**Final Project Writeup**

This writeup is to complement the ParcelUpdate.py script for the Final Project. The goal of the script is to output a **Parcels Map** and a **Parcels Update Map**. The script combines 6 different “out-of-the-box” tools from Esri, then uses the ArcGIS mapping module to display your results on a map document and pdf. I broke out the script into two parts.

1. The **tool script** combines the: Feature To Point, Spatial Join, Make Feature Layer, Add Join, Feature Class to Feature Class, and Delete tools.
2. The **map script** combines: AddLayer, UpdateLayer, and RemoveLayer functions, and the LegendElement class.

I’ll explain everything starting with my pseudocode, dropbox data, directions, script explanation, and your results.

This project is for work, where I’ve been running these series of tools every month to upload to our website. Users can click on each zoning polygon or parcel polygon and see the zoning field. Before the script, I was running each of the tools listed above separately, and checking my results after I was finished. This script cuts the time to do that dramatically.

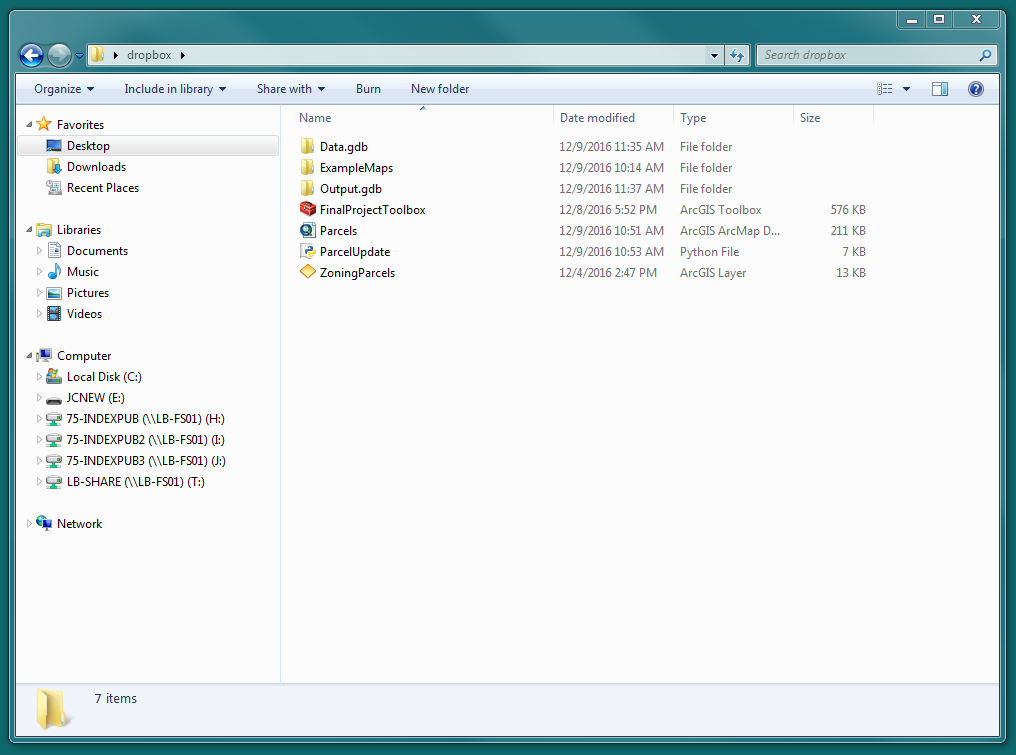
**Psuedocode**

* Make Feature Layer with Parcels
* Create Parcel Centerpoints with Parcels
* Spatial Join Parcel Centerpoints with Zoning Layer
* Add Join the Parcel Centerpoints joined with the original Parcels layer
* Create Feature Class from joined layer
* Delete excess datasets

**Ordered list of the tools and their parameters**

1. Feature To Point
   * Inside (Boolean)
2. Spatial Join
   * Join Operation (ONE TO MANY)
3. Make Feature Layer [2]
4. Add Join
   * Input Join Field
   * Join Table
   * Output Join Field
5. Feature Class to Feature Class
   * Field Map (List of fields I will keep)
6. Delete excess data

