**Python Map Automation – Thematic Map Series Sample**

**July 2020**

**ArcGIS Pro 2.5 (and later)**

**Setup**

Download and unzip the ThematicMapSeries\_Pro25.zip to: C:\Temp\ThematicMapSeries\_Pro25. If you unzip to a different location, you must update the scripts. The folder path is hard coded. See Sample Scripts section below.

**Background**

A thematic map series is when the extent of the map doesn’t change but the layers displayed modified for each page in the map series. An example might be a residential site location where you want to display parcel information on one page, then fire districts, school districts, flood zone information, an so on, all on different pages. A thematic map series currently can’t be created in the ArcGIS Pro interface , but it can be done using Python map automation. A thematic map series can be more complex than just turning layers on and off before exporting a page and that is why a solution in the application interface isn’t available. In many cases, depending on the thematic page, layers may need to be displayed in a different order, or display different symbology, definition queries, and different scale thresholds might be applied to a layer as well. The layer specific logic can be automated in a Python script more easily than an application user interface that may not always satisfy all requirements. Esri hopes to build a basic user interface for thematic maps but there may always be a need to further customize it with Python.

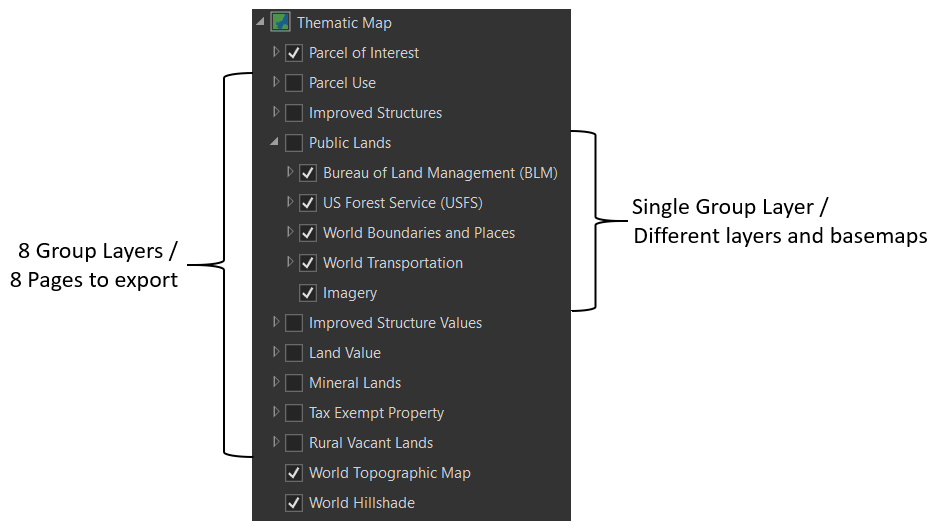
This sample application will use an approach that simplifies and minimizes the amount of Python code needed to generate a multi-page PDF thematic map series. The strategy is to build a group layer for every page in the map series. Layers may be duplicated in the various groups but this allows you to completely control many of the subtle differences like symbology, definition queries, layer order, etc. The basemap layers can remain at the bottom most position of the Contents pane and not in a group layer if the same basemap is used for all pages. But it is also common to use different basemaps in a thematic map series as well. For example, one set of thematic pages may display aerial imagery and another set may display topographic information.

***Pro TIP*:**

The Pro application interface Basemap command limits you to one basemap in a map. In other words, when you add a basemap, it replaces existing basemaps. To add multiple basemaps into a single map, add the basemap of interest and save it to a layer file (\*.lyrx). If the basemap is made up of multiple layers, you may need to group it first and then save it to a layer file. Once you have all the basemap layer files you need, then you can add them into the map using the Add Data command and browse to the layer file.

**Sample ArcGIS Pro Project**

The project contains a single map called Thematic Map with a single, local feature class called Parcels. It is labeled Parcel of Interest and located at the root level and at the top of the Contents pane. See the graphic below. All other layers are web service layers. There are also two basemap layers at the root level of the Contents pane and located below all other layers. These three layers will always be toggled on. All other layers are organized into group layers. Each group layer is designed to serve as the thematic content for each exported page. There are a couple of group layers that contain different basemap layers and because they are above the basemaps that are always toggled on, they will override the display based on priority draw order so there is no need to toggle the bottom-most basemap layers off.

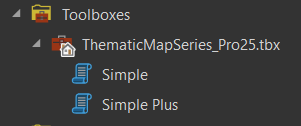


The project also contains a single layout called Thematic Map Series. It has a number of layout elements that will support the final exports. To get a sense of what each exported page will look like, simply toggle each of the group layers in the Contents pane. There are also a number of text elements that will update dynamically when the appropriate Python script is executed. These elements are the Title, Page Number, and Layer Description. The layer description comes from the group layer’s metadata Description value. Only the second, slightly more advanced Python script will update the text elements on the layout.

**Sample Scripts**

There are two script tools provided with the project: ThematicMapSeries\_Simple.py and ThematicMapSeries\_Simple\_Plus. The first script was designed to represent the bare minimum requirements for using Python to automate a thematic map series. Again, organizing each page’s thematic content into a group layer also greatly reduces the amount of code and scripting logic needed. The second script demonstrates how to incorporate additional logic to enhance the output by updating dynamic text elements for each page that gets exported. It also demonstrates appending a title page in front of all map series pages to simulate map book creation. A map book contains map series pages but it also includes additional supporting pages like a title page, table of contents, and perhaps other supporting information.

The scripts are associated with a script tool and are part of the project. To execute each script tool, open the Catalog pane and expand the Toolboxes folder. See graphic below. Next expand the ThematicMapSeries\_Pro25.tbx and double click one of the two script tools. There are no parameters needed. Click the Run command in the lower left corner.



There are two ways to view the code for each script. First, you can right-click the script tool in the Catalog pane and choose Edit from the context menu. The second way is to right-click on the \*.py files in the location this project is saved and choose Edit with IDLE (ArcGIS Pro) from the context menu.

The scripts might need to be edited to run successfully on your computer. For example, the default folder is C:\Temp so if you don’t have that folder on your system, the script will fail. This is the only critical piece that needs to be updated. Everything else will work as designed but you might want to change things like file name or if you apply this script to one of your projects that strings like map name, layout name, and the list of group layer names will need to be altered.

***Known issue***

Due to a known bug, the metadata description does NOT update in Pro 2.5 either from the script tool or run from within the Python Window. At Pro 2.6, the metadata does not appear when the script is run from a script tool but does update if you load the same script into the Python Window and execute it. This only applies to the second, Simple Plus, script. The issue will be addressed in either an ArcGIS Pro 2.6 patch or version 2.7.

To run the script from the Python Window, activate the Analysis tab. In the Geoprocessing group, click on the Python Window command. You can optionally dock the window. Right-click the lower input control and choose Load Code from the context menu. Browse to the location of the \*.py files. After the code is loaded, you will need to hit the Enter key to execute the code.

**Data sources**

There is only one local dataset called Parcels in the project. It was downloaded from the following Open Montana Cadastral site and clipped to the map frame’s extent:

<ftp://ftp.geoinfo.msl.mt.gov/Data/Spatial/MSDI/Cadastral/>

All other layers are derivative products from the above layer and are available as web maps services that are available via Esri’s ArcGIS Solutions for Local Government / Community Parcels.

<https://solutions.arcgis.com/local-government/help/aggregate-parcels/get-started/explore-maps/>