**Note**: *Even though this document primarily focuses using ArcMap, the same concept still applies if you were to author it from ArcGIS Pro 2.1 or above*.

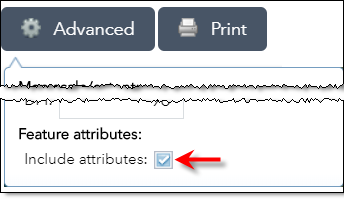
# Overall concepts:

There are two approaches that can be taken:

1. Stored selected features, of a query results, in a separate graphic layer:  
   *Applicable for both mapImage (aka dynamic map service layer) and feature layers.*

**Note**:

* Due to an accidental side effect of a fix, this approach does not work with
  + JS API ver 3.21, 3.22 & 3.23 or
  + clients rely on them such as Web Application Builder comes with ArcGIS Enterprise 10.6
* In ver 3.24, JS API introduced [an option](https://developers.arcgis.com/javascript/3/jsapi/printtemplate-amd.html#forcefeatureattributes) that allows users to choose whether attributes will be included or not in the request that goes to a print service
* Web Application Builder that is released as part of ArcGIS Online June release exposes JS API’s new option in its Print Widget’s UI.



* The web app needs to store features (that are selected because you click on to see popups or you use the query task) as graphics in an operational layer.
* Because selected features are stored in an operational layer, the Printing widget/task will include them in the JSON, when [forceFeatureAttributes](https://developers.arcgis.com/javascript/3/jsapi/printtemplate-amd.html#forcefeatureattributes) is set to true, that gets sent to the Printing Service
* The printing service, which *in this case* an arcpy based service, is designed to:
  + read the json,
  + find all graphics layers (i.e. operationalLayer with ‘featureCollection’, no URLs) and
  + add attributes for all features from all layers on a layout as text elements... finally, it
    - exports the map as a pdf
    - exports one or more layouts with text element into multiple pdf files
    - combines them all and returns a single mult-page pdf to the client

1. Use selection on a feature layer:  
   *This only works with feature layer and not applicable to mapImage layer.*

* You can call [selectFeture()](https://developers.arcgis.com/javascript/3/jsapi/featurelayer.html) or use widgets like Web Application Builder’s Select widget, to select features off that layer.
* Since the selection set is maintained with the feature layer, the Printing widget/task will object ids of those selected features in the request that gets sent to the Printing Service.

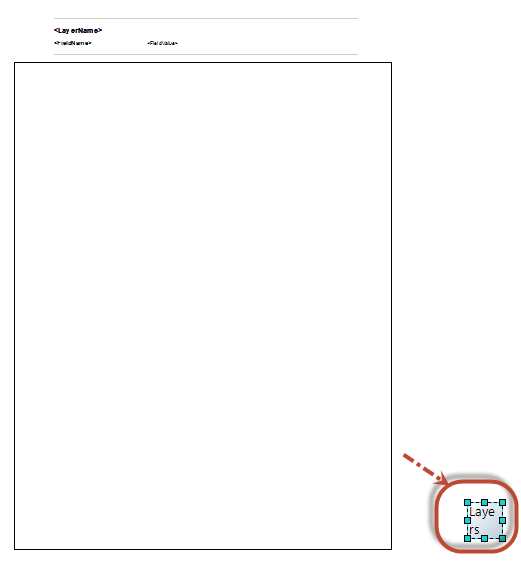
**Note**: with this approach, attributes will be included in the request unlike the first approach. As a result, forceFeatureAttributes has no impact on this.

