



2011 Esri Developer Summit

Palm Springs, CA

Python Scripting for Map Automation

Michael Grossman

Jeff Barrette



What is map scripting (arcpy.mapping)?

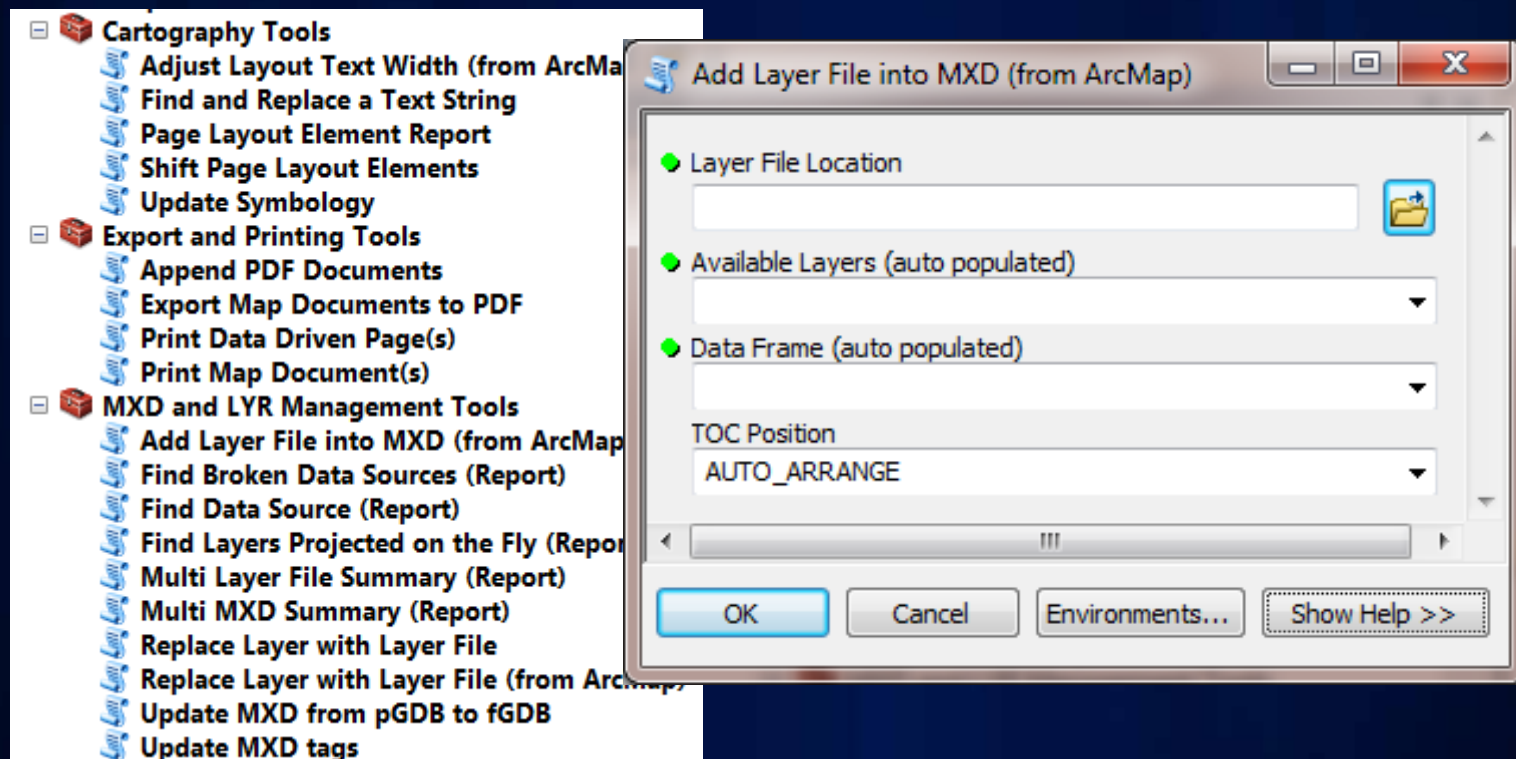
- **A new mapping module that is part of the geoprocessing ArcPy site-package**
- **A python scripting API that allows you to:**
 - **Manage map documents, layer files, and the data within them**
 - Find a layer with data source X and replace with Y
 - Update a layer's symbology in many MXDs
 - Generate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

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**

Demonstration:

