App Building Guidelines

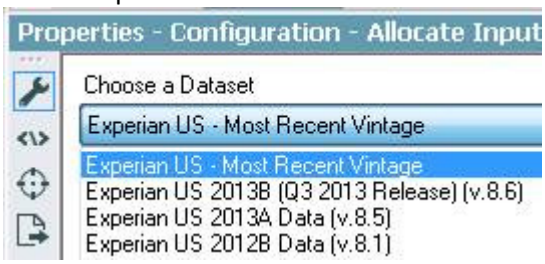
Definition of Apps

An Alteryx Analytic App is a self-contained program that performs a specific function for the user. The interface must be simple and the App designed with a single goal in mind. The App should perform a limited range of tasks, focused on an optimal user experience. Considering that users are running the Apps you are building in a web browser, the back-end design needs to be clever, elegant and efficient.

Most Recent Vintage Dataset

When authoring an App which uses a specific dataset, the selected dataset should be set to use the Most Recent Vintage version of the dataset. This means that Alteryx will automatically use the latest version of the dataset that is installed on the user's computer. This applies to Demographic Analysis tools (Allocate), Behavior Analysis tools (Solocast), Calgary tools, Drivetimes, Geocoder or for the Reference Base Map in the mapping tool.

For example:



Meta Info Tab

All of the information entered in the meta info tab will display in the App details on the web. There are quite a few things to fill in on the Meta Info tab of the Workflow Properties window.

- **Custom**– Give the App a 'friendly' name, for example the App file itself may be called Download_Weather_Data then the custom name would be the same but without the underscores, Download Weather Data.
- **Description–**The description that is entered here will be the description that gets shown on the web for the app, so please make sure it is an accurate description and is user friendly.
- **URL -**The URL field provides the ability to include a link to an external website.
- **Author –**Provide the details of the Analytic Application creator.
    - Name, Vintage (YYYY.Qtr)
    - Company
    - Copyright

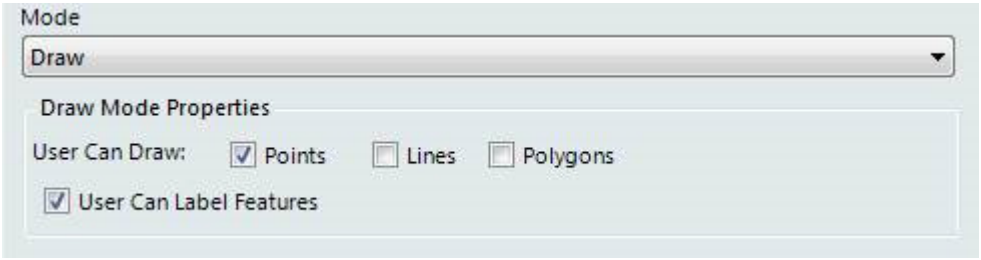
Extension yxwz not yxmd

When authoring an app, make sure you have saved the file as an yxwz file even if you changed the workflow type to Analytic App. If the file extension is left as an yxmd it will open as a workflow and not as an App. *Apps created in 9.0 can no longer be saved as *.yxmd files.*

Map Input

The Map Input tool gives the App consumer the ability to select a location or draw a polygon or line using the map input feature. Map questions should be short but descriptive and a base map should always be used. When choosing a base map, make sure you select 'Most Recent Vintage'.

Finally, if 'Draw' mode is selected, ensure 'User Can Label Features' is always checked. This gives the user the option to add a name to each of their custom polygons.



Report Output

When saving an App to the Gallery, if the output of the report is rendered as PCXML, the application provides the option for the end users to not only preview the report on the web, but to also download the report in any of our supported report formats. Supported report formats include; PDF, Microsoft Word, Microsoft Excel and HTML.

When authoring an App the output can be any of our supported report formats, to utilize PCXML set the 'Output Mode' in the Render tool to 'Choose a Specific Output File', and for the 'Output File' use *Name of the App*.pcxml, for example Geography Analysis.pcxml.

Note: When previewing a report in PCXML you will not see footers, but they will be available when downloaded as a PDF, Word or Excel document.

Reading / Writing Files, Including Using Macros in an App

For web Apps you will only be able to read and write to files that are within the same folder as your App or a lower down folder, i.e. Any folder that is within the folder that contains the app. (Can be more than 1 level down.) This rule will also apply to Macros that you wish to use for your App, unless it is a standard macro which is part of the Alteryx product installer or a data installer.

