Exercise 5: Map Automation using ArcPy Mapping

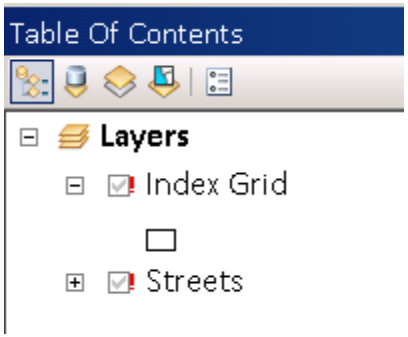
Your task in this exercise is to create a multi-page map book of the transportation infrastructure of Atlanta, GA. This map book will be used as a printed reference for utility and transportation workers in the field. The city has been divided into a grid of sixteen zones – each of these zones is to have a smaller scale (zoomed in) page in the map book showing all highways, streets, ramps, and trails with labels.

Because the creation of sixteen individual map pages would be time consuming, use the ArcPy Mapping module and data driven pages functionality to create this map book in PDF format.

Fix broken sources

The first thing that is needed for the map book is a title page that shows an overview map of the pages contained in the book. This map has already been created.

- Navigate to folder …Python_ArcGIS\exercises\exercise5_mapping
- Double-click TitlePage.mxd to open it in ArcMap, and examine the data layers that are available in the table of contents ↓
- **Whoops!** After this map document was created, a member of the GIS team converted all data from shapefiles to file geodatabase feature classes. Since the source shapefiles for the layers in this map were deleted, the table of contents shows broken links, and no data is displayed.



In the ArcMap Python window, use the ArcPy Mapping replaceWorkspaces function to change the data source of the layers from the old shapefile directory to the new geodatabase.

- Import the os module

- Create a map document object from the current map

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

- Use the replaceWorkspaces function of the map document object to correct the data source problem. Use the current map path to define the shapefile directory and geodatabase path, both of which were stored relative to the map document (alternatively, just enter the actual paths to the shapefile folder and the geodatabase).

```
>>> shppath = os.path.join(os.path.dirname(mxd.filePath), "data", "shp")
>>> gdbpath = os.path.join(os.path.dirname(mxd.filePath), "data", "gdb", "Atlanta.gdb")
>>> mxd.replaceWorkspaces(shppath,"SHAPEFILE_WORKSPACE", gdbpath, "FILEGDB_WORKSPACE")
```

- Use the arcpy.RefreshTOC() function to refresh the table of contents to remove the broken data source icons.
- Save the map document using the mxd.save() function
- Export this title page to a PDF using the ExportToPDF function

```
>>> titlepdf = os.path.join(os.path.dirname(mxd.filePath), "TitlePage.pdf")
>>> arcpy.mapping.ExportToPDF(mxd, titlepdf)
```

- Close ArcMap

To create a map book, a map document with data driven pages enabled is also needed. Data driven pages gives you the ability to generate multiple pages by taking a single map layout and iterating over a set of map extents. Any feature layer, point, line or polygon can be used to define the extents. This layer is referred to as the index layer and is the "data" that "drives the pages," thus the name. In this case, the grid that divides Atlanta into sixteen zones is the index layer. A map document has already been created that has the desired layout for the individual sheets of the map book.

- Navigate to folder …Python_ArcGIS\exercises\exercise5_mapping
- Double-click MapPages.mxd to open it in ArcMap, and examine the data layers that are available in the table of contents ↓
- **Whoops again!** The same shapefiles that were used to construct the title page map were also used in the map book pages. Perform the same steps as above to correct the problem with the data sources in this map document (but do not export to PDF quite yet).
- **Alternative –** with some clever scripting, you can fix the data sources for all your map documents in one script! It would look something like this:

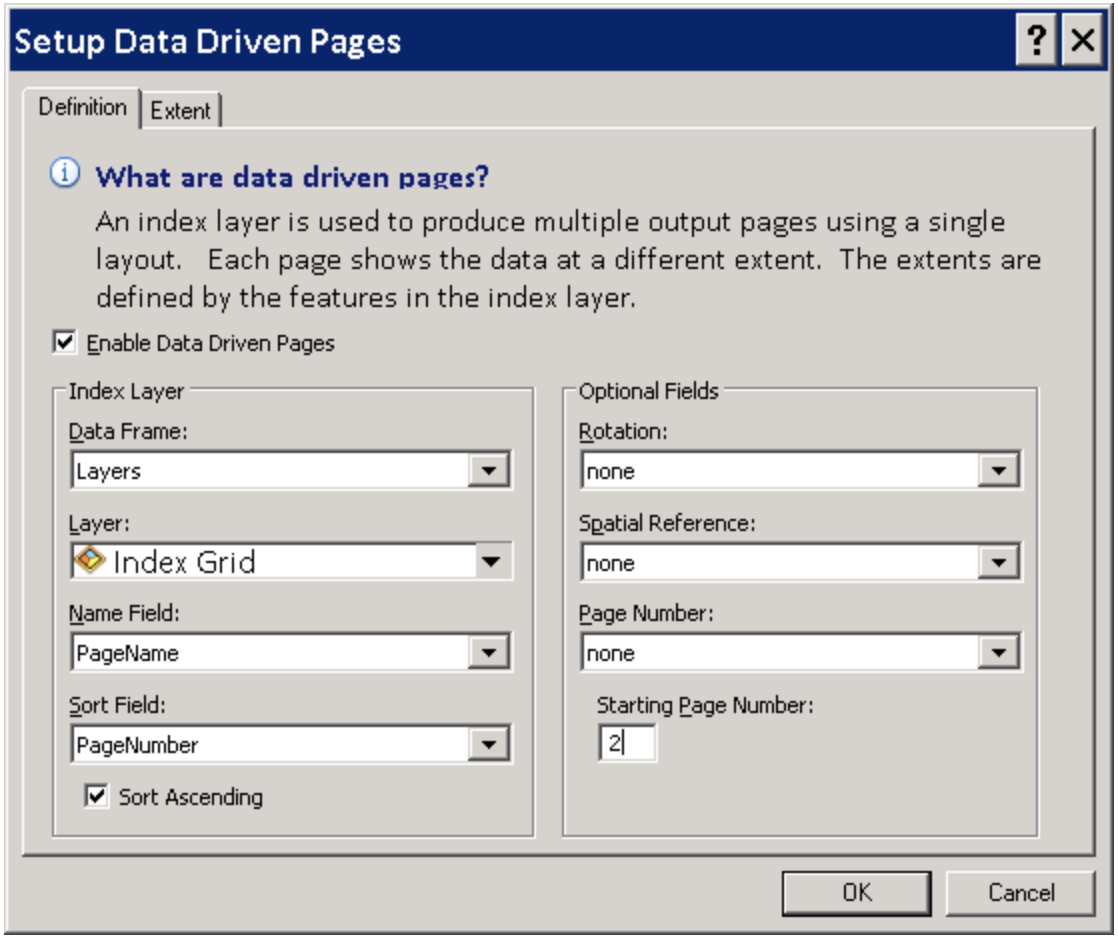
```
mapdirectory = r"C:\Maps"
olddata = r"C:\Data\shp"
newdata = r"C:\Data\gdb\File_gdb.gdb"

filelist = os.listdir(mapdirectory)
for file in filelist:
    if os.path.splitext(file)[1] == ".mxd":
        mxd = arcpy.mapping.MapDocument(os.path.join(mapdirectory, file))
        mxd.replaceWorkspaces(olddata, "SHAPEFILE_WORKSPACE", newdata, "FILEGDB_WORKSPACE")
        mxd.save()
```

