

Creating your first Virtual Earth Web Page Part 2

By Dr. Neil Roodyn

In the first part of this article you learned how to add and use the version 2 Virtual Earth map control in your own web page. In this article you will discover how to use some of the other controls that make up the Virtual Earth product.

By the end of this article we will have extended the page we build in part 1 to use new images for the compass and zoom bar controls and include a custom panel. The end result should appear as seen in Figure 1.

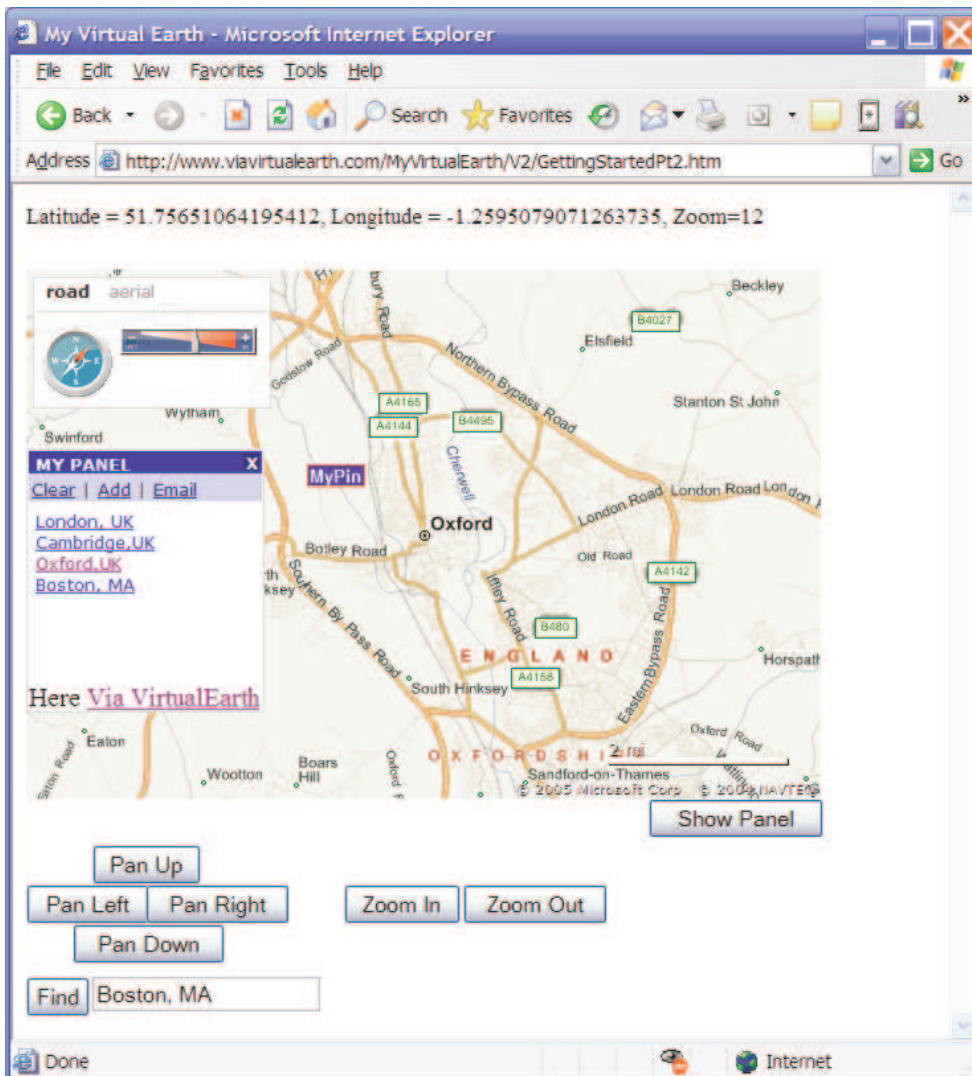


Figure 1

Setting up the script for the other controls

In part 1 we discovered that the script for the Virtual Earth Map Control can be found at <http://dev.virtualearth.net/standard/v2/MapControl.js> In the first version of the Virtual Earth control you would have had to include another VE.JS script file to use the basic controls, such as the compass or the zoom bar. These controls are now included by default in the new dashboard on the VirtualEarth MapControl.

