Applications of Python Scripting: Creating Custom Map Books in ArcGIS 10

JENNIFER SYLVESTER

GIS ANALYST II

TXDOT

TRANSPORTATION, PLANNING & PROGRAMMING DIVISION

Outline

- Identify Key Changes in Map Book Production
- VBA vs. Python – Why its Easier
- Introduce the Arcpy Mapping Module
- Introduce Data Driven Pages
- Breakdown of Custom Scripts and their Map Components
- Examples of the Finished Product
- Resources
- Contact Information

Changes in Mapbook Production

- For previous years we used lengthy VBA Code
- 246 Lines Total…
- Code was harder to customize and manipulate

Why Python Makes Things Easier….

- Less code to write
- Large well rounded module library that simplifies and shortens the amount of code written
- Dynamically typed
- Scripts can be published as geoprocessing tools and used again for future projects
- The ArcPy module builds on and is the successor to the arcgisscripting module which could be used in previous versions of Arc

The ArcPy Mapping Module

- ArcPy provides access to:
  - Geoprocessing Tools
  - Functions
  - Classes
  - Other Modules
- This access allows the user to create simple or complex workflows quickly.
- Each python script used for producing a map or map book will begin with by importing the arcpy module in line 1.

ex: import arcpy

How ArcPy is Organized

Modules, Functions, Tools, & Classes

- Module
  - Python file that includes functions and classes
  - Can be shared with others working on similar projects
  - arcpy.mapping is the module we will focus on

- Function
  - Performs a specific task
  - Incorporated into a larger programs with other functions to create custom scripts
  - All **geoprocessing tools** are **functions**
  - Examples of functions within this presentation:
    - arcpy.RefreshActiveView,
    - arcpy.mapping.ExportToPDF()
    - arcpy.mapping.ListLayourElements

- Class
  - Blueprints or framework of how something is created
  - Can be used to create objects (instances)
  - dataFrame.extent is a property of the dataFrame class in this presentation

A Few Key Elements within a Map Book Script

- Map:
  - MapDocument
- dataframe :
  - ListDataFrames, dataFrame.extent
- Cursor:
  - arcpy.SearchCursor("Layer")
- Extent:
  - row.shape.extent
- textElement:
  - Many, many options to create custom layouts
- outDir:
  - Output directory used when creating multipage pdf's
- finalPdf:
  - arcpy.mapping.PDFDocumentCreate(finalPDF_filename) - use when creating multipage pdf's
- mxdPath
  - mxdPath = r"C:\MapbookName\Mapbook.mxd"

Data Driven Pages

How it works:

- A feature layer or index layer divides the map into sections and generates a map page per index feature.
- A single layout defines the map composition and it consistent for all map pages.
- Dynamic elements of the layout will change with each page.
- Maps can be exported as individual pages or multipage PDFs.
- Title pages, indices, and any other supplemental information can be appended with a few simple lines of code.

Choose Your Best Fit…

Can be Created from:

- A grid (grid index features)
  - Uses a set scale for each map page based on the size of each grid cell.
- A route (strip map index features)
  - The rotation setting can be used to center the each strip (linear feature) from bottom to top in the center of the page.
- Map features
  - The scale will change depending on the map feature.
  - Ex: county, state, lake, city etc.

Additional Settings:

- Spatial Reference – The map will use the spatial reference of the main data frame unless otherwise state by the user in the DDP setup.
  - For More information on setting the spatial reference visit the ERSI Arc 10 Help desktop ‘Creating Data Driven Pages’
- Inset maps can also be included using extent rectangles within the DDP map set that will change dynamically for each map page

How to set up Data Driven Pages

Image Removed to reduce file size; will be shown in presentation

How to Setup Data Driven Pages

- The tools are part of the data driven pages toolset.
  - The ‘page name’, ‘page number’ and a given ‘page count’ can be added and controlled from the Data Driven Pages Toolbar.

Image Removed to reduce file size; will be shown in presentation

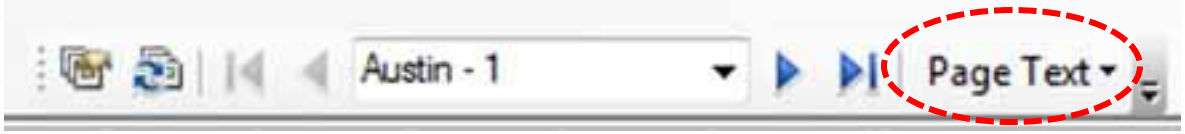
Austin - 1

Map Template Example

2010

Inset Map

- Two dynamic text elements are set with the Data Driven Pages in this example
  - The name “Austin – 1” is derived from the CITY_NAME field of the Index Grid
  - The Page number is derived from the PAGE_NUM field of the index grid.
  - Both elements are added to the page through the Data Driven Pages toolbar below:



Scale 1:44,430

Setup: Definition

- Data Frame:
  - Choose the data frame being displayed by the map
- Layer:
  - Grid, Strip, or Feature Layer for which each record and extent will produce a single map page
- Name Field:
  - Will define the dynamic text included as the 'Page Text' → 'Data Driven Page Name' element
- Sort Field:
  - Field used by the DDP toolset to loop through the layer (grid, strip or feature)
- Rotation:
  - Can be set when the feature layer used to create each map has different orientations for different map pages
- Spatial Reference:
  - Automatically takes the spatial reference of the main data frame unless otherwise noted.
- Page Number:
  - Can be set by a field from the layer used to create each map page; Will define the dynamic text included as the 'Page Text' → 'Data Driven Page Number' element
- Starting page Number:
  - Map pages can be set to begin printing as a specific number; this will leave space for titles, indices, and supplementary graphs or other information

Setup: Extent

- Best Fit
  - Use this setting for Strip or Feature Maps using a feature layer with different scales and orientations
  - Gives you the option to set your margins
- Center and Maintain Current Scale
  - Use this option when using symmetrical grids or similarly sized features with the same scale for all map pages
- Data Driven Scale
  - Can be derived from scale values within the index layer. Valid fields will be either short integer, long integer, float, or double

How to Create Your Pages…

Two options:

(1) Run from the export map option

- Must choose all pages from the 'Options' → 'Pages' Tab from File → Export Map menu
- This will give you a separate PDF page for each map in your map book

(2) Run a custom data driven pages script through the python window with customizations and appended pages

- This will create a single map book with appended pages such as title, index, and supplemental information and maps.

Option (1)

Exports Individual PDFs based on setting s from ‘Page and Print Setup’

Option (2)

Exports a Single Multi-page PDF with appended pages

Images Removed to reduce file size; will be shown in presentation

Data Driven Pages Script

Image Removed to reduce file size; will be shown in presentation

Custom Data Driven Pages Script:

Exports Pages and Appends Additional Information…

Example of the Final Data Driven Pages Custom Script Output Map

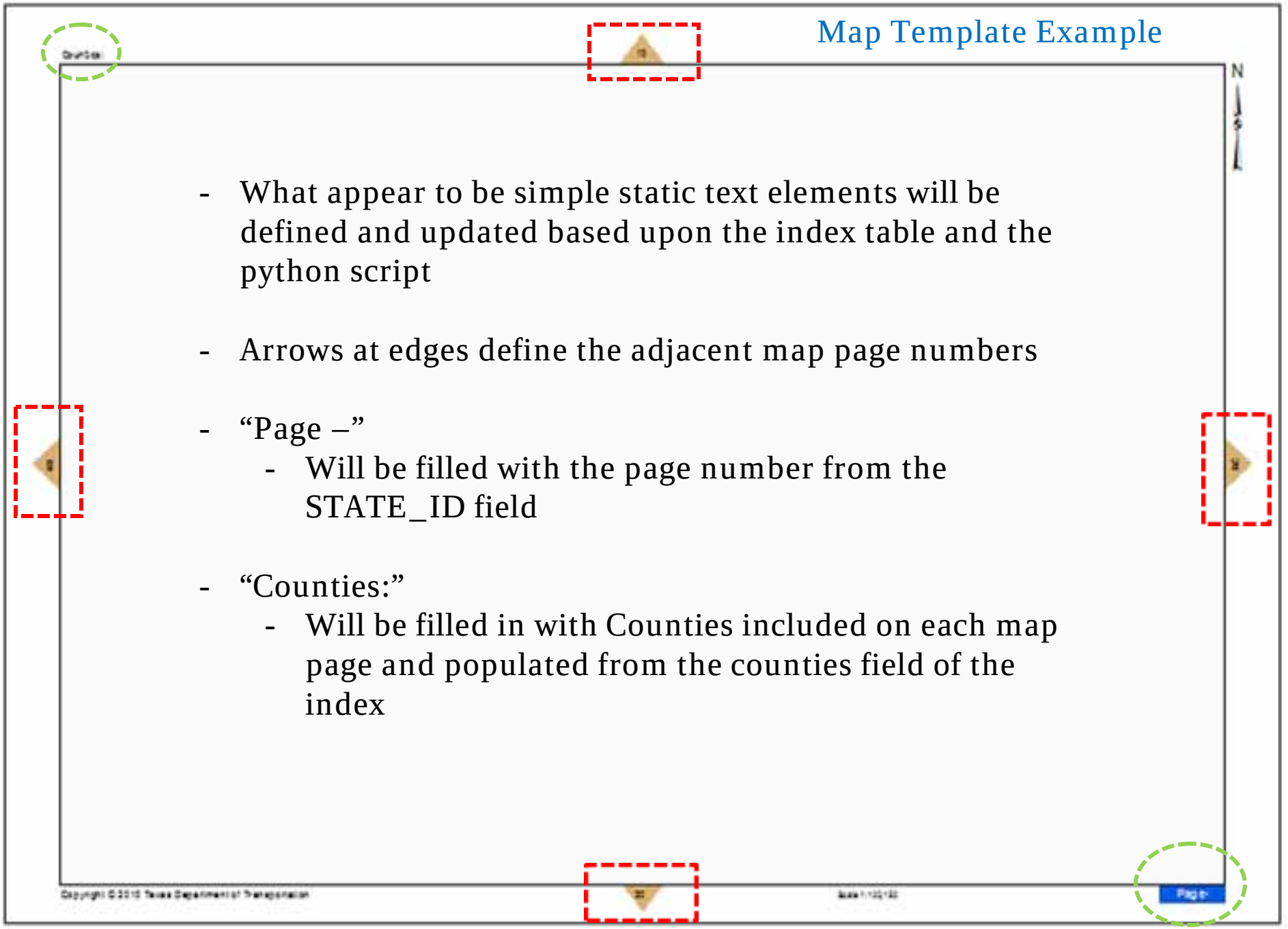
File Removed to reduce file size; will be shown in presentation