Brief introduction to arcpy.mapping sample script tools



Samples available on the Resource Center

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

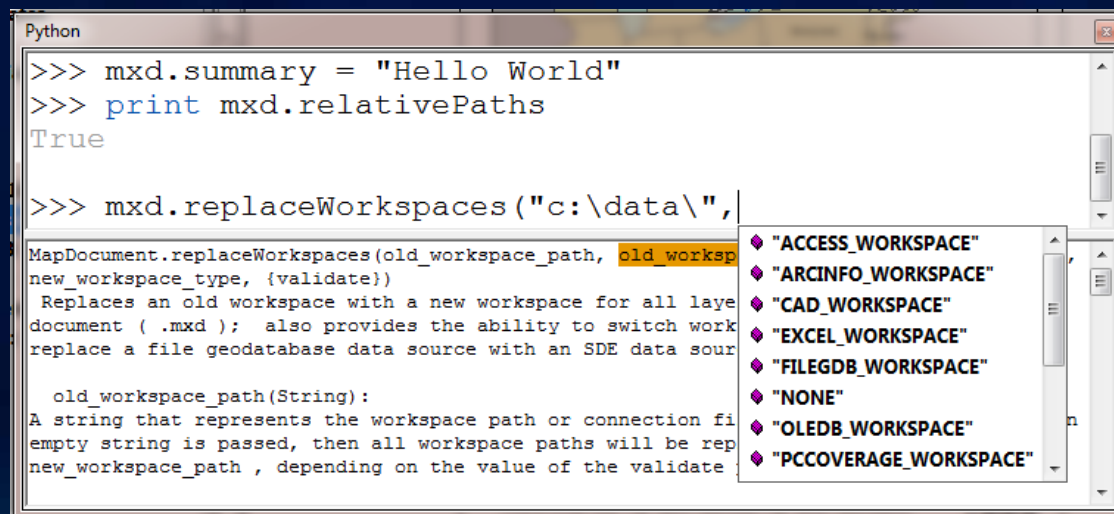
DataDrivenPages
PDFDocument

FUNCTIONS

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

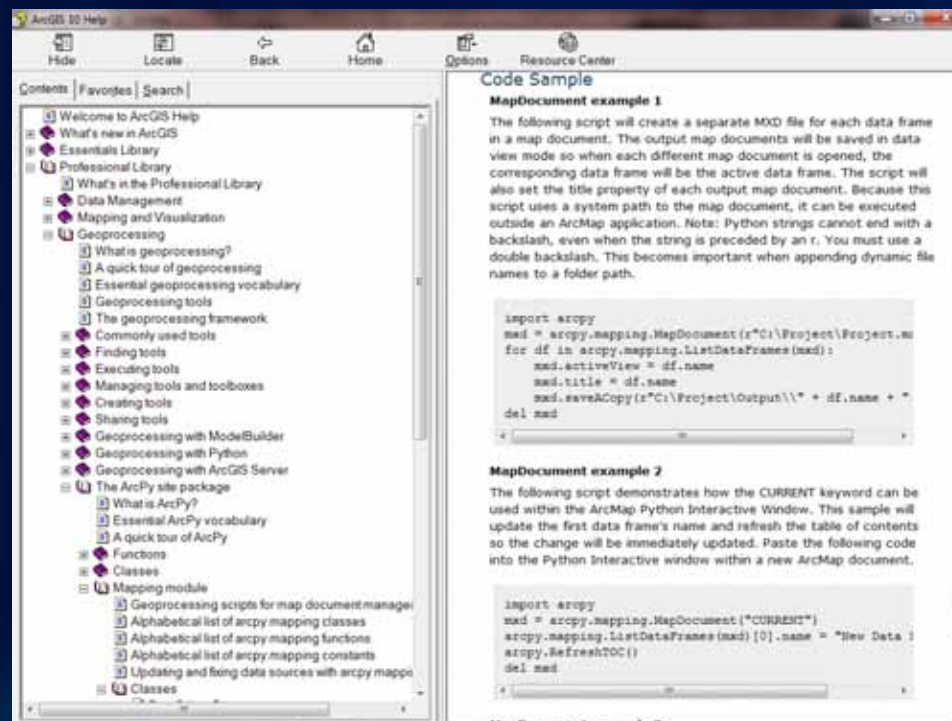
Python Window

- Command Line becomes the Python Window
- Quick and easy access to Python and arcpy
 - Gateway for new users to learn Python
 - Intellisense for all tools and methods/properties & help window
 - Quickly and efficiently execute tools