**Dropbox Data**



In the dropbox you will find:

* Data.gdb
  + Parcels (Input Variable)
  + Zoning (Input Variable)
* Output.gdb
* ParcelUpdate.py (the Script)
* ZoningParcels.lyr (“sourceLayer” in the UpdateLayer function)
* Parcels.mxd (for the mapping module “mapDocument1”)
* ExampleMaps Folder Parcels Map- example and ParcelsUpdate Map- example (these are what your output PDFs should look like)
* FinalProjectToolbox.tbx
  + ParcelUpdate (Model explained at the end of the write-up)

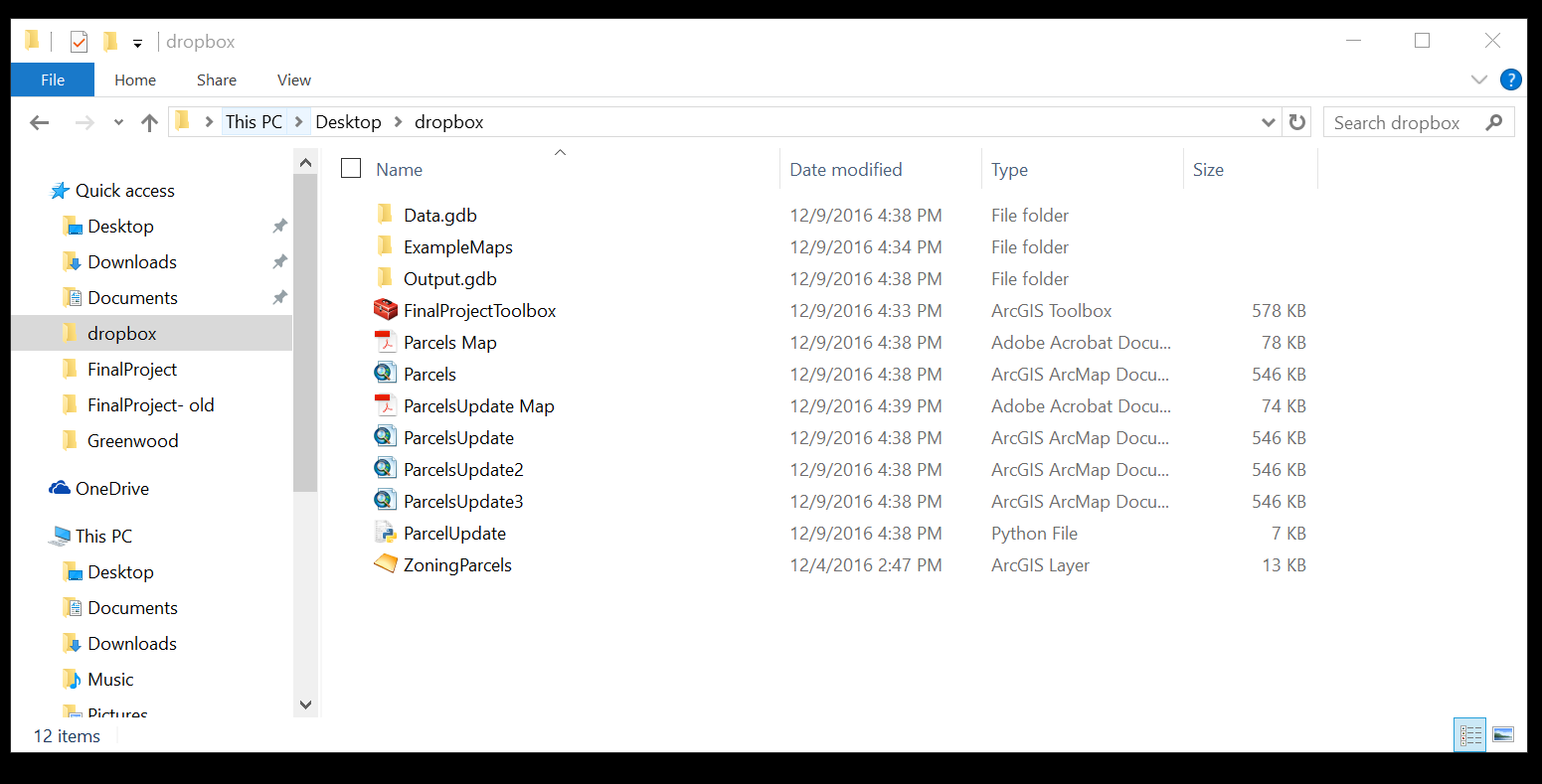
**Directions**

All you should need to do to successfully run the scriptis change the environment workspace at the front of the script to the folder with the dropbox content.