%temp%

Using %temp% to write to the temp directory when building workflows or apps to be saved to the Gallery is not recommended. Keep all file paths within the workflow. If developing a chained app, it is especially important to use only the file name rather than any file path (e.g.: '.\fileOutput.yxdb' and NOT

'%temp%\fileOutput.yxdb' or something similar. As a best practice users should select a file by browsing to it and then change the file dependency path to a relative path using the Workflow Dependencies window.

Prohibited Tools and Events

The following tools and events are prohibited in applications saved in the Analytics Gallery.

- Download tool
- Email tool
- R tool
- Run Command tool
- Run Command and Email events (from Events tab in Workflow properties)

If you have a good reason to need to use one of these tools or the events option, then you may apply for an exemption, which will allow your App to run. More information on applying for an exemption can be found here: http://gallery.alteryx.com/#!exemption

Please note that the Predictive Macros included with the Designer Desktop which use of the R tool ***are*** allowed in the Analytics Gallery.

Unsupported App features for Web

The following are not supported in the Analytics Gallery web environment.

- Folder Browse
- File Browse (Save as). Upload works fine on the web.
- If an App throws any errors in the Desktop environment, it will not be able to be saved to the Gallery.

Best Practices for Building Apps

Update Actions

When using the Action tool to Update Value, unless actually needed, don't use the 'Replace a specific string:' option. If you change the tool configuration in your workflow due to continued development, you may end up breaking that action tool, because that specific string may no longer exist. Obviously there will be times when you need to use that option but just bear in mind that if you change that string in the tool then you may need to also update the action tool.

Updating Detours

If you are using an Action tool to update a detour try to be complete and update the detour for both possibilities, i.e. detour left and detour right. Otherwise if you just update for the one condition and you change the workflow while developing then you haven't accounted for when you need the workflow to go the other way.

Ending Detours

All detours must end, especially before joining any data streams from a detour to anywhere else in the workflow. Detour End tools require no configuration, so they are easy to use. Make sure you use them, or end the detour with an Output tool.

Using the Error Message Tool

When writing questions, consider your user by anticipating common mistakes. You should be issuing error messaging using an Error Message tool. For example if the user must select an option, then throw an error when they don't select anything, this can avoid the engine throwing errors which may not be meaningful enough to the user to understand how to fix an error. Ideally have a throw error for every question that the user needs to fill in. You can also do more complex conditions to make sure they have filled things in correctly. For example, if they need to fill in a text box with up to 5 trade areas separated by commas (for example 1, 2, 3, 4, 5), you could use a regex condition to make sure there aren't more than 4 commas in the text box.

Issuing Errors using the Message Tool

Along these same lines, try to anticipate cases where the App may fail even when configured properly. For instance, the user may enter in an address that fails to geocode and renders no results. You can easily handle messaging in your App by Filtering bad geocodes and then use the message tool to return a message to the user: "The address you have provided did not geocode. Please check the address is valid and make appropriate changes to it or enter a different address and re-run."

Updating raw XML, escaping HTML Metacharacters

If you are updating raw xml for a tool or using special characters in your Apps then bear in mind that on the web this may not work as anticipated. For example a DropDown/ListBox Selection on the engine could contain text such as:

> Age By Sex Summary Report:<Report Type="summary">Age By Sex Summary Report</Report>
>
> Basic Demographic Summary Chart:<Report Type="summary">Basic Demographic Summary Chart</Report>

But when this is used on the web it will not actually display the question correctly and therefore when this gets used in an action it will probably not have the desired effect. A solution is to change the text to be:

> Age By Sex Summary Report:<Report Type="summary">Age By Sex Summary Report</Report>
>
> Basic Demographic Summary Chart:<Report Type="summary">Basic Demographic Summary Chart</Report>

Then if you are using the question response in an action you will need to update it to be:

EscapeXMLMetacharacters([AllocateSummaryReport])

The Escape XML Metacharacters function was added to the Formula Library in version 8.0. You can access it from the Specialized category. This function will replace all XML metacharacters with their escaped versions.

