

Esri Southeast Regional User Group Conference

May 2-4, 2011 | Jacksonville, FL



Python in ArcGIS: An Introduction

Presented by
Ben Ramseth



Audience

- **New to Python scripts, beginner**
- **Comfortable with ArcGIS GP tools but want to become more efficient**
- **Move to Python from other scripting language**
- **Interested in what's new at 10**

Agenda

- **The Basics**
 - **What is Python**
 - **Python and ArcGIS**
 - **ArcPy Site Package**
 - **Geoprocessing**
- **Scripting for Map Automation**
 - **MapScripting**
 - **Tour ArcPy Mapping**
 - **MapDocuments**
 - **Layouts, Dataframes, Elements**
 - **Map Output and Map Books**
 - **SA and GA Modules**
- **Looking Ahead: 10.1**

Why Python?

- **Esri has embraced Python for ArcGIS 10**
- **Python is the language that fulfills the needs of our user community**
 - **Easy to learn**
 - **Excellent for beginners and experts**
 - **Suitable for large projects or small scripts**
 - **Cross-platform**

Python 101



- **Clear, easy to read syntax**
- **Easy to use, makes it simple to get started**
- **Variety of basic data types**
 - Including lists and dictionaries
- **Comes with a large standard library**
- **Supports raising and catching exceptions**
- **Code can be grouped into modules and packages**
- **Supports object-oriented programming**

<http://www.python.org/about/>

Python at 10.0

Esri has fully embraced Python as its language for automation

1. ArcPy site-package

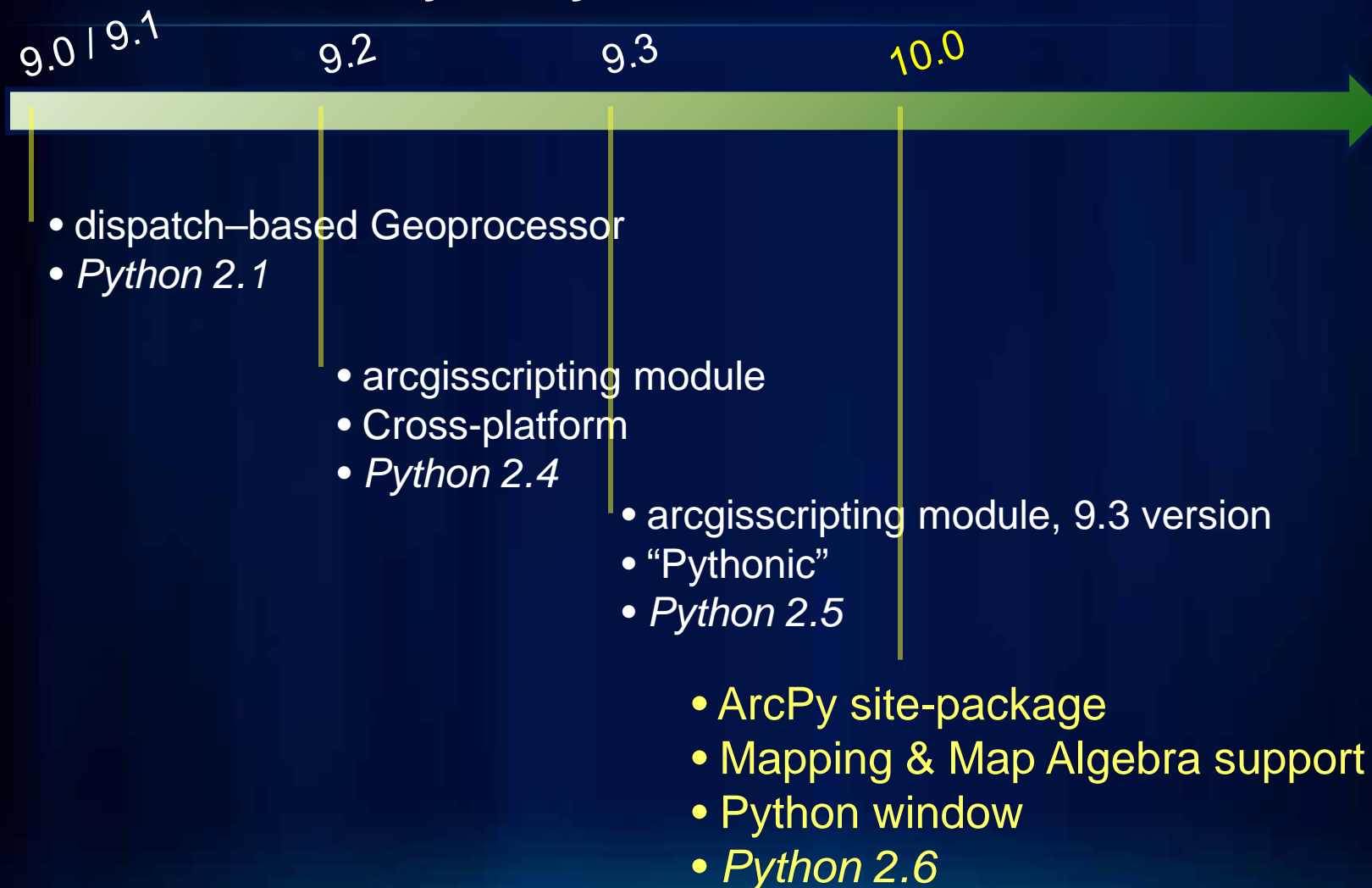
- Includes mapping and Map Algebra support
- Successor to the arcgisscripting module

