|  |  |
| --- | --- |
| esrihdr  ***ESRI*** | ***Web App Builder for Image Services (WABIS) V 1.0Beta*** |
| * ***Overview*** |
| Item | Details | |
| **Title** | **Web App builder v.1.0.0** | |
| **Authors** | **Shitij** | |
| **Creation Date** | **23/06/2014** | |
| **Last Updated** | **27/05/2015** | |
| **Software Release** | **v.1.0Beta** | |

Revision & Sign-off sheet

**Change Record**

| Date | Author | Change Reference |
| --- | --- | --- |
| 23/06/2014 | Shitij | Base draft of the Web App Builder User Doc. |
| 28/06/2014 | Peter | General review |
| 07/07/2014 | Shitij | Limitations/Comments added |
| 13/08/2014 | Shitij | Information updated |
| 15/04/2015 | Kushagra | Information updated |
| 27/05/2015 | Kushagra | Information updated |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[Revision & Sign-off sheet 1](#_Toc395706817)

[Overview 4](#_Toc395706818)

[Understanding the framework for Image Service widgets 5](#_Toc395706819)

[Get ArcGIS Webapp Builder 5](#_Toc395706820)

[Setup ArcGIS WebApp Builder 5](#_Toc395706821)

[Create a basic web app in the builder 6](#_Toc395706822)

[Create a basic web app (for image services) in the builder 7](#_Toc395706823)

[Widget Usage 12](#_Toc395706824)

[ISLayers 12](#_Toc395706825)

[Overview 12](#_Toc395706826)

[Configuration 12](#_Toc395706827)

[ISDisplay 13](#_Toc395706828)

[Overview 13](#_Toc395706829)

[ISDisplayOrder 13](#_Toc395706830)

[Overview 13](#_Toc395706831)

[ISParameters 13](#_Toc395706832)

[Overview 13](#_Toc395706833)

[ISTimeFilter 14](#_Toc395706834)

[Overview 14](#_Toc395706835)

[ISCompare 14](#_Toc395706836)

[Overview 14](#_Toc395706837)

[ISPrimaryAcquisitionDate 15](#_Toc395706838)

[Overview 15](#_Toc395706839)

[ISSecondaryAcquisitionDate 15](#_Toc395706840)

[Overview 15](#_Toc395706841)

[ISSpectralProfile 15](#_Toc395706842)

[Overview 15](#_Toc395706843)

[ISChangeDetection 16](#_Toc395706844)

[Overview 16](#_Toc395706845)

[ISMensuration 16](#_Toc395706846)

[Overview 16](#_Toc395706847)

[ISClassification 17](#_Toc395706848)

[Overview 17](#_Toc395706849)

**WebApp Builder for Image Services**

# Overview

The WebApp Builder Image Services widgets have been built to provide a simplified way for users to incorporate imagery functionality into web apps. The design goal has been to create a set of widgets that work together, can be incorporated into a wide range of applications and do not clutter the applications with functionality that is not required.

Typically web applications have base maps that provide general context and operational layers which contain the key information with which the application interacts. In many cases it is advantageous to have intelligent imagery as a layer between the base map and the operational layer and be able to operate efficiently on these images. The imagery layer can provide imagery or raster data such as elevation that provides additional context or can be analyzed to extract additional data.

When working with imagery in such apps the following rules typically hold

* Typically image services cover the extent of the Area of Interest, so users should not need to select and work with individual image.
* Although a user may be interested in many different image services typically only one needs to be shown. Having a legend showing many layers that can be turned on simultaneously is not intuitive
* There are many cases where two image layers need to be viewed. Typical examples are when looking at change between images.
* The date of the imagery is important to most applications.

The WABIS widgets enable users to easily add imagery capabilities to their apps. The key concept is that a Primary and optionally Secondary image layer is added to the app. The WABIS tools all interact with these two layers. Using very simple interfaces a user can change the Primary image to a new image service, by simply selecting from a drop down list of preconfigured image layers.

The list of layers drop down layers does not come from the current web map, but instead the Image Layers tool takes as input the ID of a separate WebMap. This enables organizations to create a webmap that contains a list of all the relevant image services for their applications and main this list independently from the webmaps that are used to define the applications. This method makes is also simpler to add the imagery functionality to apps without needing to modify the associated webmap with a list of possible imagery layers.

The properties of the primary layer can be change these include changing functions or setting properties such as the mosaic method. A simple time filter can be used to look through temporal selections of imagery be better review change. The widgets enable the primary and secondary layers to be quickly changed. Tools also enable the transparency or swipe between the layers to be modified to enable visual comparison. A range of additional tools are also being developed to perform change detection, which will create a third ‘Results’ layer. Tools for mensuration as well as tasks such as supervised classification will be developed based on the concept of working with a primary and secondary layer.

# Understanding the framework for Image Service widgets

The widgets for image services are dynamic and need to communicate with each other. In order to achieve this nature, they assume a framework for the app on which they are added. The widgets assume that following layers are present on app start up.