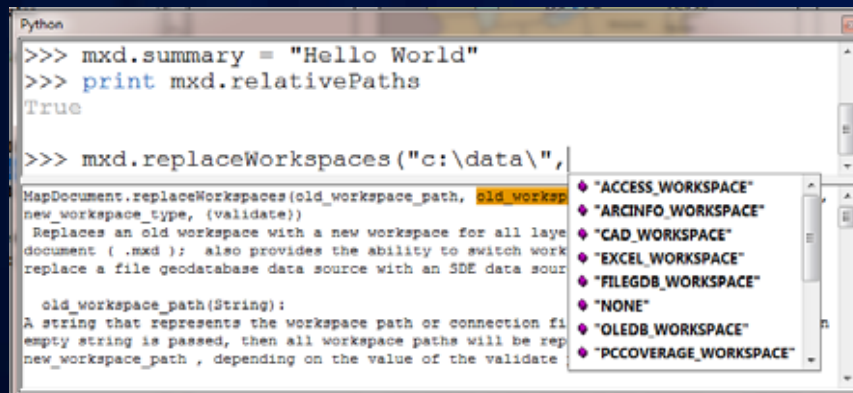
arcpy.mapping help

- Go to Professional Library → Geoprocessing → The ArcPy site package → Mapping module



Demonstration

The Python Window and using the Desktop Help System



Python

```
>>> mxd.summary = "Hello World"
>>> print mxd.relativePaths
True

>>> mxd.replaceWorkspaces("c:\data\","
```

MapDocument.replaceWorkspaces(old_workspace_path, old_workspace_type, new_workspace_type, (validate))
Replaces an old workspace with a new workspace for all layers in the map document (.mxd). Also provides the ability to switch workspace types. For example, replace a file geodatabase data source with an SDE data source.

old_workspace_path(String):
A string that represents the workspace path or connection file name. If an empty string is passed, then all workspace paths will be replaced with new_workspace_path, depending on the value of the validate parameter.

- ACCESS_WORKSPACE
- ARCINFO_WORKSPACE
- CAD_WORKSPACE
- EXCEL_WORKSPACE
- FILEGDB_WORKSPACE
- NONE
- OLEDB_WORKSPACE
- PCCOVERAGE_WORKSPACE



arcpy.mapping for Map Documents

- MapDocument function

MapDocument

- **MapDocument Class**

Methods

save

saveAsCopy

UpdateDataSources

...

Properties:

author

credits

...

Referencing Map Documents (MXDs)

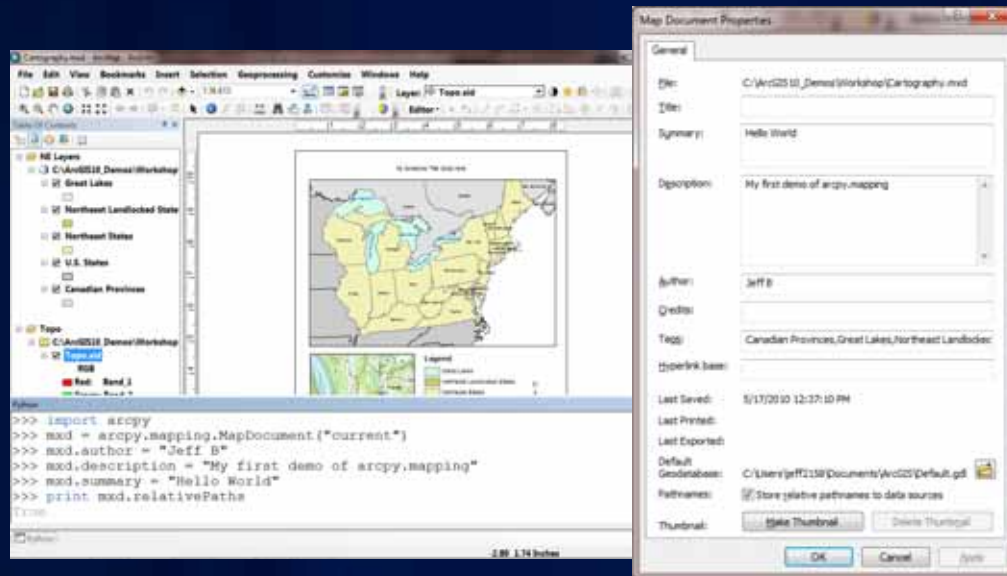
- Opening Map Documents (MXD) with `arcpy.mapping`
- Use the `arcpy.mapping.MapDocument` function
- Takes a path to MXD file on disk or special keyword "CURRENT"
- Reference map on disk
`mxd = arcpy.mapping.MapDocument(r"C:\some.mxd")`
- Get map from current ArcMap session
`mxd = arcpy.mapping.MapDocument("CURRENT")`

Referencing Map Documents (MXDs), cont.

