WICKED

Efficient Map Book Production Using ArcGIS 10 Data Driven Pages and ArcPy

Greg Bonynge
Geospatial Extension Program
University of Rhode Island
greg@edc.uri.edu

THE UNIVERSITY OF RHODE ISLAND

v2, last updated 20110518

Overview

1. Background Information (~5 minutes)
2. Data-Driven Pages (~20 minutes)
3. ArcPy.mapping (~30 minutes)

Photo by Paul Fusco, USDA Natural Resources Conservation Service.

Partners

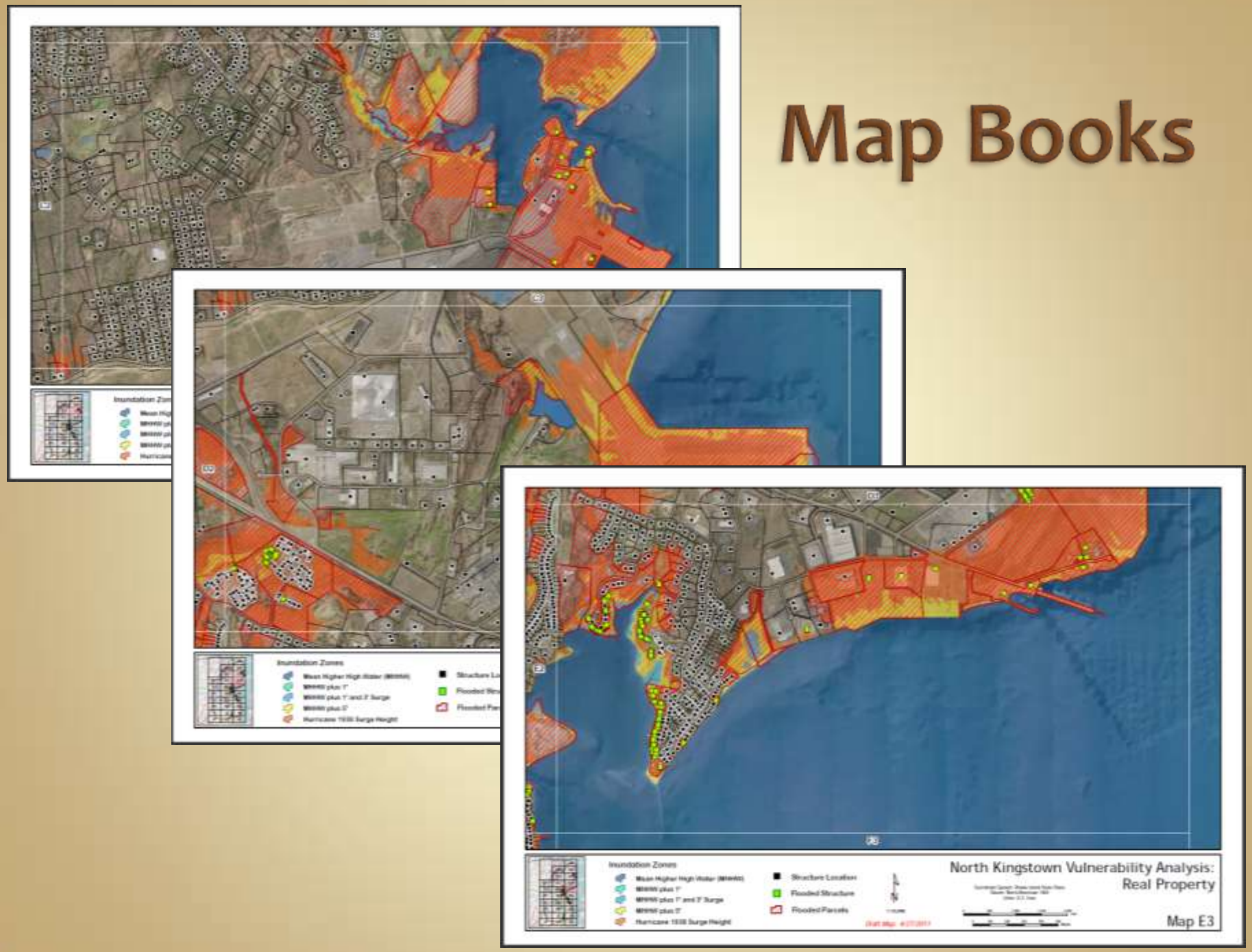
- USDA Natural Resources Conservation Service
- RI Community Resource Inventory Project
  - CICEET
  - UConn CLEAR
  - RI NEMO
- RIEDC Broadband Rhode Island

USDA NRCS
United States Department of Agriculture
Natural Resources Conservation Service

University of Connecticut

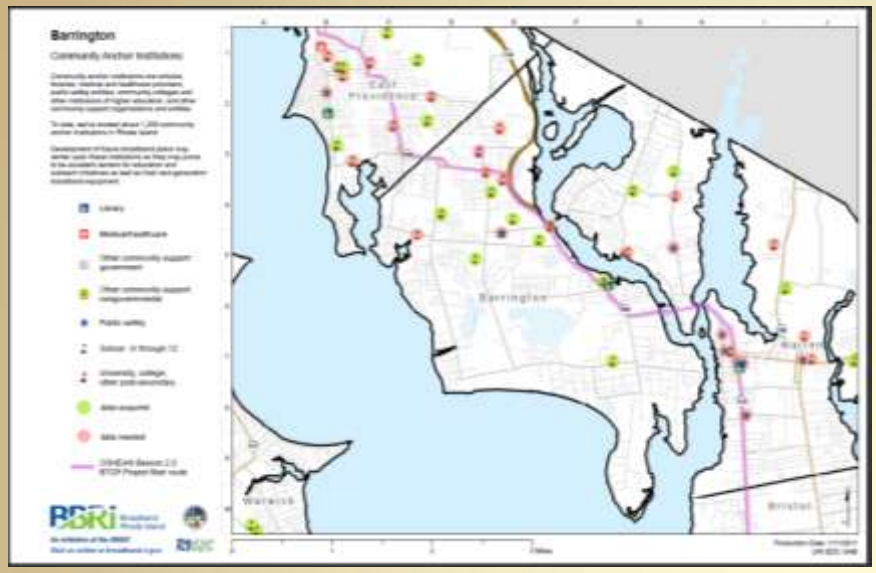
BBRI Broadband Rhode Island
An initiative of the RIEDC