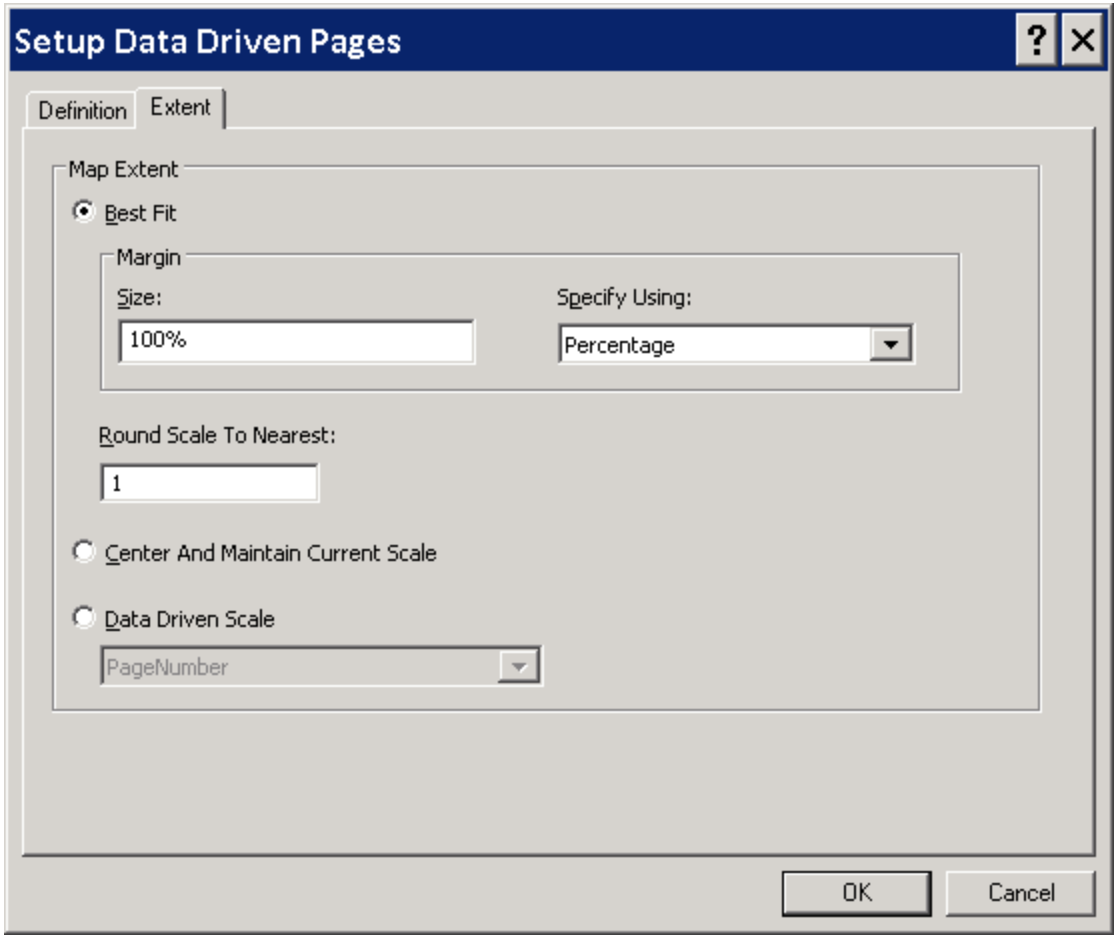
Go-go data driven pages!

- Now that the map data sources are corrected, enable data driven pages by clicking the **Data Driven Page Toolbar** button on the Layout toolbar

- Click the **Data Driven Page Setup…** button on the Data Driven Pages toolbar

- Setup the Data Driven Pages panes as shown below


Setup Data Driven Pages
Definition
Extent
What are data driven pages?
An index layer is used to produce multiple output pages using a single layout. Each page shows the data at a different extent. The extents are defined by the features in the index layer.
Enable Data Driven Pages
Index Layer
Data Frame:
Layers
Layer:
Index Grid
Name Field:
PageName
Sort Field:
PageNumber
Sort Ascending
Optional Fields
Rotation:
none
Spatial Reference:
none
Page Number:
none
Starting Page Number:
2
OK
Cancel



- Save the map document
- Create a DataDrivenPages class from the current map document object

```
>>> ddp = mxd.dataDrivenPages
```

- Use the exportToPDF function of the DataDrivenPages class to export all of the pages to a pdf

```
>>> mappagespdf = os.path.join(os.path.dirname(mxd.filePath), "MapPages.pdf")
>>> ddp.exportToPDF(mappagespdf)
```

- Now create a final copy of the map book that contains the title page and the map pages. First, create an empty PDF document using the PDFDocumentCreate function.

```
>>> mapbook = os.path.join(os.path.dirname(mxd.filePath), "MapBook.pdf")
>>> finalpdf = arcpy.mapping.PDFDocumentCreate(mapbook)
```

