Web AppBuilder for Image Services 2.0

User Documentation

Updated July 7, 2017

Disclaimer

Applicable for WABIS Version 2.0

Note: The functionality of WABIS has not been exhaustively tested and is not currently covered under ArcGIS Support. Questions or suggestions related to these widgets should be addressed to the GitHub repo associated with the download or sent to ImageManagementWorkflows@esri.com.

Contents

Overview

What is Web AppBuilder for Image Services (WABIS)?

WABIS widgets make it easy for you to incorporate imagery into a wide range of web apps created using ArcGIS Web AppBuilder. Image services widgets have several features, including:

- Managing, analyzing, and visualizing imagery in your app.
- Working within the existing Web AppBuilder for ArcGIS framework.
- Modular design that easily integrates with other GIS and image services widgets.
- Implemented with no coding required.

What do you get?

The Web AppBuilder for Image Services ZIP file downloaded from GitHub contains the following:

- A Web AppBuilder theme designed to work smoothly with imagery
- A variety of image services widgets for Web AppBuilder (see **WABIS Widget Descriptions** below)

What's new in WABIS 2.0?

New concepts

- The IS Image Selector widget streamlines time-enabled image selection.
- App creators can no longer identify auxiliary web maps. All layers needed in the app must be available in the apps primary web map.
- IS widgets are no longer dependent on the IS Layers widget. When the app does not contain IS Layers or IS Layers is not initialized, all widgets will default to using the topmost visible imagery layer in the web map.
- Users can now identify any imagery layer in a web map as a Results layer by adding un underscore ("_") to the end of the imagery layer title. Only IS Compare and IS Export will work on Results layers.

Widgets have been added, renamed, and deprecated

- IS Image Selector and IS Export are new
- IS Spectral Profile is now IS Profile; Scatterplot is now IS Scatterplot; and IS Parameters is now IS Display Parameters
- IS Time Filter, IS Time Select, IS Primary Date and IS Secondary Date have been deprecated and replaced by IS Image Selector
- IS Time Filter Profile has been deprecated and replaced by IS Image Selector and IS Profile
- IS Split Tool has been deprecated and replaced by IS Compare

What are the system requirements?

- Web AppBuilder for ArcGIS Version 2.4 installed (See **Installing WABIS** below)
- ArcGIS Online account

Note: WABIS widgets are compatible with all versions of Web AppBuilder (1.0 – 2.4).

What's in this document?

- An overview of WABIS
- Instructions for how to install WABIS
- Instructions for creating a basic app using WABIS widgets
- A guide to the different widgets' capabilities, configuration parameters, and user interfaces

Introduction

With WABIS, app creators can add the ability to manage, analyze, and visualize imagery to their web apps. Typically, web apps have (1) base maps, which provide general context for the user, and (2) operational layers, which contain key information with which the application interacts. By including imagery in an app's operational layers, the user can access additional context and new options for analysis.

Among other things, image services widgets enable users to do the following in an app:

Manage imagery. Add and change imagery layers; choose mosaic, interpolation, and compression parameters; and set visualization options for your imagery.

Incorporate time. Select dates to display and analyze, filter imagery by date, and use a slider to view images chronologically.

Analyze imagery. Visually compare two layers, calculate change between two dates, and generate spectral profiles.

For WABIS widgets to function properly, a webmap with at least one image layer must be associated with the app in Web AppBuilder when the app is created. If two or more image layers are present, a secondary layer can be added to your app for further visualization and analysis capabilities.

Since image services typically cover the extent of the area of interest (in this case, the app window), there is no need for users to select and work with individual images—the appropriate image will be generated and visualized on-the-fly by the app.

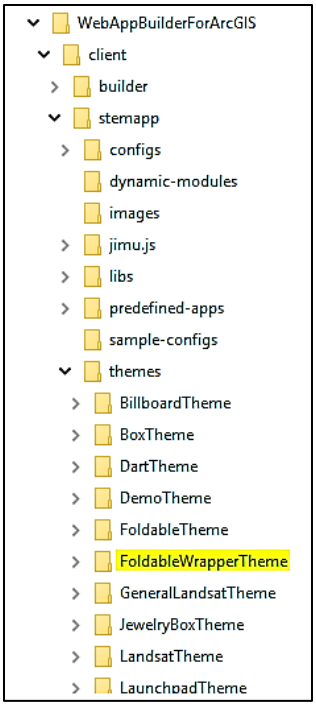
Installing WABIS

Web AppBuilder for ArcGIS (Developer Edition) Version 2.4 should be installed and set up on your computer before proceeding. Instructions for accomplishing this can be found at https://developers.arcgis.com/web-appbuilder/guide/getstarted.htm.

To set up WABIS, complete the following steps.

Note: To update from WABIS 1.0 to WABIS 2.0, first delete the WABIS 1.0 image services widgets from the widgets folder (i.e. …/WebAppBuilderForArcGIS/client/stemapp/widgets). Image services widgets include Scatterplot and those prefixed with an IS.

1. From the WABIS-2.0-Beta branch, download the **WAB-Image-Services-Widgets-WABIS-2.0-Beta ZIP file** by clicking https://github.com/Esri/WAB-Image-Services-Widgets/archive/WABIS-2.0-Beta.zip.
2. **Unzip** WAB-Image-Services-Widgets-WABIS-2.0-Beta.zip to your computer navigate to **…/WAB-Image-Services-Widgets-WABIS-2.0-Beta/ WAB-Image-Services-Widgets-WABIS-2.0-Beta /theme**.
3. Copy the **FoldableWrapperTheme folder** to the themes folder in your Web AppBuilder installation location (…/WebAppBuilderForArcGIS/client/stemapp/themes).



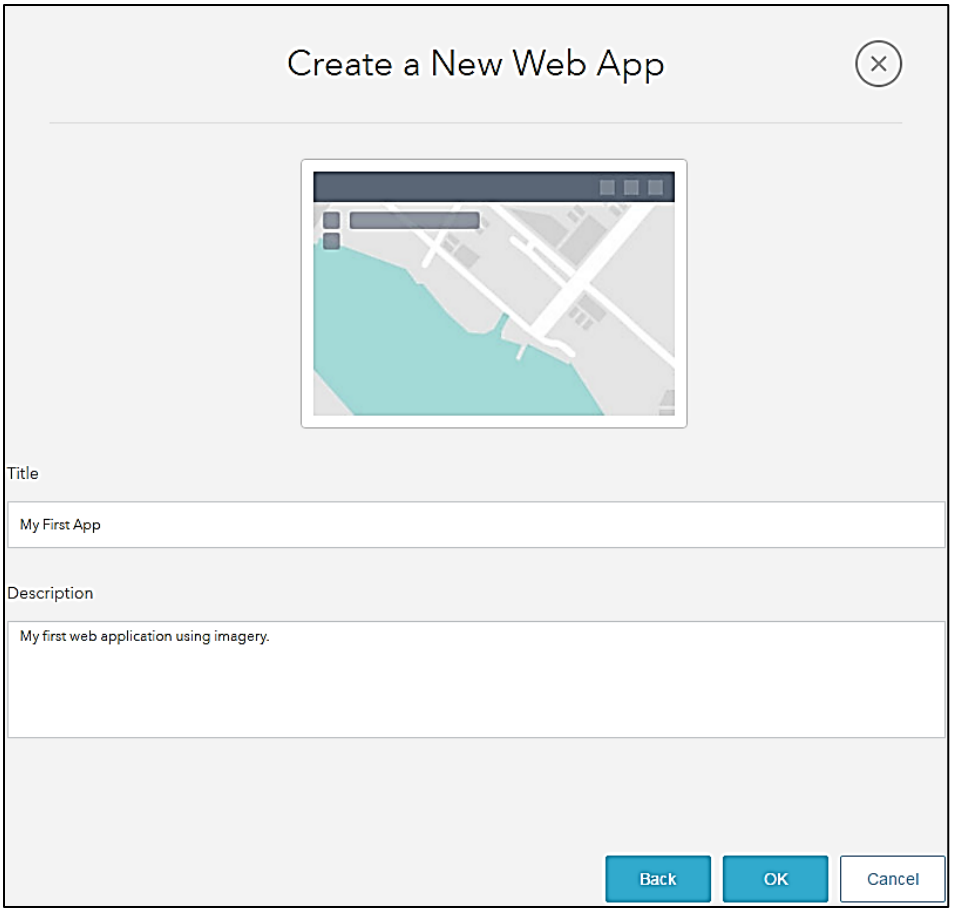
4. Return to the unzipped **WAB-Image-Services-Widgets-master** directory. Open the **imagery_widgets** folder.
5. Copy the folders found in the **imagery_widgets** folder to the **widgets** folder in your Web AppBuilder installation location. (i.e. …/WebAppBuilderForArcGIS/client/stemapp/widgets).
6. If it's still running, **close and re-open Web AppBuilder** to refresh the Image Services themes and widgets before developing an imagery app.

Create a Basic Web App Using WABIS

Note: Restart Web AppBuilder if you just set up WABIS. This will ensure that all the new imagery options are available.

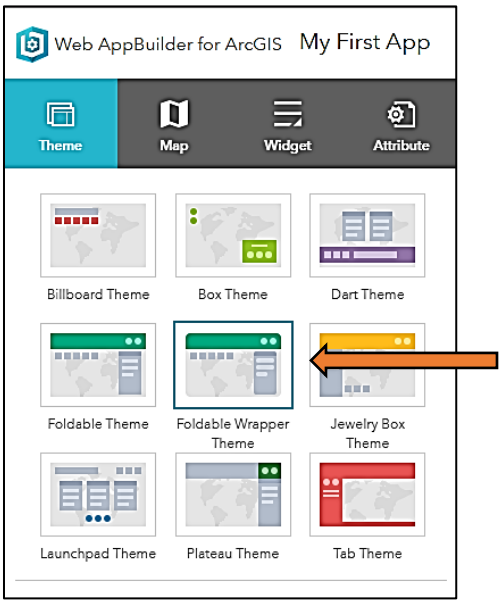
1. Navigate to your **Web AppBuilder installation location** (for example, C:/arcgis-web-appbuilder-2.4/WebAppBuilderForArcGIS/).
2. **Double-click the startup.bat file** to start Web AppBuilder. If prompted, click **Run**.

