Bing Maps V7 - Custom Infoboxes and Modular Design

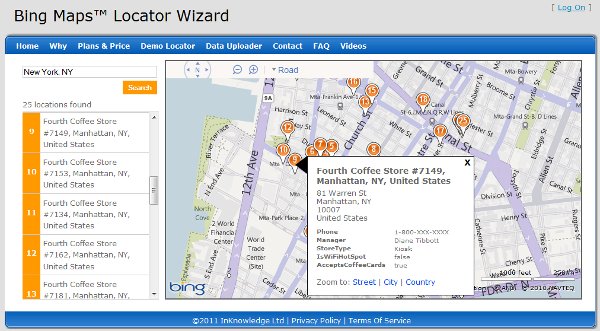
Recently I was working on the [Bing Maps Locator Wizard](http://bingmapslocators.cloudapp.net/) and required a highly customizable infobox control that would likely be put in an iframe. For those who are not familiar with the [Bing Maps Locator Wizard](http://bingmapslocators.cloudapp.net/), it is a portal where you can navigate through a series of steps to generate Bing Maps Locator in minutes and have it hosted in Windows Azure for free. One of the steps allows you to customize the style of various elements of the locator. For this wizard I needed an infobox control that users could easily customize. This article highlights some of the key features of the modular plugin that was created and how they came about. The end result far exceeded our expectations and [InKnowledge](http://inknowledge.co.uk/) has since released the code to the Bing Maps community for use in your own applications. You can download the complete source code with sample implementations and documentation [here](http://cid-0d35222484a76a01.office.live.com/self.aspx/BingMapsv7/BMv7.CustomInfobox.zip).

**Modular Design**

The Bing Maps V7 control was released last fall and is a complete rebuild from the ground up when compared to previous versions of Bing Maps. With this version of Bing Maps we have seen huge improvements in the performance of the API and the size of the control. The map control was redesigned to use a modular framework which allows you to register and load modules as they are needed rather than loading everything at once. By using this approach the loading speed of web sites hosting Bing Maps will load much faster than previous versions of Bing Maps as only the resources that are needed are loaded. For those interested in creating your own reusable modular plugins for Bing Maps have a look at the [documentation](http://msdn.microsoft.com/en-us/library/hh125836.aspx).

**Custom Infoboxes**

When the V7 control was originally released there were no Infoboxes. Not long after the release Microsoft added these in. The infobox control that was added is a bit different from past versions of Bing Maps and was designed to be more flexible than those that were in previous versions of Bing Maps. This is a great improvement what was available before but there are a couple caveats to using this infobox control. The first is that the infobox always points up and to the right and does not reposition itself such that the content is towards the center of the map. This means that infoboxes that are anchored near the top of the map end up being displayed outside of the map. This can be an issue if the map is being put in an iframe or the map is placed at the end of a web page as the infobox would be cut off. The second caveat is that using custom HTML requires you to also generate the infobox arrow and content area in addition to the content you want to display. Ideally we would be able to put custom HTML into the frame of the built in infobox.



**The Infobox Arrow**

One common issue that developers run into when trying to create customizable infoboxes is with the Infobox arrow. In the past these were almost always images which required you edit them if you wanted to change something as simple as the color. After a lot of research and experimenting I found an ingenious solution, CSS borders. Years ago when the web was young web pages used to use really big borders to make things look like they were in a frame. As time went on the borders got smaller and smaller to the point where they are either not used or only 1 or 2 pixels wide. So where am I going with this? Well, if you look at a picture frame you might notice that the sides of the frame are connected in the corner using a 45 degree angle. This angle also occurs with CSS borders. So taking this and using a bit of fancy styling we can modify the color and width of four edges of the border CSS class and generate an arrow. Not only can we generate an arrow but with a little bit of logic and CSS styling we can control the width, length and color of the arrow. A good article on creating angled shapes using the border style can be found [here](http://www.howtocreate.co.uk/tutorials/css/slopes).

**Positioning**

The built in infobox always points up and to the right. This might not be an issue for some but anyone hosting a map in an iframe or placing a map near the edge of a web page will quickly fine that this is not ideal. What we want is the content to be positioned towards the center of the map. This will help ensure that the infobox is in a place where the user can see it. The solution, break the map into quadrants and modify the layout of the infobox such that the content is towards the middle of the map. For example, if the anchor point of an infobox is in the bottom left quadrant of the map we want the infobox to be positions to the right and up, it is in the top right quadrant, position it so that it’s to the left and down. So now we just need to figure out which quadrant the anchor is in. The easiest way to do this is to convert the anchor coordinate to a pixel coordinate relative to the top left corner of the map. To figure out if the anchor is to the left or right of the center of the map simply check to see if the x pixel coordinate is more or less than half the width of the map. Similar logic can be used to figure out if the anchor is above or below the middle of the map. Once we know if it is left or right of the center and above or below the center we simply need to infobox so that it is in the opposite direction.

**Additional features**

The API for this custom infobox control is pretty basic. There are only four public methods that are exposed which are; hide, show, getOptions and setOptions. The show method is pretty straight forward; it takes in a coordinate and content. The content can be either text or HTML. The content will be placed inside the infobox frame rather than replacing it. By making the input this generic we are able to use this infobox with pushpins, polyline, polygons, or on its own. As for options not only can we set the dimensions and color of the arrow but you can also set the background color of the content area of the infobox, set the color of the border on the infobox, specify minimum width and heights for the content area, specify an offset distance from the anchor point, hide the arrow, hide the default close button, and finally tether the infobox. The tether option allows one of two user experiences. When enabled the infobox will stay open and reposition it’s self as you move the map around until you close it. When disabled the infobox will close when the map moves.

**Implementing the Plugin**

Implementing the custom infobox is easy. The first thing you must do is register the plugin with the map control. You can do this using code similar to the following:

|  |
| --- |
| var map, customInfobox;  function GetMap() {  map = new Microsoft.Maps.Map(document.getElementById("myMap"), { credentials: "Your\_Bing\_Maps\_Key” });  //Register and load the Custom Infobox Module  Microsoft.Maps.registerModule("CustomInfoboxModule", "scripts/V7CustomInfobox.min.js");  Microsoft.Maps.loadModule("CustomInfoboxModule", { callback: function () {  //Create an instance of the custom infobox control  customInfobox = new CustomInfobox(map);  }  });  } |

To display an infobox you simply need to specify a coordinate and some content like the following:

|  |
| --- |
| customInfobox.show(map.getCenter(), "<b>Hello World</b>"); |

**Source Code and Documentation**

Complete source code, with documentation and sample web pages is available [here](http://cid-0d35222484a76a01.office.live.com/self.aspx/BingMapsv7/BMv7.CustomInfobox.zip).

Ricky Brundritt  
[InKnowledge](http://inknowledge.co.uk/)