* Primary Imagery Layer
* Secondary Imagery Layer

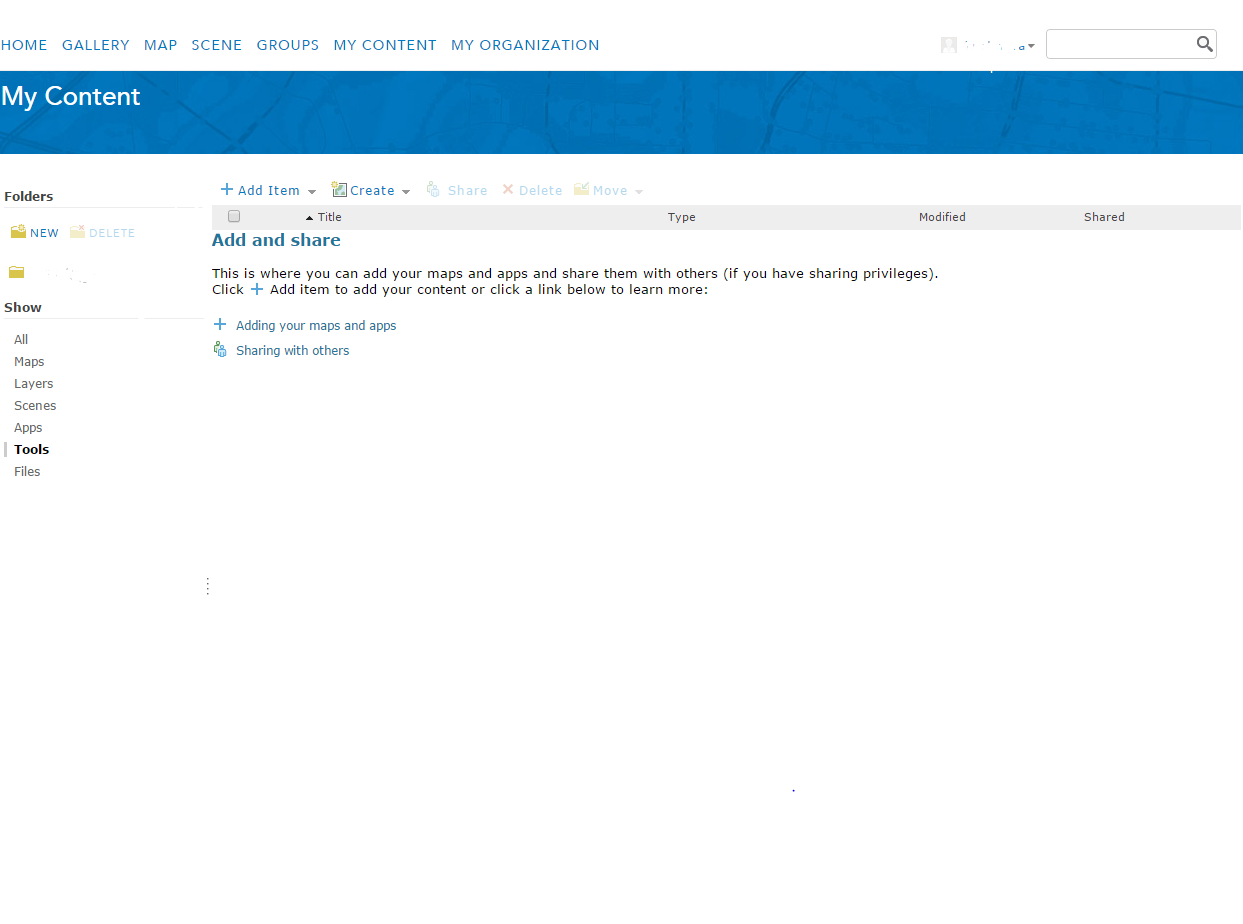
Thus, at the time of app creation, a web map with 2 image layers will have to be added through the builder in order for the proper functioning of the widgets. The document will provide guidance on how to add the web map at a later stage.

# Get ArcGIS Webapp Builder

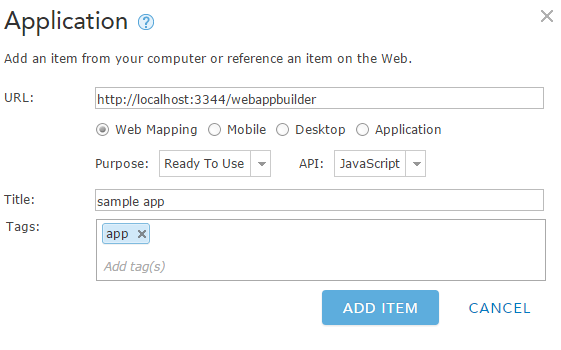
For the 1.0 beta release, ArcGIS WebApp Builder is available in a ZIP file named arcgis-webapp-builder-1.0beta.zip, which contains everything you need to start using ArcGIS WebApp Builder, and developing custom widgets and themes.

# Setup ArcGIS WebApp Builder

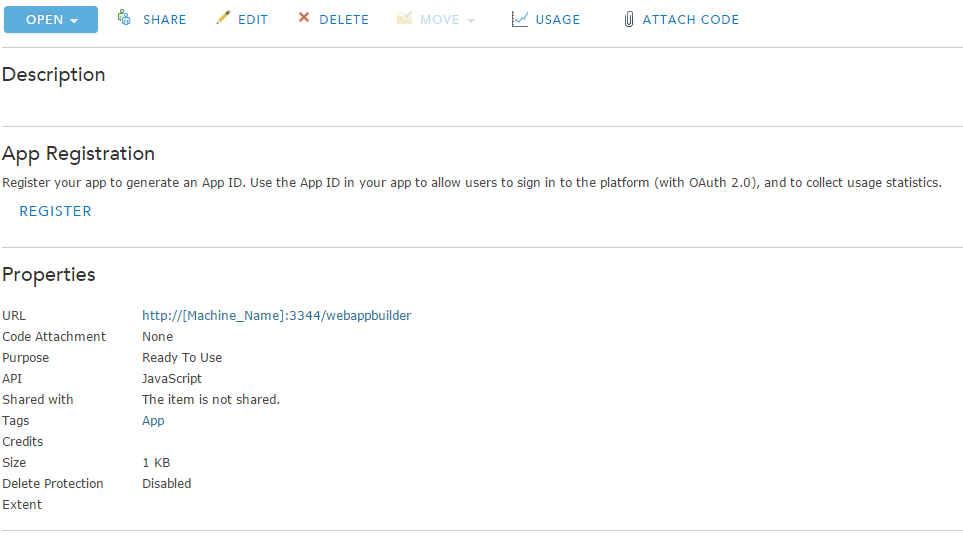
1. Download arcgis-webapp-builder-1.0beta ZIP file from https://developers.arcgis.com/web-appbuilder.
2. Unzip arcgis-webapp-builder-1.0beta ZIP file to your computer.
3. Open the unzipped. Click on the startup.bat file. The Builder server starts and listens to port 3344.
4. Create a new ArcGIS Online Account or Sign in if you have one.
5. Click on the my content option
6. Click on add item option