   Note: The Web AppBuilder interface should automatically open in a browser window. If it doesn't, restart Web AppBuilder, or use your browser to navigate to: http://<**your_machine_name**>:3344/webappbuilder/.
3. On the AppBuilder homepage, click **Create New**, select **Default (2D)**, and click **Next**.
4. Enter an **app name** ("My First App," for example) and a **description**. Select **OK**.

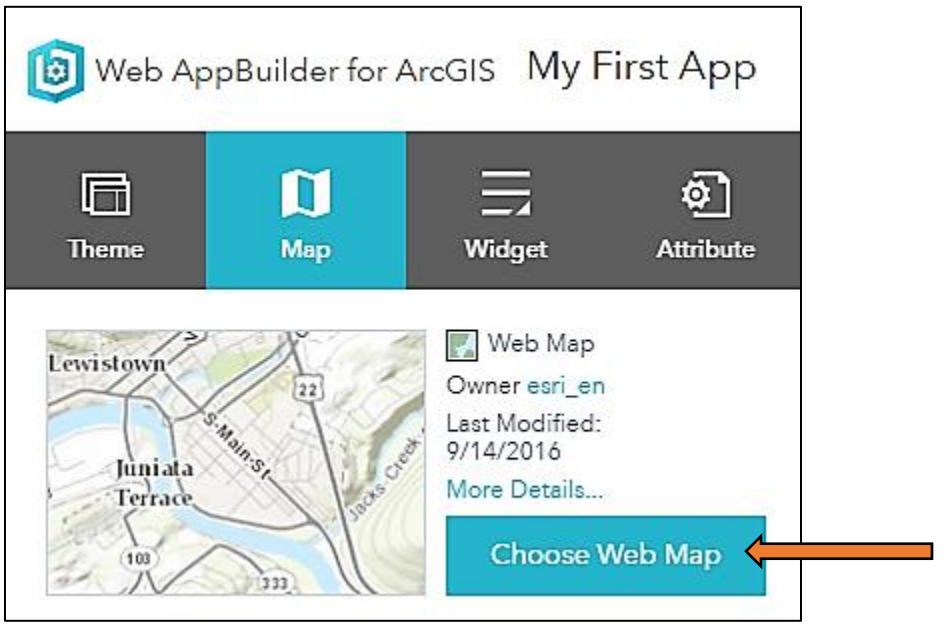


5. Select the **Foldable Wrapper Theme**. Leave the rest of the Theme options as-is.

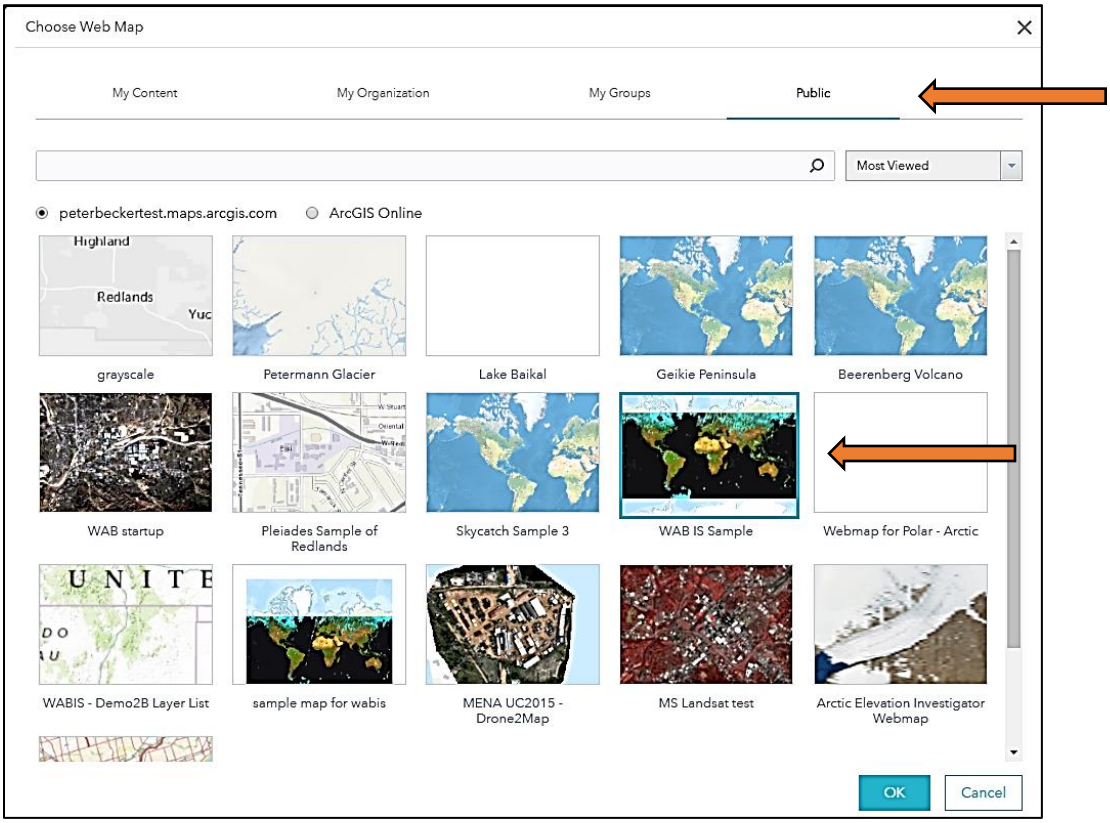
   Note: The Foldable Wrapper Theme ensures proper positioning of widgets specific to image services and won't work when all the widgets are grouped together.



6. Select the **Map** tab at the top left and click **Choose Web Map**.



7. Select **Public** at the top, select **WAB IS Sample** from the available web maps, then click **OK**. (You may need to search for "WAB IS Sample.") The layers in the web map will be added to the map in the builder.

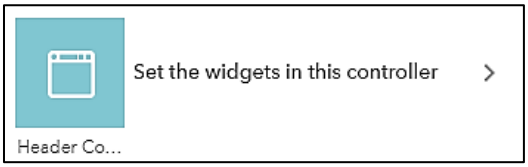


Note: This sample web map has been made public to facilitate this guide. However, any web map with one or more operational imagery layers can be added at this step and will work with WABIS widgets.

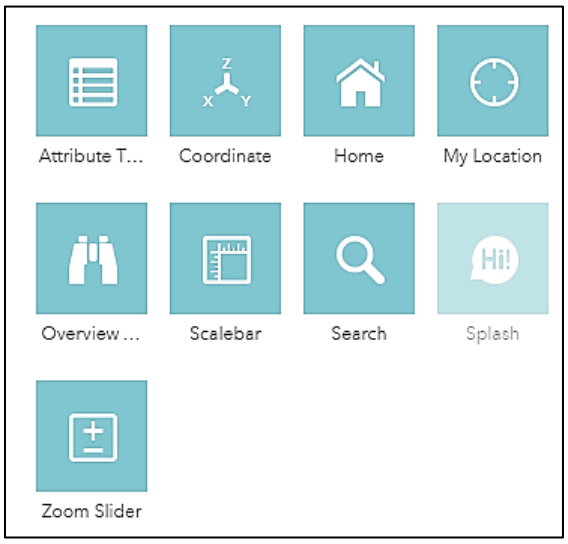
8. Select the **Widget** tab at the top left. Here you can select which widgets to include in your app.

   Note: Web AppBuilder has three widget sections:

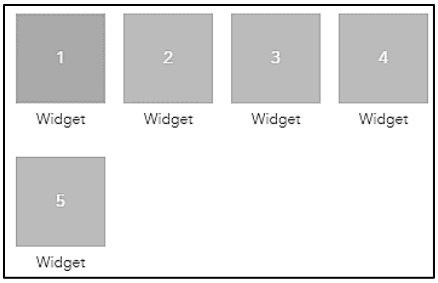
   a. Header Controller Widgets. The first section allows the app creator to add widgets to the header by clicking **Set the widgets in this controller**. By default, the Legend and the Layer List widget controls show up here. Additional widgets can be added to the header as desired.



   b. Theme Widgets. The second section lists the default widgets included on the body of the app. Hover over each to see the widget control highlighted on the app, and to find the icon that turns the widget on/off and the icon that allows the app creator to configure the widget.



   c. On-Screen Widgets. The third section allows the app creator to add up to five widgets that are displayed on the screen of the app itself. Click on any of the numbered gray boxes to see a list of options.



9. Add the IS Renderer widget to your app by **clicking the numbered gray boxes** and looking for widget names that start with "**IS**."

   Note: Image services widgets will all start with IS.

10. Click **Save**.
11. Click **Launch** to view your finished web app in a new browser tab. Your widget icon will appear in the upper left corner of the app (identified with an arrow below).

12. Click the icon to try out your widget to control how your imagery is visualized.
13. Return to the **AppBuilder** tab in your browser. Click **Previews** to explore what your app will look like to a user on a variety of devices.

WABIS Widget Descriptions

WABIS widgets offer a variety of tools for manipulating and analyzing imagery in your applications.

Notes:

- Be sure to load **all** widgets in the header controller before opening any of the header controller widgets in the app.
- Some widgets require the creator of the app to configure them during the first-time load. However, once the widget is configured, the only way to change the configuration is to remove the widget (by clicking the icon) and then reloading it. **While the widget icons have a pencil symbol that suggests the configuration can be edited, this will not work properly for all WABIS widgets.**
- Some problems you encounter with widgets—too much white space, window placement, unable to activate multiple widgets, etc.—may actually be an issue with the theme you've chosen. Learn more about customizing themes in the Web AppBuilder documentation.
- If the title of the imagery layer ends with an underscore, widgets will identify it as a Results layer and it will be available from the secondary layer list if the IS Layers widgets is activated.
- **Always save your application after making changes in the AppBuilder by clicking Save at the bottom left of the screen.**

IS Layers

The IS Layers widget sets and changes the primary and secondary imagery layers in the app. After the app user performs analysis, it can also be used to add the Results layer to the secondary layer list.