- When using CURRENT
 - Always run in foreground (checkbox in script tool properties)
 - Be wary of open conflicts, file contention
 - May need to refresh the application

```
arcpy.RefreshActiveView()  
arcpy.RefreshContents()
```
- Limitations and pre-authoring
 - No "New Map" function, so keep an empty MXD available
 - Can't create new objects (e.g., north arrow, text elements)

Demonstration: Working with Map Documents (MXDs)



- Use Python Window to change map document property info
- Evaluate relative paths, last saved, etc.
- Change the active view
- Save changes out to a new file

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

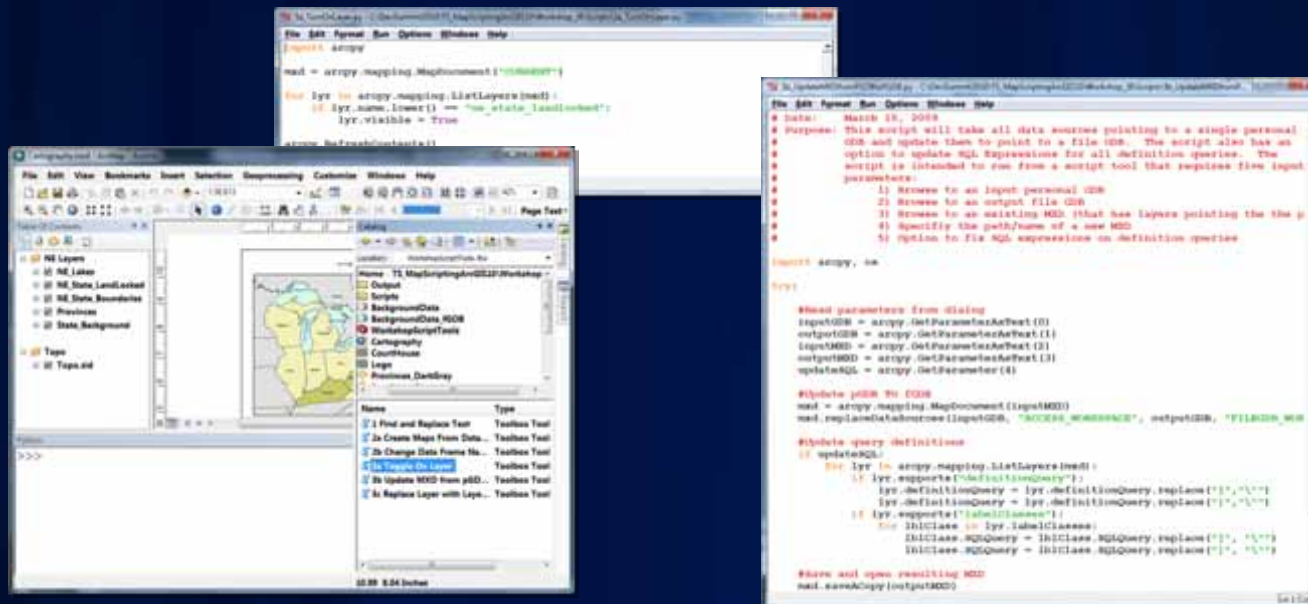
AddLayer
AddLayerToGroup
InsertLayer
MoveLayer
RemoveLayer
UpdateLayer
```

- Data Frame Class

```
Methods
    panToExtent(extent)
    zoomToSelectedFeatures()