Map Books


Inundation Zones
Mean Higher High Water (MHHW)
MHHW plus 1'
MHHW plus 1' and 3' Surge
MHHW plus 5'
Hurricane 1938 Surge Height
Structure Location
Flooded Structure
Flooded Parcels
North Kingstown Vulnerability Analysis:
Real Property
Map E3

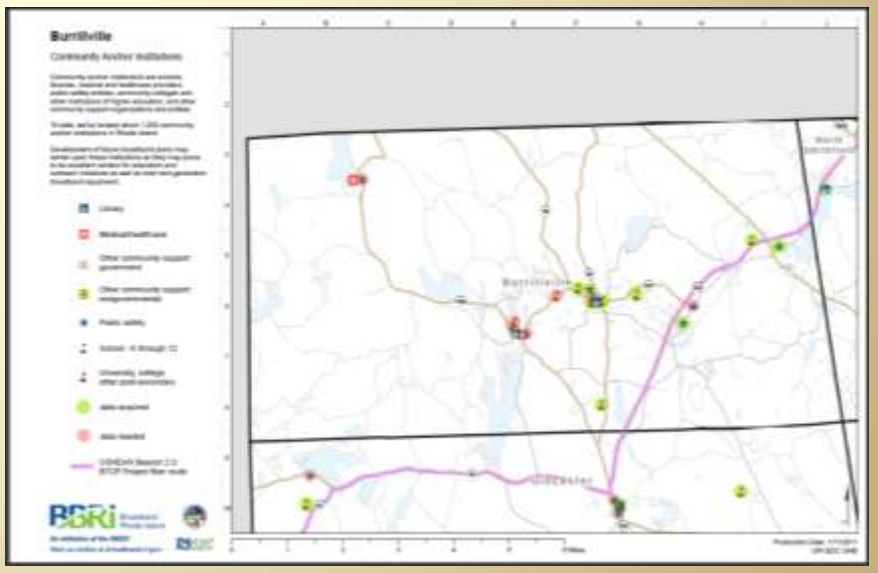
Typical Map Book Styles

Single map, many geographic areas



Tools

- Mandatory: ArcGIS Desktop Data Driven Pages
- Optional: ArcPy

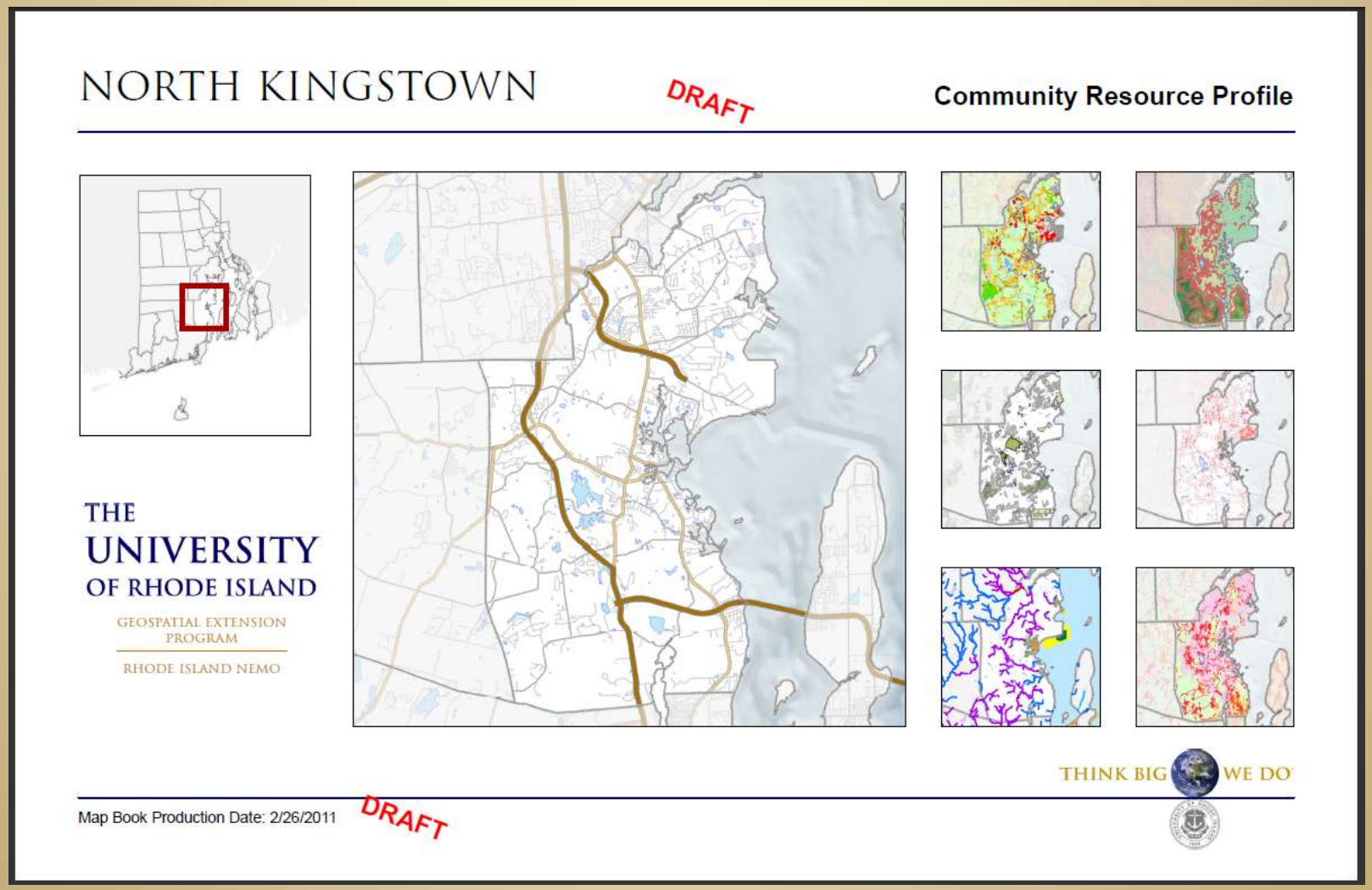


Typical Map Book Styles

Many maps, single geographic area

Tools

Mandatory: ArcGIS Desktop Data Driven Pages, ArcPy



The Challenge

25 maps + 1 municipality = 1 map book

… and repeat 39 times

The Challenge

25 maps + 1 municipality = 1 map book

… how to export 25 individual maps clipped to a single custom extent?