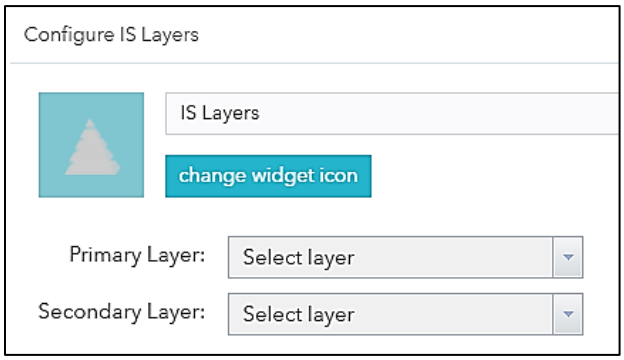
Notes:

- In WABIS 2.0, IS widgets are no longer dependent on the IS Layers widget. When the app does not contain IS Layers or IS Layers is not initialized, all widgets will default to using the topmost visible imagery layer in the web map.
- In WABIS 2.0, the app creator can no longer define an auxiliary image services web map to be associated with the app.
- If the title of the imagery layer ends with an underscore, IS Layers will identify it as a Results layer and it will only be available from the secondary layer list.
- If the user performs analysis within the app, the visibility of the new Results layer can be set using IS Layers.
- Once the IS Layers widget is initialized within the app, all imagery layers in the app will be invisible, except those layers set as primary and secondary layers. As a result, this widget should be initialized when the app is opened.

The IS Layers widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

During configuration, the IS Layers widget gives you the option to choose the default primary and secondary layers used in the app. App users will also be able to select or change these layers from within the app.



The IS Layers widget should be activated automatically when the app starts. This is accomplished during app creation by clicking the bottom left-hand circle to “on” in the IS Layers widget icon.

User interface

This is an image of the IS Layers widget interface, with explanations of the numbered features below:

1.  is the **IS Layers icon**. Click the icon to toggle the widget dialog box on and off.
2. Two **dropdown menus** allow the user to choose primary and secondary layers.

   Note: The dropdown menu will include (1) the image services associated with the Webmap ID identified during IS Layers widget configuration and (2) the image services associated with the app's default webmap.
3. Toggle the layers on and off using the **checkboxes** next to each dropdown menu.
4. Click  to **switch** the primary and secondary layers.
5. Click  to **copy** the primary layer to the secondary layer.

IS Image Selector

The IS Image Selector widget allows the app user to search an imagery layer by a field (chosen during widget configuration), as well as set primary and secondary layers. This is useful when performing change detection or comparing layers.

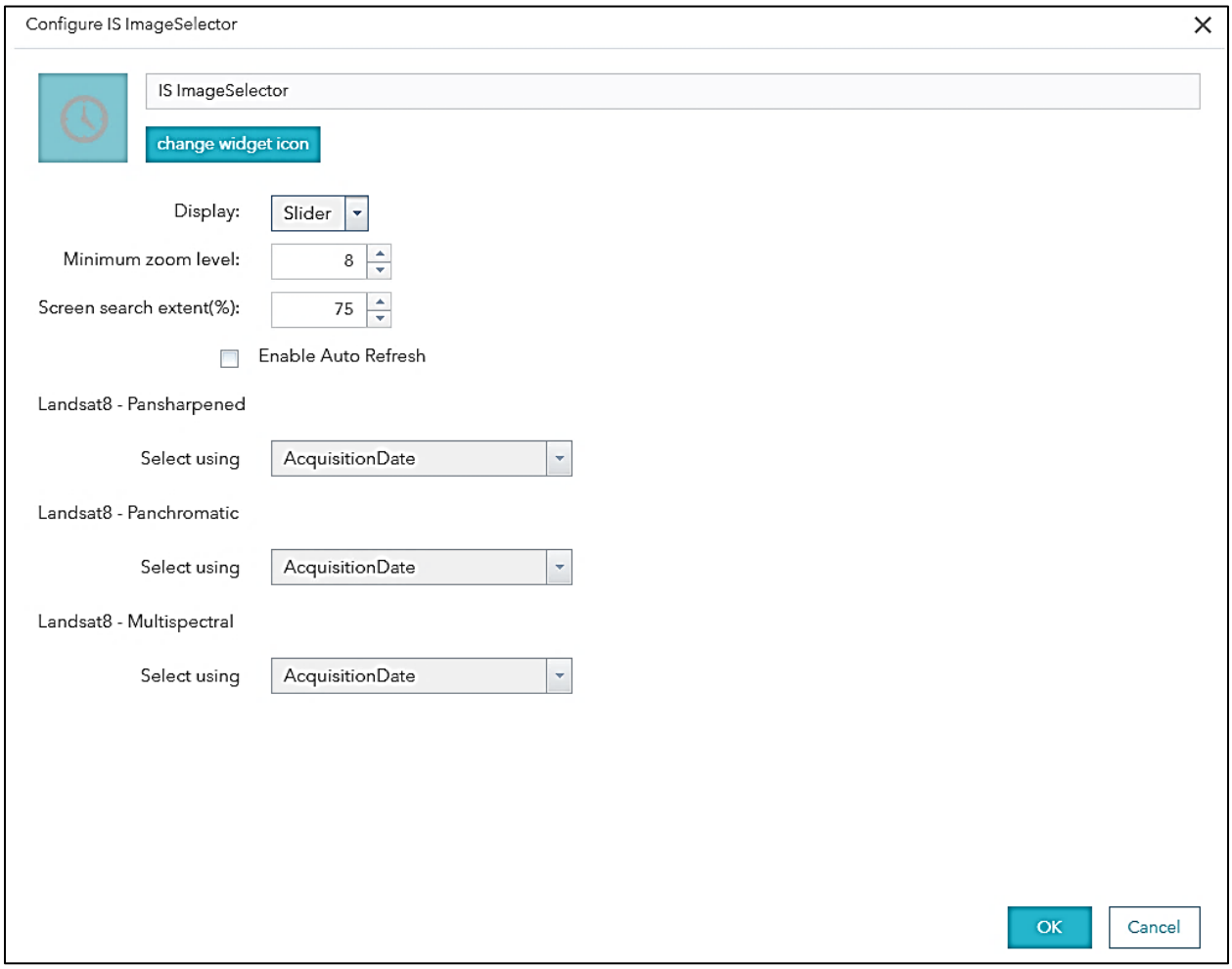
The IS Image Selector widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

During configuration, the IS Image Selector widget gives you the option to choose the following:

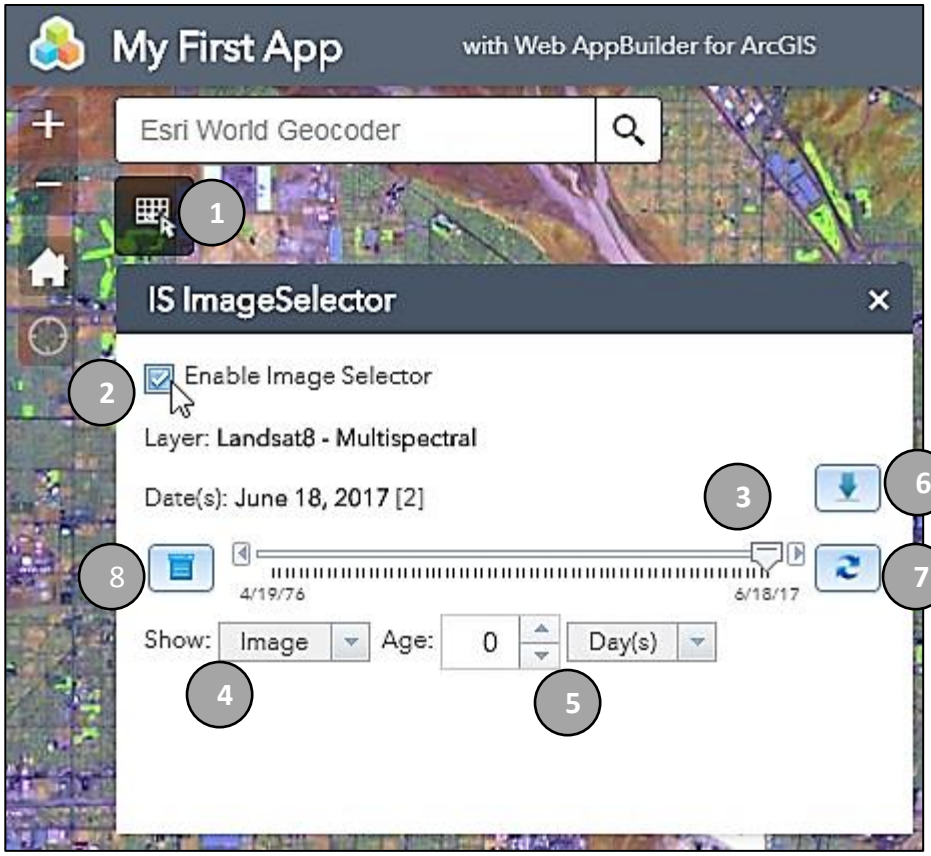
- Display (the app creator can select a Slider, a Dropdown List, or both)
- Minimum zoom level
- Screen search extent (%)

- Enable autorefresh (on or off)
- For each imagery layer available in the app, the field used to select images (Acquisition Date is the default)



User interface

This is an image of the IS Image Selector widget interface, with explanations of the numbered features below:



1.  is the **IS Image Selector icon**. Click this icon to toggle the widget on and off.
2. Click the **Enable Image Selector** checkbox to activate the widget.
3. Use the **slider** to find your desired image (depending on configuration, this may be a dropdown list, or both).
4. Use the **dropdown menu** to select whether you see the image or its footprint.
5. Use the **Age** fields to identify the date range you wish to make visible. The dropdown options include Day(s), Week(s), Month(s), and Year(s).

