

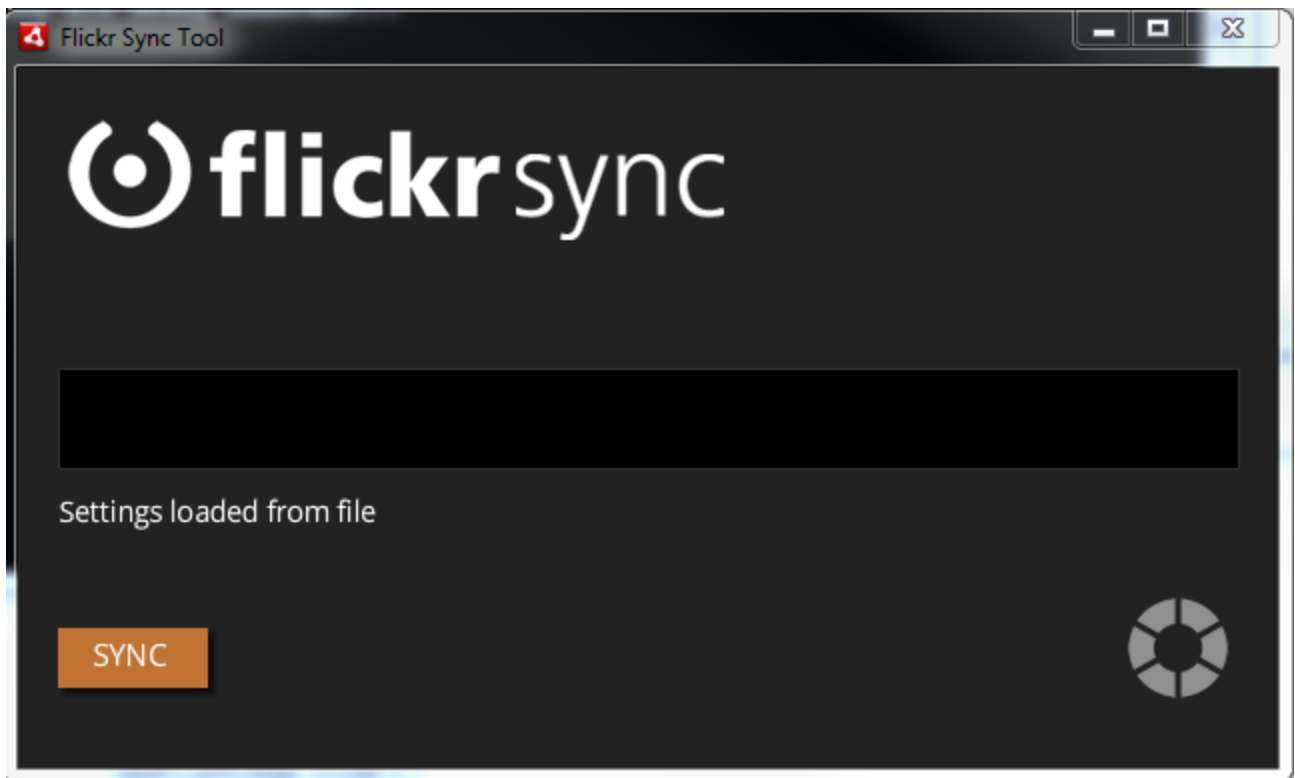
Flickr Sync Tool

Introduction

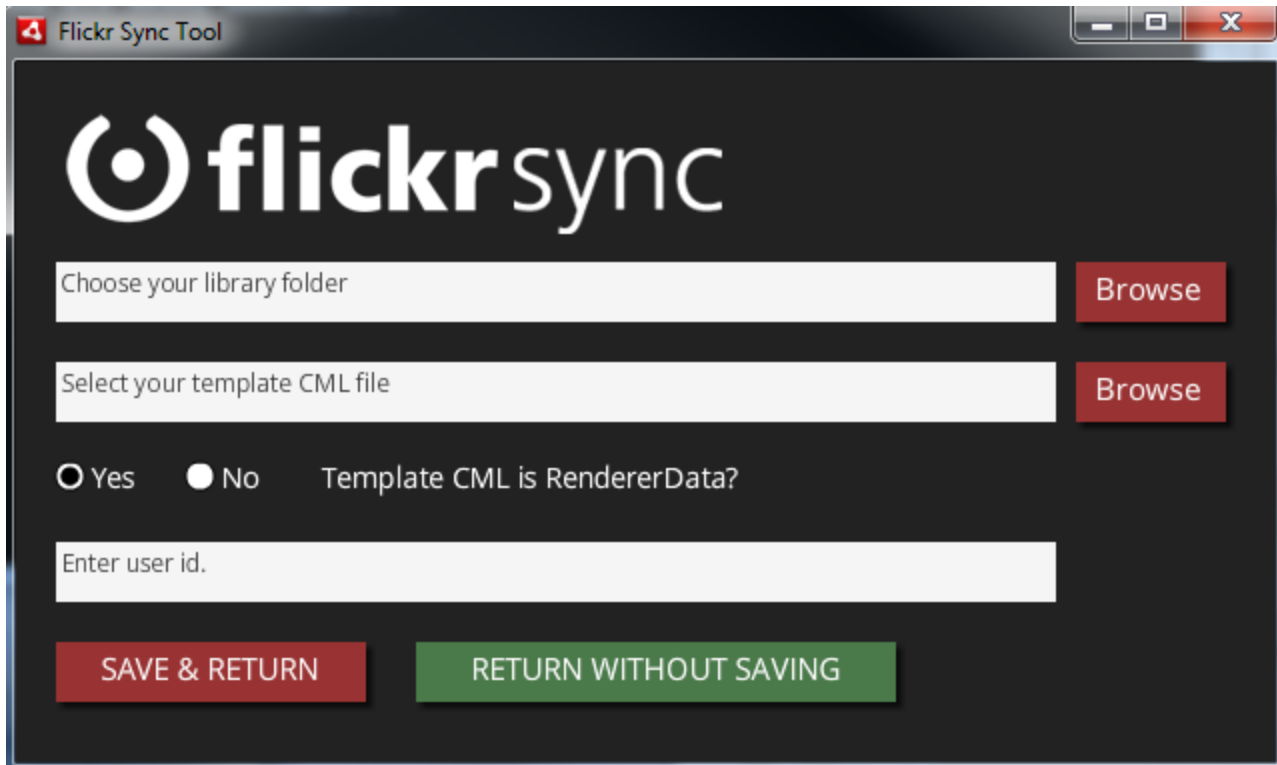
The Flickr Sync Tool is a utility with which you can download pictures and videos from Flickr and dynamically generate CML files based on user-defined templates for use in your CML driven projects.

How To Use

When you initially start up the tool you should see a screen similar to this



On the first run, instead of seeing "Settings loaded from file", you will probably see "Settings file not found. Please configure your settings". To configure your settings click on the Gear in the bottom right hand corner. It should take you to a screen that looks like this



You will notice that there are 4 form fields that you have to set..

1. You need to enter the library folder which exists in the bin folder of the CML project you wish to use this tool with. If you are familiar with creating CML projects then you will understand that is the “project root folder/bin/library/” directory. Please select this directory. We will cover the required directory structure later in this document.
2. Please enter the template CML file. This is the template file from which we will be dynamically creating CML files with the data we receive from Flickr. We will go over how the tool can use this template files later in this document.
3. Select the option if the template CML file is a RenderData type CML file or not. We will cover examples of each.
4. Enter your Flickr user id. This is the id of the user you want to get images from. We can only get public images for this user.

Once you have entered in the information required, press “Save and Return”. This will return you to the previous screen. If you click “Sync” this tool will start querying Flickr for the User’s pictures and sets, download them, and create CML files out the metadata.

Required Directory Structure

You will need a CML folder in your Library folder, therefore your CML directory will end up having a path similar to this - “project root folder/bin/library/cml”.

