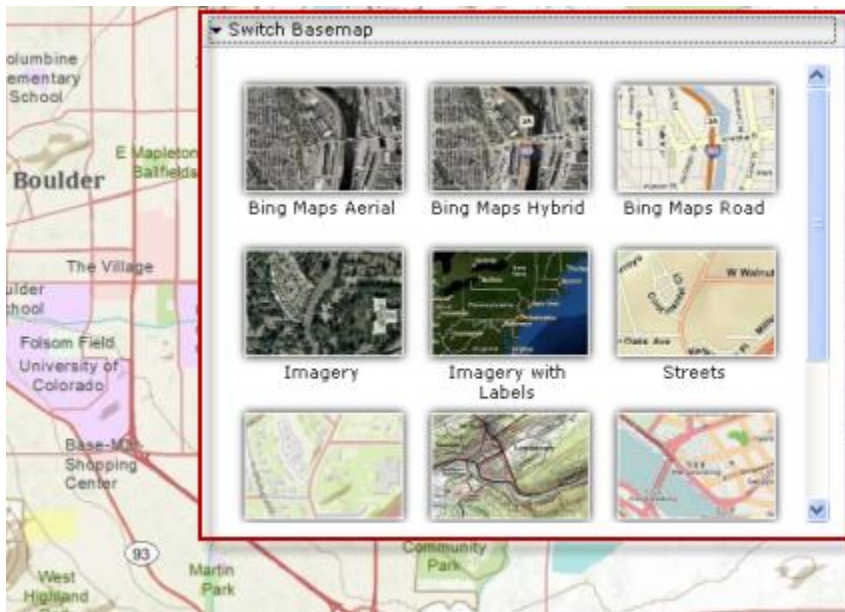


Chapter 9: Integrating Widgets

The API for JavaScript comes with several out of the box widgets that you can drop into your application for enhanced productivity. Included are the BasemapGallery, Legend, Scalebar, OverviewMap, Measure and Bookmarks widgets.

Basemap Gallery Widget

The BasemapGallery widget displays a collection of basemaps from ArcGIS.com and/or a user-defined set of map or image services. When a basemap is selected from the collection the current basemap is removed and the newly selected basemap appears. When adding custom maps to the basemap gallery they will need to have the same spatial reference as the other layers in the gallery. When using layers from ArcGIS.com this would be the Web Mercator reference with wkids 102100, 102113, or 3857. It is also recommended that all basemaps be tiled layers for performance reasons.



When creating a BasemapGallery there are a number of parameters that you can supply in the constructor including the ability to show ArcGIS basemaps, define one or more custom basemaps for inclusion in the gallery, supply a Bing maps key, and a reference to the map where the gallery will be placed. After creating the BasemapGallery widget you need to call the startup() method to prepare it for user interaction.

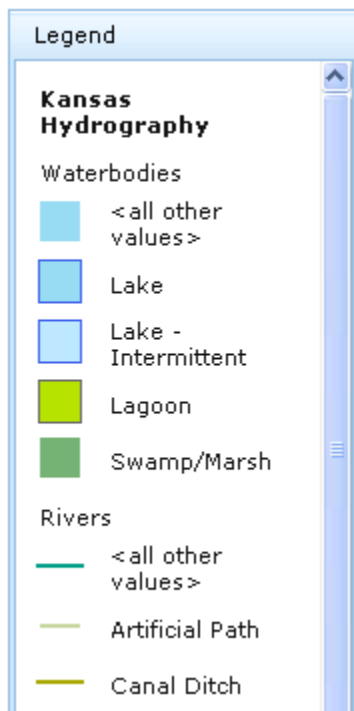
```

var basemaps= [];
var waterBasemap = new esri.dijit.Basemap({
  layers:[waterTemplateLayer],
  title:"Water Template",
  thumbnailUrl:"images/waterThumb.png"
});
basemaps.push(waterBasemap);

```

Legend Widget

The Legend widget displays a label and symbols for some or all layers in the map. It does have the ability to respect scale dependencies so that as you zoom in or out in the application the legend updates to reflect layer visibility at various scale ranges. The Legend widget supports ArcGISDynamicMapServiceLayer, ArcGISTiledMapServiceLayer, and FeatureLayer. If using ArcGIS Server 10.0 SP1 or higher the legend is generated using the REST map service legend resource. If using ArcGIS Server 10.0 or lower the legend is generated using the ArcGIS.com legend service.