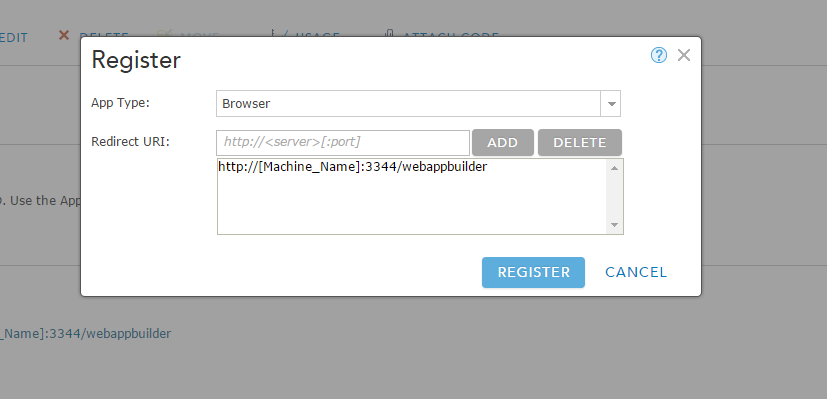
1. From Dropdown list, select an application option.
2. A dialog box will open.
3. Fill the details



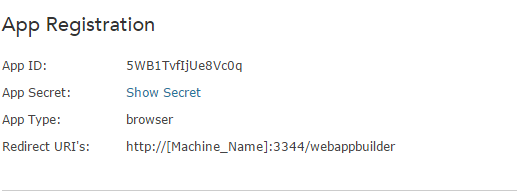
1. Click on Add button.
2. After this step, the following tab will open



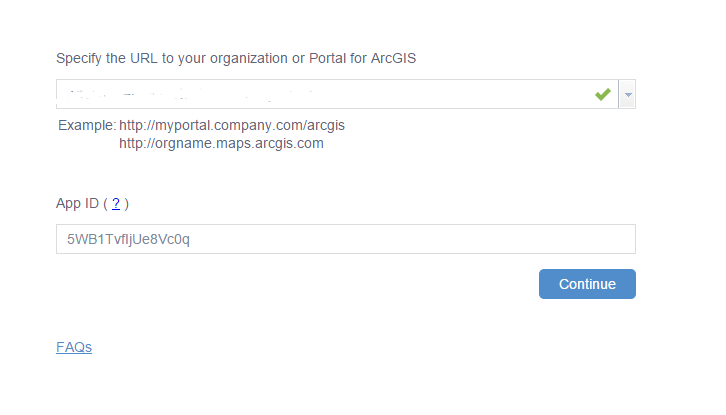
1. Click on Register for registering the app and getting the app id.
2. Add the URLs which will redirect to your app. (you can add as many URLs you want)



1. Click on register. Copy the App ID



1. Click on open button. The App will open. Add the following details



1. Click on continue.
2. Click on approve.
3. Now you can create your own app on web app builder.
4. Access ArcGIS WebApp Builder in the web browser: <http://[Machine_Name]:3344/webappbuilder>. Eg. <http://localhost:3344/webappbuilder>

# Create a basic web app in the builder

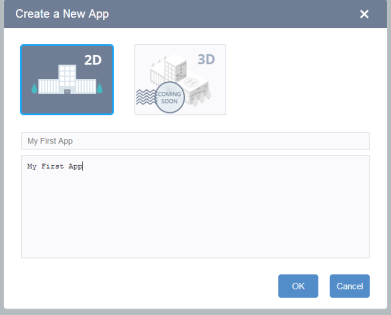
1. On the Builder homepage, click the “Create New” button.
2. Type the app name, for example My First App, and click “OK”.
3. Keep using the default theme, map, and widgets.
4. Under the Attributes tab, change the title for example My First App.
5. Save your changes and click the Home button to return to the homepage.
6. Click Launch button beside the app item to view your app in the web browser.

# Create a basic web app (for image services) in the builder

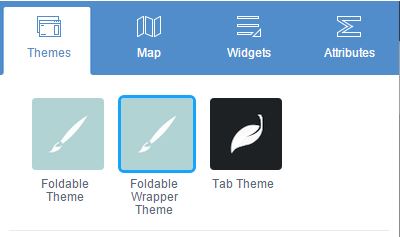
1. On the Builder homepage, click the “Create New” button.



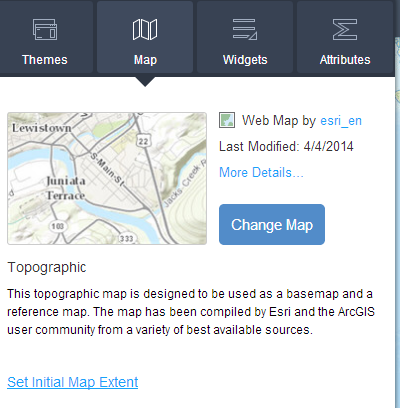
1. Type the app name, for example My First App, and click “OK”.