2. Python window

- Python access and interaction from within ArcGIS

3. Python script tool framework

A brief history of Python in ArcGIS



What is ArcPy?

- **A cornerstone for automation in ArcGIS**
 - data analysis
 - data conversion
 - data management
 - map automation

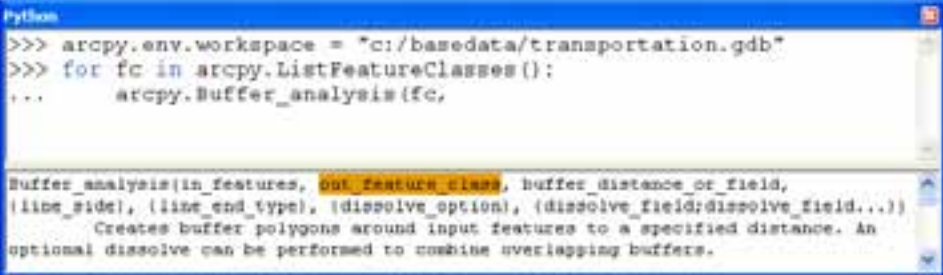
- **ArcPy is a native Python site-package**
 - Access to 800+ geoprocessing tools
 - Provides embedded reference documentation for each function, class and module
 - Code completion for ArcGIS components in your favorite Python editor
 - Familiar to arcgisscripting users

ArcPy improvements

- **Improved coding experience, such as:**
 - **Cursors**
 - **Classes**
 - **Multi-value parameters can be expressed as Python lists**
 - **Ability to convert rasters to and from NumPy arrays**
- **ArcPy is supported by modules, including:**
 - **A mapping module (arcpy.mapping)**
 - **A Spatial Analyst module (arcpy.sa) to support map algebra**
 - **A Geostatistical Analyst module (arcpy.ga)**

What is the Python window?

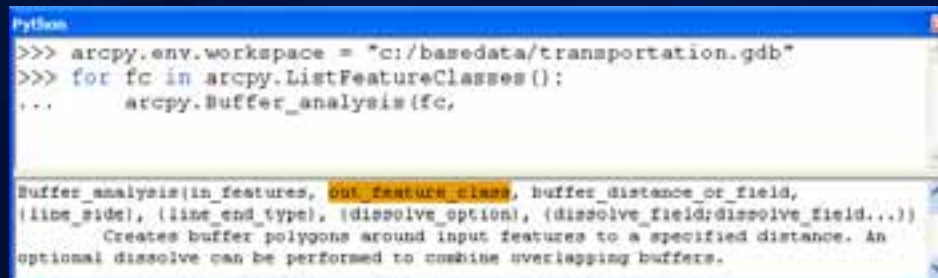
- **An embedded Interactive Python window within ArcGIS**
 - Can access ArcPy, including tools and environments
 - Can access any other Python functionality,
 - Better code completion and intelligence



```
Python
>>> arcpy.env.workspace = "c:/basedata/transportation.gdb"
>>> for fc in arcpy.ListFeatureClasses():
...     arcpy.Buffer_analysis(fc,
Buffer_analysis(in_features, out_feature_class, buffer_distance_or_field,
(line_side), (line_end_type), (dissolve_option), (dissolve_field:dissolve_field...))
Creates buffer polygons around input features to a specified distance. An
optional dissolve can be performed to combine overlapping buffers.
```

What is the Python window?