When creating a new instance of the Legend widget you can specify various parameters that control the contents and display characteristics of the legend. The arrangement parameter can be used to specify the alignment of the legend within its container HTML element. The layerInfos parameter is used to specify a subset of layers to use in the legend, and respectCurrentMapScale can be set to 'true' to trigger automatic legend updates based on scale ranges for each layer. Finally, you need to call the startup() method to display the newly created legend.

```

var legend = new esri.dijit.Legend({
  map:map
}, "legendDiv");
legend.startup();

```

Scalebar Widget

The Scalebar widget is used to add a scalebar to the map or a specific HTML node. The scalebar displays units in either english or metric values. You can also control scalebar positioning through the 'attachTo' parameter. By default, the scalebar is positioned in the bottom left hand corner of the map.



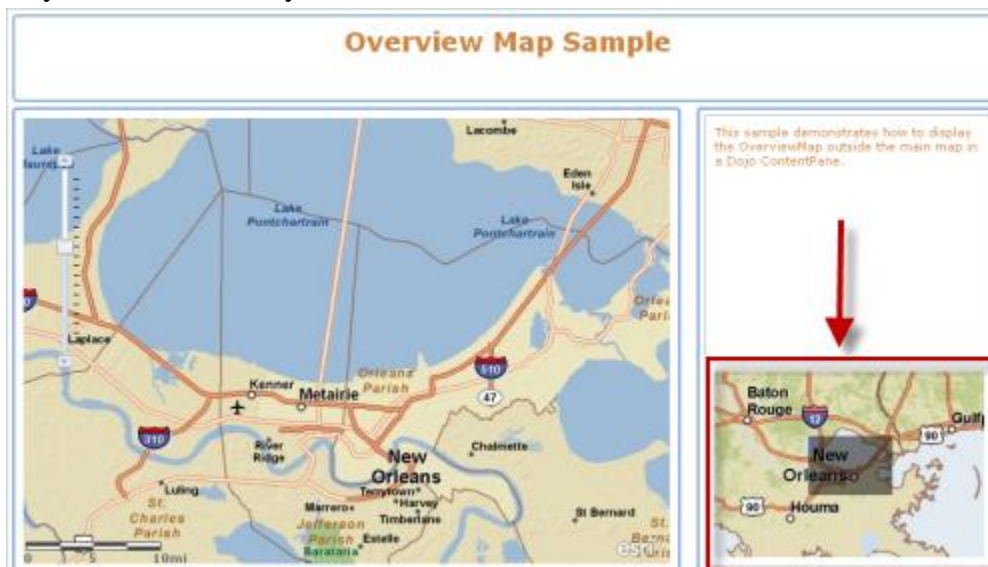
```

var scalebar = new esri.dijit.Scalebar({
  map: map,
  scalebarUnit: 'english'
});

```

OverviewMap Widget

The OverviewMap widget is used to display the current extent of the main map within the context of a larger area. This overview map updates each time the main map extent changes. The extent of the main map is represented as a rectangle in the overview map. This extent rectangle can also be dragged to change the extent of the main map. An overview map can be displayed in a corner of the main map and also hidden from display when not in use. It can also be placed inside a <div> element outside the main map window or temporarily maximized for easy access to far away areas of interest.