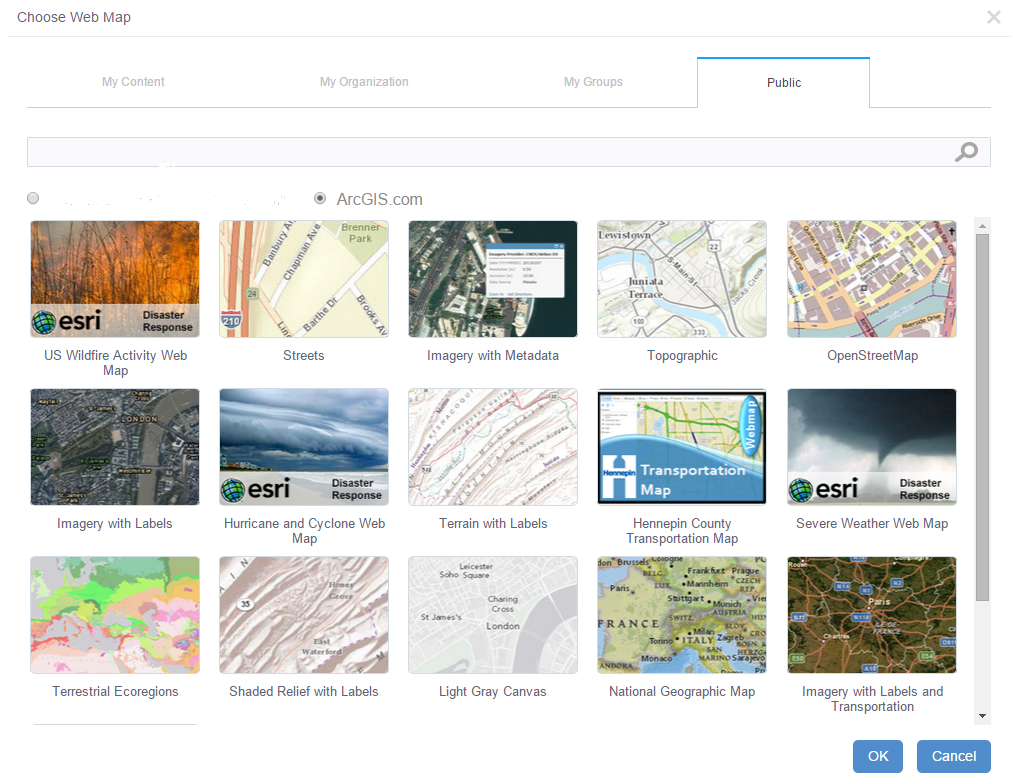
1. Use the Foldable Wrapper Theme. (The theme has been created for proper positioning of widgets specific to image services and won’t work when all the widgets are grouped together).If you want to form the group of widgets then use the Foldable Theme.



1. Click on the “Map” tab and then click on “Change Map”.



1. A sample web map (with 2 operational layers) has been made public. Search for “sample map for wabis”. Click on it and then click on “OK”. Any web map, having 2 operational layers, can be added at this step for the proper functioning of the image services widgets. (You can add the web map with more than 2 layers but make sure only the top two layers are on. Even if all the layers are made visible on the web map, make sure to set the visibility of all layers (except the top two layers) to off through LayerList widget.)

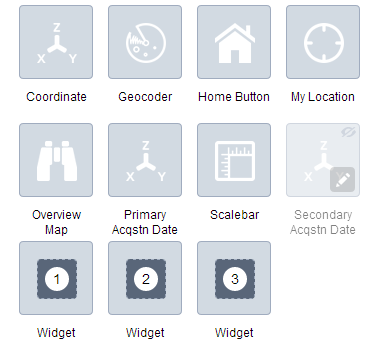


The layers in the web map will be added to the map in the builder.

1. Under the Widgets tab, one can select/deselect the widgets listed. These widgets are automatically added to the app when the custom theme is selected.



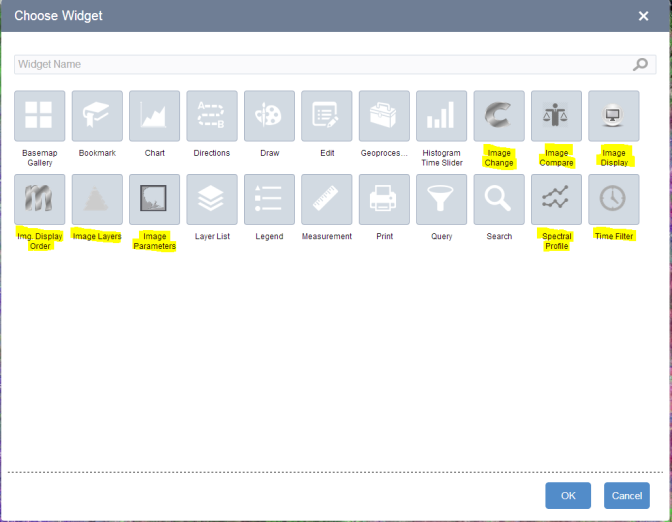
The widgets marked in yellow are specific to Image services. The widgets can be turned on/off by clicking on the  on the top right corner of the widget.



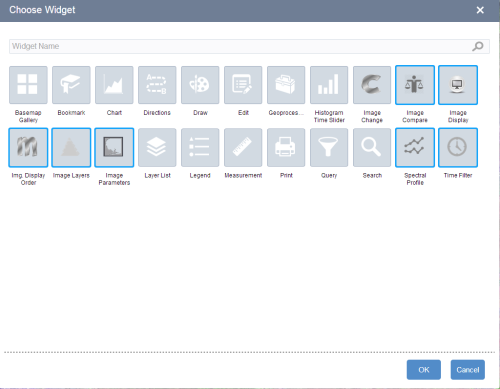
To add/remove controlled widgets click on “Set controlled widgets”.



Click on “Click here to add widget”. It will then show a list of all the widgets that can be added to the app header.



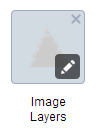
The ones marked in yellow are the image service widgets. Select the widgets to be added to the app and the click on “OK”.