- The Python window is for:
 - Testing ideas
 - Experimenting with and learning Python
 - Simple execution of tools
 - Building quick and easy workflows in Python



```
Python
>>> arcpy.env.workspace = "c:/basedata/transportation.gdb"
>>> for fc in arcpy.ListFeatureClasses():
...     arcpy.Buffer_analysis(fc,
Buffer_analysis(in features, out_feature_class, buffer distance or field,
(line_side), (line_end_type), (dissolve_option), (dissolve_field;dissolve_field...))
Creates buffer polygons around input features to a specified distance. An
optional dissolve can be performed to combine overlapping buffers.
```

Running Tools

- Tools are accessed as functions on arcpy
- Environments as properties from arcpy.env class

```
# ~~~ PYTHON CODE ~~~
```

```
import arcpy
```

```
# Set the workspace
```

```
arcpy.env.workspace = "c:/st_Johns/GISData.gdb"
```

```
# Execute Geoprocessing tool
```

```
arcpy.Intersect_analysis(["roads", "urban_area", "urban_roads"], 5, "join")
```

Getting tool syntax

Results window, 'Copy as Python Snippet'

Export Model to Python script

Drag tool into Python window

Tool documentation

```
arcpy.Usage("Buffer_analysis")
```



*A note on tool organization

- Tools can be accessed directly from arcpy

```
import arcpy  
arcpy.GetCount_management(fc)
```

- Or from arcpy 'toolbox' modules

```
from arcpy.management import as dm  
dm.GetCount(fc)
```

- *Matter of preference – functionally no difference*

Environments

- **Script writers set the environment and tools use them**
 - **General settings**
 - **Current Workspace, Output Spatial Reference, Extent**
 - **Raster analysis settings**
 - **Cell Size, Mask**
 - **Many more**




arcpy.env.workspace

arcpy.env.outputCoordinateSystem

arcpy.env.extent

arcpy.env.cellSize

Tool Messages

- Tools return 3 types of messages
 -  Informative messages (severity = 0)
 -  Warning messages (severity = 1)
 -  Error messages (severity = 2)

```
# start try block
```

```
try:
```

```
    arcpy.Buffer_analysis("c:/ws/roads.shp", "c:/outws/roads10.shp", 10)
```

```
# If an error occurs when running a tool, print the tool messages
```

```
except arcpy.ExecuteError:
```

```
    print arcpy.GetMessages(2)
```

```
# Any other error
```

```
except Exception as e:
```

```
    print e.message
```


Results Window

- **Each result contains, in order**
 - **Output datasets and values**
 - **Input datasets and values**
 - **Environment settings used in tool execution**
 - **All messages**
- **Using the Results window**
 - **View tool execute status**
 - **Copy and paste or drag and drop, a result into a model**
 - **Copy a code snippet into your Python script**
 - **Manage results and output datasets**
 - **Add a result's output to a map display, Python window or Model**
 - **Review a result's processing messages**

Results Window

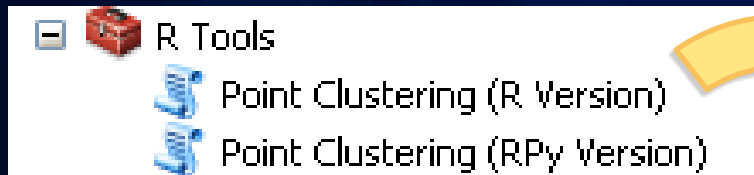
DEMO



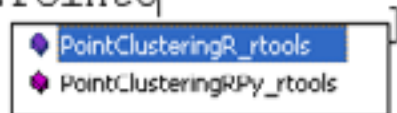
Sample Script Tools

Geoprocessing Tools

- Tools are the fundamental unit of geoprocessing
- There are hundreds of tools at your disposal
 - You can create your own tools (ModelBuilder, Python, etc.)
- Any tool, once created, can be called in Python by using the `arcpy.ImportToolbox` function



```
>>> arcpy.ImportToolbox(r'c:\tools\RTools\R Tools.tbx')  
>>> arcpy.PointC
```



Agenda

- **The Basics**
 - What is Python
 - Python and ArcGIS
 - ArcPy Site Package
 - Geoprocessing
- **Scripting for Map Automation**
 - MapScripting
 - Tour ArcPy Mapping
 - MapDocuments
 - Layouts, Dataframes, Elements
 - Map Output and Map Books
 - SA and GA Modules
- **Looking Ahead: 10.1**

What is map scripting (aka arcpy.mapping)?