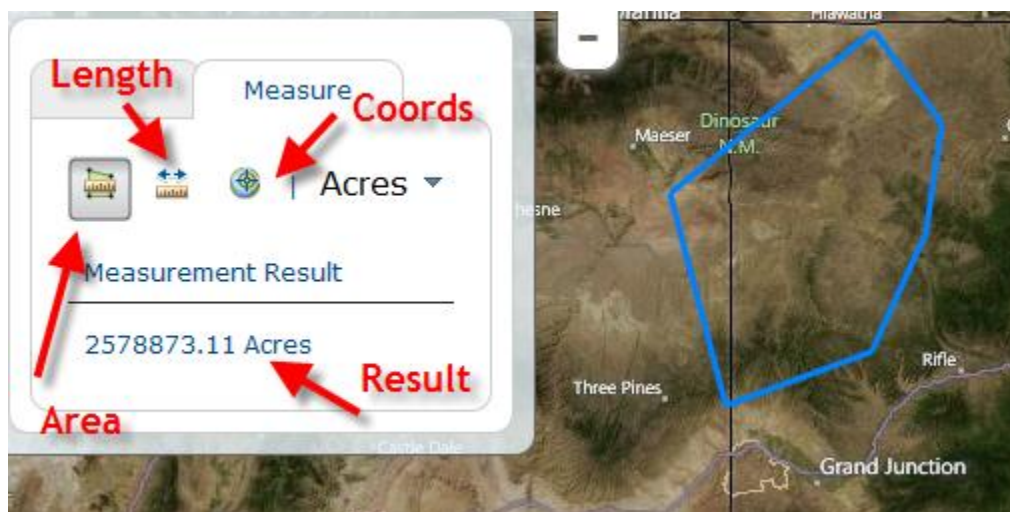


The OverviewMap widget takes a number of optional parameters in the constructor for the object. These parameters allow you to control things like where the overview map is placed in relation to the main map, the base layer to use for the overview map, the fill color for the extent rectangle, the appearance of a maximize button, and the initial visibility of the overview map.

```
var overviewMapDijit = new esri.dijit.OverviewMap({
  map: map,
  visible:true
});
overviewMapDijit.startup();
```

Measurement Widget

The Measurement widget can be used to measure area and length as well as report the geographic location of the cursor. Dropping this widget into your application is very simple just like the other widgets. You can also define custom symbols for the lines and polygons that are drawn on the screen during the process as well as custom picture symbols for the points.



First, import the resource for the widget.

```
dojo.require("esri.dijit.Measurement");
```

Then create a new instance of the Measurement object, passing in an option that defines a <div> tag where the widget should be placed.

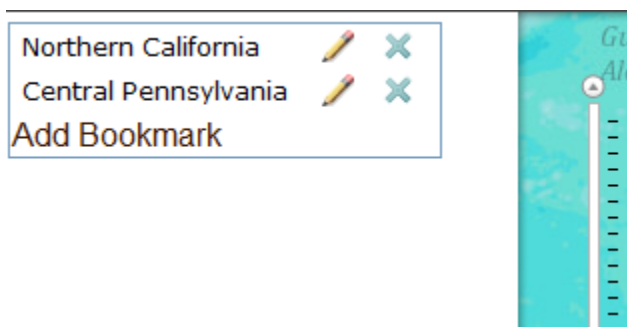
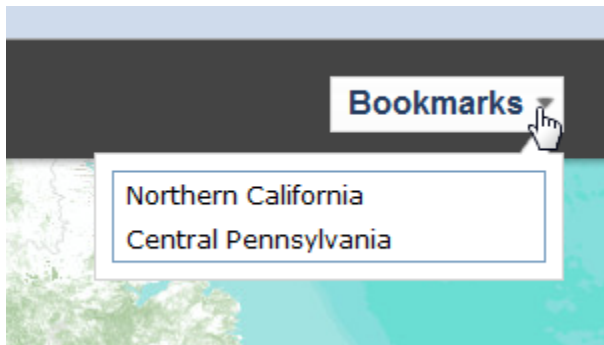
```
var measurement = new esri.dijit.Measurement({
  map: map
}, dojo.byId('measurementDiv'));
```

Finally, call the startup() method to display the widget.

```
measurement.startup();
```

Bookmarks Widget

The Bookmarks widget allows you to define specific geographic extents for named areas. For example, in the figure below there are two bookmarks: Northern California and Central Pennsylvania. When selected from the list the map will reposition the geographic extent to the extent attached to the bookmark.



To use the Bookmarks widget first import the resource along with a container for the bookmarks. This is normally a drop-down list or combo list as seen in the figures above.

```
dojo.require("esri.dijit.Bookmarks");
```

The individual bookmarks that you add to this widget are defined as an array of JSON objects. Each bookmark should be given a name, and a geographic extent composed of both a spatial reference and the coordinates that define the extent.

```

var bookmarks = [{
  "extent": {
    "spatialReference": {
      "wkid": 102100
    },
    "xmin": -14201669,
    "ymin": 4642975,
    "xmax": -13021482,
    "ymax": 5278931
  },
  "name": "Northern California"
}, {
  "extent": {
    "spatialReference": {
      "wkid": 102100
    },
    "xmin": -8669334,
    "ymin": 4982379,
    "xmax": -8664724,
    "ymax": 4984864
  },
  "name": "Central Pennsylvania"
}
];

```

Finally, create the new Bookmarks object, passing in the variable that references the bookmarks defined in the JSON object above.

```

// Create the bookmark widget
bookmarks = new esri.dijit.Bookmarks({
  map: map,
  bookmarks: bookmarks
}, dojo.byId('bookmarks'));

```

The `dojo.byId` method is used to reference the `<div>` container for the bookmark.

```

<div id="bookmarks-wrapper">
  <div data-dojo-type="dijit.form.DropDownButton">
    <span>Bookmarks</span>
    <div data-dojo-type="dijit.TooltipDialog">
      <div id="bookmarks"></div>
    </div>
  </div>
</div>

```