… how to efficiently maintain 25 MXDs?

The Challenge

25 maps + 1 municipality = 1 map book

… how to auto-cycle through all 39 municipalities for all the maps?

The Challenge

25 maps + 1 municipality = 1 map book

… how to assemble the final map book?

The Plan

1. **Create a template MXD using ArcMap.**
2. **Duplicate and update the template MXD using ArcMap.**
3. **Export and compile custom map book using ArcPy.**
4. **Maintain the MXDs *en masse* using ArcPy.**

The Plan

1. **Create a template MXD using ArcMap.**
2. Duplicate and update the template using ArcMap.
3. Export and compile custom map book using ArcPy.
4. Maintain the MXDs *en masse* using ArcPy.

Introducing Data Driven Pages

- Map Book Production!
  - Replaces the map book developer sample
- Out-of-the-box capabilities:
  - Automatically generated subsets of a larger map
  - Dynamically-generated text
  - Dynamic page numbers

Demo

Review: Data Driven Page Tips

- Try to predict how template will be used.
- Use a multi-part feature class when necessary to correctly subset the map.
- Take advantage of page definition queries to highlight your area of interest.
- Add single word, easy-to-understand labels to all layout elements if planning to use ArcPy.

Data Driven Page Resources

- Esri Help Documentation
- Esri Mapping Center
- Esri Blogs

The Plan

1. Create a template MXD using ArcMap.
2. **Duplicate and update the template using ArcMap.**
3. Export and compile custom map book using ArcPy.
4. Maintain the MXDs *en masse* using ArcPy.

Working With the Template

- Duplicate, rename, reuse template as needed.
- Keep it simple
  - Avoid adding new layout elements.
  - Initially avoid moving layout elements.

The Plan

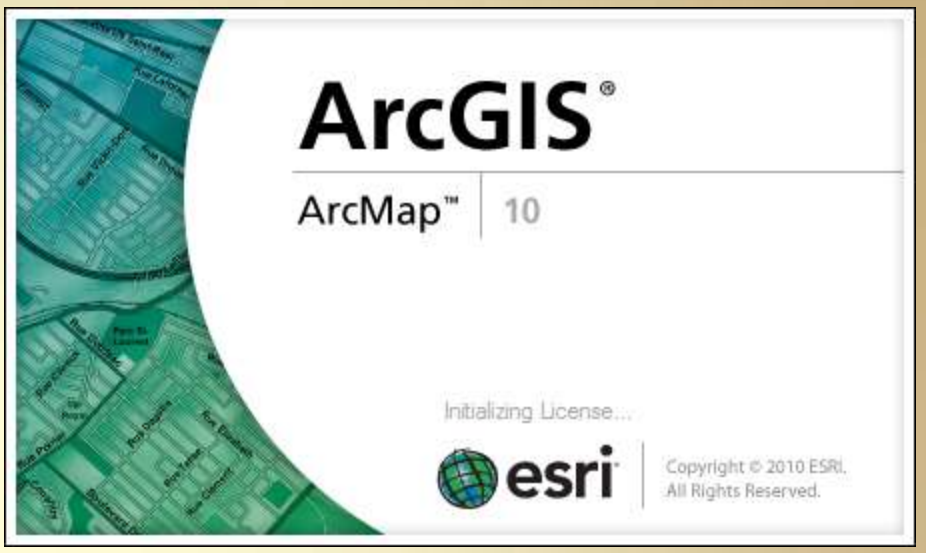
1. Create a template MXD using ArcMap.
2. Duplicate and update the template using ArcMap.
3. **Export and compile custom map book using ArcPy.**
4. Maintain the MXDs *en masse* using ArcPy.

The Challenge

25 maps + 1 municipality = 1 map book

… and repeat 39 times

ArcPy



- New With ArcGIS 10
- Automates tasks in ArcGIS Desktop & Server
- Can (theoretically) access all ArcGIS geoprocessing tools.

ArcPy.mapping

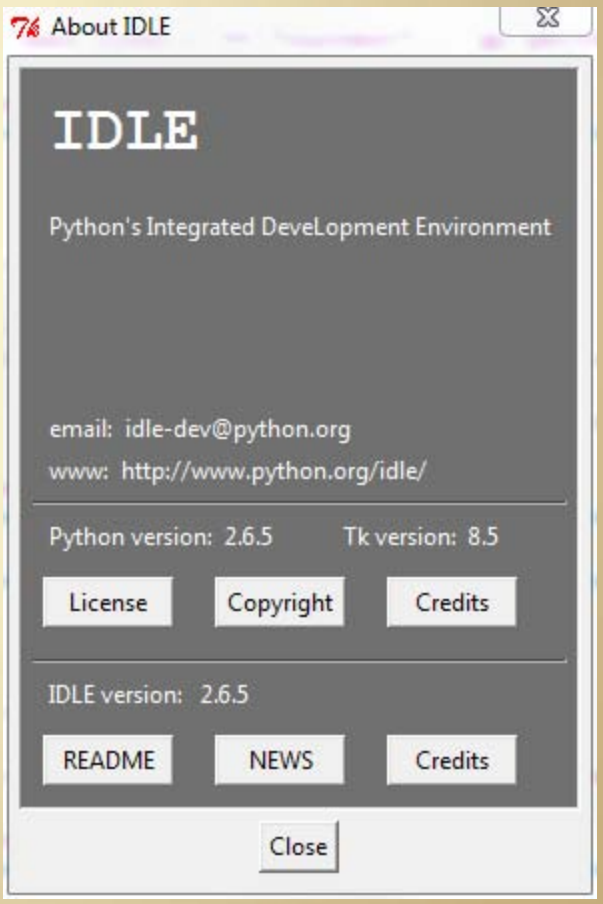
- A component (a.k.a. module) of the ArcPy package.