The Compass control

The first control we are going to modify is the compass control. This provides a great way for navigating around the map. The compass is represented by an image, it is best to use a gif with transparency so that the compass does not cover any more of the map than is needed. You can create your own image or copy the image used with this example. The compass is defined in the Mapcontrol.css cascading style sheet as

```
.Compass{ width:54px; height:54px; background:url(i/compass.gif); margin:0px; cursor:pointer }
```

In order to use a different image for the compass you will need to add your own definition for the .Compass style after you import the MapControl.css file.

```
<link href="http://local.live.com/css/MapControl.css"
      type="text/css" rel="stylesheet" />
<script src="http://local.live.com/MapControl.ashx">
</script>
<STYLE TYPE="text/css" MEDIA=screen>
<!--
.Compass{ width:54px; height:54px; background:url(i/compass.gif);
margin:0px; cursor:pointer }
-->
</STYLE>
```

This is the image we use for this exercise



The page should now display your compass image in the dashboard on the map.

The Zoom control

The zoom control provides a slick user interface for zooming in and out of the map. It also provides visual feedback as to the amount a map can be zoomed and where the current map is on that scale.

In order to modify the zoom control to the page you must add some style classes that describe how the control will look.

The control consists of 2 graphical components, the bar and the slider. For this exercise we are using these images

Bar: 

Slider: 

In the same way you defined a different image for the compass you will need to add your own definition for the .ZoomBar and .ZoomBar_slider styles after you import the MapControl.css file.

```
<link href="http://local.live.com/css/MapControl.css"
      type="text/css" rel="stylesheet" />
<script src="http://local.live.com/MapControl.ashx">
</script>
<STYLE TYPE="text/css" MEDIA=screen>
<!--
.Compass{ width:54px; height:54px; background:url(i/compass.gif);
margin:0px; cursor:pointer }
.ZoomBar{ position:relative;background:url(i/zoom/bar.gif);
width:103px; height:20px; margin:2px; overflow:hidden; }
.ZoomBar_slider { position:absolute; background:url(i/zoom/slider.gif);
width:7px; height:20px; overflow:hidden; display:block; }
-->
</STYLE>
```

At this stage we have a web page with a map and great looking controls for interacting with the map. The page should look something like that shown in figure 2.