Any generated CML files will be placed directly in this CML folder.

Dynamic Content from Templates

Let's cover two types of templates that we support. We support RendererData templates and non-RendererData templates. An example of the RendererData template is the directory that contained this document and is called "template.cml". The non-RendererData template is called "ImageViewer.cml". The former template can handle set and picture metadata, whereas the latter can only handle picture metadata. The former does not support include statements which can include other CML files, whereas the latter does support include statements. It just depends on which template style is better suited for your projects.

Let's cover the attributes that each template type supports and how to use each one.

RenderData Template

First let's cover a RendererData template file.

```
<?xml version="1.0" encoding="UTF-8"?>
<cml>

<RenderKit>
<Hotspot>
    <label>{set:title}</label>
    <x>{picture:tags:Lon}</x>
    <y>{picture:tags:Lat}</y>
    <link>{set:id}</link>
    <hotspot_id>HID_{set:id}</hotspot_id>
</Hotspot>

<Viewer>
    <id>{set:id}</id>
    <title>{set:title}</title>
    <panel_path>library/cml/locations/{set:id}.cml</panel_path>
</Viewer>

<Panel multiple="true" location="set:pictures">
    <media>library/assets/{picture:fileName}</media>
    <width>426</width>
    <height>320</height>
</Panel>
```

```
<Description multiple="true" location="set:pictures">
    <desc_text>{picture:description}</desc_text>
</Description>
</RenderKit>

</cml>
```

If we use this type of template file, then we the tool will iterate over all Sets that a User has in their Flickr account and get the corresponding Pictures in those Sets. If a User has any Pictures not in a Set, then this tool will put it in a Set by itself. This sample template file has been used to generate CML files for Album Viewers.