**You also need to re-source and save the Parcels.mxd as follows BEFORE you run the script:**

* **Blank Parcels – Data.gdb/Parcels**
* **Zoning – Data.gdb/Zoning**

This is what the folder should look like after the script is run:



Toggle the **Parcels Map**, and **ParcelsUpdate Map**. The zoning is now appended to the parcels!

**Script Explanation**

(Lines 6-9) First, I begin setting the input and output workspace and setting the overwrite to true.

(Lines 11-13) These are your input variables, the blank parcels and your zoning data.

(Lines 15-18) These are your temporary layers created by the tools.

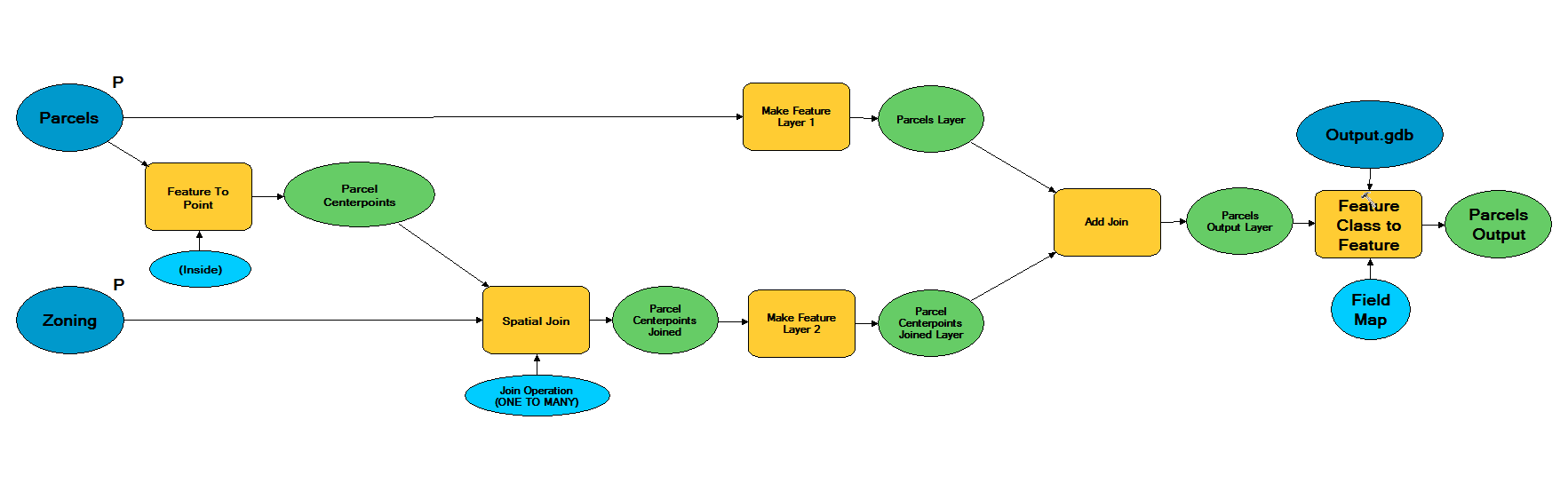
(Lines 20-23) These are your output variables. Only the “Parcels” in your Output.gdb will survive.