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

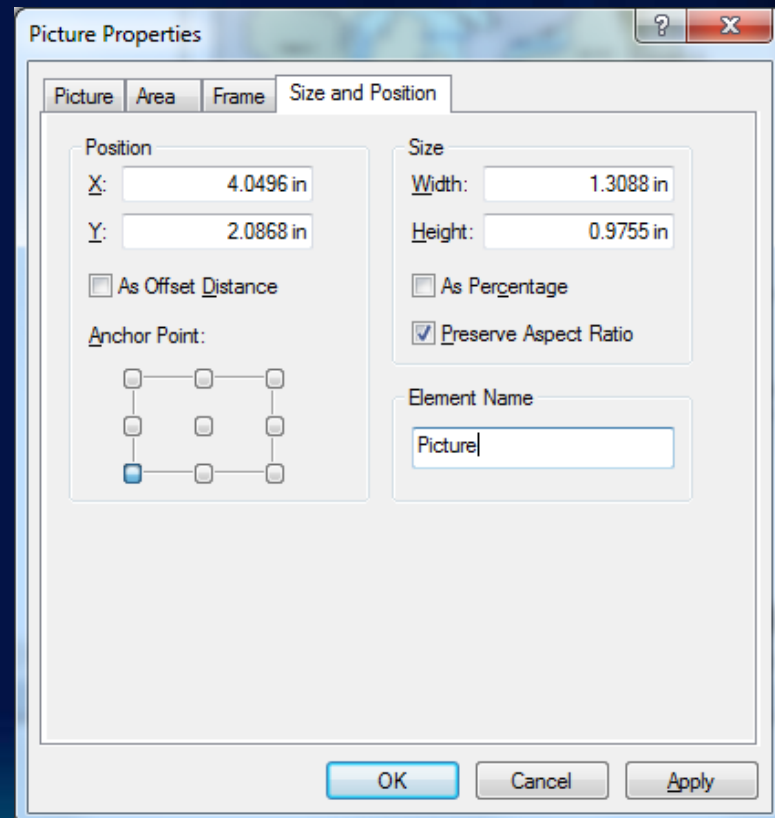

Demonstration: Working with Map Layers and Data Frames



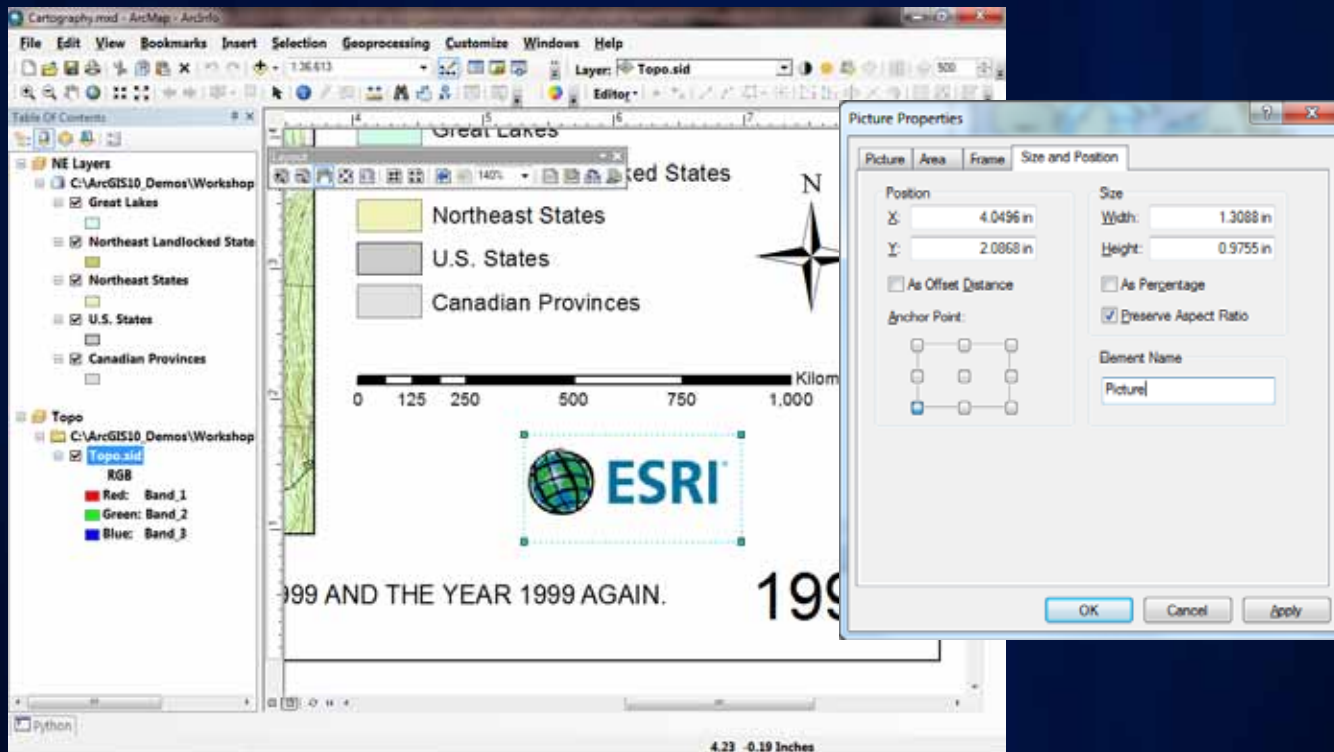
- Find a layer and turns it on or off
- Update a layer's symbology using an existing layer file
- Modify the scale/rotation of a data frame
- Zoom to selected features

arcpy.mapping for the Page Layout

- **When and what to pre-author for layout manipulation scenarios**
 - **Name your layout elements**
 - **Set the appropriate anchor**
 - **Cannot add new elements, so pre-author and hide**