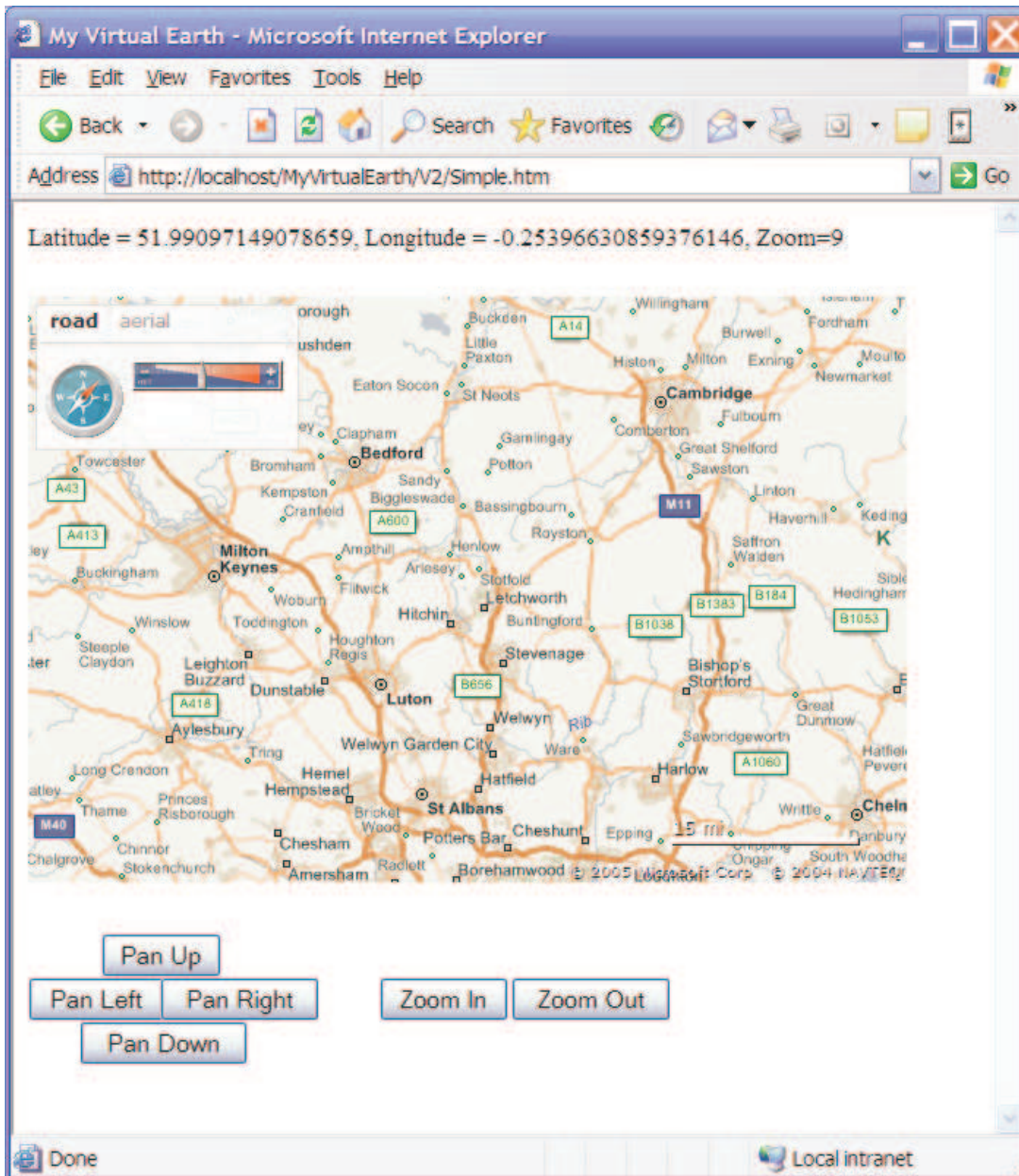


Figure 2

The next control we are going to explore is the most complicated and the one that has the most compelling features; the panel. We will discover how to create a panel that behaves in a similar way to the VirtualEarth scratchpad.

Creating a Panel

As with some of the other controls we need to set the styles for a Panel. Most of the code for creating a Panel is in a script file located here <http://local.live.com/JS/VE.js> You will need to import that into your page

```
<script src="http://local.live.com/JS/VE.js">
```

</script>

We will start using the same style as Virtual Earth uses for the ScratchPad.

```
.VE_Panel_el
{
overflow:hidden;
z-index:31;
border:1px solid #cbcbcb;
padding:0;
margin:0;
background:white;
}

.VE_Panel_title
{
position:absolute;
padding-top:2px;
padding-left:5px;
overflow:hidden;
z-index:32;
font-family:Verdana,sans-serif;
font-size:8pt;
font-weight:bold;
color:rgb(230,250,255);
text-transform:uppercase;
cursor:default;
white-space:nowrap;
text-overflow:ellipsis;
}
.VE_Panel_title_blue{background:#0030cc}

.VE_Panel_cb
{
padding-left:1px;
width:18px;
height:18px;
color:white;
text-align:center;
font-size:7pt;
font-family:Verdana;
font-weight:bold;
overflow:hidden;
cursor:pointer
}

.VE_Panel_cb_blue{background:#001d7a;border:solid 2px #0030cc}

.VE_Panel_tb
{
height:18px;
padding-top:3px;
padding-left:2px;
font-family:Verdana,sans-serif;
font-size:8pt;
overflow:hidden
```

```

}

.VE_Panel_tb_blue{background:#ccd8ff}
.VE_Panel_tb td
{
font-family:Verdana,sans-serif;
font-size:8pt
}

.VE_Panel_tb a{color:#000080}
.VE_Panel_tb a:hover{color:#ff9900}

.VE_Panel_body
{
padding:5px;
font-family:Verdana,sans-serif;
font-size:8pt;
overflow:auto
}