- Scripting possibilities:
  - Exporting Map Books
    - Excels at PDF document management!
  - Printing Map Books
  - Move any map elements within a layout
  - Change selected properties

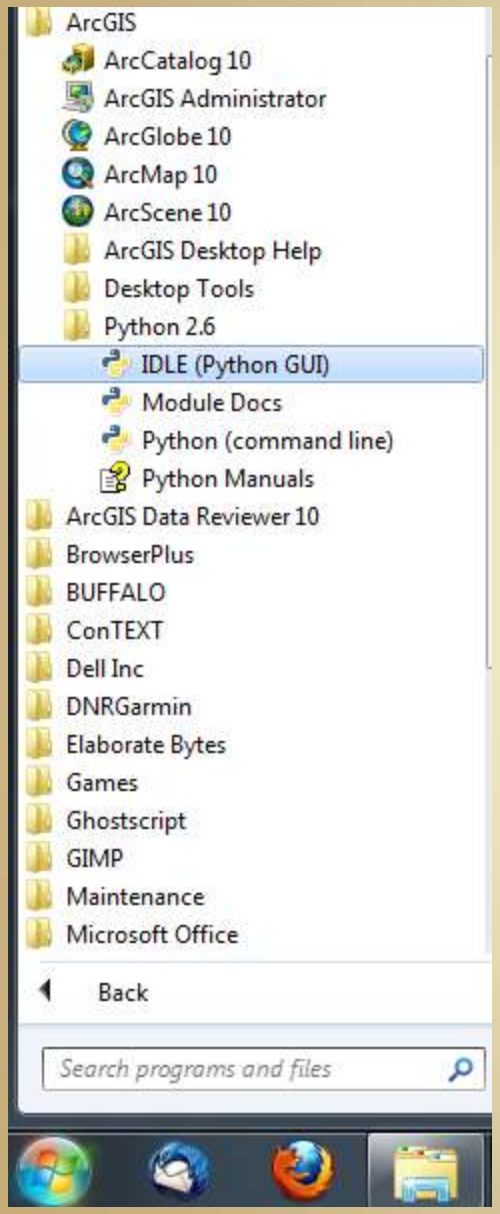
ArcPy Tools

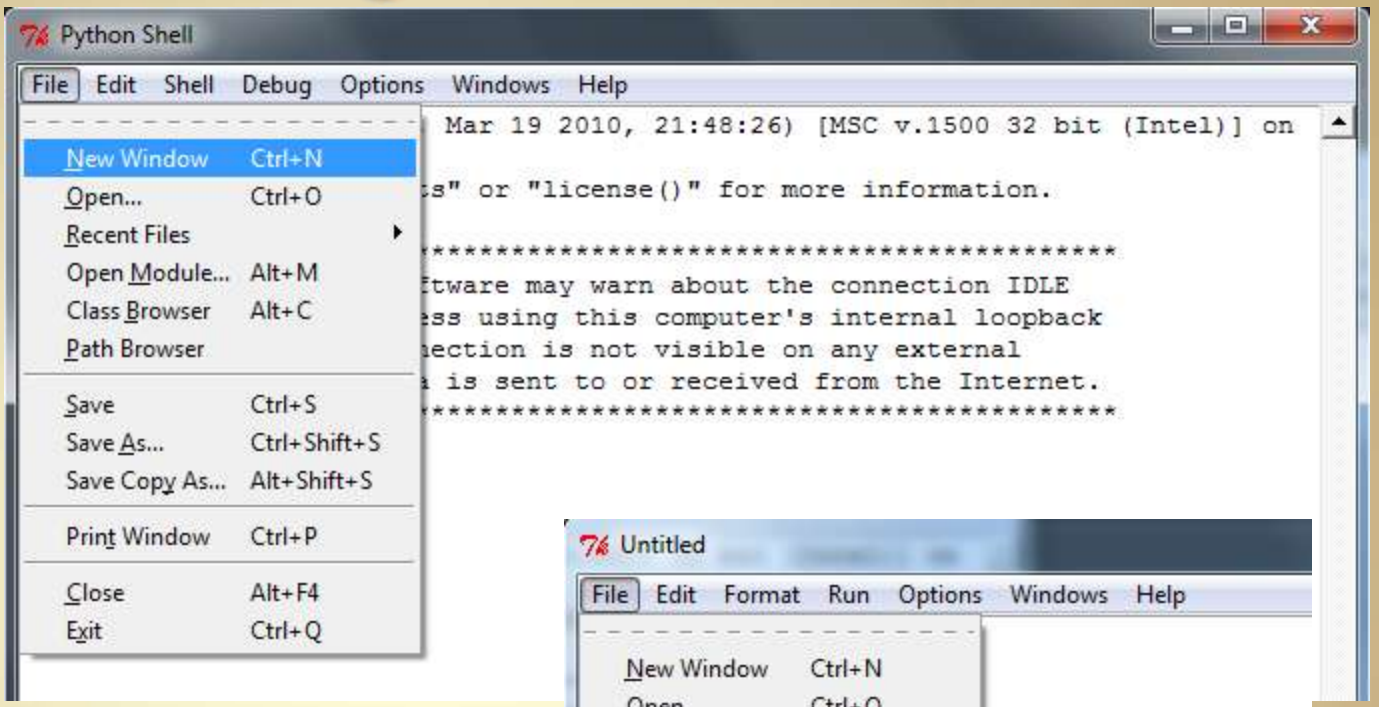
- *Everything you need is provided with the ArcGIS 10 Desktop install.*

- Other Python editors
  - ConText
  - Komodo IDE
  - Notepad
  - Notepad++
  - UltraEdit
  - More listed at: http://wiki.python.org