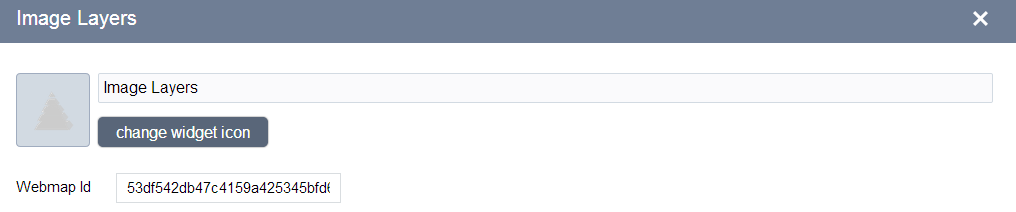
The selected widgets will show up in the app header as well as the header controller.



If configs need to be specified for any of the widgets, click on the  button on the bottom left corner of widget in the header controller. Eg.



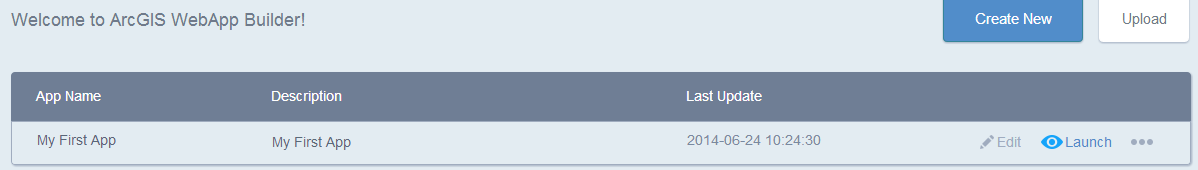
On clicking the button in the Image Layers widget, the settings page for the Image Layers widget will open where the configs for the widget can be edited. The web map Id can be changed using this interface. (The editable config properties for each widget will be discussed in the widget documentation.)



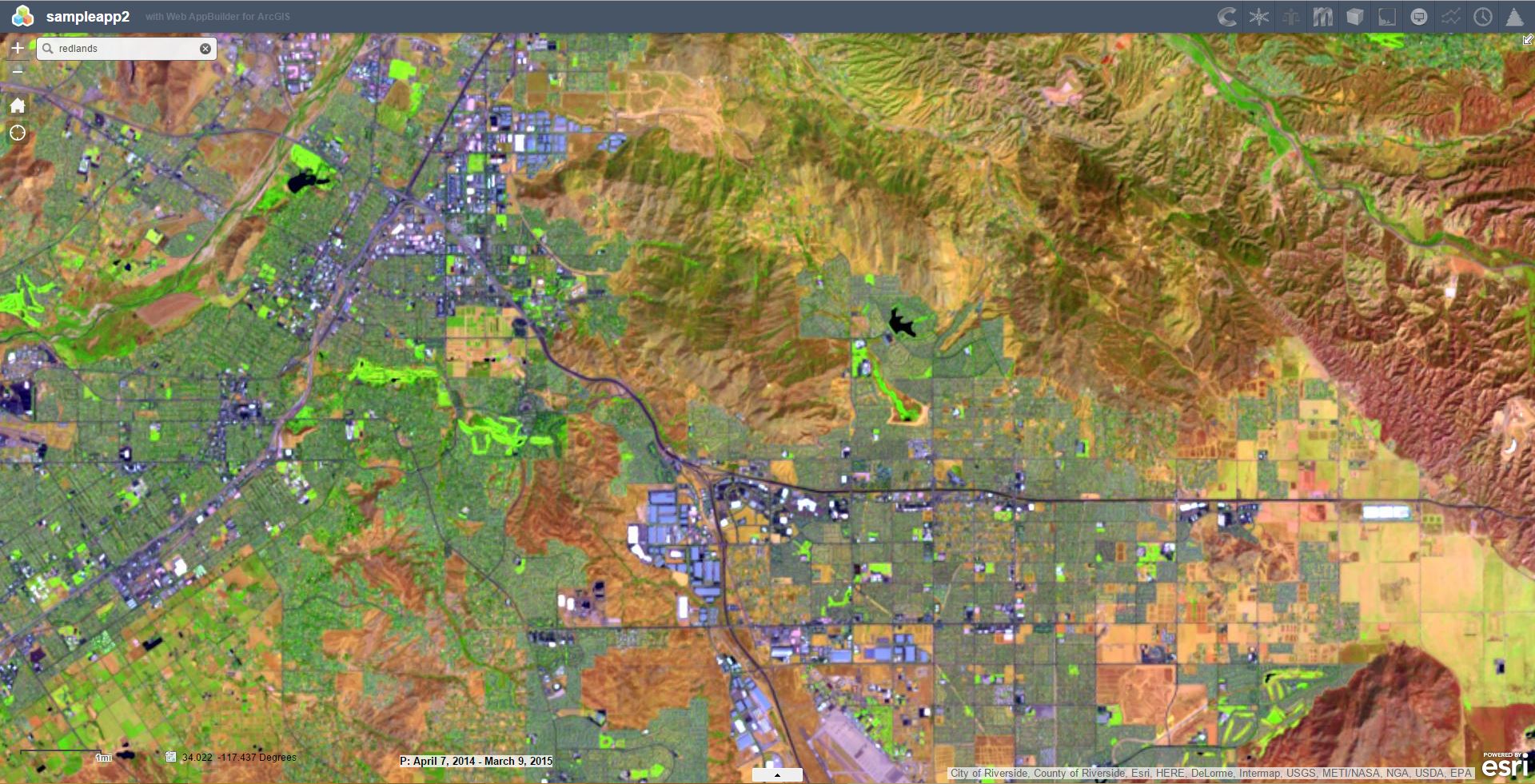
1. Save your changes and click the Home button to return to the homepage.



1. Click Launch button beside the app item to view your app in the web browser.



The final app created would look like the image below.