```

Now you can define a global page variable for the panel.

```
var p;
```

At the end of the OnPageLoad function you can then create and show a Panel. You need to initialise the IOsec variable as the code in the VE.JS file that creates the VE_Panel uses an OutputEncoder.

```
IOsec=new OutputEncoder();

p=new VE_Panel("MyPanel",10,200,
180,200,
"blue",
31,
"My Panel",
"Nothing here yet!",
"footer",
"Mouse Over!",true);

p.onCloseClick=function(e)
{
p.Hide();
};

```

If you view the page in your browser you will now see a Panel, figure 3.

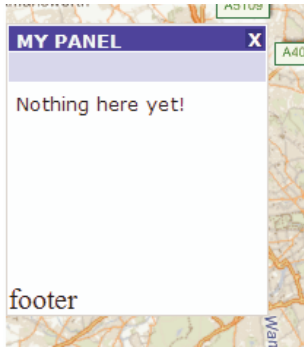


Figure 3

You can now set some properties of the Panel to change the way it appears. Let's start with the footer. It can be set by adding the following simple line after you have created the panel:

```
p.SetFooter("Here <a href='http://ViaVirtualEarth.com'>Via  
VirtualEarth</a>  ");
```

You can also set the Panel menu items.

```
p.SetToolbar (GetToolbar ());
```

Each part of the panel is simply HTML so you need a GetToolBar function that returns the HTML you want in the menu.

```
function GetToolbar ()
{
    var html="<table cellpadding=\"0\" cellspacing=\"0\" ";
    html+="border=\"0\" align=\"left\">";
    html+="<tr><td valign=\"top\" align=\"center\">";
    html+="<a href=\"javascript:ClearPanel ();\" ";
    html+="oncontextmenu=\"return false;\">Clear</a>&nbsp;|&nbsp;";
    html+="<a href=\"javascript:AddPoint ('Custom');\" ";
    html+="oncontextmenu=\"return false;\">Add</a>&nbsp;|&nbsp;";
    html+="<a href=\"javascript:EmailPanel ();\" ";
    html+="oncontextmenu=\"return false;\">Email</a>&nbsp;";
    html+="</td></tr><tr><td>&nbsp;</td></tr></table>";
    return html;
}
```

This GetToolbar function simply returns an HTML string of the contents of the menu at the top of the Panel. In this example we will have Clear and Email (as per the Virtual Earth Scratch Pad) and also an Add menu item.

You now need to write the code that implements the functions called by the menu. Let's start with the AddPoint function. This will add an entry to the Panel with a name and a location (LatLong). The name in this example will be "custom", we will extend this later on.

Let's start by defining a type called a PanelPoint that contains a name and a LatLong variable. The panel will contain a collection of these points so also define an array.

```

PanelPoint = function(n, pt)
{
    this.Name = n;
    this.LatLong = pt;
}
//array of PanelPoint objects
var panelPoints = new Array();

```

The AddPoint function can then be written to use this array. This function takes the center point of the map and adds that location to the panel array with the name supplied.

```

function AddPoint(name)
{
    var body = "";
    var lat = map.GetCenterLatitude();
    var lon = map.GetCenterLongitude();
    var pt = new Msn.VE.LatLong(lat,lon);
    var newPoint = new PanelPoint(name, pt);

    panelPoints.push(newPoint);
    var pts = panelPoints;
    for(var i=0;i<pts.length;i++)
    {
        var pt = panelPoints[i];
        body += "<a href='javascript:map.PanToLatLong(";
        body += pt.LatLong.latitude + "," + pt.LatLong.longitude;
        body += ");'>" + pt.Name;
        body += "</a> <br/>";
    }
    p.body.innerHTML = body;
}

```

You should now be able to add points to the Panel, move around the map and add oints. Clicking on the points in the panel should take you back to those points.

