BirdOne Fly Macros

In BirdOne Fly version 1.40, BirdOne introduced flight macros. Flight macros are a way to monitor different events during flight and then executing an action that is designated by the pilot. Macros are built to easily distribute to your entire team automatically. Pilots can choose different macros they want to use in their flight settings.

Macro Structure

Macros are broken down into two main categories. Those categories are **triggers** and **actions**. Triggers are listeners that monitor for a specific event to happen. Once these events occur, this will trigger an action, something you want to execute based on that trigger. Each trigger and action have an object they are associated with i.e., a feature layer. Actions also have a value that is associated with that feature layers field. For example, you may want to change the field *status* value in your data to *complete*. In this case, *complete* is the value for the field *status*.

Syntax

The syntax used for the macros uses only 3 characters. They are **<,>**, and **%**. These characters were chosen because they cannot conflict with Esri's data naming structure. Each trigger, action, field, and value are bracketed by the **<** and **>** symbols. The **%** symbol is used to begin an action sequence. If you want, you can stack multiple actions together from a single trigger. To do so, just separate each action sequence using **%**. An example of that Syntax is shown below:

Example 1 (Simple trigger and action sequence)
<ESRI_ON_ATTACH_PHOTO> <Address_Points> %<ESRI_EDIT_FIELD> <ADDR_NUM> <12345>
<ESRI_ON_CREATE_FEATURE> <Address_Points> %<ESRI_EDIT_FIELD> <ADDR_NUM2> <34567>

Example 2 (Stringing actions together)
<ESRI_ON_ATTACH_PHOTO> <Address_Points> %<ESRI_EDIT_FIELD> <ADDR_NUM> <12345>
%<ESRI_EDIT_FIELD> <ADDR_NUM2> <34567> %<ESRI_EDIT_FIELD> <ADDR_NUM3> <4982>

Current Programmable Triggers

<ESRI_ON_ATTACH_PHOTO>

This trigger is used to monitor when a photo has been associated with a specific feature. This trigger should be followed by a field that is being monitored. For example, **<my_gis_feature_layer>.**

<ESRI_ON_CREATE_FEATURE>

This trigger is used to monitor when a new GIS point is added to a feature layer. The trigger should be followed by the field that is being monitored such as **<Address_Points>.**
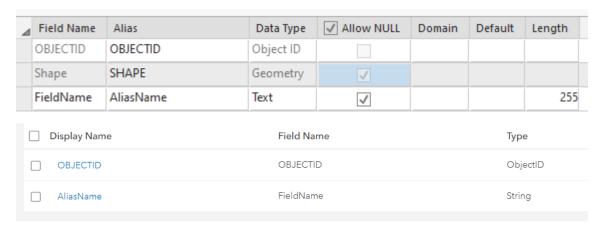
Current Programmable Actions

<ESRI_EDIT_FIELD>

This action is used to edit the contents of your attribute table and takes two parameters. The parameters are the field that is being edited such as **<ADDR_NUM>** and the value you wish to set such as <My custom value from macros>

Esri field names, Alias, and Values

When setting a features field name and data value, it is important to distinguish between the Field or Layer name and the Alias (also known as the Display Name).

Field Name	Alias	Data Type	✓ Allow NULL	Domain	Default	Length
OBJECTID	OBJECTID	Object ID	☐			
Shape	SHAPE	Geometry	✓			
FieldName	AliasName	Text	✓			255

	Display Name	Field Name	Type
☐	OBJECTID	OBJECTID	ObjectID
☐	AliasName	FieldName	String

When possible, use the Alias/Display name for both the Layer and the Field. The one exception to this rule is domains. For domains, you are required to use the Field Name.

Error logs

BirdOne Fly keeps a log which includes any crashes or errors you may receive. If your macros are not working correctly for any reason, please refer to your flight logs. Logs can be found at: **…\Android\data\com.birdone.robinar.parrotapp\files\BirdOne\Logs** A new log is created every day.

Using Macros

Once you have set up your macro based on your data, save it as a .txt file. You can name it anything unique to you and your project. Copy the macro to your macro folder here:
…\Android\data\com.birdone.robinar.parrotapp\files\BirdOne\macros

If you have multiple users and you want to deploy automatically to all users, please send your macro file to support@b1rd.io with the subject line macros.

To use the Macro, go to settings and select the dropdown for Active Macro File. By default, it should be set to none. Select the file you wish to use and then fly. Your macro should now automatically perform the action you want. You should be ready to go.