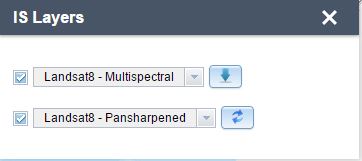


# Widget Usage

## ISLayers

### Overview

The ISLayers widget sets/changes the primary and secondary layer in the app. It allows users to turn on/off the primary and secondary layers. It also allows users to change the primary layer and the secondary layer to one of the services listed in the dropdown. The primary layer can be moved to replace the secondary level ,or the primary and secondary layers can be swapped. The result layer after doing change detection can be saved by clicking on save icon beside the result checkbox. The user can give any name to the layer and the layer would be added in the dropdown list of both the primary and secondary layer.



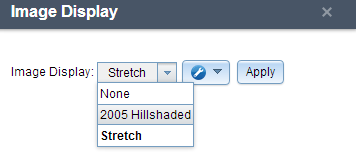
### Configuration

Configuration enables the definition of webmapId. This is the ID of the Webmap used to define the list of image services that are to be populated in the drop down in the widget. Typically an organization would create such a webmap and populate it with different image layers. The properties of each image layer can be set as part of that webmap. The webmap and content of the webmap must be accessible by (shared with users of) the current app. This Webmap can then be maintained independently of the app.

## ISDisplay

### Overview

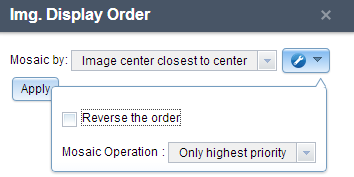
The IS Display widget sets the service functions/stretch on the primary layer. The drop down in the widget is automatically populated with the service functions of the primary layer.



## ISDisplayOrder

### Overview

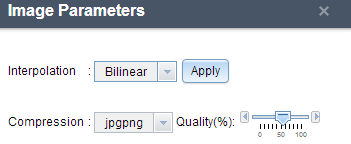
The IS Display Order widget sets the mosaic rule on the primary layer.



## ISParameters

### Overview

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

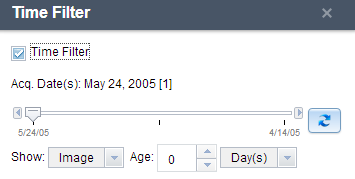


## ISTimeFilter

### Overview

The IS TimeFilter widget displays all the raster on the current AOI of the primary layer, ordered by date field, on a slider. As the user moves the slider, the corresponding raster is selected. The user has the option to see the image or footprint of the raster. If the user enters an age, then all the rasters, till the age specified, from the current acquisition date, are selected.

Note the selection is defined at the time when the Time Filter is turned on. If you zoom to a different location you need to press the refresh button on the right side to have the tool reselect the overlapping images for the current extent.

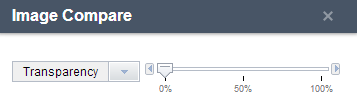


This tool only becomes active, if the Primary image layer has TimeInfo. The selection set is based on the current AOI.

## ISCompare

### Overview

The IS Compare widget sets transparency or swipe on the topmost imagery layer. This enables comparison between the topmost image layer and those below. Typically this would be between the primary and secondary layer, but if there is a results layer then this would be between the results layer and the primary layer.





## ISPrimaryAcquisitionDate

### Overview

The ISprimaryacquisitiondate widget shows the date range for the visible rasters, for the current AOI, on the primary layer.



## ISSecondaryAcquisitionDate

### Overview

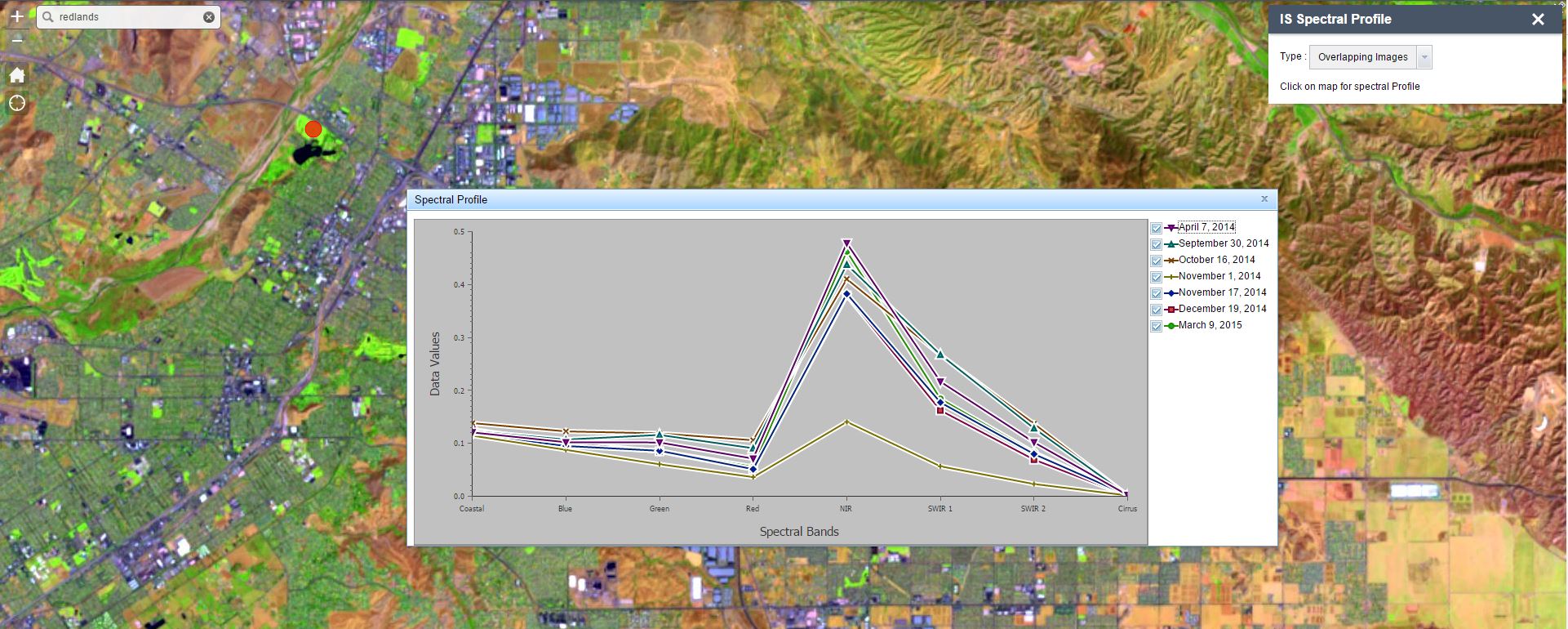
The ISsecondaryacquisitiondate widget shows the date range for the visible rasters, for the current AOI, on the secondary layer.