   Note: This will only be available if the selection field chosen during widget configuration is a date field.

6. Click  to make the selected image the secondary layer.
7. Click  to refresh the time slider.
8. Click  to toggle between the slider and dropdown list (if both are enabled).

IS Change Detection

The IS Change Detection widget allows users to calculate the difference between the primary and secondary layer rasters. The result is added as a new Results layer, which can be added to the dropdown list of image services available to your app using the IS Layers widget. Assuming the primary date is later than the secondary date, increases are shown in green and decreases are shown in magenta.

This widget requires that the IS Image Selector widget (and optionally IS Layers) be loaded in your app.

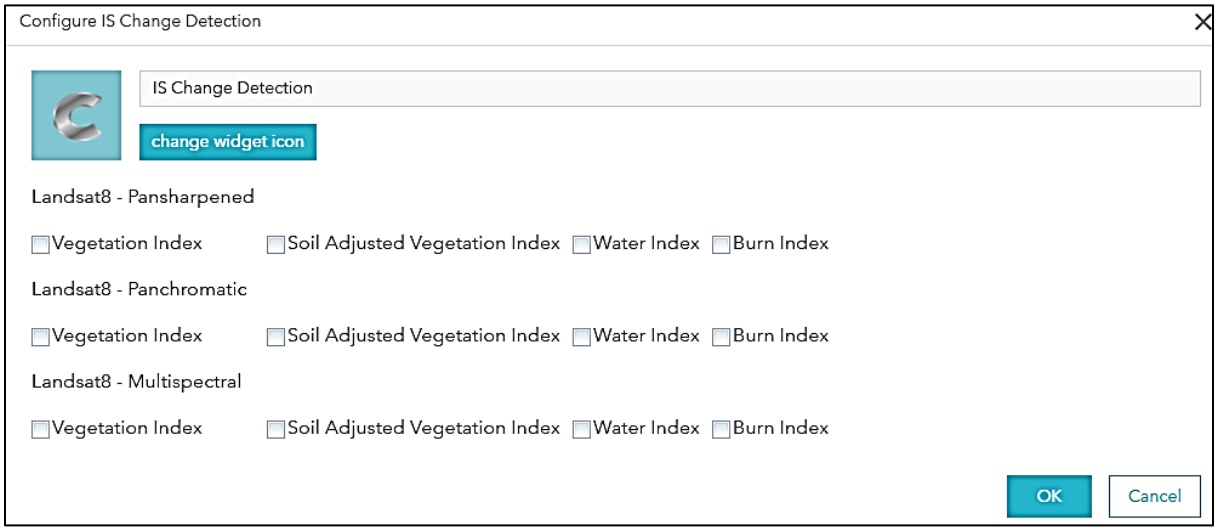
The IS Change Detection widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

During configuration, the IS Change Detection widget gives you the option to add four additional indexes to your app's IS Change Detection widget. Index options for each band include:

- **Vegetation Index** (NDVI) (requires infrared and red bands)
- **Soil-Adjusted Vegetation Index** (SAVI) (requires infrared and red bands)
- **Water Index** (requires green and short-wave infrared bands)
- **Burn Index** (requires infrared and short-wave infrared bands)

If none is selected, only the Difference option will be available in the app.



User interface

This is an image of the IS Change Detection widget interface, with explanations of the numbered features below:

1.  is the **IS Change Detection icon**. Click the icon to toggle the widget dialog box on and off.

   Note: The IS Image Selector widget (and optionally IS Layers) should also be available.

2. Use the **Method** dropdown list to choose the change detection method. Depending on how you configured the widget, options include:
   - **Difference** (no specific bands required)
   - **Vegetation Index** (NDVI) (requires infrared and red bands)
   - **Soil-Adjusted Vegetation Index** (SAVI) (requires infrared and red bands)
   - **Water Index** (requires green and short-wave infrared bands)
   - **Burn Index** (requires infrared and short-wave infrared bands)
3. Use the **Mode** dropdown list to select the type of change detection. Options include:
   - **Difference Image**: Illustrates all the changes in the selected index between the two dates—increases are shown in green, and decreases are shown in magenta.
   - **Difference Mask**: Calculates the difference between the two images. However, you can set the positive and negative difference range to adjust how big the change between two images must be in order to show up in green or magenta.
   - **Threshold Mask**: The user sets a threshold for what counts as change. The app will only identify change from the user-set lower threshold to the upper threshold. You can also set the minimum magnitude of the change you want to detect (for example, detect only changes in NDVI bigger than 0.1).
4. If bands are required for the index you've chosen, they will be identified and auto-populated here (assuming they are available. If not, use the dropdowns to select the appropriate bands.
5. Click **Apply** to initiate change detection.

How to use the widget

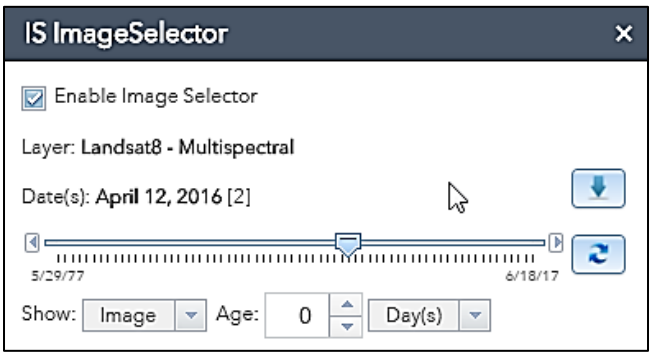
The Change Detection widget is used in conjunction with the IS Image Selector widget (and optionally the IS Layers widget).

Set the earlier date for change detection

1. (Optional) Activate the **IS Layers** widget to select the layer you wish to use for change detection. Select the primary layer using the dropdown menu. If no layer is selected, or IS Layers is not available, the IS Change Detection widget will default to the topmost visible layer.

   Note: Once IS Layers is activated, the user must select a primary layer or no image layers will be visible or available in the app.

2. Activate the **IS Image Selector** widget. Click on the **check box** to display the slider.



3. Use the Time Filter slider to **select an earlier scene** for change detection. **Click** on the  button to set the scene you've selected as the secondary layer.

Set the later date for change detection

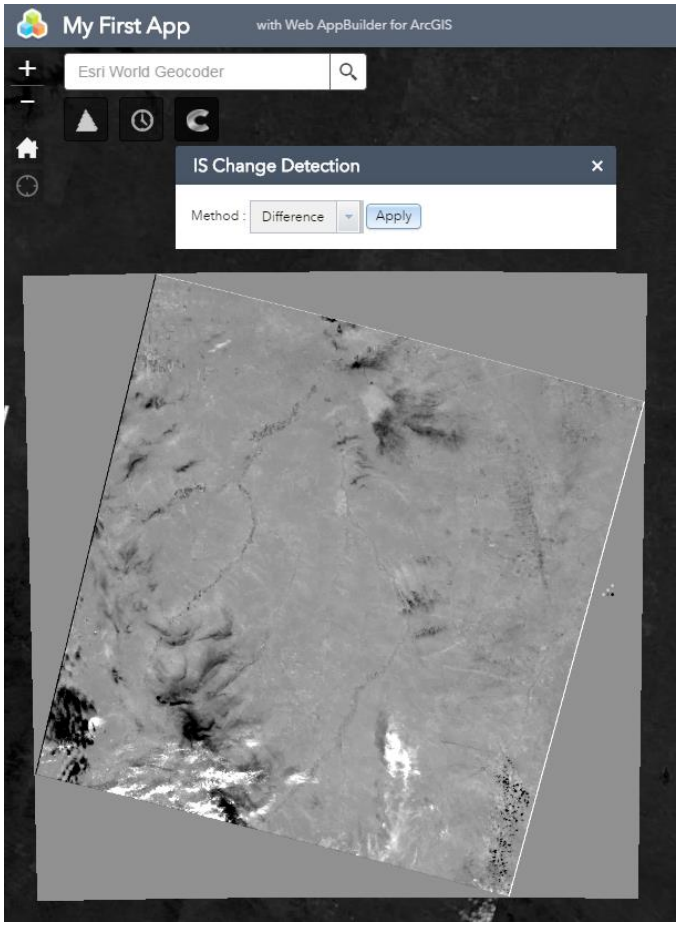
4. Using the time slider again, **select a later scene** for change detection. The date on the time slider will be automatically set as the primary date—no need to click on anything.

   Note: The two images selected must overlap.

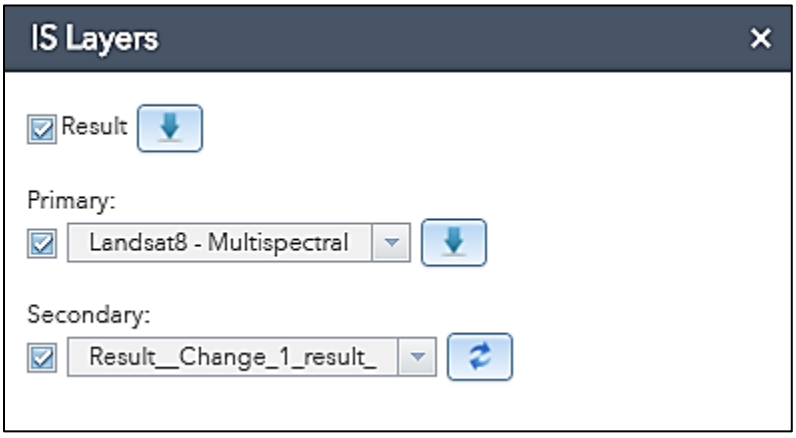
Detect change

5. Open the **Change Detection** widget. Choose the desired change detection method, mode, and bands (if needed) using the dropdown menus.