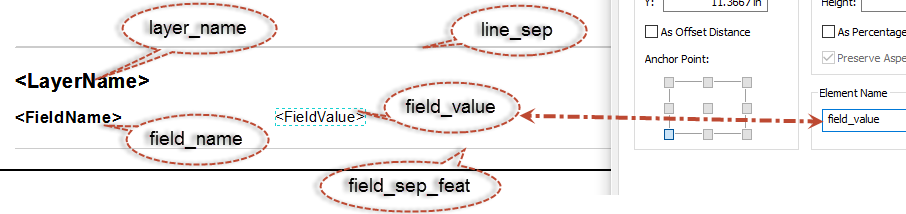
* The printing service, which *in this case* an arcpy based service, is designed to:
  + read the json,
  + find selected features object ids
    - use those ids to retrieve attributes by submitting a query request to the layer’s source map or feature service
  + add attributes on a layout as text elements... finally, it
    - exports the map as a pdf
    - exports one or more layouts with text element into multiple pdf files
    - combines them all and returns a single mult-page pdf to the client

# How to create layout templates to show popups?

In this case, an additional layout template (.mxd) needs to be created (*that is the Popups.mxd you will find in this folder*). This template layout is not exposed to the end client via the printing service... it is internal to the printing service.

Here are the steps used to create Popups.mxd:

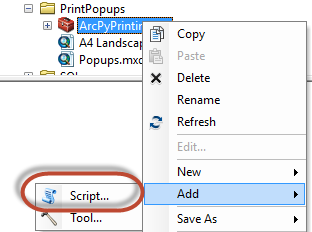
1. Create a new document in ArcMap
2. Switch to the ‘Layout’ mode
3. Move the ‘Data frame’ element outside of the page outline  
   
4. Add few elements on the layout
   1. 3 text elements
   2. 2 line elements
5. Move them on top of the page (*outside of the page area as you see in the above screen shot*)
6. For text elements, change their font size, style, name etc.
7. For line elements, change their width, color, style etc.
8. Make sure those elements are ‘positioned correctly in x-axis (horizontally)’
   1. The script duplicates them and move them vertically; therefore it is important that you position them correctly in the x-axis (horizontally)  
      
9. Open each elements (both text and line) ‘Properties’ page
   1. Switch to ‘Size and Position’ tab
   2. and provide ‘Element Name’s
      1. Names of line elements:
         1. line\_sep
         2. line\_sep\_feat
      2. Names for text elements:
         1. layer\_name
         2. field\_name
         3. field\_value

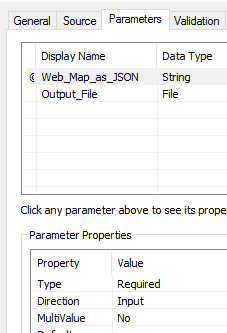


Note: values for these text elements could be anything... they get modified by the script anyways.

# Steps to create an arcpy script based printing service?

The script tool is already created for you in the same folder with this readme.doc and it is inside ‘ArcPyPrinting.tbx’ file.

1. Create a python script tool in Desktop referring to the .py file  
   
2. Add the following two parameters set to the script tool.  
   Web\_Map\_as\_JSON (type: string, direction: INPUT)

Output\_File (type: FILE, direction: OUTPUT)  
  
*please see a note in the Appendix.*

1. Execute the script tool by
   1. passing a space i.e. ' ' to ‘Web\_Map\_as\_JSON’ parameter and
   2. choosing an output file name
2. From ‘Geoprocessing Result’ window (Menu: Geoprocessing >> Results), right-click on the result and start the publishing process.
3. Publish it as a ‘synchronozied’ gp service

Note: for more information and step by step instructions on creating python based printing service, read the help and check out tutorials link from <http://server.arcgis.com/en/server/latest/create-web-apps/windows/advanced-printing-for-web-maps.htm>

# How to use that in the web app?

These steps are for Web AppBuilder for ArcGIS. For custom JS API app, you need to write code to perform similar operations:

1. Get the REST URL of the only Task inside the newly published service and use that with the Printing Task or Widget in the web app
2. In web application builder, you can use Select widget to select features
3. Bring up the Print widget and click OK to print.
4. The output should be a multi-page pdf

# Appendix:

* There is [bug](https://support.esri.com/en/bugs/nimbus/QlVHLTAwMDExNDE1NA==) in Web App Builder’s Print widget that makes the widget not recognize a print service with missing optional parameters. If it happens to you, please add ‘Format’ and ‘Layout\_Template’ optional parameters, with minimally one place holder value for each of them, to your script tool.