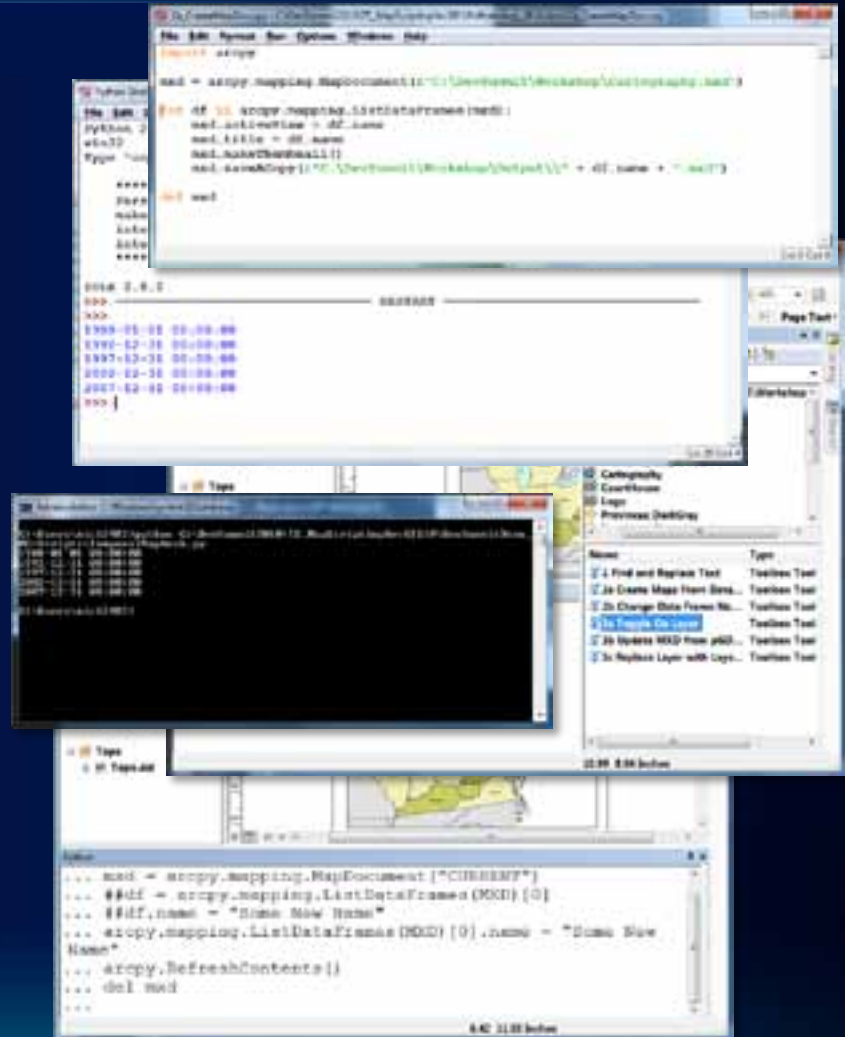
- A new mapping module that is part of the geoprocessing ArcPy site-package
- A python scripting API that allows our users to:
 - Manage map documents, layer files, and the data within them
 - Find a layer with data source and replace
 - Update a layer's symbology in many MXDs
 - Generate reports that 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**
 - **Courser 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**

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 (cont.)

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 (cont.)

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

Referencing Map Documents (MXDs)

- Opening Map Documents (MXD) with `arcpy.mapping`
- Use the `arcpy.mapping.MapDocument` function
- Takes a path to MXD file on disk or special keyword "CURRENT"
- Reference map on disk
`mxd = arcpy.mapping.MapDocument(r"C:\some.mxd")`
- Get map from current ArcMap session
`mxd = arcpy.mapping.MapDocument("CURRENT")`
- When using CURRENT
 - Always run in the foreground
 - Pay attention to open conflicts, file contention
- Limitations and pre-authoring
 - No "New Map" function, so keep an empty MXD available