6. Click **Apply**. The result is then visualized in the app as a new **Results** layer, shown below.



7. (Optional) Activate the **IS Layers** widget. **Click** on  to add the Results layer to the secondary layer list in your app.



IS Compare

The IS Compare widget uses a vertical swipe to compare the topmost imagery layer with a secondary imagery layer. Optionally, users can also use transparency or a horizontal swipe to compare the primary and secondary imagery layers with a Results layer.

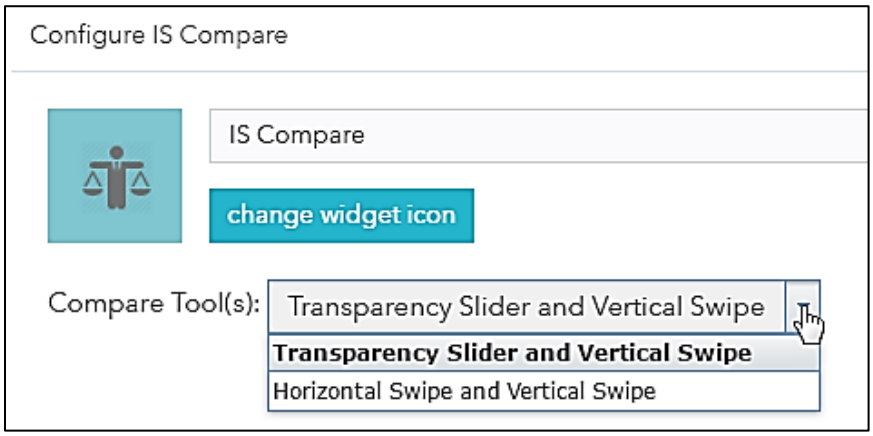
The IS Compare widget should be added as an On-Screen widget or a Header Controller widget. It is often used in conjunction with IS Image Selector and IS Layers widgets.

Configuration

During configuration, the app creator can select between two options for the end user: 1) Transparency Slider and Vertical Swipe, or 2) Horizontal Swipe and Vertical Swipe.

Vertical swipe will always be used to compare the primary and secondary layers.

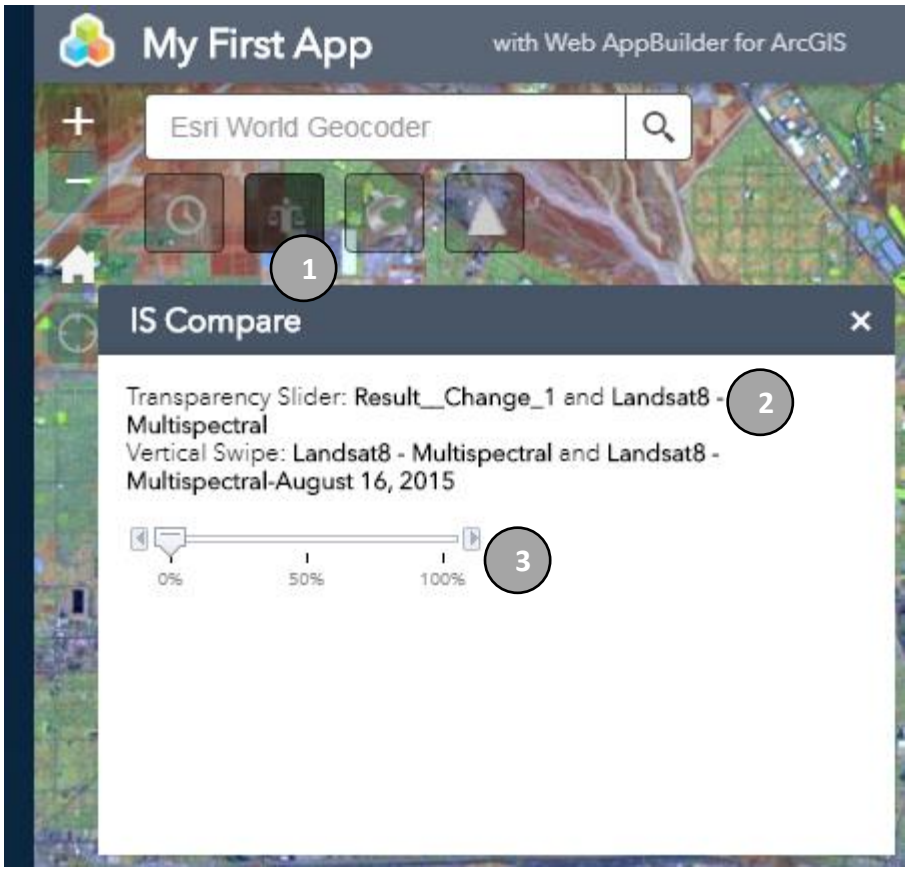
If a Results layer is available, the transparency slider OR the horizontal swipe can be used to compare the Results layer with the primary and secondary layers.



Note: If you (1) want the user to be able to compare primary, secondary, AND Results layers, and (2) the user will perform analysis and add the ensuing Results layer to the layer list beforehand, then the IS Layers widget must be activated automatically when the app starts. This is accomplished during app creation by clicking the bottom left-hand circle to "on" in the IS Layers widget icon.

User interface

This is an image of the IS Compare widget interface, with explanations of the numbered features below:



1.  is the **IS Compare icon**. Click the icon to toggle the widget dialog box on and off.
2. The layers illustrated using the Vertical Swipe and either the Horizontal Swipe or Transparency Slider are identified.
3. When configured for transparency, use the **Transparency slider** to set the degree of transparency. (If Swipe is selected, the slider will disappear.)

How to use the widget

The Compare widget is used in conjunction with the IS Image Selector widget, the IS Layers widget, or both.

You can use the IS Image Selector widget to compare imagery from different dates (*See IS Image Selector, below*). You can also use IS Layers to compare different layers, or use both widgets together to compare different dates from different layers.

Additionally, if you wish to compare and primary, secondary, AND Results layers, you will perform your analysis, use IS Layers to add the Results layer to your Secondary Layer list, then activate the IS Compare widget to see all three layers at once. The IS Layers widget must be activated when the widget opens.

Note: If the primary and secondary layers are the same, the vertical swipe will compare the primary layer and the basemap.

IS Display Order

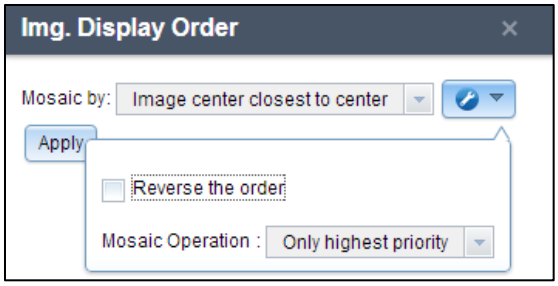
The IS Display Order widget sets the mosaic rule for the primary layer, which determines which image in the mosaic will be visualized if images overlap.

The IS Display Order widget should be added as an On-Screen widget or a Header Controller widget.

User interface

This is an image of the IS Display Order widget interface, with explanations of the numbered features below:

1.  is the **IS Display Order icon**. Click the icon to toggle the widget dialog box on and off.
2. The **Order By** dropdown menu lists the mosaicking options, some of which also take require additional selections. Options include:
    - Only scale
    - An attribute: The user must select an attribute to sort by and enter a priority value (i.e. the value from the chosen attribute field you wish to appear on top).
    - Image center closest to center
    - Fixed order with most NW on top
    - Sensor location closest to center
    - Closest to View Point: User must click a view point on the map, then click **Apply**.
    - Defined seamlines
    - A list of images: The user must enter image IDs as a comma separated list.
3. Click on  to choose the **Mosaic Operation** available for your selected mosaic method (a **checkbox** that will reverse the order of your chosen mosaic method will also be available).



4. Click Apply to visualize your primary layer using the selected mosaicking parameters.

IS Display Parameters (formerly IS Parameters)

The IS Display Parameters widget allows users to set interpolation and compression for the primary layer.

The IS Display Parameters widget should be added as an On-Screen widget or a Header Controller widget.

User interface

This is an image of the IS Display Parameters widget interface, with explanations of the numbered features below:

1.  is the **IS Display Parameters icon**. Click the icon to toggle the widget dialog box on and off.
2. The **Interpolation** dropdown menu lists the interpolation options for imagery layers, including:
    - Default (as defined by the image service)
    - Bilinear
    - Cubic convolution

- Majority
- Nearest neighbor

3. The **Compression** dropdown menu lists the compression options for imagery layers, including:
    - Default (as defined by the image service)
    - None
    - Jpgpng
    - Png
    - Jpg
4. Use the **Quality** dropdown to determine the acceptable level of data loss due to compression (in increments of five).
5. Click 

to save the selected interpolation and compression parameters.

IS Export

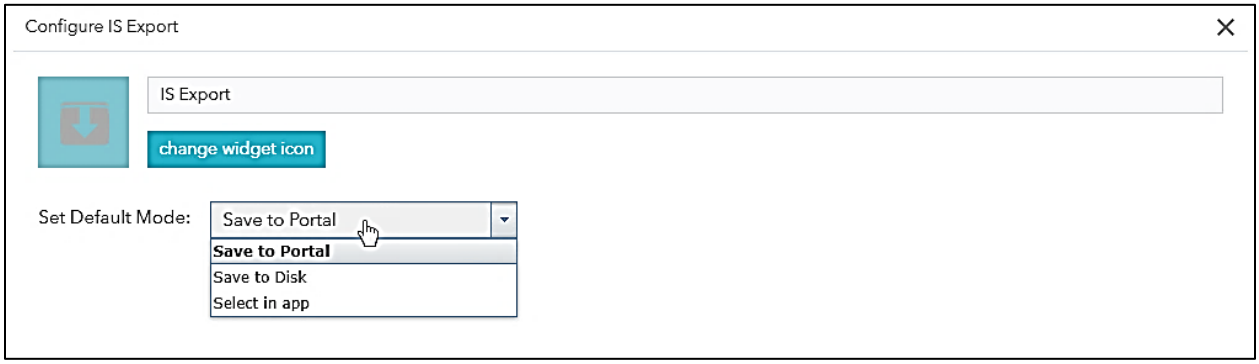
The IS Export widget allows the app user to either save the topmost visible imagery layer to the app user's content in Portal, or to export the same image locally as a TIFF.

