books
- The PDFDocument class

## CLASSES

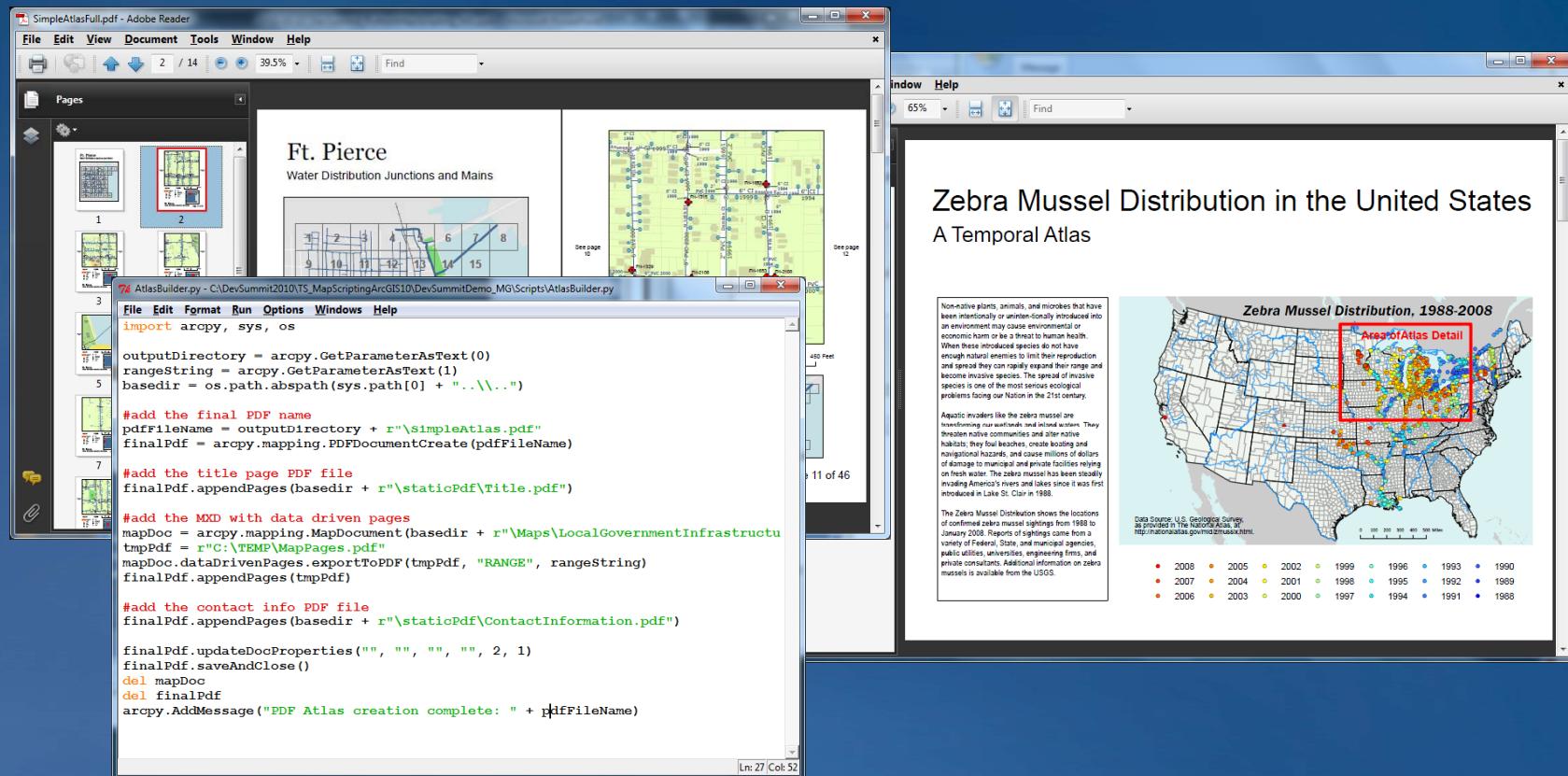
DataDrivenPages  
PDFDocument

## FUNCTIONS

ExportToAI  
ExportToBMP  
ExportToEMF  
ExportToEPS  
ExportToGIF  
ExportToJPEG  
ExportToPDF  
ExportToPNG  
ExportToSVG  
ExportToTIFF  
PDFDocumentCreate  
PDFDocumentOpen  
PrintMap  
PublishMSDToServer  
AnalyzeForMSD  
ConvertToMSD

# Demonstration #5

## Map output and map books



- A script tool to export data driven pages to multipage PDF
  - A script tool to create a temporal map book
  - A full custom application ported from VBA

# Resources available

- **Desktop help**
  - **Geoprocessing → The ArcPy site package → Mapping module**
    - **Alphabetical lists of classes and functions**
    - **Detailed discussions**
    - **Multiple sample scripts for each class and function topic**
- **ArcGIS Resource Center**
  - **Download sample script tools**
    - **see the Geoprocessing Resource Center Gallery**
  - **Watch video demonstrations**
  - **Monitor the user forums/discussions**

# We want your feedback

- Get access to ArcGIS 10 Pre-release
- Try using it for you day to day tasks
- Help us identify future requirements

Thank you for your attention.

Questions?