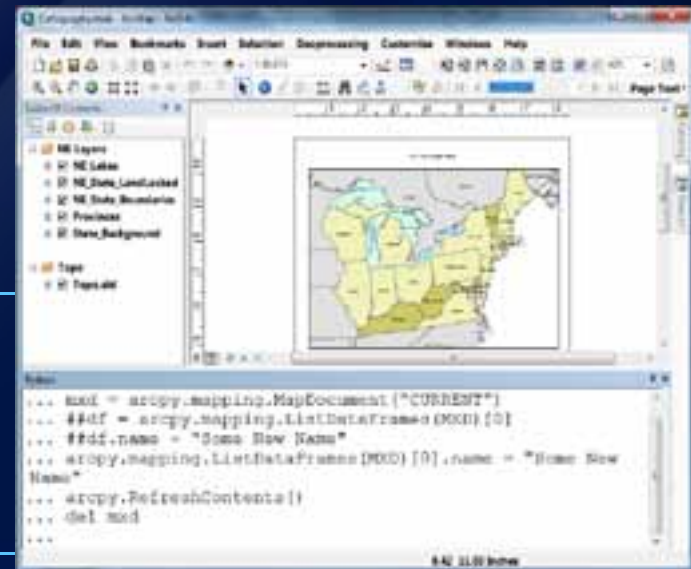
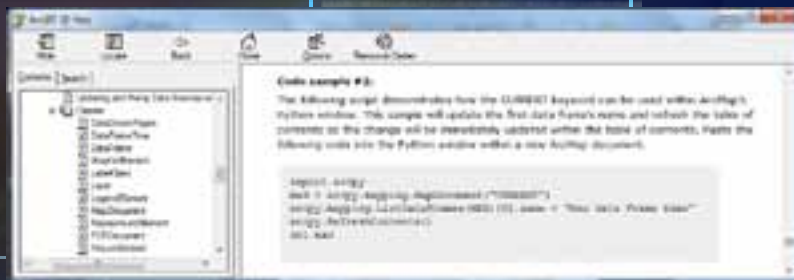
Working with Map Documents DEMO

```
File Edit Format View Options Windows Help
Python 2.7.10

mod = arcpy.mapping.MapDocument("C:\Users\user\workspace\GetTopology.mxd")

for df in arcpy.mapping.ListDataFrames(mod):
    mod.setLayerView(df.name)
    mod.title = df.name
    mod.makeThumbnail()
    mod.saveAsCopy("C:\Users\user\workspace\Output\\" + df.name + ".mxd")

del mod
```



*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]`

- Layer properties

- Common properties are available (e.g., def query, visible)
- All properties can be updated via layer (.lyr) files

- DataFrame properties and methods

- Map Navigation
- DataFrameTime

arcpy.mapping for Map Layers and Data Frames

- Layer functions

```
Layer  
ListLayers  
ListTableViews
```

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

- Data Frame Class

```
Methods  
panToExtent(extent)  
zoomToSelectedFeatures()
```

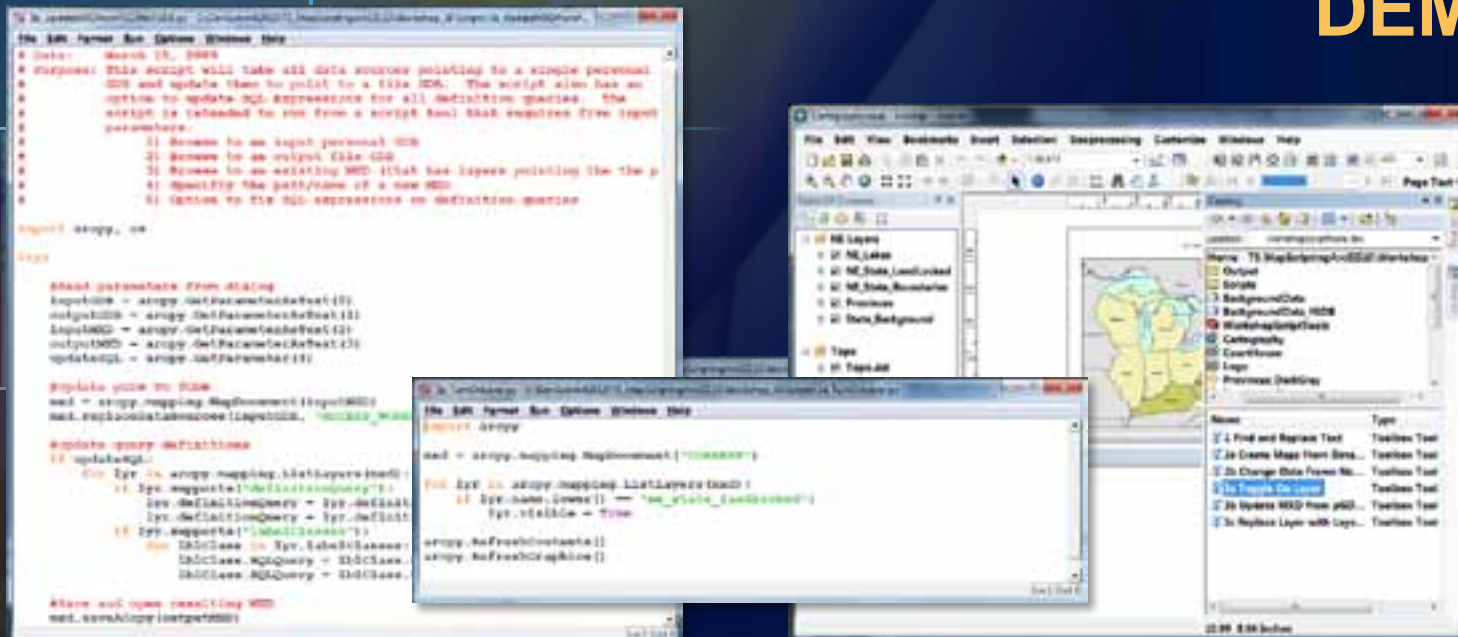
```
Properties:  
credits  
description  
displayUnits  
elementHeight  
elementPosition  
...
```