The IS Export widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

During configuration, the IS Export widget gives you the option to choose the default in-app save option: Save to Portal, Save to Disk, or Select in app. (Both options will be available in-app regardless.)

In Save to Portal mode, the widget will save the topmost visible imagery layer to the app user's content in Portal as an imagery layer. In Save to Disk mode, the widget will export the image in TIF format to the app user's computer.



User interface

This is an image of the IS Export widget interface, with explanations of the numbered features below:

1.  is the **IS Export icon**. Click the icon to toggle the widget on and off.
2. Use this **dropdown** to select the widget's mode: Save to Portal, or Save to Disk.
3. Complete the **text boxes** with an image title, description, and tags.
4. Click **Submit** to either save the image as an imagery layer to the user's content, or export the image to the user's computer as a TIFF.

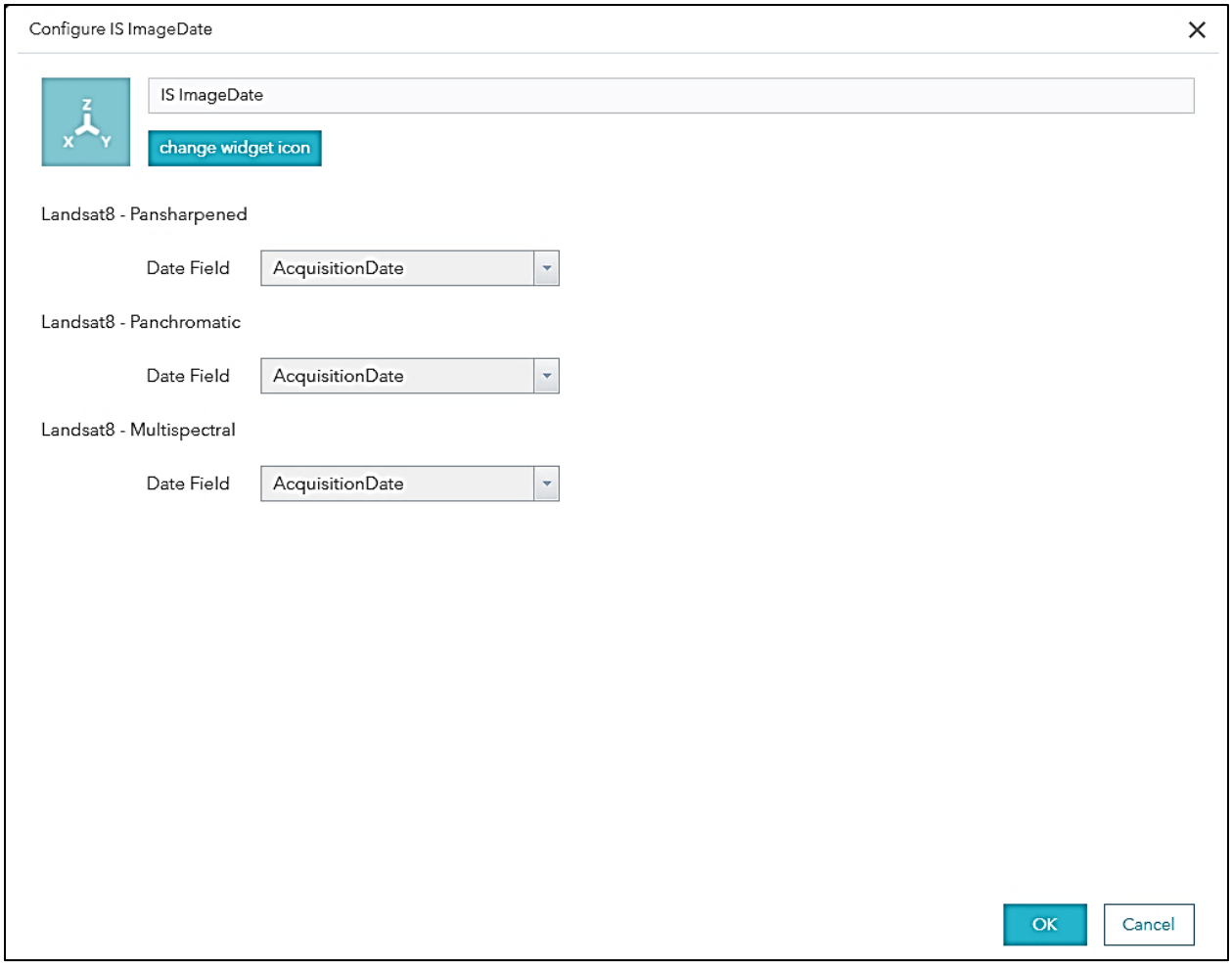
IS Image Date

The IS Image Date widget will show the date of the primary or secondary layer (whichever is visible in the app).

The IS Image Date widget should be added as an On-Screen widget or a Theme widget (not as a Header Controller widget).

Configuration

During configuration, the IS Image Date widget gives you the option to choose the date field that will be displayed for each layer available in the app.



The IS Image Date widget should be activated automatically when the app starts. This is accomplished during app creation by clicking the bottom left-hand circle to "on" in the IS Image Date widget icon.

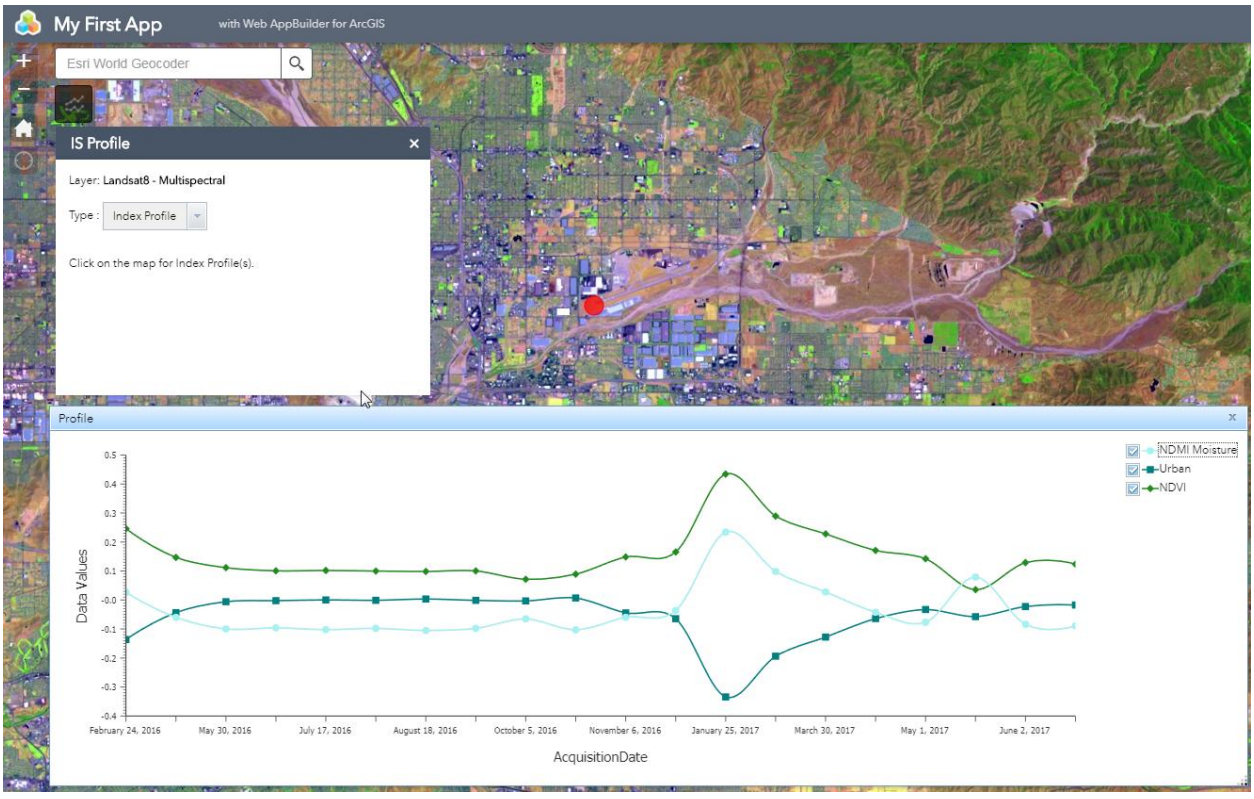
User Interface

This is an image of the IS Image Date widget interface, with explanations of the numbered features below:

1.  is the **IS Image Date** icon. Click the icon to toggle the widget dialog box on and off. If you configured the widget to load automatically when the app starts, the widget will automatically be toggled on.
2. The **image date** of the visible layer will appear below the icon when the widget is toggled on.

IS Profile (formerly IS Spectral Profile)

The IS Profile widget shows a spectral or index profile (NDVI, NDMI Moisture Index, or Urban Index) for a selected point (in red, below) on the primary layer.