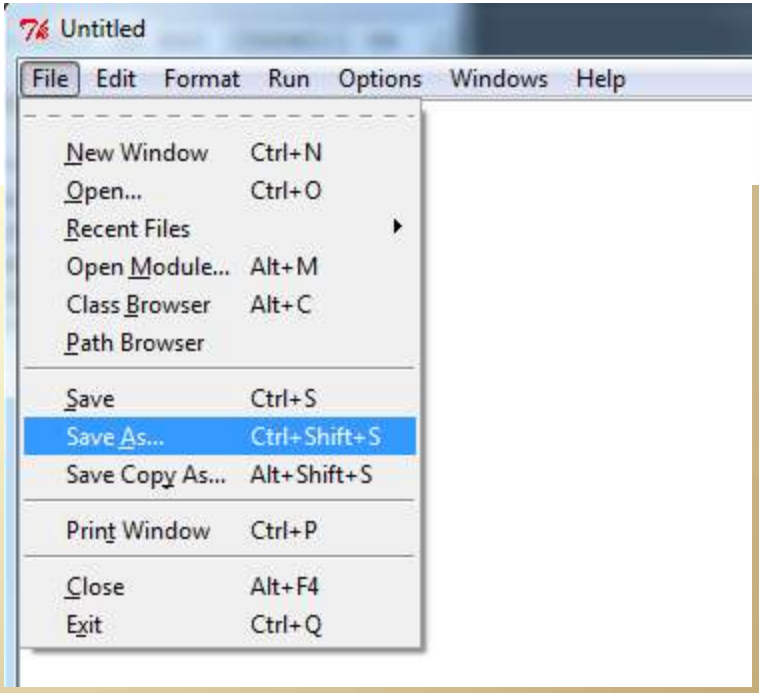


Getting Started





Create a new script…



Getting Started: Blank Script

Getting Started: The First Line

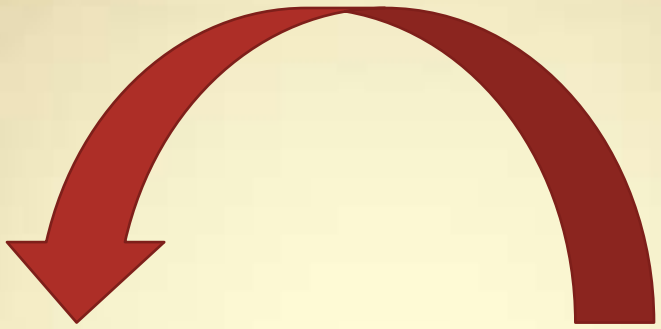
```
import arcpy, os, sys
```

Getting Started: One More Line

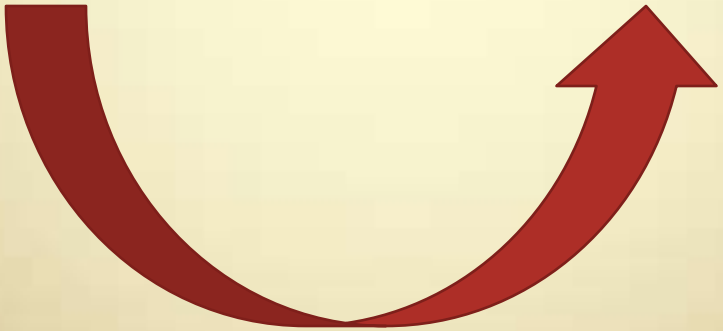
```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])
```

Our first variable! More to come…

The Challenge



25 maps + 1 municipality = 1 map book



The Challenge: Breaking it Down

Start simple…

How to export single, standalone map?

Export a Single Map

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mxd = arcpy.mapping.MapDocument(path + r"/MyMap.mxd")

arcpy.mapping.ExportToPDF(mxd, path + r"/MySingleMap.pdf")
```

The Challenge: Breaking it Down

Keep it simple…

How to export a standard map book?

1 map + 39 municipalities = 1 map book

Export a Standard Map Book

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mxd = arcpy.mapping.MapDocument(path + r"/MyMap.mxd")

mxd.dataDrivenPages.exportToPDF( r"/MyMapBook.pdf")
```

The Challenge: Breaking it Down

Take the next small step…

25 maps + 1 municipality = 1 map book

… how to export a single map book page for a single extent?

Select & Export a Single Map Book Page

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mxd = arcpy.mapping.MapDocument(path + r"/MyMap.mxd")

pageName = "Providence"

ddp = mxd.dataDrivenPages
pageID = ddp.getPageIDFromName(pageName)
ddp.currentPageID = pageID

arcpy.mapping.ExportToPDF(mxd, path + r"/MySingleMap.pdf")
```

The Challenge: Breaking it Down

Take the next small step…

25 maps + 1 municipality = 1 map book

… how to export 25 individual maps clipped to a single custom extent?

Export Multiple Maps: Set the Scene

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageName = "Providence"

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]
```

Export Multiple Maps: Create the Loop

```
mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")

        ddp = mxd.dataDrivenPages
        pageID = ddp.getPageIDFromName(pageName)
        ddp.currentPageID = pageID

        arcpy.mapping.ExportToPDF(mxd, path + r"/" + str(pageName) + "_" +
str(mapName) + ".pdf")
```

Export Multiple Maps: Putting it All Together

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageName = "Providence"

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")

        ddp = mxd.dataDrivenPages
        pageID = ddp.getPageIDFromName(pageName)
        ddp.currentPageID = pageID

        arcpy.mapping.ExportToPDF(mxd, path + r"/" + str(pageName) + "_" +
        str(mapName) + ".pdf")
```

Where do we stand now?

Using ArcPy, we've exported:

- A standard map.
- A standard map book.
- An individual map book page.
- Individual map book pages.

The Challenge: Breaking it Down

25 maps + 1 municipality = 1 map book

… how to compile these individual map book pages?

Custom Map Book: Create a Blank PDF File

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

finalPDF = arcpy.mapping.PDFDocumentCreate(mxd, path + r"/MyMapBook.pdf")
```

Custom Map Books: Append PDF Files

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

finalPDF = arcpy.mapping.PDFDocumentCreate(path + r"/MyMapBook.pdf")

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        finalPDF.appendPages(path + r"/" + str(mapName) + ".pdf")

finalPDF.saveAndClose()
```

Custom Map Books: Cleanup Crew

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

finalPDF = arcpy.mapping.PDFDocumentCreate(path + r"/MyMapBook.pdf")

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        finalPDF.appendPages(path + r"/" + str(mapName) + ".pdf")
        os.remove(path + r"/" + str(mapName) + ".pdf")

finalPDF.saveAndClose()
```