Allocate Variable Trees

The web and the engine return different values when an Allocate variable tree is left empty. If the App has an Allocate variable tree as a question type then you may want to write a condition which checks if the user selected anything or left this empty. For the engine you could write something like:

[AllocateVariables] == "<Variables />"

Because when the variable tree has nothing selected <Variables /> is returned. However on the web the value returned is actually nothing, so you would need the condition to be:

isempty([AllocateVariables])

Obviously we want the Apps to work on both the web and the desktop so the condition actually needs to be:

[AllocateVariables] == "<Variables />" or isempty([AllocateVariables])

If we are also worried about using the <, > and / characters on the web then we could change the condition to be:

REGEX_CountMatches([AllocateVariables], "Variables") == 1 or isempty([AllocateVariables])

The Union Tool

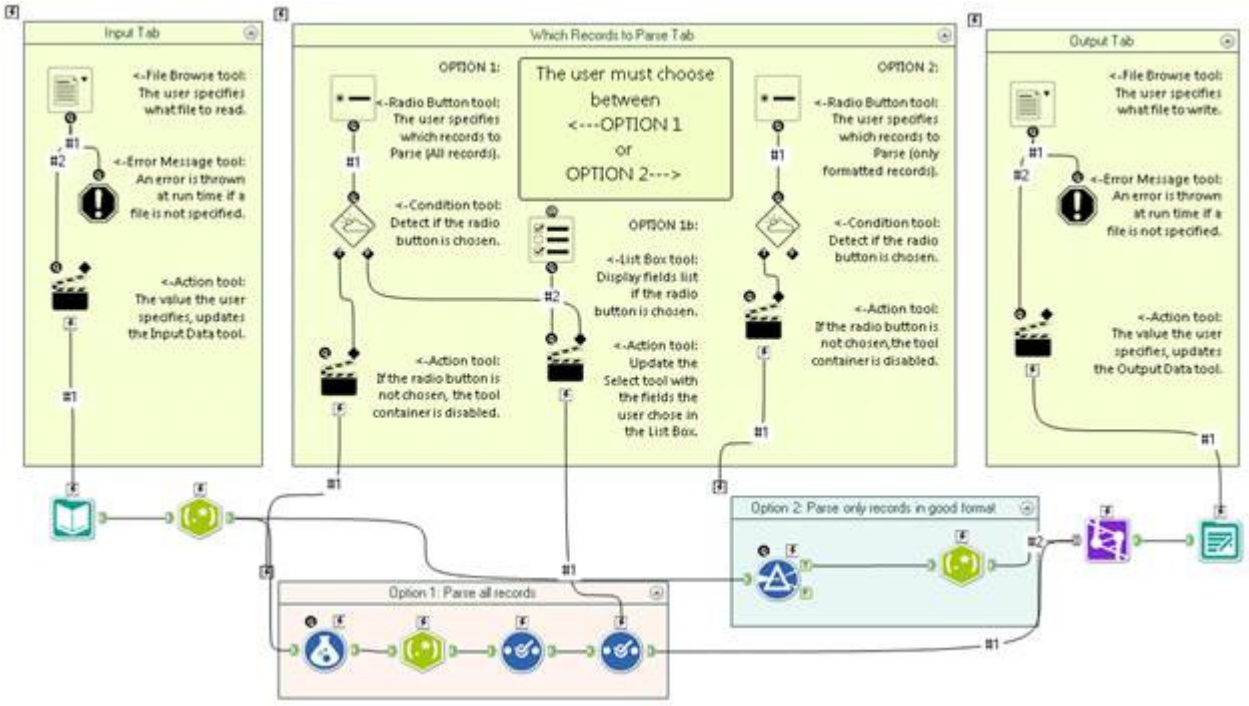
It is best to keep manually configured Union tools out of any App because in most cases, the workflow changes at runtime. Instead of configuring the Union tool in manual mode, insert Select tools into each connection feeding into the Union tool. When configuring the Select tool, rename and reorder fields as necessary taking proper care to NOT include Dynamic/Unknown fields. Configure the Union tool using either Auto Config by Name or Auto Config by Position.

Tool Containers and App Organization

App organization is done in part by using different tool containers for different sections of the app. Tool containers can be color coded based on the function they are highlighting. For example, all the reporting tools are typically the last section of an App.

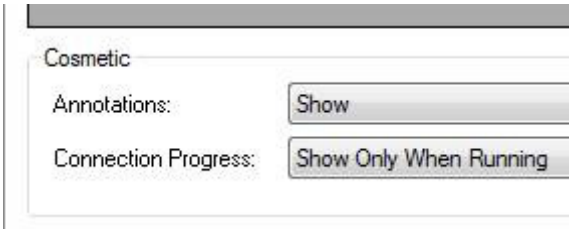
Color-coding tool containers make it easier to understand what an app is doing, especially when viewed at a small zoom. This is helpful when troubleshooting or de-bugging an app that someone else has authored.