The IS Profile widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

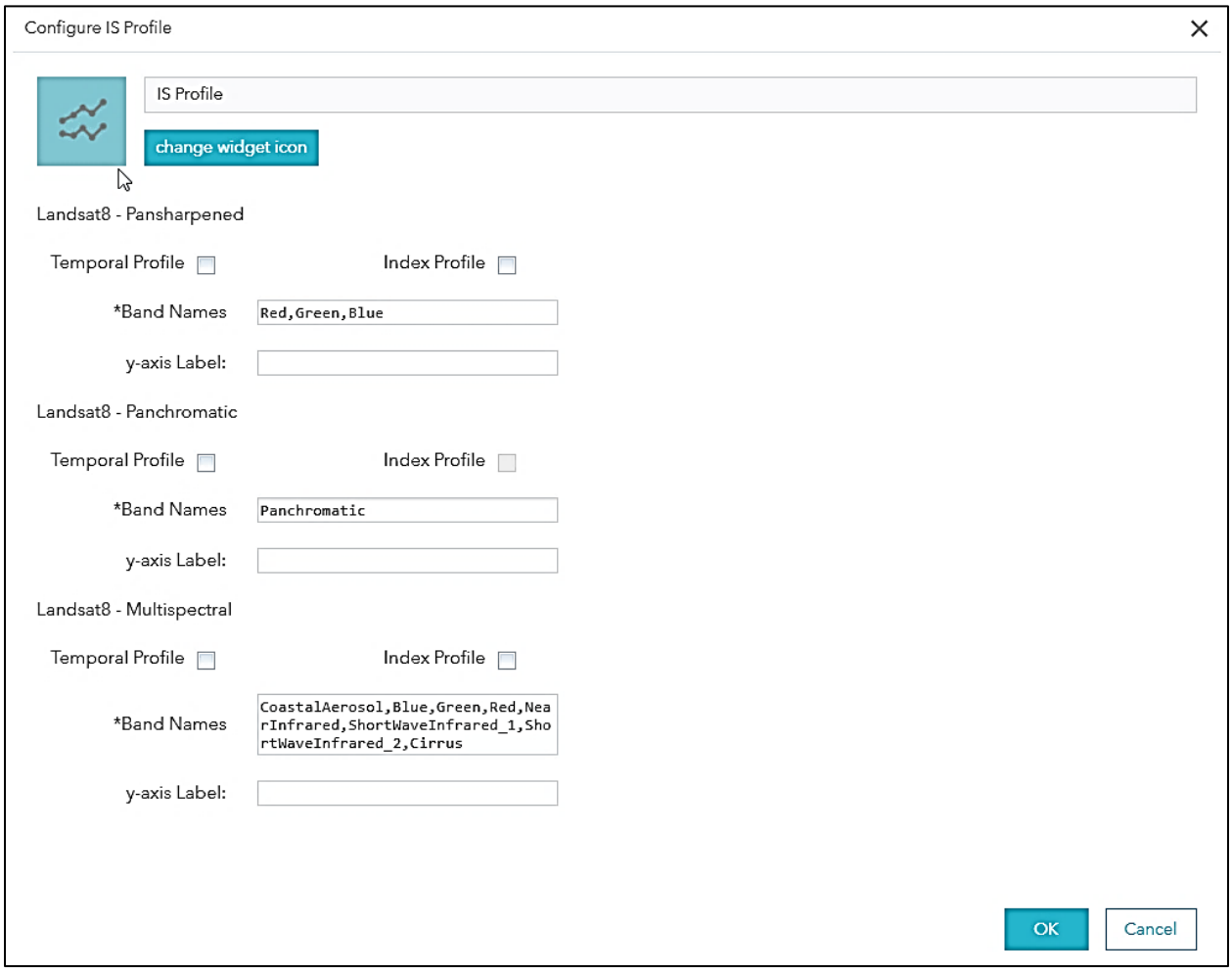
During configuration, the app creator can identify the **band names** to be used for each available imagery layer.

Additionally, for each imagery layer, the app creator should specify whether a temporal profile and/or an index profile will be enabled using the checkboxes.

- If **temporal profile** is enabled for a layer, the app creator should choose the date field to be used by the widget. The widget will choose AcquisitionDate by default.
- If **index profile** is enabled for a layer, the app creator should specify a date field, as well as the near-infrared, red, and shortwave infrared bands.

  Note: If a date field or the required index bands are unavailable, the index profile option will not be available.

Finally, the app creator can also specify the **y-axis label** to be used for each layer when the specified profile is displayed.



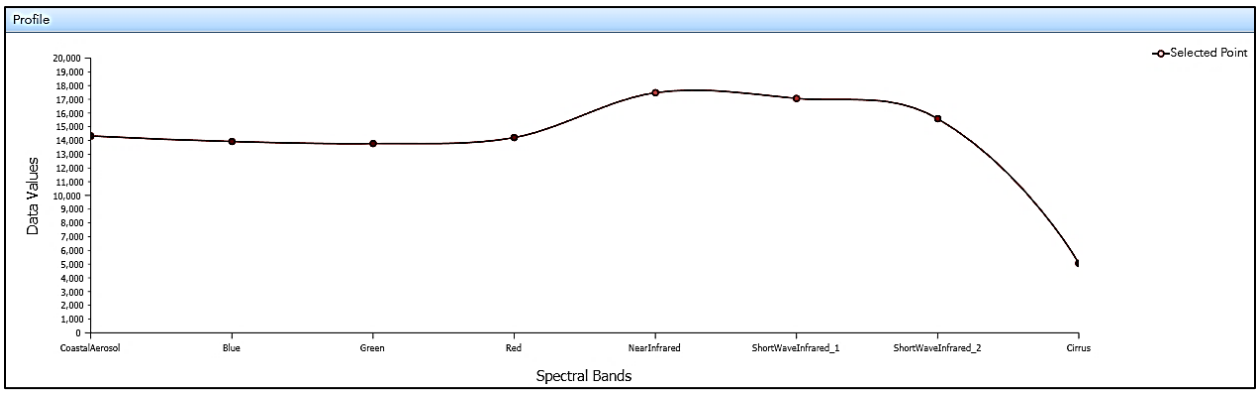
User interface

This is an image of the IS Profile widget interface, with explanations of the numbered features below:

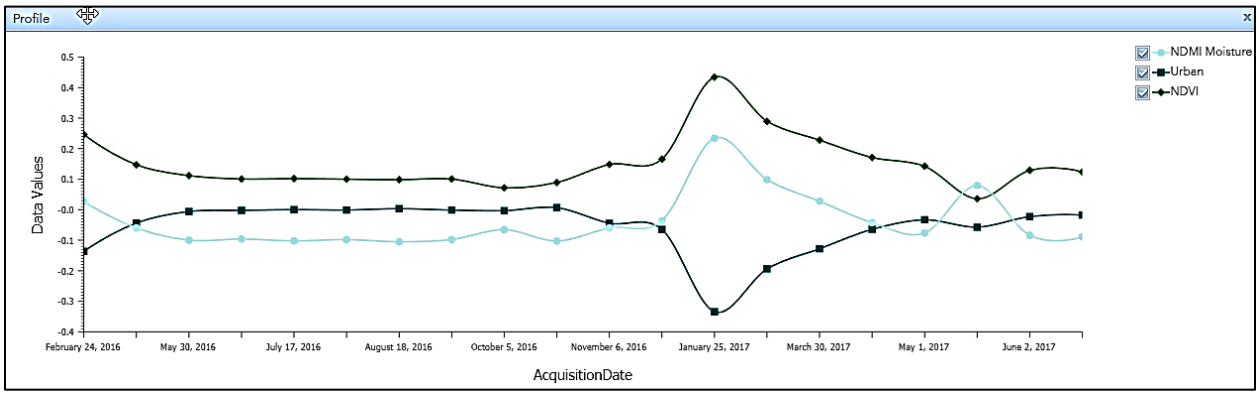
1.  is the **IS Profile icon**. Click the icon to toggle the widget dialog box on and off.
2. The **dropdown** menu allows you to choose between Spectral Profile or Index Profile.

How to use the widget

1. Select Spectral Profile or Index Profile using the dropdown menu.
2. **Click** on a point on the map. The resulting spectral profile will look something like this:



And the resulting index profile will allow you to add NDMI Moisture, Urban, and NDVI index profiles:



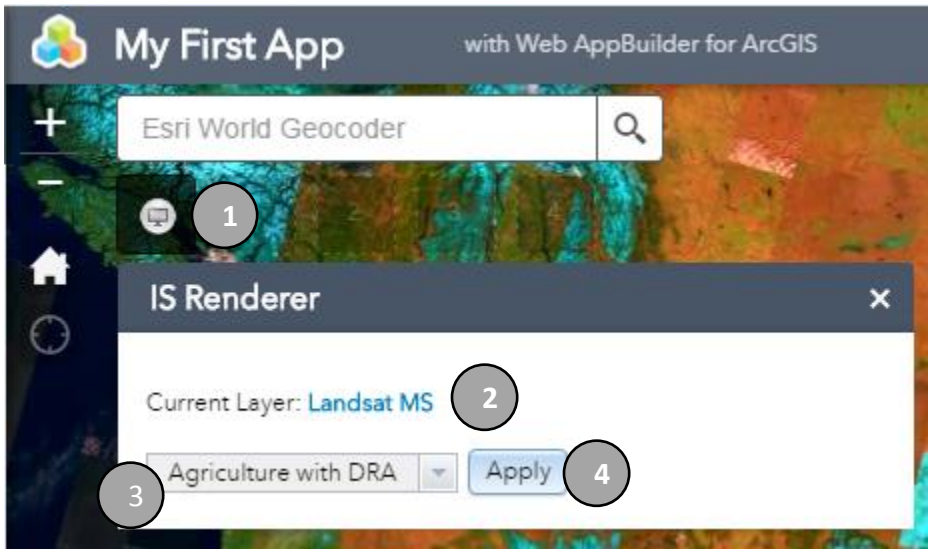
IS Renderer

The IS Renderer widget sets the service functions and stretch on the primary layer. The dropdown menu in the widget is automatically populated with the service functions associated with the primary layer.

The IS Renderer widget should be added as an On-Screen widget or a Header Controller widget.

User interface

This is an image of the IS Renderer widget interface, with explanations of the numbered features below:



1.  is the **IS Renderer icon**. Click the icon to toggle the widget dialog box on and off.
2. **Current Layer** identifies the layer being visualized by the widget.

   Note: This will be the primary layer, which can be changed using the IS Layers widget.

3. The dropdown menu lists the available visualization options based on the service functions set up in the associated imagery webmap.