Custom Map Book: Putting it All Together

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageName = "Providence"

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

finalPDF = arcpy.mapping.PDFDocumentCreate(path + r"/MyMapBook.pdf")

for mapName in mapNameList:

        mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")

        ddp = mxd.dataDrivenPages
        pageID = ddp.getPageIDFromName(pageName)
        ddp.currentPageID = pageID

        arcpy.mapping.ExportToPDF(mxd, path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

        finalPDF.appendPages(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

        os.remove(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

finalPDF.saveAndClose()
```

The Challenge: Breaking it Down

25 maps + 1 municipality = 1 map book

… how to automatically cycle through all municipalities for all maps?

Export Many Map Books: Another Loop!

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageNameList = ["Providence", "Pawtucket", "North Providence"]

for pageName in pageNameList:

        finalPDF = arcpy.mapping.PDFDocumentCreate(mxd, path +
        r"/finalMapBook_" + str(pageName) + ".pdf")

        (then insert the loop that exports individual maps…)

        finalPDF.saveAndClose
```

Export Many Map Books: Loop-the-Loop!

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageNameList = ["Providence", "Pawtucket", "North Providence"]

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for pageName in pageNameList:

        finalPDF = arcpy.mapping.PDFDocumentCreate(mxd, path +  r"/finalMapBook_" + str(pageName) + ".pdf")

        for mapName in mapNameList:

                mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")

                ddp = mxd.dataDrivenPages
                pageID = ddp.getPageIDFromName(pageName)
                ddp.currentPageID = pageID

                arcpy.mapping.ExportToPDF(mxd, path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

                finalPDF.appendPages(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

                 os.remove(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")

        finalPDF.saveAndClose()
```

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

pageNameList = ["Providence", "Pawtucket", "North Providence"]

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for pageName in pageNameList:
    finalPDF = arcpy.mapping.PDFDocumentCreate(path + r"/finalMapBook_" + str(pageName) + ".pdf")
    for mapName in mapNameList:
        mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")
        ddp = mxd.dataDrivenPages
        pageID = ddp.getPageIDFromName(pageName)
        ddp.currentPageID = pageID
        arcpy.mapping.ExportToPDF(mxd, path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")
        finalPDF.appendPages(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")
        os.remove(path + r"/" + str(pageName) + "_" + str(mapName) + ".pdf")
    finalPDF.saveAndClose()
```

Where do we stand now?

We've exported:

- A standard map.
- A standard map book.
- An individual map book page.
- A group of individual map book pages.
- A map book constituted of individual map book pages.

And we've looped the entire process!

The Plan

1. Create a template MXD using ArcMap.
2. Duplicate and update the template using ArcMap.
3. Export and compile custom map book using ArcPy.
4. **Maintain the MXDs *en masse* using ArcPy.**

MUNICIPALITY NAME

Community Resource Profile

Land Use 2003-2004

DRAFT

Rhode Island Digital Atlas
http://www.edc.uri.edu/atlas

- High - Medium High Density Residential
- Medium - Medium Low Density Residential
- Low Density Residential
- Commercial or Industrial
- Transportation
- Waste Disposal
- Institutional, Developed Recreation, Cemetery
- Pasture, Idle Agriculture or Power Lines
- Cropland, Orchards and Nurseries
- Forested wetlands, Forest and Brushland
- Sandy/Beaches, Quarries and Outcrop
- Water and Sewage Treatment Facilities
- Vacant Land (Urban) and Transitional Areas
- Water
- Wetland