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 (change to online basemap)
- **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.

Working with Map Layers and Data Frames

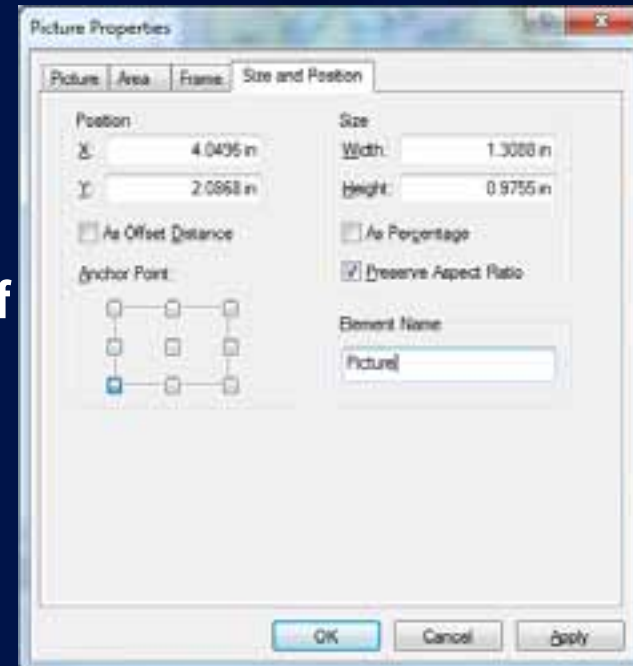
DEMO



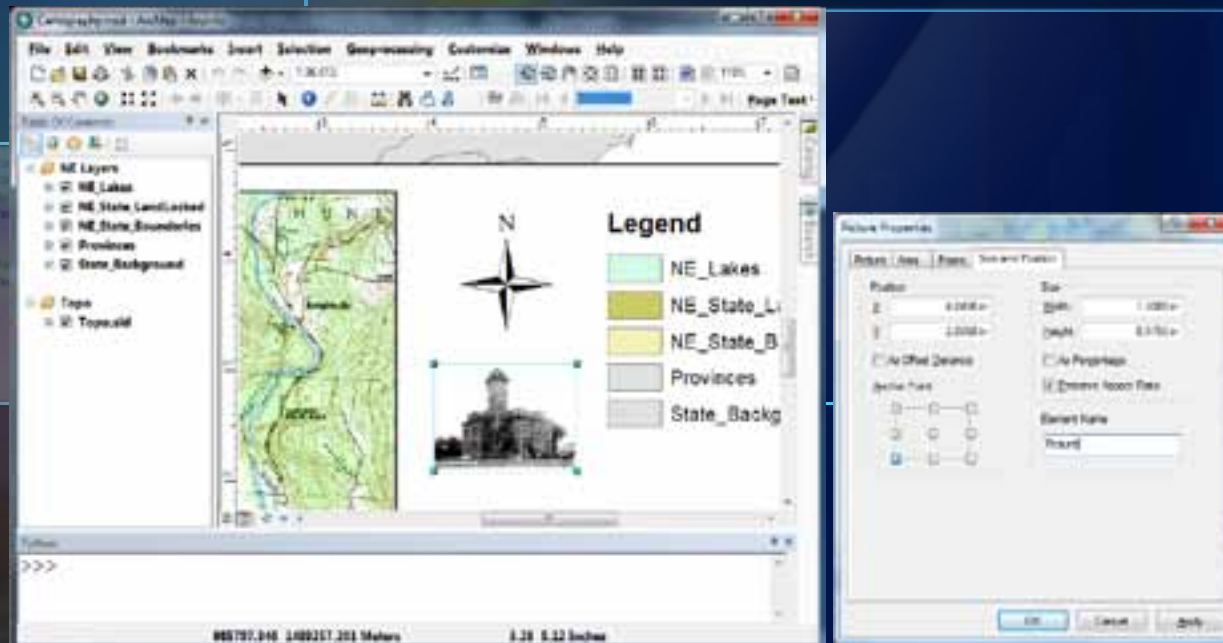
A script that finds a layer and turns it on
A script that updates layer data sources
A script that replaces a layer