Let's go over the supported replacement tags - most of these are taken from how Flickr names their metadata tags.

```
{set:id}
{set:title}
{set:description}
{set:date_create}
{set:date_update}
```

```
{picture:id}
{picture:dateuploaded}
{picture:title}
{picture:description}
{picture:posted}
{picture:lastupdate}
{picture:owner}
```

```
{picture:fileUrl}
{picture:fileName} - this is the name of the media as will be saved on your file drive. This name will be the media name plus the media id. (If you have a picture called Picture A with id "234987249", the fileName will be "Picture A 234987249". The file extension will be .jpg if it is an image or .mp4 if it is a video.
```

```
{picture:media} - this will either be "photo" or "video"
```

```
{picture:tags:*} - this is a User defined field in Flickr. The tool will look at whatever is written where the "*" is located and try to find a tag that starts with that String. If it finds a match, then it will replace everything in between (and including) the brackets. For example if you use the String {picture:tags:Lat} and you have a User defined tag in Flickr that is "Lat:120N", you will end up with Lat:120N.
```

You may also notice that the Panel and Description tags have attributes called multiple and location. The multiple tag means that this CML element may need to be repeated multiple times. For example if we are looking at a Set with 2 pictures we would end up with two Panel elements and two Description elements. The second attribute of location, just tells us to look at all pictures within a set. This is the only tag that is supported currently for this attribute with this tool.

Let's look at a sample CML file of what would be generated if we used the template file.

```
<?xml version="1.0" encoding="utf-8"?>
<cml>
  <Hotspot>
    <label><![CDATA[Black Butte Reservoir]]></label>
    <y><![CDATA[Lat:39°47'31.63N]]></y>
    <hotspot_id><![CDATA[HID_72157633472423199]]></hotspot_id>
    <link><![CDATA[72157633472423199]]></link>
    <x><![CDATA[Lon:122°21'40.21W]]></x>
  </Hotspot>
  <Description>
    <desc_text><![CDATA[Sample Description 1]]></desc_text>
  </Description>
  <Description>
    <desc_text><![CDATA[Sample Description 2]]></desc_text>
  </Description>
  <Viewer>
    <title><![CDATA[Sample Viewer Title]]></title>
    <id><![CDATA[72157633472423199]]></id>
  </Viewer>
  <panel_path><![CDATA[library/cml/locations/72157633472423199.cml]]></panel_path>
  </Viewer>
  <Panel>
    <height>320</height>
    <media><![CDATA[library/assets/Sample Title 1.jpg]]></media>
    <width>426</width>
  </Panel>
  <Panel>
    <height>320</height>
    <media><![CDATA[library/assets/Sample Title 2.jpg]]></media>
    <width>426</width>
  </Panel>
</cml>
```

Non Render Data Template

Now let's cover the non RenderData file.

This template file supports iterating over pictures only. Lets go over the tags that are supported:

```
{id}
{dateuploaded}
{title}
{description}
{posted}
{lastupdate}
{owner}
{fileUrl}
{fileName}
```

However, this template method supports include statements. This means that this can go into the files specified in the include statement and replace tags in those files as well. Please make sure that your included CML files have simple CML tags (with no attributes).