## ISSpectralProfile

### Overview

The IS spectral profile widget shows temporal spectral, non-temporal spectral or NDVI profile of a point on the primary layer.



This tool assumes that Band names and band values are present in key properties at item level. The data values are scaled between the min and max of the service.

## ISTimeFilterProfile

**Overview**

IS TimeFilterProfile is an combination of IS TimeFilter and IS SpectralProfile widget. The user can move through time by clicking on the graph points of temporal profile.

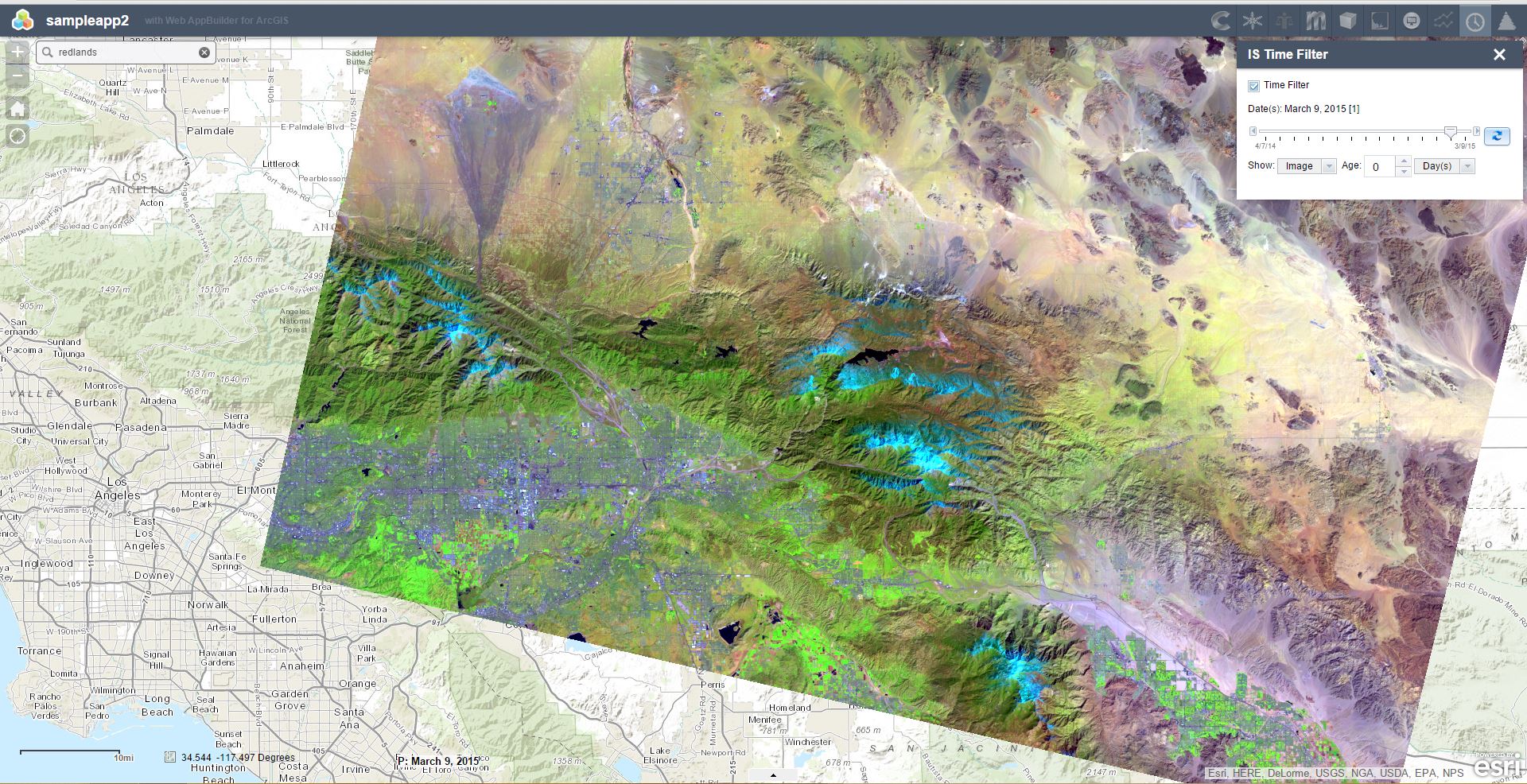
## ISChangeDetection

### Overview

IS Change Detection widget allows user to detect the difference between 2 rasters on the primary layer and secondary layer. The change is remapped to one color and the rest is remapped to another color. The result is shown as a new layer called “Result Layer”.

For change detection widget to function, following workflow should be followed –

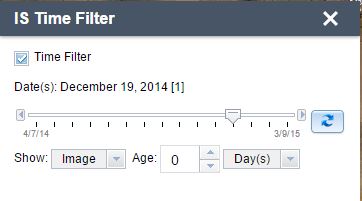
* 1. On the primary layer, go to the IS timeFilter widget and activate it. After activating select the scene for which you want to do change detection.