Demonstration: Working with layout elements



- Find a picture element and change its data source
- Find and replace text in an ArcMap layout

arcpy.mapping for Printing and Exporting

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

CLASSES

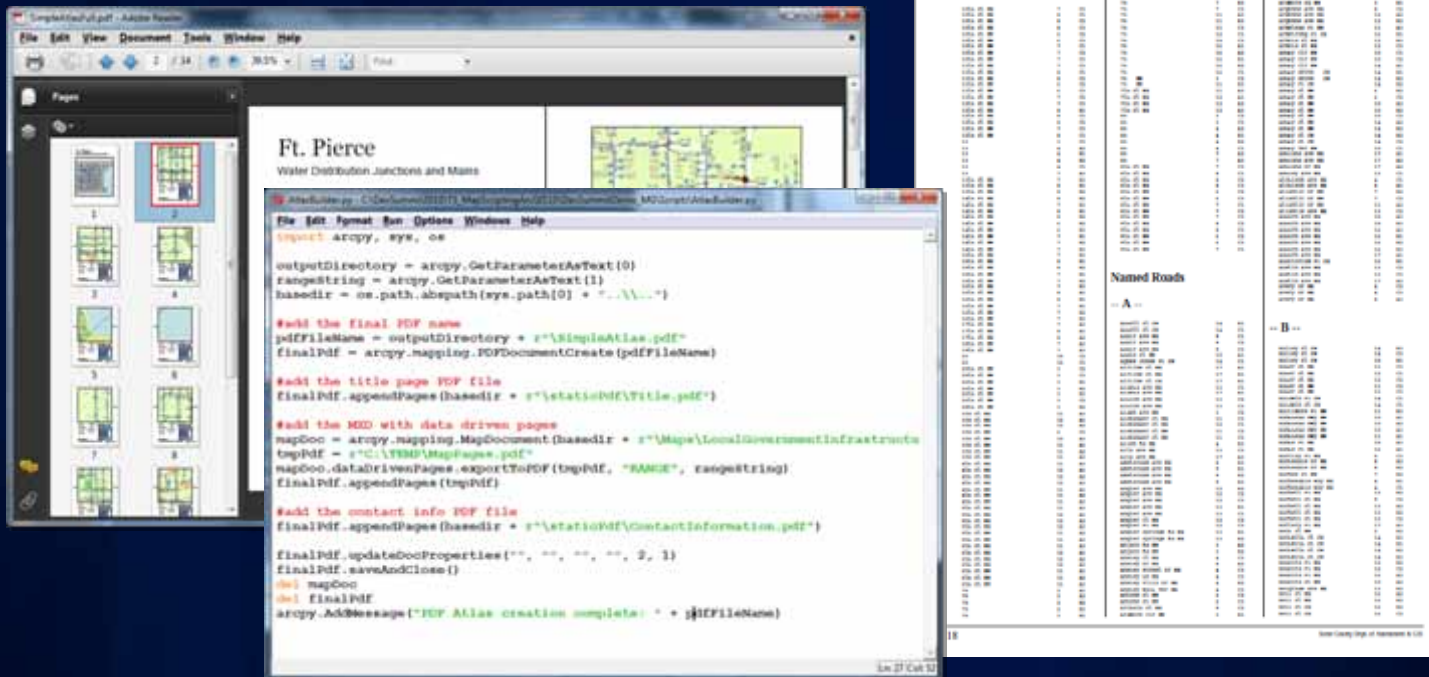
DataDrivenPages
PDFDocument

FUNCTIONS

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

Demonstration:

Map output and map books



The image is a composite of three screenshots. The top-left screenshot shows a map application window titled 'SampleFull.pdf - Adobe Reader'. It displays a map of 'Ft. Pierce' with 'Water Distribution Junctions and Mains'. The bottom-left screenshot shows a Python script in a text editor. The script uses arcpy to create a multipage PDF map book. It sets an output directory, gets parameters for range and base directory, and then appends various PDF files to a final PDF document. The files include a title page, a map document with data-driven pages, and a contact information page. The rightmost screenshot shows a page from a map book index. It lists 'Numbered Roads' and 'Named Roads' with corresponding page numbers. The 'Numbered Roads' section lists roads like '1000 N. US HWY 1' through '1000 N. US HWY 100'. The 'Named Roads' section lists roads like 'A - 1000 N. US HWY 1' through 'B - 1000 N. US HWY 100'.

```
File Edit Format Run Options Windows Help
import arcpy, sys, os

outputDirectory = arcpy.GetParameterAsText(0)
rangeString = arcpy.GetParameterAsText(1)
basedir = os.path.abspath(sys.path[0] + "..\\..")

#add the final PDF name
pdfFileName = outputDirectory + r"\\SingleAtlas.pdf"
finalPdf = arcpy.mapping.PDFDocumentCreate(pdfFileName)

#add the title page PDF file
finalPdf.appendPages(basedir + r"\\staticPDF\\Title.pdf")

#add the MDD with data driven pages
mapDoc = arcpy.mapping.MapDocument(basedir + r"\\Maps\\LocalGovernmentInfrastructure.mxd")
tmpPdf = r"C:\\TEMP\\MapPages.pdf"
mapDoc.dataDrivenPages.exportToPDF(tmpPdf, "RANGE", rangeString)
finalPdf.appendPages(tmpPdf)

#add the contact info PDF file
finalPdf.appendPages(basedir + r"\\staticPDF\\ContactInformation.pdf")

finalPdf.updateDocProperties("", "", "", "", 2, 1)
finalPdf.saveAndClose()
del mapDoc
del finalPdf
arcpy.AddMessage("PDF Atlas creation complete: " + pdfFileName)
```

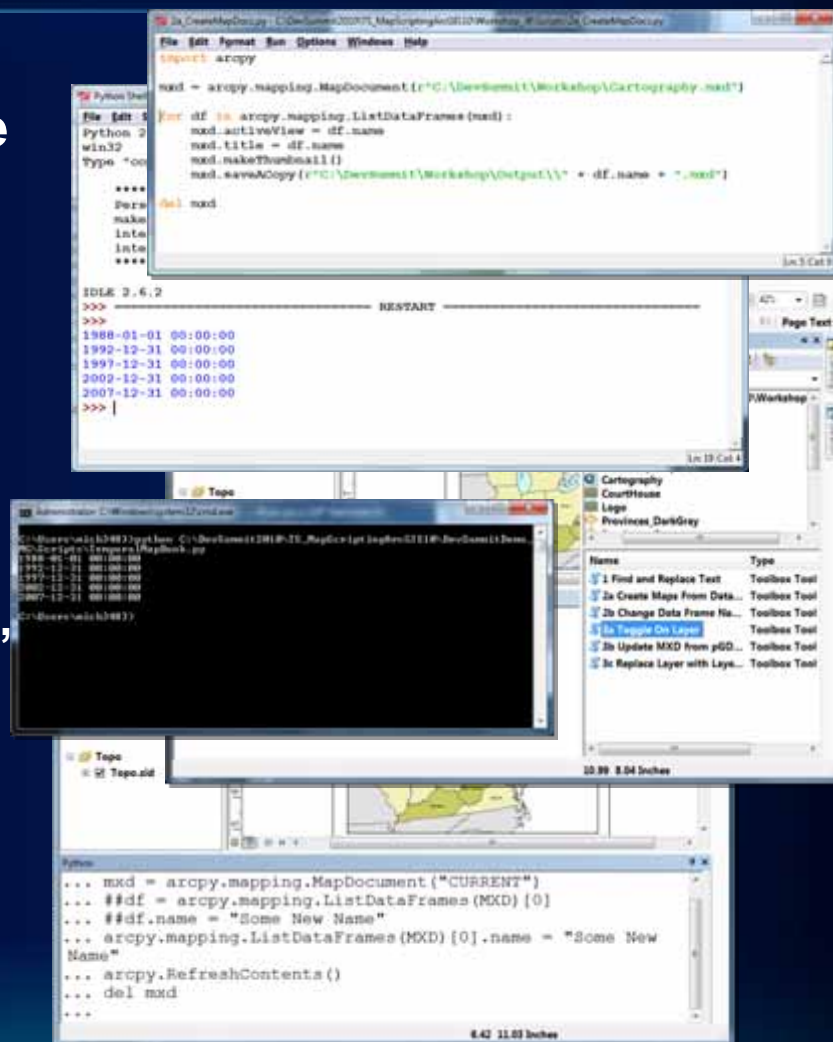
- Export data driven pages to multipage PDF
- Map book that includes index pages using Python ReportLab
- Custom thematic map application ported from AML