Here is a sample template file.

```
<?xml version="1.0" encoding="UTF-8"?>
<cml gml="library/gml/my_gestures.gml"
css="library/css/module/my_styles_modules.css" simulator="false">

    <LibraryKit>
        <Library type="swf"
src="library/assets/swfs/openexhibits_assets.swf" />
    </LibraryKit>

    <ImageViewer id="iv" gestureReleaseInertia="true" image="#img1" front="#front"
menu="#menu" back="#back1,#back2" background="#info-bg" timeout="10"
fadeoutDuration="5" dropShadow="true" blur="true" mouseChildren="true">

        <TouchContainer id="front" visible="true" targetParent="true"
mouseChildren="false">
            <Image id="img1" src="library/assets/{fileName}" resample="true"
width="1000" scale=".5"/>
        </TouchContainer>

        <TouchContainer id="back1" targetParent="true" mouseChildren="false"
clusterBubbling="false" visible="false">

            <Graphic id="info-bg" class="info_bg" shape="rectangle"/>
            <Text id="title" class="info_title" text="{title}" />
```

```

    </TouchContainer>

    <ScrollPane id="back2" visible="false" width="600" height="400"
paneStroke="0" scrollThickness="7" mouseChildren="true" targetParent="false"
clusterBubbling="true">
        <ScrollBar buttonFill="0x000000" fill="0x555555"
thumbFill="0x303030" cornerHeight="15" cornerWidth="15"/>
        <Text id="description" scrollable="false" paddingLeft="20"
paddingRight="20" paddingBottom="10" class="info_description" multiline="true"
selectable="false" text="{description}"/>
        <TouchContainer disableNativeTransform="true">
            <GestureList>
                <Gesture ref="n-drag" gestureOn="true"/>
            </GestureList>
        </TouchContainer>
    </ScrollPane>

    <Menu id="menu" class="menu_container" alpha="0.6" position="bottom"
paddingLeft="20" paddingBottom="20" paddingRight="20" autoHide="true"
visible="false" >
        <Include src="library/cml/buttons/closeButton.cml"/>
    </Menu>

    <GestureList>
        <Gesture ref="n-manipulate-inertia-boundary" gestureOn="true"/>
    </GestureList>

</ImageViewer>

</cml>

```

Here is what could be generated -

```

<?xml version="1.0" encoding="UTF-8"?>
<cml gml="library/gml/my_gestures.gml"
css="library/css/module/my_styles_modules.css" simulator="false">

    <LibraryKit>
        <Library type="swf"
src="library/assets/swfs/openexhibits_assets.swf" />
    </LibraryKit>

<ImageViewer id="iv" gestureReleaseInertia="true" image="#img1" front="#front"
menu="#menu" back="#back1,#back2" background="#info-bg" timeout="10"
fadeoutDuration="5" dropShadow="true" blur="true" mouseChildren="true">

```

```

        <TouchContainer id="front" visible="true" targetParent="true"
mouseChildren="false">
            <Image id="img1" src="library/assets/8735172421.jpg"
resample="true" width="1000" scale=".5"/>
        </TouchContainer>

        <TouchContainer id="back1" targetParent="true" mouseChildren="false"
clusterBubbling="false" visible="false">

            <Graphic id="info-bg" class="info_bg" shape="rectangle"/>
            <Text id="title" class="info_title" text="Sample Title" />

        </TouchContainer>

        <ScrollPane id="back2" visible="false" width="600" height="400"
paneStroke="0" scrollThickness="7" mouseChildren="true" targetParent="false"
clusterBubbling="true">
            <ScrollBar buttonFill="0x000000" fill="0x555555"
thumbFill="0x303030" cornerHeight="15" cornerWidth="15"/>
            <Text id="description" scrollable="false" paddingLeft="20"
paddingRight="20" paddingBottom="10" class="info_description" multiline="true"
selectable="false" text="Sample Text"/>
            <TouchContainer disableNativeTransform="true">
                <GestureList>
                    <Gesture ref="n-drag" gestureOn="true"/>
                </GestureList>
            </TouchContainer>
        </ScrollPane>

        <Menu id="menu" class="menu_container" alpha="0.6" position="bottom"
paddingLeft="20" paddingBottom="20" paddingRight="20" autoHide="true"
visible="false" >

<Button id="close-btn" class="close_btn" dispatch="down:close" hit="close-hit"
initial="close-up" down="close-down" up="close-up" out="close-up">
    <Container id="close-up" class="up" x="660" y="10" >
        <Graphic alpha="1" shape="rectangle" width="30" height="23"
lineStroke="1" lineColor="0x666666" color="0x4D4D4D" />
        <SWF alpha="1" x="8" y="5" color="0xbdbdbd" scale="1.3"
classRef="org.openexhibits.assets.Close"/>
    </Container>
    <Container id="close-down" class="down" x="660" y="10" >
        <Graphic alpha="1" shape="rectangle" width="30" height="23"

```



```
lineStroke="1" lineColor="0x666666" color="0x4D4D4D" />
    <SWF alpha="1" x="8" y="5" color="0xbdbdbd" scale="1.3"
classRef="org.openexhibits.assets.Close" />
    </Container>
    <Container id="close-hit" class="hit" x="660" y="10" >
        <Graphic id="close-hit-bg" shape="rectangle" alpha="0" width="30"
height="23" />
    </Container>
</Button>

</Menu>

<GestureList>
    <Gesture ref="n-manipulate-inertia-boundary" gestureOn="true"/>
</GestureList>

</ImageViewer>

</cml>
```

Best Usage

This tool can work really effectively if you want to sync with Flickr and have the tool generate fairly simple CML files. More complicated features may added if there is demand.