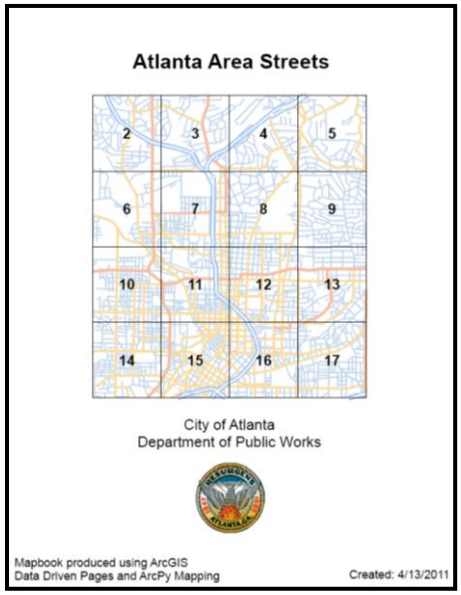
- Finally, append the title and map pages to the map book, then save and close the PDF.

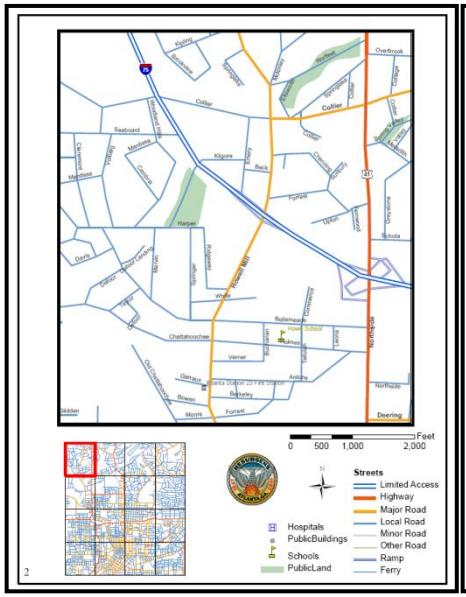
```
>>> finalpdf.appendPages('C:\\USGS11\\Python_ArcGIS\\exercises\\exercise5_mapping\\TitlePage.pdf
>>> finalpdf.appendPages('C:\\USGS11\\Python_ArcGIS\\exercises\\exercise5_mapping\\MapPages.pdf'
>>> finalpdf.saveAndClose()
```

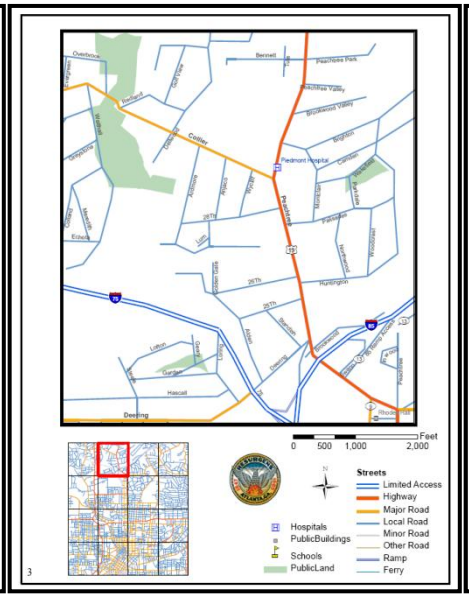
- Close ArcMap

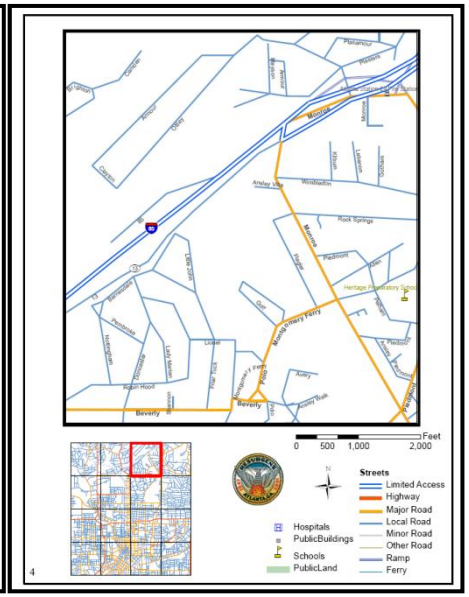
Results

The ArcPy Mapping module provides an efficient way to automate map management and authoring tasks like fixing broken data sources in maps and creating multiple page map books using data driven pages.









...