Updating Data Sources

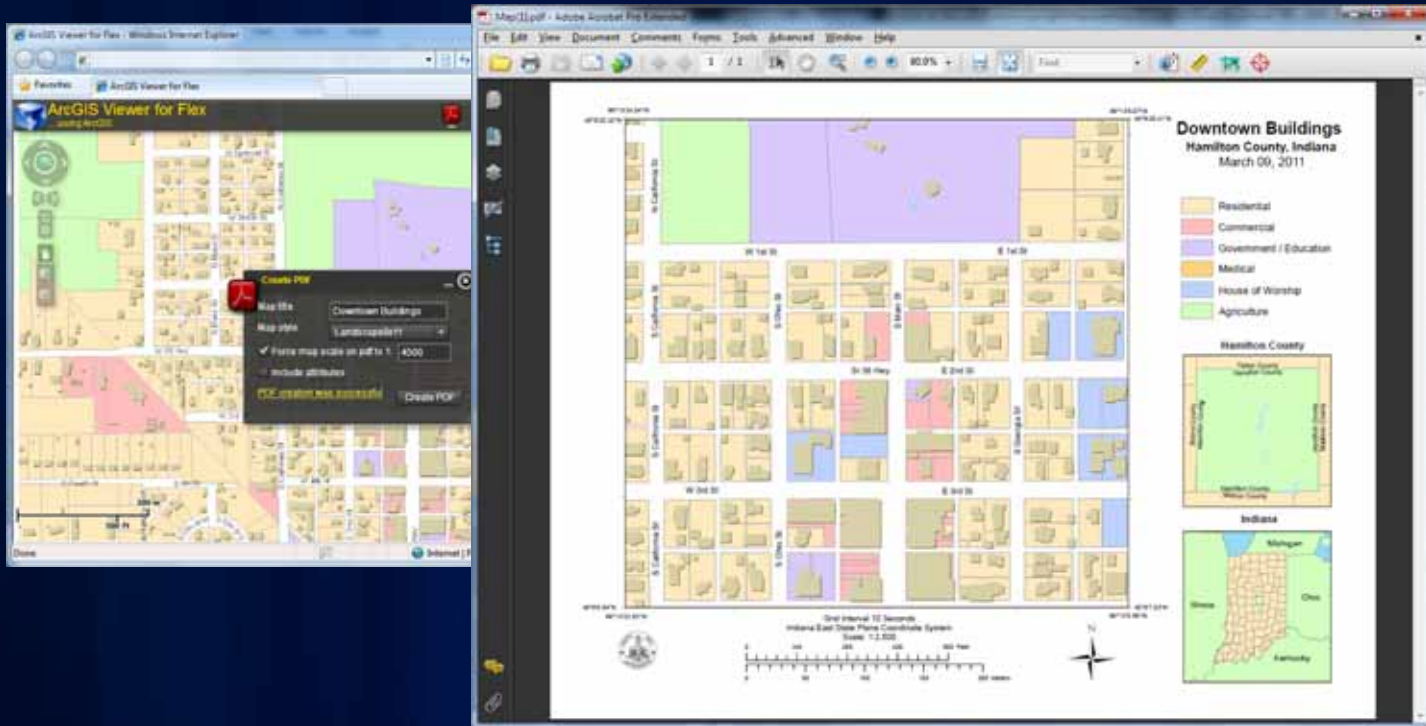
- Use `arcpy.mapping` for migrating Map Documents and Layer files to new data sources
- Fancier scripts can help mitigate migration pain: SQL syntax changes, field name changes, etc
- A complete concept document is dedicated to this topic
 - **“Updating and fixing data sources with `arcpy.mapping`”**
- Many capabilities:
 - Update all layers in an MXD or specific tables and layers
 - Works with all file and GDB types
 - Update joins and relates
 - Migrate from different workspace types
 - Etc

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
 - Standalone script - from IDE, from the command line, or as a scheduled task
 - Python Window
 - Run as a GP Service in ArcGIS Server



Demonstration: arcpy.mapping on ArcGIS Server



- GP Services
- Exporting a Map Layout to PDF from the server

Resources available

- **Desktop\Web help**
 - **Geoprocessing → The ArcPy site package → Mapping module**
 - **<http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.htm>**
 - **Alphabetical lists of classes and functions**
 - **Detailed discussions**
 - **Multiple sample scripts for each class and function topic**
- **ArcGIS Resource Center**
 - **<http://resources.esri.com/content/geoprocessing>**
 - **Download sample script tools from the Model and Script Tool Gallery**
 - **Watch video demonstrations**
 - **Monitor user forums and blog discussions**

arcpy.mapping 10.1 Road Ahead

- **Symbology**
 - automate layer symbology (renderer) properties
 - Feature Layers: graduated colors, graduated symbols, unique values
 - Raster Layers: raster classified
- **Export Report**
 - automate the generation of reports without having to open ArcMap.
- **Layer time**
 - access a layer's time properties
 - enable time on layers
- **Miscellaneous improvements to the API**
 - reading bookmarks, setting text size, setting relative paths, reading page size, etc.



esri