**Mkpopup.py Documentation**

Four utilities have been created to make creating popup.xml files easier for individuals configuring the FlexViewer. Popup.xml files are used by the ESRI Flex Viewer to allow users to click on features and bring up information about these features, in much the way the traditional identify tool works in ArcView or ArcMap. The ESRI documents how to use these popup.xml file along with the viewers configuration file at the following link:

<http://help.arcgis.com/en/webapps/flexviewer/help/index.html#/Pop_up_configuration_files/01m30000002q000000/>

The tools created allow for basic configuration of the popup.xml file. The person implementing the viewer will most likely need to slightly tweek the xml files produced by this utility, but it should provide a good starting point. There are two python scripts provided, and they do the following:

**mkPopup.py:**

This tool is run from the command line like such:

*Python myPopup.py c:/Data/GISData/blank\_canvec.gdb testFC PopuptestFC.xml*

Where the arguments to the tool are:

1. The geodatabase where the feature class is located
2. The name of the feature class found in the prior geodatabase
3. The outbput xml file

As a result the PopuptestFC.xml produced looks like this:

<?xml version="1.0" ?>

<!-- File Created with mkPopup.py by Frank Roberts - froberts@innovateteam.com -->

<configuration>

<!-- Add field name with curly brackets for the title -->

<title></title>

<fields>

<field name="uglyname" alias="uglyname" visible="true"/>

<field name="uglycode" alias="uglycode" visible="true"/>

<field name="uglyblob" alias="uglyblob" visible="true"/>

</fields>

<!-- Show attachments = either true or false -->

<showattachments>false</showattachments>

</configuration>

In the above file you will need to add the field used for the title in the pop up if you wish to have a title in a popup. So for example I would then add {uglyname}, if I wanted the field called uglyname to be added as the popup title. In addition, if you only want a subset of the attributes listed then you will want to delete some of the entries from the xml file. Lastly you may want to change the alias values for the fields, if you have better more user friendly names.

**lisFC.py**

The other file provided is called listFC.py. This file is used in conjunction with the above python file.

This tool is run from the command line like such:

*python listFC.py c:/Data/GISData/blank\_canvec.gdb RoadMap*

The result of running the above command would generate popup.xml file for all feature classes found within the “*blank\_canvec.gdb*” geodatabase. In addition the popup.xml file would have the name “RoadMap” tacked on to each file name generated between Pupup and the feature class name (e.g. PopupRoadMapRoads.xml). This last item is added to assist the individual configuring the flexviewer, since all poup.xml files are put in one directory for all flexviewer maps, this sometimes can get confusing with what pupups go with which maps.

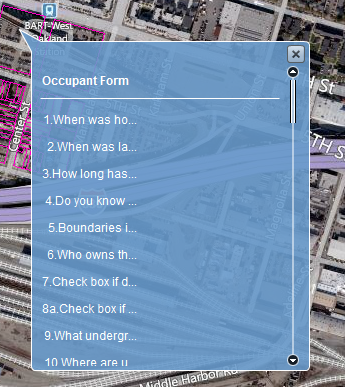
When listFC.py is run, the resultant xml files will be placed in the directory that the command is being run from. The resultant xml files can then be copied up the appropriate place in

**NEW Improvement !!**

In addition, two other files have been added to this collection. They are listFCDesc.py and mkPopupDesc.py. They are very similar to the two above files but they are use to write the popup xml files so the popups utilize the description part of the popup tag. In the event that you have long descriptors in the titles of your columns, I have seen where the popups are not readable, since the default popup display does not wrap the description of each item (see graphic 1 below). To fix this problem you then write both the description and value of the attribute using html code to the description section of the xml.

For more on this topic see thread: <http://forums.arcgis.com/threads/47831-Popup-Truncated-Description?p=163632#post163632>

Graphic 1. Example of long attribute names and lack of values being displayed in the popup.



**Python from the command window help**

In the event that you want to run python from the command line windows (aka the dos prompt), you may need to add the location of python to your path variable on your computer. On my Windows 7 pc with ArcGIS 10 installed on it, the location of python is at “"C:\Python26\ArcGIS10.0\python.exe"”. Follow the instructions below to add the python path to your path variable.

1. From the [desktop](http://en.wikipedia.org/wiki/Desktop), right-click **My Computer** and click **Properties**.
2. In the System Properties [window](http://en.wikipedia.org/wiki/Windows), click on the **Advanced** [tab](http://www.computerhope.com/jargon/t/tab.htm).
3. In the Advanced section, click the **Environment Variables** [button](http://www.computerhope.com/jargon/p/pushbutt.htm).
4. Finally, in the Environment Variables window (as shown below), highlight the **Path** variable in the Systems Variable section and click the **Edit** button. Add or modify the path lines with the paths you wish the computer to access. Each different directory is separated with a semicolon as shown below.  
     
   C:\Program Files;C:\Winnt;C:\Winnt\System32; C:\Python26\ArcGIS10.0\python.exe