THE UNIVERSITY OF RHODE ISLAND

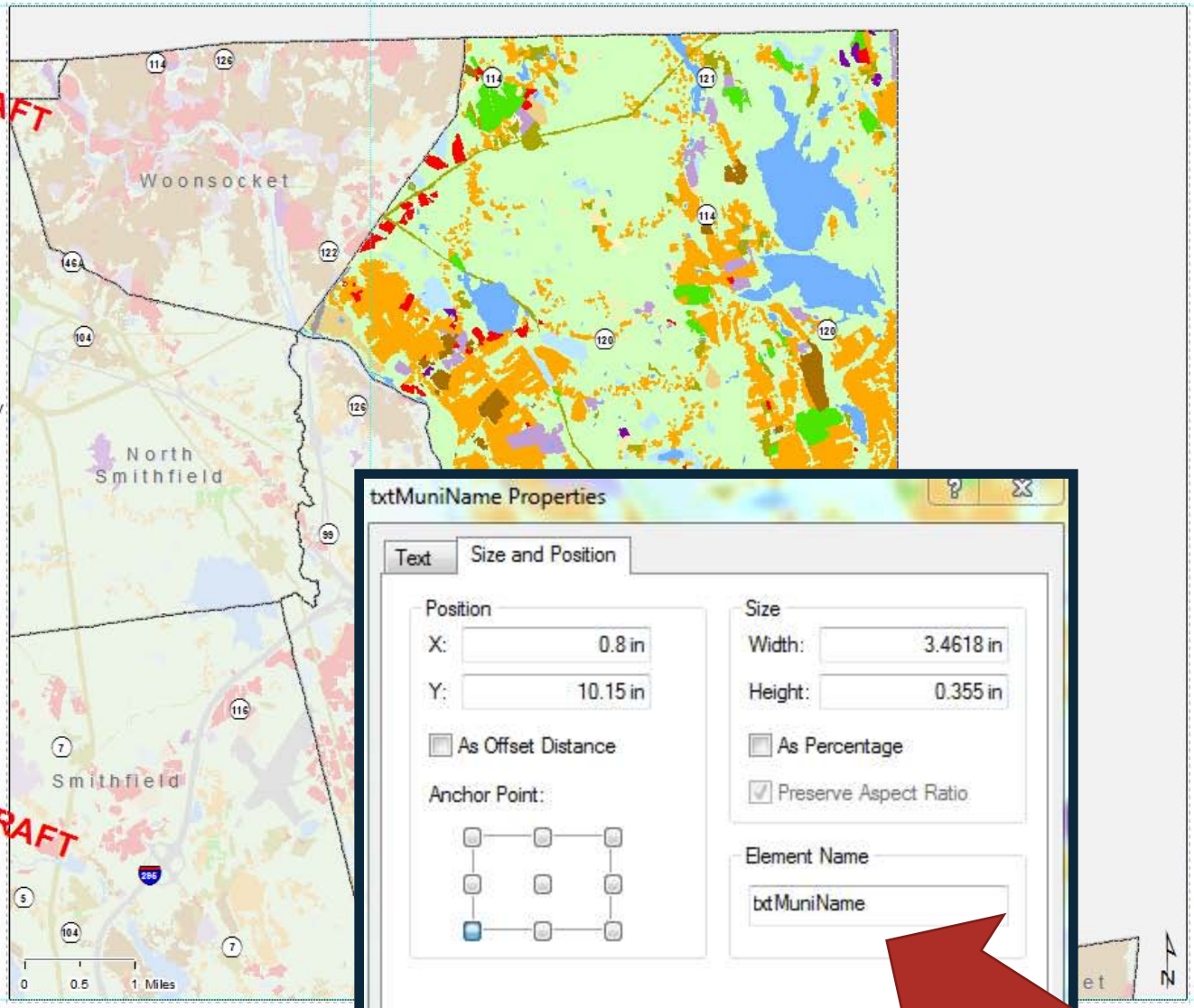
GEOSPATIAL EXTENSION PROGRAM

RHODE ISLAND NEMO

DRAFT

Note: This is not a survey. Map production date is NOT indicative of the age of the underlying data. Please review this profile's accompanying guide for more information.

Date Produced: 5/16/2011
URI EDC GHB



txtMuniName Properties

Text | Size and Position

Position
X: 0.8 in
Y: 10.15 in
As Offset Distance
Anchor Point:

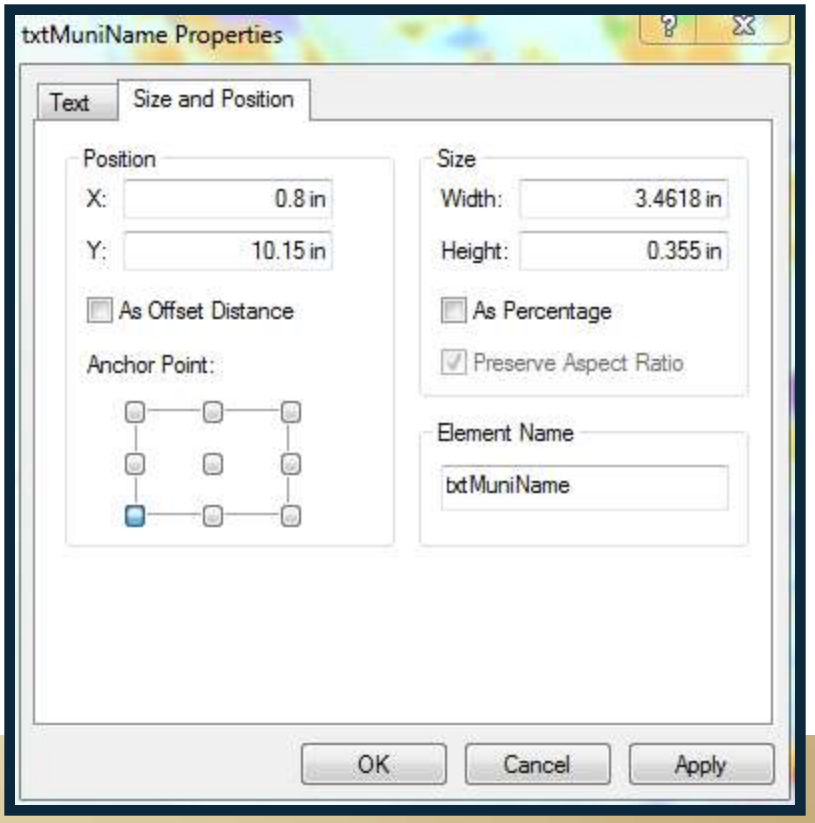
Size
Width: 3.4618 in
Height: 0.355 in
As Percentage
Preserve Aspect Ratio

Element Name
txtMuniName

OK | Cancel | Apply

Adjusting Layout Element Properties

```
elm = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "txtMuniName")[0]

elm.elementPositionX, elm.elementPositionY = 0.8, 10.15

del elm
```



Moving Layout Elements (Note: this is a new script.)

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        mxd = arcpy.mapping.MapDocument(path + r"/" + str(mapName) + ".mxd")

#move title of map book page
        elm = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "txtMuniName")[0]
        elm.elementPositionX, elm.elementPositionY = 0.8, 10.15
        del elm

        #move and resize dataframe
        df = arcpy.mapping.ListDataFrames(mxd, "Layers") [0]
        df.elementPositionX, df.elementPositionY = 4.36, 0.5
        df.elementHeight, df.elementWidth = 10, 11.84
        del df

        mxd.save()
```

More ArcPy.mapping Possibilities

- Adjust PDF export properties.
- Print maps and map books.
- Add/remove layer files to/from data frames.
- Update layer symbology (using a separate layer file).
- List broken data sources; update data sources.
- Update MXD metadata.
- More planned for ArcGIS 10.1…

Review: ArcPy Tips

- Start small & simple!
- Careful of your indents.
- Add comments to your scripts.
- If your python script crashes, you may need to kill the pythonwin process using Windows Task Manager.

Review: ArcPy Tips

- Start small & simple!
- Careful of your indents.
- Add comments to your scripts.
- If your python script crashes, you may need to kill the pythonwin process using Windows Task Manager.

Loops, Indents, & Python

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

    #move map title
    elm = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "txtMuniName")[0]
    elm.elementPositionX, elm.elementPositionY = 0.8, 10.15
    del elm

    #move and resize dataframe
    df = arcpy.mapping.ListDataFrames(mxd, "Layers") [0]
    df.elementPositionX, df.elementPositionY = 4.36, 0.5
    df.elementHeight, df.elementWidth = 10, 11.84
    del df

    mxd.save()
```

Loops, Indents, & Python

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mapNameList = ["MyMap[illegible]MyMap2", "MyMap3[illegible]

for mapName in ma[illegible]

#move map title
elm = arcpy.mapping.ListLay[illegible]iName")[0]
elm.elementPositionX, e[illegible]
del elm

#move and resize datafra[illegible]
df = arcpy.mapping.List[illegible]
df.elementPositionX[illegible]
df.elementHeight[illegible],
del df

mxd.save()
```

Loops, Indents, & Python

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        #move map title
        elm = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "txtMuniName")[0]
        elm.elementPositionX, elm.elementPositionY = 0.8, 10.15
        del elm

        #move and resize dataframe
        df = arcpy.mapping.ListDataFrames(mxd, "Layers") [0]
        df.elementPositionX, df.elementPositionY = 4.36, 0.5
        df.elementHeight, df.elementWidth = 10, 11.84
        del df

        mxd.save()
```

Review: ArcPy Tips

- Start small & simple!
- Careful of your indents.
- Add comments to your scripts.
- If your python script crashes, you may need to kill the pythonwin process using Windows Task Manager.