(Lines 25-32) These are the Mapping Variables. The source (ZoningParcels.lyr) and mapDocument1 (Parcels.mxd) you already have on file. The mapDocument2, mapDocument3, mapDocument4, pdf, and pdf2 are created after you run the script.

\*Think of your pdf’s and the Parcels feature class in the Output.gdb as your final product.

(Lines 34-60) There are three optional parameters utilized in the script and I decided to call these out as variables instead of having them in your script parameters. **I purposely did this for explanation in this write-up.** The v\_Inside\_ is the Boolean “true” for the Feature To Point tool. Join\_Operation\_\_ONE\_TO\_MANY\_ is used in the spatial join. The Field Map selection is used in the Feature Class to Feature Class. The Field Map allows you to choose the fields you want in the output. I used backslashes to separate a very long line of code for this. The long structured strings represent each field map parameter you want in the output.

(Lines 62-84) Here are the seven process tools you will run. They are also displayed in the following graphic from my project proposal below.



\*Two highlights to take note of are: the “PIN14” field in Process 5 (Lines 76-77), and the deletes in Process 7 (Lines 82-84). “PIN14” is the join field for the target and join feature tables. Process 7 deletes the centerpoint data after the script is finished. It is not in the graphic above.

(Lines 86-91) Handles errors for the tool portion of the script.

(Lines 93-96) This distinguishes the second part of the script using the mapping module. It starts with the sourceLayer we defined above on line 26.

(Lines 98-103) Adds the Parcels\_Output layer to our map document. This is what we created in the tool script!

(Lines 105-114) Updates the layer and legend based on the sourceLayer and the Esri style, “Horizontal Single Symbol Layer Name and Label”.

(Lines 116-124) Removes the beginning layers from the first map based on layer names. The Python string lower() converts the layer names to lower case.

(Lines 126-130) Exports the maps to a pdf (Parcels Map and ParcelsUpdate Map).

(Lines 132-137) The last lines handle errors for the map portion of the script.