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**



Working with Layout Elements DEMO



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

arcpy.mapping for Printing and Exporting

- **PDF & DDP classes**
- **Export and print functions**
- **Map Server Publishing**
- **Map Book generation**

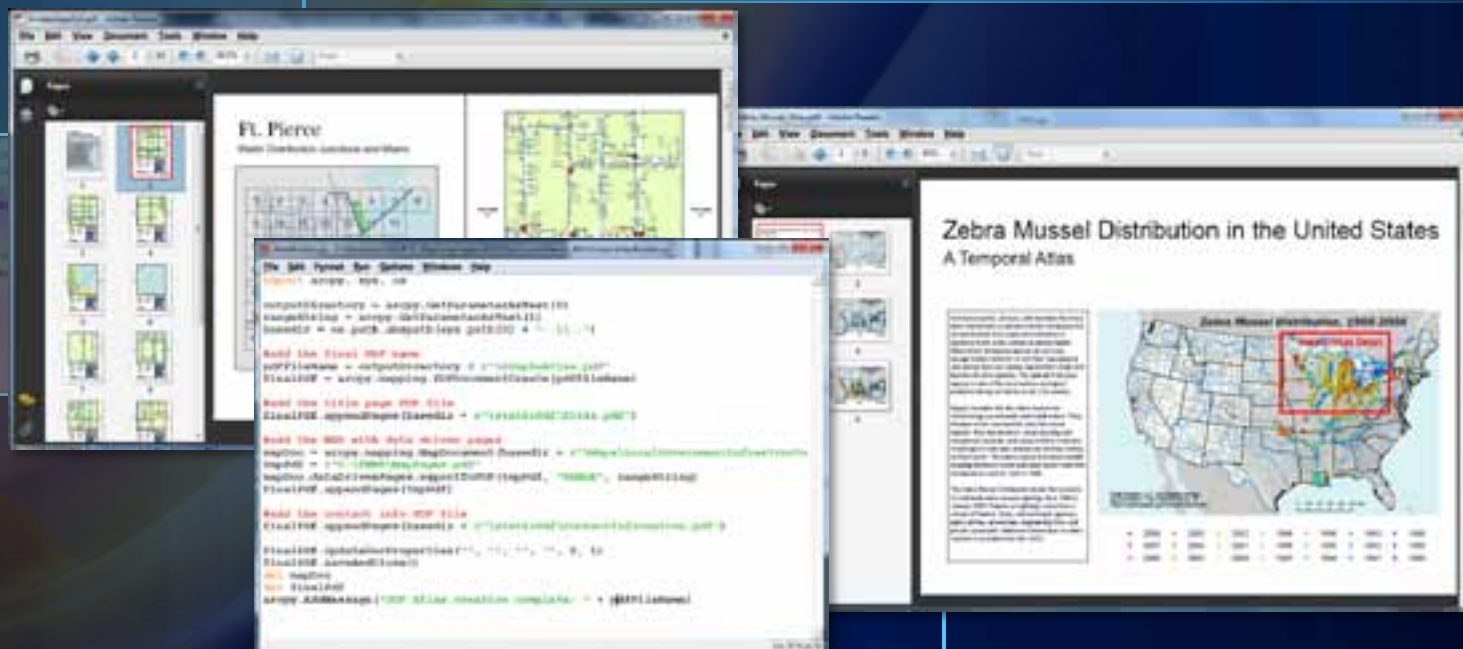
CLASSES

DataDrivenPages
PDFDocument

FUNCTIONS

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

Map Output and Map Books DEMO



- 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*

Spatial Analyst module

- Automate workflows and create new tools
- Auto completion, expanded scriptability, deferred execution
- Integrates Map Algebra into Python
 - Includes all Spatial Analyst tools
 - Supports operators in Map Algebra expressions
 - Support complex parameters
 - Map Algebra expressions syntax is similar to previous tools

```
from arcpy.sa import *  
demmm = Raster("DEM") / 3.28  
slpdeg = Slope(demmm, "DEGREE")  
  
demfs = FocalStatistics("DEM", NbrRectangle(3,3), "MEAN")
```