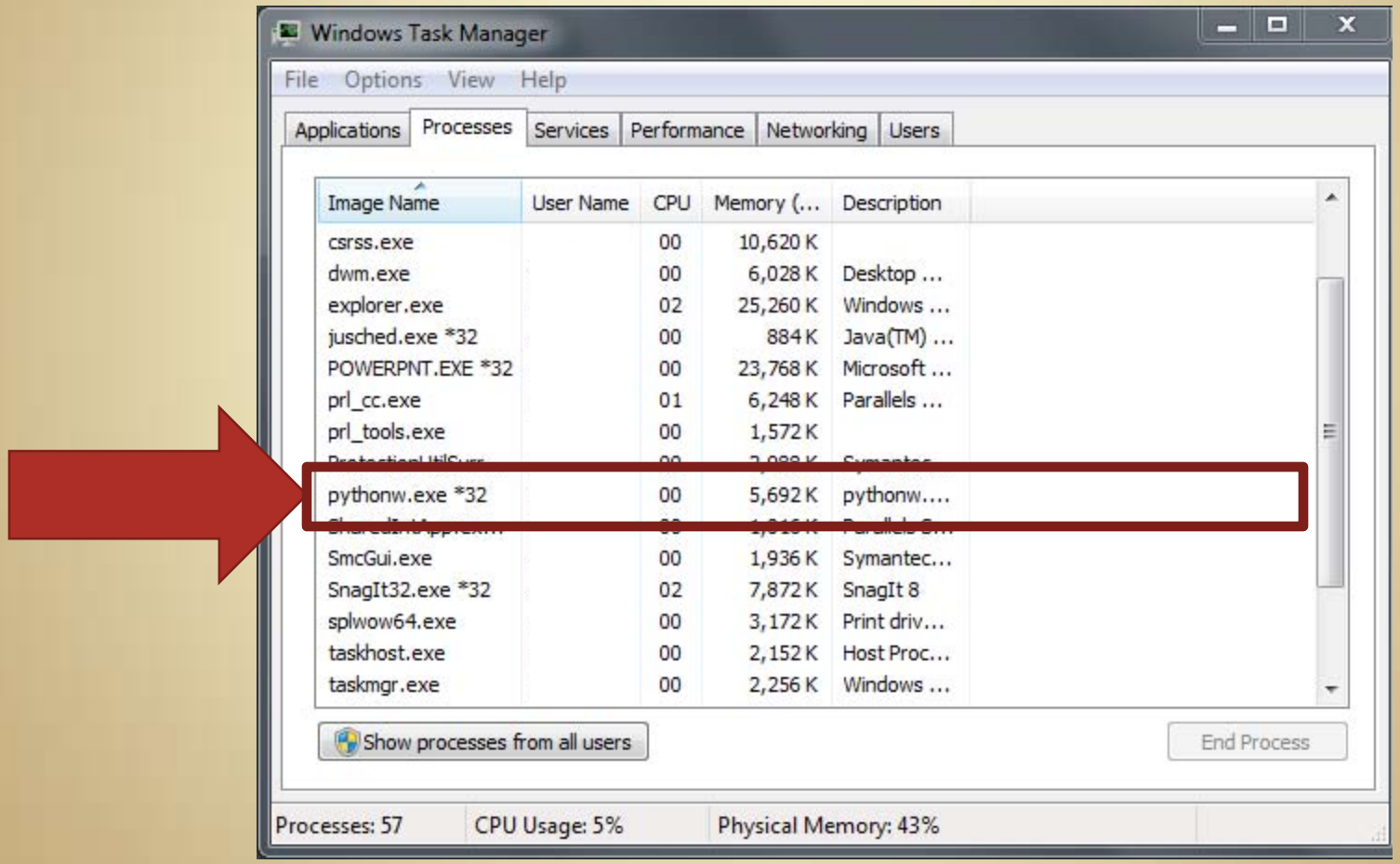
Comments & Python

```
import arcpy, os, sys

path = os.path.dirname(sys.argv[0])

mapNameList = ["MyMap1", "MyMap2", "MyMap3"]

for mapName in mapNameList:

        #move map title
        elm = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "txtMuniName")[0]
        elm.elementPositionX, elm.elementPositionY = 0.8, 10.15
        del elm

        #move and resize dataframe
        df = arcpy.mapping.ListDataFrames(mxd, "Layers") [0]
        df.elementPositionX, df.elementPositionY = 4.36, 0.5
        df.elementHeight, df.elementWidth = 10, 11.84
        del df

        mxd.save()
```

Review: ArcPy Tips

- Start small & simple!
- Careful of your indents.
- Add comments to your scripts.
- If your python script crashes, you may need to kill the pythonwin process using Windows Task Manager.

Ending the ‘pythonwin’ Process



ArcPy.mapping Resources

- ArcGIS Resource Center
  - Look for "arcpy.mapping sample script tools" by username "jbarrette"
- Esri Help Documentation
- Esri Blogs
- New 3-hour Esri Online Class:
  Python Scripting for Map Automation in ArcGIS 10
- A nice introductory Python talk:

  http://www.youtube.com/user/GoogleDevelopers

  (Look for "Google I/O 2011: Python@Google")

THE
UNIVERSITY
OF RHODE ISLAND

Greg Bonynge
Geospatial Extension Program
University of Rhode Island
greg@edc.uri.edu