Now you can write the EmailPanel function to send an email with the points in the Panel. This function uses the Live Local site to display any points you send in the email.

```

function EmailPanel()
{
    var body = "http://ViaVirtualEarth.com/MyVirtualEarth/V2/ \n\n";
    var pts = panelPoints;
    for(var i=0;i<pts.length;i++)
    {
        var pt = panelPoints[i];
        body += pt.Name + "\n";
        body += "http://local.live.com/default.aspx?v=2&cp=";
        body += pt.LatLong.latitude + "~" + pt.LatLong.longitude;
        body += "&lvl=12";
        body += "\n\n";
    }
    var url = 'mailto:?subject=My%20Virtual%20Earth%20Panel&body='
        + escape(body);
    window.open(url);
}

```

```
}
```

Finally you can write the function to clear the items from the panel.

```
function ClearPanel()  
{  
    while (panelPoints.length)  
    {  
        panelPoints.pop();  
    }  
    p.SetBody('');  
}
```

It would be good to add points to the Panel based on a search, that what you will learn in the next section.

Finding

In order to do a search we can use the Virtual Earth Search Manager. The only downside of this is that we need to access content that is outside of our domain and that will bring up a security warning to the end user.

IMPORTANT: This will only work if the browser allows cross domain data access. By default many browsers don't allow this.

In Internet Explorer you can change the access permissions to allow cross domain data access by following these steps:

1. From the Tools menu select Internet Options
2. Select the security tab
3. Click Custom Level
4. In the list find the Miscellaneous section
5. Change the 'Access data sources across domains' to Prompt