**Results**

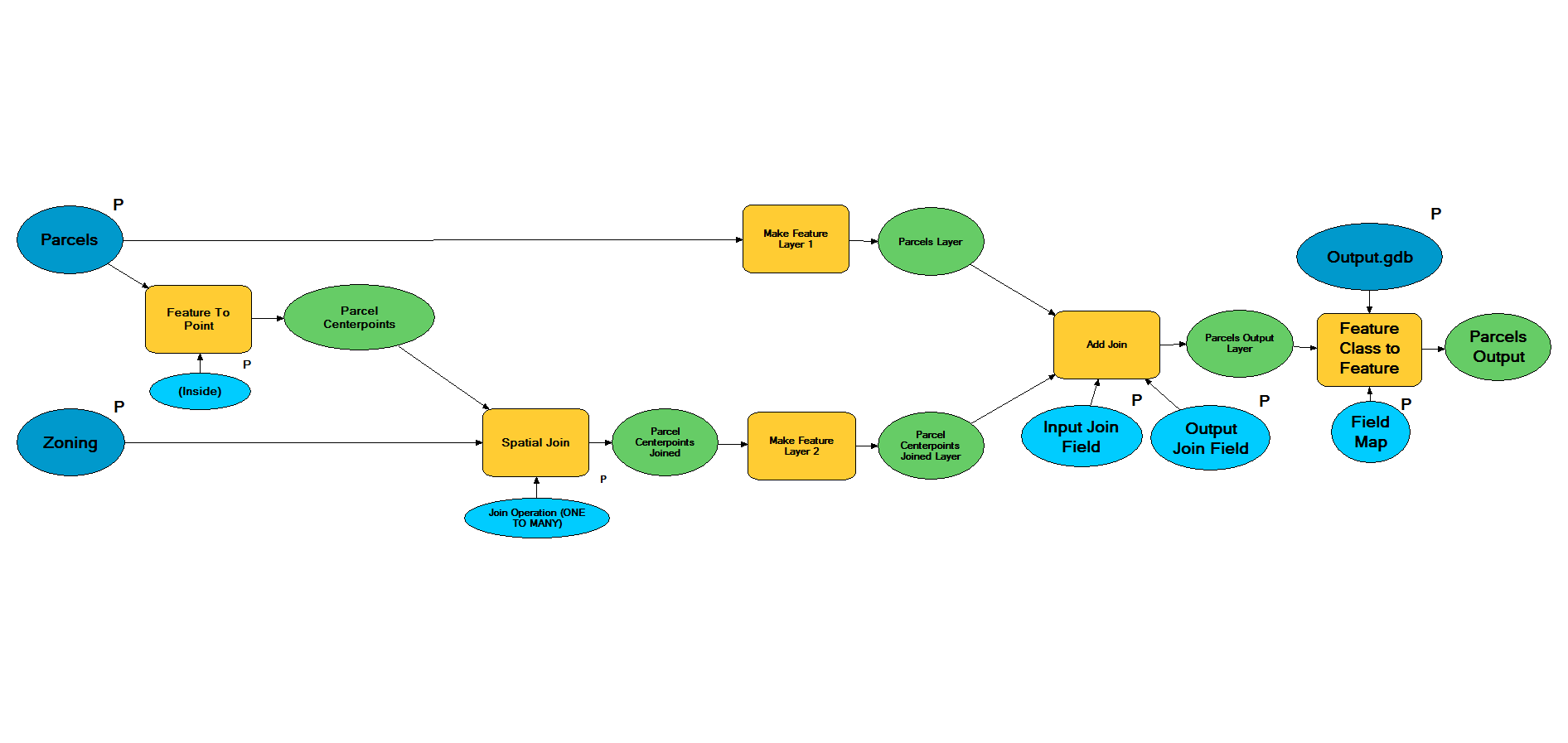
The script should run quickly and you should see three new map documents in the folder called ParcelsUpdate, ParcelsUpdate2, ParcelsUpdate3. You can disregard these, but they are a window into the four-step process of creating the ParcelsUpdate pdf document.

Looking at the Parcels Map pdf document first, you can see first a map with blank parcels overlaid on a zoning layer. Your script joins the zoning fields in the zoning layer to the blank parcels layer to create the ParcelsUpdate Map. Now the parcels themselves are symbolized by Zoning Type, which we get from the ZoningParcelsy.lyr file.

If you were wondering, the purpose of multiple joins is so the parcel’s centerpoint takes the intersecting zoning instead of the entire parcel. Those centerpoints are joined back to the existing parcel polygons after they intersect the zoning. It’s rare, but sometimes one parcel will fall between two zoning districts, so the centerpoint is most suitable. The Green (Park) on the attached maps is a good example of this. This is an important thing to note about data accuracy and limitation of the script.

Since the Field Map parameter is data-specific, you won’t want to use this tool with other data because the Feature Class to Feature Class conversion has those fields linked to my example data. If I wanted to share this script out, I’d make all three parameters, and the input and output join field, a choice in an ArcToolbox tool. This model is also in the dropbox if you would like to try it out. This doesn’t do the mapping module portion of the script or delete the centerpoint data.

Here is an image of the ParcelUpdate Model:



**Conclusion**

The use of this script is joining zoning data to parcel data. Although your sample data is small and the script takes very little time to run, consider how much time it saves on larger areas such as Chicago. This script cuts about a 4 hour process down to about 10 minutes.

I came up with this analogy whilst explaining this project to my girlfriend.

*There is a road from point A to point B. The road has seven traffic signals and you must stop and start at each one to get to your destination. Python scripting is like building a highway to bypass those stoplights, so you only must start and stop once.*

Let me know if you run into any problems running the script. Enjoyed the course. Have a great holiday!

Sincerely,

James Carpenter

(919) 889-3708