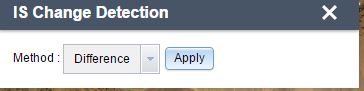
* 1. Go back to the IS Layers widget and click on the ‘copy primary to secondary’ button. The current primary layer with the scene you selected becomes the secondary layer.

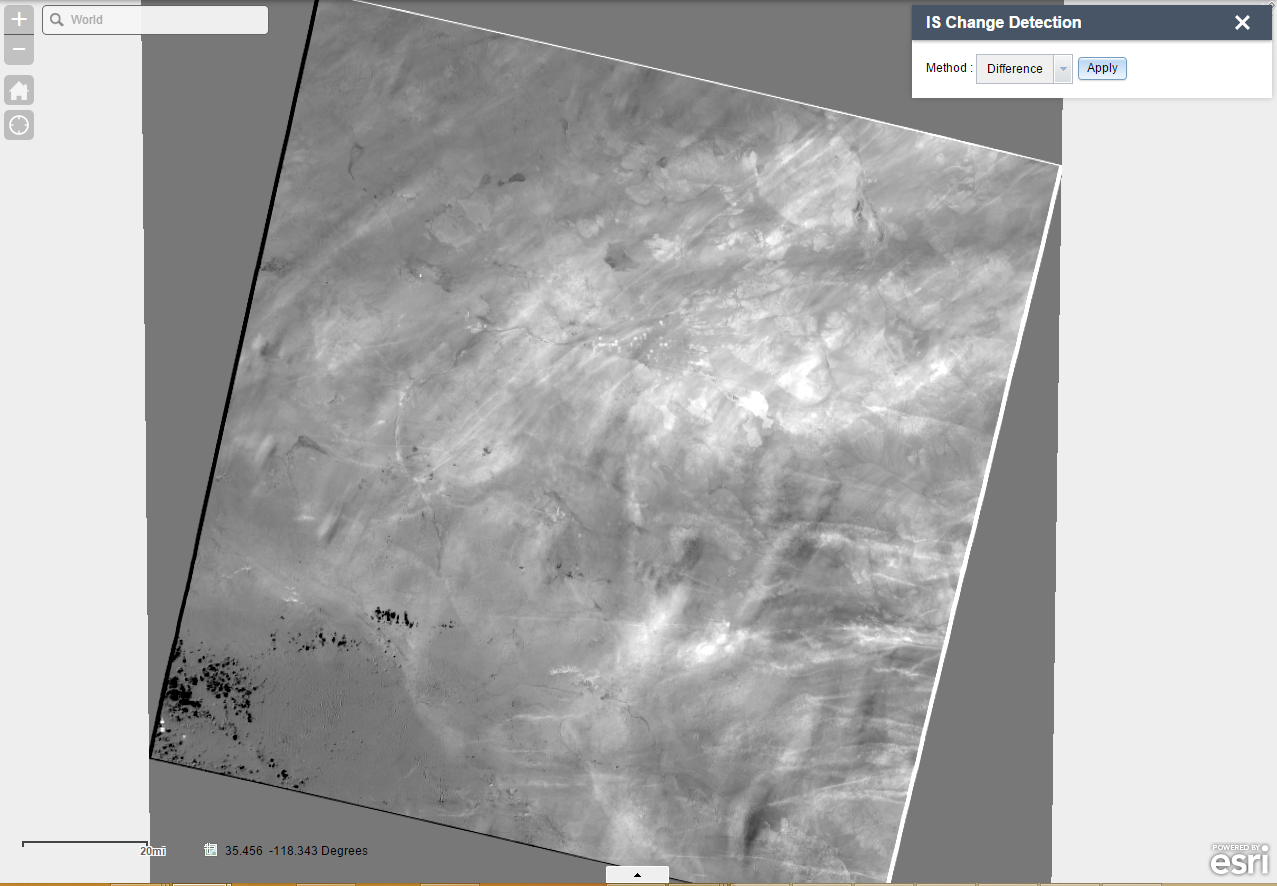


* 1. On go back to the IS timeFilter widget and select the new scene for which the change detection will be performed.



* 1. Open IS change detection widget. Choose the method and click on apply.

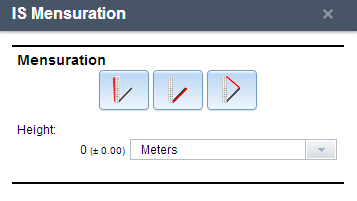




## ISMensuration

### Overview

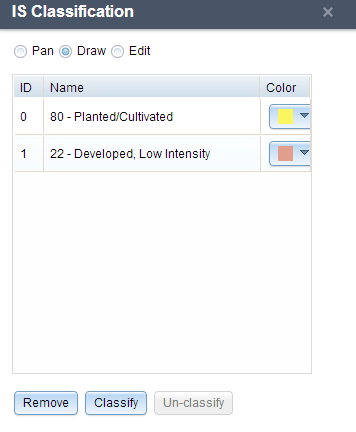
IS Mensuration widget is similar to the tool in Arcmap. It enables user to measure heights(typically buildings). It works on the primary image. It becomes active only when the primary image has mensuration capabilities.



## ISClassification

### Overview

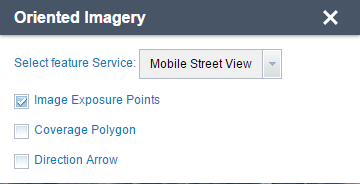
IS Classification widget enables user to draw features, associate categories and then perform classification. The result is then added as the “result layer”.



**Oriented Imagery widget**

**Overview**

Oriented Imagery widget enables user to view images taken through mobile device, drone etc. When the user clicks on the map, the nearest image source of the clicked point is selected and a popup window displays the image. The widget reads the feature service from the config.



**ISSplitTool**

**Overview**

IS SplitTool widget enables the user to compare result, primary and secondary layer. User can move the swipe widget in all directions i.e. top, bottom, right and left.

Result

Primary Secondary