Geostatistical Analyst Module

- Automate workflows
- Auto completion, expanded scriptability, deferred execution
- Define varying and complex parameters
- Access output properties

```
import arcpy

arcpy.env.workspace = "C:/gapyexamples/data"
Arcpy.GaussianGeostatisticalSimulations_ga("C:/gapyexamples/data/kriging.lyr","10","C:/gapyexamples/output","ggs","","","","2000","","","","MEAN")
```

Getting more help

Resource Center

<http://resourcesbeta.esri.com/content/geoprocessing>

The screenshot shows the ArcGIS Resource Center website. The header includes the ArcGIS logo, navigation links for Help, Blogs, and Forums, and a search bar. The main content area is titled "Geoprocessing" and features a sidebar with categories like Automation, Modeling and Analysis, Tools and Framework, and Emerging. The main text describes geoprocessing as an essential part of daily work with ArcGIS, providing tools for analysis and data management. It also mentions a large suite of tools for GIS tasks and methods for automating tasks. A "Get Support" button is visible at the bottom left.

Desktop help

The screenshot shows the ArcGIS Desktop help content for Geoprocessing. The content is organized into a tree view with the following items:

- Geoprocessing
 - What is geoprocessing?
 - A quick tour of geoprocessing
 - Essential geoprocessing vocabulary
 - Geoprocessing tools
 - The geoprocessing framework
 - Commonly used tools
 - Finding tools
 - Executing tools
 - Managing tools and toolboxes
 - Creating tools
 - Sharing tools
 - Geoprocessing with ModelBuilder
 - Geoprocessing with Python
 - What is Python?
 - Essential Python vocabulary
 - A quick tour of Python
 - Accessing tools
 - Working with sets of data in Python
 - Accessing geographic data in Python
 - Geoprocessing with ArcGIS Server
 - The ArcPy site package
 - What is ArcPy?
 - Essential ArcPy vocabulary
 - A quick tour of ArcPy
 - Functions
 - Classes
 - Mapping module
 - Geostatistical Analyst module
 - Spatial Analyst module
 - Geoprocessing environment settings
 - Geoprocessing tool reference

Agenda

- **The Basics**
 - What is Python
 - Python and ArcGIS
 - ArcPy Site Package
 - Geoprocessing
- **Scripting for Map Automation**
 - MapScripting
 - Tour ArcPy Mapping
 - MapDocuments
 - Layouts, Dataframes, Elements
 - Map Output and Map Books
 - SA and GA Modules
- **Looking Ahead: 10.1**

Python Programming Features:

- ***Use Python to Create ArcGIS Desktop Add-Ins***
 - *ArcGIS 10.1 will support Python as an option for authoring Desktop add-ins (i.e. collection of tools on a toolbar).*
- ***Python Toolboxes***
 - *ArcGIS 10.1 will include support for Python toolboxes. Python toolboxes provide a new way of creating script tools entirely in Python.*
- ***Data Access Module (da)***
 - *The data access module, arcpy.da, is a new python module for working with data. It provides improved cursor support (including considerably faster performance), allows control of the edit session and edit operation, support for versioning.*

ArcMap Page Layout and Data Driven Pages improvements (cont'):

- ***Dynamic Legend***
 - *Display of only features in the visible extent*
 - *Features counts*
 - *New fixed frame option, legend items will be dropped if they do not fit inside the legend frame.*
 - *Options to automatically adjust the number of columns*
- ***North Arrow***
 - *New "True North" option has been added as well as calibration angle enhancements.*
- ***Scale bar***
 - *Enhancements that allow you to set the zero point as the anchor so that multiple, different unit, scale bars can be aligned with each other.*

ArcMap Page Layout and Data Driven Pages improvements:

- ***Symbology API –***
 - *Automate symbology properties (graduated colors, graduated symbols, unique values, and classified rasters)*
- ***Export Report -.***
 - *Automate the generation of reports without having to open ArcMap*
- ***Layer time –***
 - *Access to a layer's time properties in order to perform analysis over time. You can also enable time on layers.*
- ***Other improvements –***
 - *Setting text size, setting relative paths, reading page size, etc.*

Questions?

Thank You

Arcpy Sample Scripts Available at:

<http://resources.arcgis.com/gallery/file/geoprocessing/details?entryID=A910AB18-1422-2418-3418-3885D388EF60>