4. Click  to visualize your primary layer using the selected option from the dropdown menu.

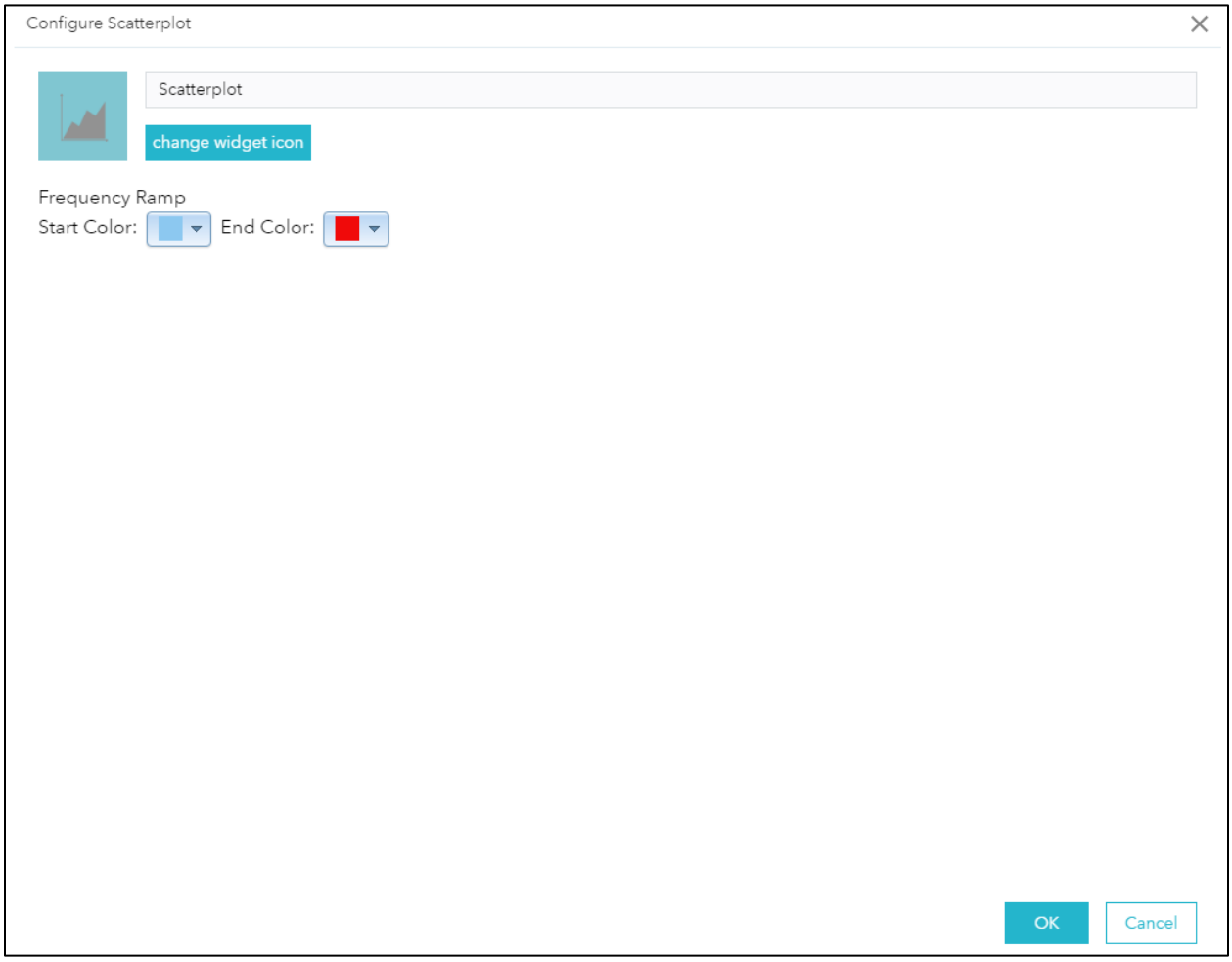
IS Scatterplot (formerly Scatterplot)

The IS Scatterplot widget allows the user to select two bands from the image service layer and plot their values on a graph. The user can (1) highlight a region on the map by drawing and selecting the points on the plot, (2) click on the map to highlight that point on the scatterplot and get the pixel values, and (3) define an optional area of interest for which the plot should be drawn. Additionally, if both bands selected are the same, the widget will plot the frequency of that band.

The IS Scatterplot widget should be added as an On-Screen widget or a Header Controller widget.

Configuration

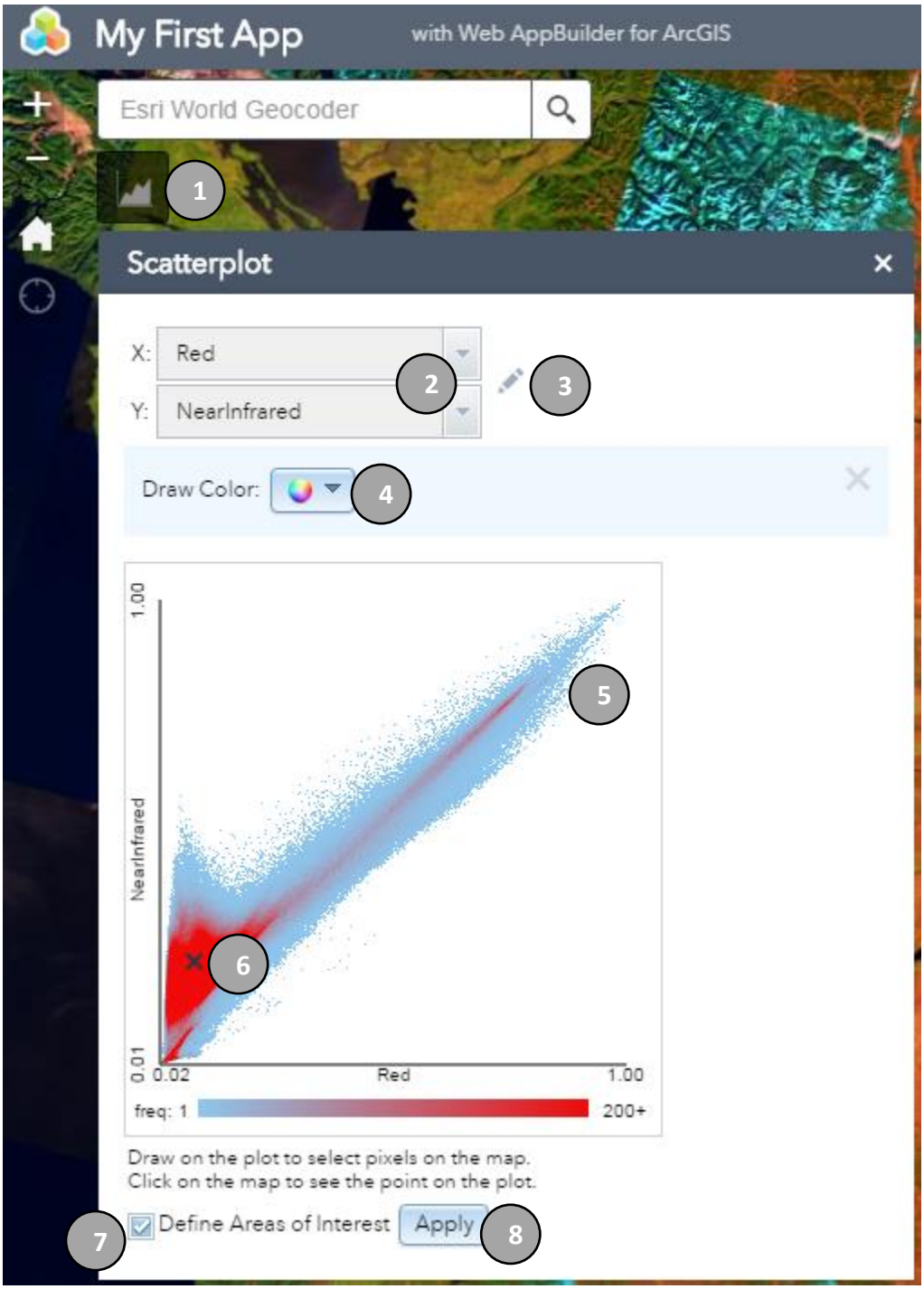
During configuration, the user can specify the start and end colors for the color ramp used to illustrate how often different values occur in the scatterplot.



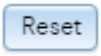
User Interface

This is an image of the Scatterplot widget interface, with explanations of the numbered features below.

Note: The Scatterplot window may need to be resized manually if the Define Area of Interest checkbox is not visible initially.



1.  is the **Scatterplot** icon. Click the icon to toggle the widget dialog box on and off.
2. Use the **drop down lists** to select the bands to be plotted on the X and Y axes. If both bands selected are the same, the widget will plot the frequency of that band.
3. Click the  icon to view the **draw color**.
4. Use the  button to change the **draw color** from the default cyan (will appear after you click the  icon).
5. The **scatterplot** illustrates the band value for each pixel in the AOI. The color at each point indicates the number of pixels in the AOI with that given value, based on the color ramp chosen during configuration.
6. Clicking on the map places a **marker** on the scatterplot that indicates the band values of the clicked point. Hover over the marker to view the band values.
7. Check the **Define Areas of Interest checkbox** to draw on the map and select a custom area of interest.
8. Click on the 

button to **redraw** the plot for the defined AOI. Click the 

button that appears to **remove** the area of interest.

IS Classification (broken)

The IS Classification widget allows the user to draw features, associate categories, and perform classification. The result is added as a new "Result Layer," which can be added to the dropdown list of image services available to your app, or can be saved to ArcGIS Online.

Note: This widget has known issues. It will not function in ArcGIS 10.4+, and the output using ArcGIS 10.3 will be incorrect.

Widgets deprecated in WABIS 2.0

Deprecated widget	Widget(s) to use instead
IS Time Select	IS Image Selector
IS Split Tool	IS Compare
IS Time Filter Profile	IS Image Selector and IS Profile
IS Time Filter	IS Image Selector
IS Primary Acquisition Date	IS Image Selector
IS Secondary Acquisition Date	IS Image Selector