Custom Scripts Based on Map Elements

Custom Maps

- For map books with many variable text elements and customizations a simple python script will suffice.
- For the TxDOT County Mapbook a python script is used to export and update various map text elements across the page.
- This option works best if you want complete control over your customizations and style.

Map Template Example



- What appear to be simple static text elements will be defined and updated based upon the index table and the python script

- Arrows at edges define the adjacent map page numbers

- “Page –”
  - Will be filled with the page number from the STATE_ID field

- “Counties:”
  - Will be filled in with Counties included on each map page and populated from the counties field of the index

Setup: Dynamic Text Elements

Image Removed to reduce file size; will be shown in presentation

- Adjacent page numbers are set by the North, South, East and West fields
- Page Number is set by a static text element + the STATE_ID field.
- The Counties shown per page is set a static text element “Counties:” + the ‘Counties’ field.

Custom Map Layouts: Setting Up Your Map

1) Fixed extents do not work, you will get an extent error when running your code.
2) Center your text elements so they do not move.

- Page numbers will be left aligned and padded with spaces

3) Remember your anchor points

```
textElement.text = " Page-" + str.center(str(row.STATE_ID),3)
```

- Can use the str.center(textelement) function to pad your page numbers to get even spacing for each page number whether it is 1,2,3 or more characters
  - Example: 'Page- 1' 'Page- 11' 'Page-111'
    - All page numbers have the same text length independent of their own length by using the str.center(textelement) code

4) If your data frame is the same size as your grid then you will not have to set a scale in your python code, if it is not an exact fit you will add a scale reference in the script before the refresh active view and before the pdf is exported.

Text Elements

Examples from the TxDOT County Map Book:

North (adjacent page element)

- Text Element determined by the 'North' Field of the index grid.
- The same theory applies to South, East and West adjacent page elements

```
if textElement.name == "North":
    textElement.text =  row.North
```

Counties

- Fixed Text "Counties:" + Variable Text Element determined by the 'Counties' Field of the index grid.

```
if textElement.name == "CountyName":
    textElement.text = " Counties: " + row.Counties
```

Page

- Fixed Text "Page –" + variable text element from 'State_ID' field of the index grid.

```
if textElement.name == "Page":
    textElement.text = " Page-" + str.center(str(row.STATE_ID),3)
```

For Pages with no adjacent map

- Text element is set to "-" by a line of code in the script.

```
if row.South == 0:
    textElement.text = "-"
```

County Map Book Code

Image Removed to reduce file size; will be shown in presentation

Custom Script for the 2010 County Map Book

Creates a set of numbered map pages with custom text and layout elements.

Sample County Mapbook Example

File Removed to reduce file size; will be shown in presentation

Some Basic Syntax

- Python code is very sensitive to its indention, in fact it will not work if even one indention error exists.

- You cannot concatenate string variables to numeric variables. Must set numeric variables as strings in your code.

```
textElement.text = " Page-" + (str(row.STATE_ID))
```

- '=' signs set your variables
  - Use '=' when setting your location variables (Directories/Paths)

```
outDir = r"C:\NewMap"
```

  - Directories/Paths can also be written as:

```
outDir = "C:\\NewMap"
```

- '= =' is equivalent to 'is equal to' and Semicolons are equal to 'then'

```
if textElement.name == "West":
    textElement.text =  row.West
    if row.West == 0:
        textElement.text = "-"
```

Resources

- How to Create Custom Tools and Toolbars in Arc 10
- How too Add/Run Python Scripts in Arc 10
- ESRI Arc 10 Desktop Help
  - Data Driven Pages
  - Create Grid Index Features
  - What is ArcPy?
  - Importing ArcPy
  - Essential ArcPy Vocabulary
  - A quick Tour of ArcPy
- Python Resources
  - http://www.python.org/about/gettingstarted/
  - http://www.effbot.org/zone/librarybook-index.htm
  - http://diveintopython.org/

Files Removed to reduce file size; will be shown in presentation

Contact Information

Jennifer Sylvester
GIS Analyst II
Jennifer.Sylvester@txdot.gov
(512) 486 - 5139

or

Central TPP-GIS Account:
TPP-GIS@txdot.gov
(512) 486 - 5052

For further information and to view the data we maintain visit:

Transportation, Planning and Programming Division
http://www.txdot.gov/about_us/administration/divisions/tpp.htm

The Online County Map Book
http://www.txdot.gov/travel/county_grid_search.htm

The Statewide Planning Map
http://www.txdot.gov/apps/statewide_mapping/StatewidePlanningMap.html