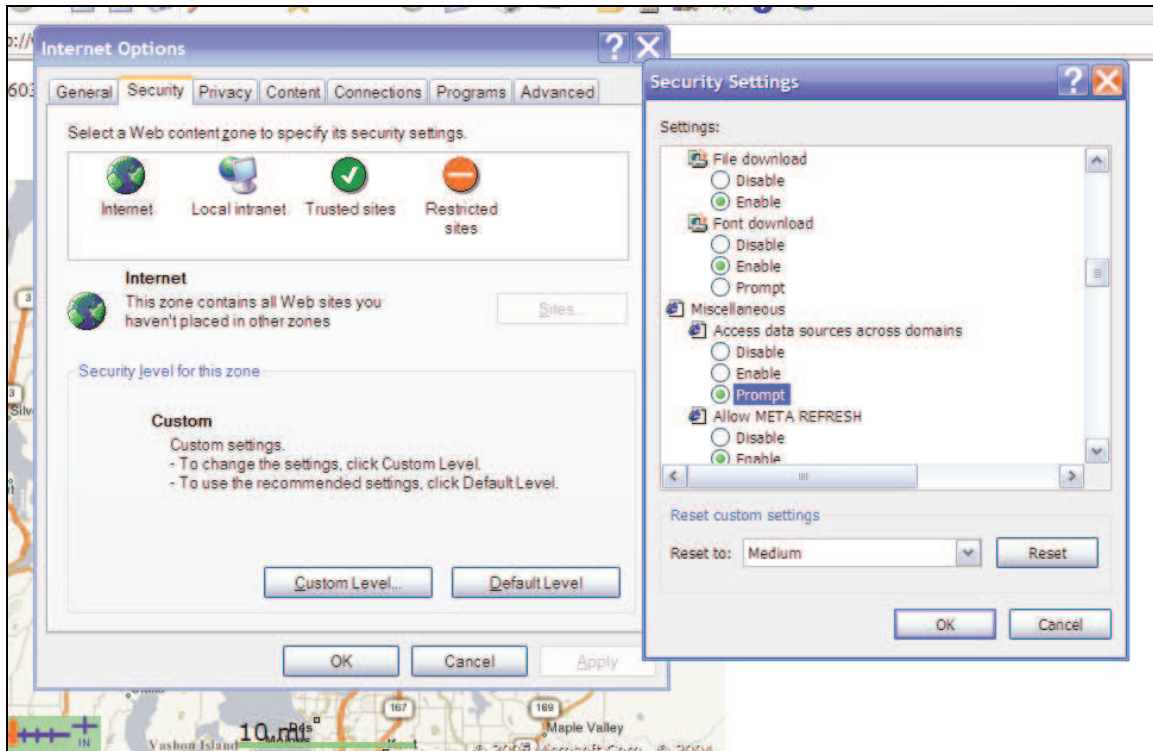


Figure 4 – Setting the access permissions to allow cross domain access

At the end of the body section of the HTML page add an input and button control to allow the user to enter some text to search for and click the button to start the search.

```

<input type="button" value="Find" onclick="DoFind()"
ID="FindButton" NAME="FindButton"
style="position:absolute;left:10px;top:600"/>
<input type="text" name="WhereText" size="20"
ID="WhereText"
style="position:absolute;left:60px;top:600"/>

```

To the script section of our page add a DoFind function.

```

function DoFind()
{
    var where = document.getElementById("WhereText").value;
    VE_SearchManager._ResetPaging();
    VE_SearchManager._CancelAllRequests();
    VE_SearchManager.searchPage="http://local.live.com/search.aspx";
    VE_SearchManager._DoSearch(where, where);
}

```

This page will perform a search and center the map at location found. It would now be great to add the found location to the Panel. In order to achieve this we will need to do more of the search ourselves and create a function to get called when the search has completed.

The new DoFind method now performs much of the activity done in the VE_SearchManager._DoSearch function.

```
function DoFind()
{
    var where = document.getElementById("WhereText").value;
    VE_SearchManager._ResetPaging();
    VE_SearchManager._CancelAllRequests();

    VE_SearchManager.searchPage="http://local.live.com/search.ashx";
    VE_SearchManager._DoSearch(where, where);

    var where = document.getElementById("WhereText").value;
    var a="";
    var b=escape(where);
    var c=map.GetLatitude(0);
    var d=map.GetLongitude(windowWidth);
    var e=map.GetLatitude(windowHeight);
    var f=map.GetLongitude(0);
    var g="";
    var i="";
    var r=0;
    var url="http://local.live.com/search.ashx"+
        "?a="+a+
        "&b="+b+
        "&c="+c+
        "&d="+d+
        "&e="+e+
        "&f="+f+
        "&g="+g+
        "&i="+i+
        "&r="+r;

    if(!VE_SearchManager.xmlHttp)
    {
        VE_SearchManager.xmlHttp=GetXmlHttp();
    }
    var xmlHttp=VE_SearchManager.xmlHttp;
    if(xmlHttp)
    {
        xmlHttp.open("POST",url,true);
        xmlHttp.onreadystatechange=FindResponseHandler;
        VE_SearchManager.searching=true;
        xmlHttp.send("");
    }
}

FindResponseHandler=function()
{
    var x=VE_SearchManager.xmlHttp;
    if(x.readyState==4)
    {
        VE_SearchManager.searching=false;
        var code=x.responseText;
        try
        {
```

```

        eval (code);
    }
    catch (ex)
    {
        //alert (ex);
        //alert ("Return:" + code);
    }
    var where =
        document.getElementById ("WhereText").value;
    AddPoint (where);
}
}

```

Conclusion

We have now created a web page that allows us to search for a location and add that location to our panel.

The full listing for the page is shown in the listing below.

```

<html>
<head>
    <title>My Virtual Earth</title>
<link href="http://local.live.com/css/MapControl.css"
    type="text/css" rel="stylesheet" />
<script src="http://local.live.com/MapControl.ashx">
</script>
<script src="http://local.live.com/js/ve.js">
</script>

<STYLE TYPE="text/css" MEDIA=screen>
<!--
.Compass{ width:54px; height:54px; background:url(i/compass.gif);
margin:0px; cursor:pointer }
.ZoomBar{ position:relative;background:url(i/zoom/bar.gif);
width:103px; height:20px; margin:2px; overflow:hidden; }
.ZoomBar_slider { position:absolute; background:url(i/zoom/slider.gif);
width:7px; height:20px; overflow:hidden; display:block; }
-->
</STYLE>

<STYLE TYPE="text/css" MEDIA=screen>
<!--
.pin
{
width:44px;height:17px;
font-family:Arial,sans-serif;
font-weight:bold;font-size:8pt;
color:White;overflow:hidden;
cursor:pointer;text-decoration:none;
text-align:center;background:#0000FF;
border:1px solid #FF0000;
z-index:5}