Interface tools should be placed in their own containers and color coded as such. For multi-tabbed apps, we have found it helpful to place the tools that make up each tab in their own container.

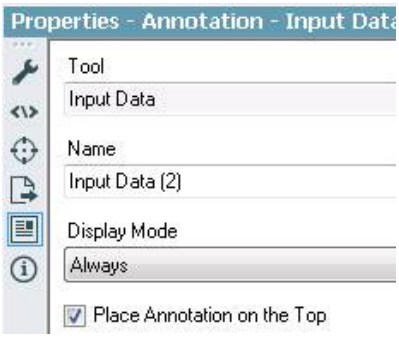


Annotations for Workflows/Apps

Annotations are text boxes that are attached to a specific tool and can be very helpful in describing the role for individual tools. One major benefit is if you move the tool, the annotation moves with it. Under the workflow/App properties, set the Annotations to 'Show'.



If needed, you can also set individual annotations to be placed on either the top or bottom of a tool. This can be achieved by clicking on the tool, then clicking on the 'Annotation' setting, then selecting or deselecting 'Place Annotation on the Top'.



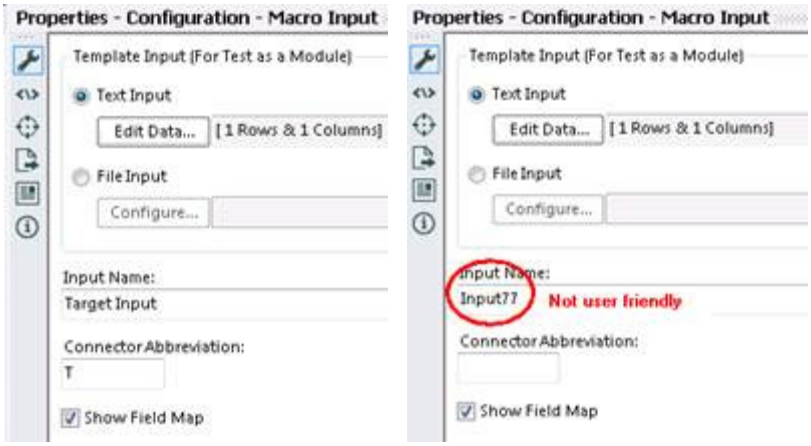
Macro Specific Guidelines

All previous guidelines apply to Macros as well as Apps, with the following exceptions that apply only to Macros.

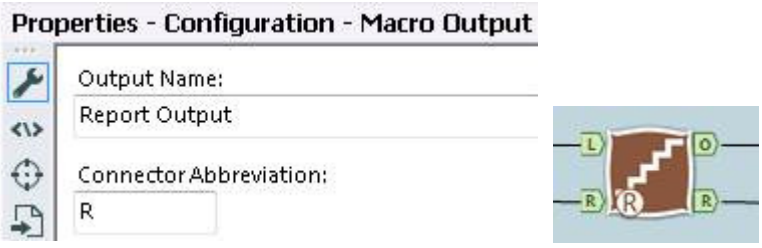
Macro Inputs

Macro Inputs should include data. The inclusion of data makes it so much easier to troubleshoot or debug if something goes wrong. There is a text input embedded inside the Macro Input tool. This is the preferred method for macro inputs. If you need a bigger file to serve as the data input, it should be included with the macro where possible. Data file dependencies for a macro should be named as such:
NameOfMacro.NameOfDataFile.yxdb

Input and Output Names should not have a tool number associated with them. From the Macro Input Properties tab, ensure the name is descriptive enough. The Name specified here will be visible to the user when they configure the Macro tool.



When there are multiple inputs and outputs adding an abbreviation will help the user configure the tool:



Naming Tabs

Tab names should be descriptive for ease of use while the user is configuring the macro tool. Tabs will be visible to the user at configuration time. The default Tab name is 'Questions' – not a very descriptive name and this is usually an oversight when developing a macro (or app) with a single tab.

Supporting Macros

Supporting Macros used in a parent workflow should be present with the parent macro or in a supporting directory and named as such: ***NameOfParent.NameOfSupporting.yxmc***