```

```

.VE_Panel_el
{
overflow:hidden;
z-index:31;
border:1px solid #cbcbcb;
padding:0;
margin:0;
background:white;
}

.VE_Panel_title
{
position:absolute;
padding-top:2px;
padding-left:5px;
overflow:hidden;
z-index:32;
font-family:Verdana,sans-serif;
font-size:8pt;
font-weight:bold;
color:rgb(230,250,255);
text-transform:uppercase;
cursor:default;
white-space:nowrap;
text-overflow:ellipsis;
}
.VE_Panel_title_blue{background:#0030cc}

.VE_Panel_cb
{
padding-left:1px;
width:18px;
height:18px;
color:white;
text-align:center;
font-size:7pt;
font-family:Verdana;
font-weight:bold;
overflow:hidden;
cursor:pointer
}

.VE_Panel_cb_blue{background:#001d7a;border:solid 2px #0030cc}

.VE_Panel_tb
{
height:18px;
padding-top:3px;
padding-left:2px;
font-family:Verdana,sans-serif;
font-size:8pt;
overflow:hidden
}

.VE_Panel_tb_blue{background:#ccd8ff}
.VE_Panel_tb td
{

```

```

font-family:Verdana,sans-serif;
font-size:8pt
}

.VE_Panel_tb a{color:#000080}
.VE_Panel_tb a:hover{color:#ff9900}

.VE_Panel_body
{
    padding:5px;
    font-family:Verdana,sans-serif;
    font-size:8pt;
    overflow:auto
}

-->
</STYLE>

<script>
var map = null;
var p;
PanelPoint = function(n, pt)
{
    this.Name = n;
    this.LatLong = pt;
}
//array of PanelPoint objects
var panelPoints = new Array();

function UpdateInfo(e)
{
    document.getElementById("info").innerHTML =
        'Latitude = ' +
        e.view.latlong.latitude +
        ', Longitude = '
        + e.view.latlong.longitude +
        ', Zoom=' +
        e.view.zoomLevel;
}

function MouseClick(e)
{
    map.RemovePushpin('pin');
    map.AddPushpin('pin',
        e.view.latlong.latitude,
        e.view.latlong.longitude,
        88,
        34,
        'pin',
        'MyPin',
        1);
}

function AddPoint(name)
{
    var body = "";
    var lat = map.GetCenterLatitude();

```

```

var lon = map.GetCenterLongitude();
var pt = new Msn.VE.LatLong(lat,lon);
var newPoint = new PanelPoint(name, pt);

panelPoints.push(newPoint);
var pts = panelPoints;
for(var i=0;i<pts.length;i++)
{
    var pt = panelPoints[i];
    body += "<a href='javascript:map.PanToLatLong(";
    body += pt.LatLong.latitude + "," +
        pt.LatLong.longitude;
    body += ");'>" + pt.Name;
    body += "</a> <br/>";
}
p.body.innerHTML = body;
}

function ClearPanel()
{
    while (panelPoints.length)
    {
        panelPoints.pop();
    }
    p.SetBody('');
}

function EmailPanel()
{
    var body = "http://ViaVirtualEarth.com/MyVirtualEarth/V2/
\n\n";
    var pts = panelPoints;
    for(var i=0;i<pts.length;i++)
    {
        var pt = panelPoints[i];
        body += pt.Name + "\n";
        body += "http://local.live.com/default.aspx?v=2&cp=";
        body += pt.LatLong.latitude + "~" +
pt.LatLong.longitude;
        body += "&lvl=12";
        body += "\n\n";
    }
    var url =
'mailto:?subject=My%20Virtual%20Earth%20Panel&body='
        + escape(body);
    window.open(url);
}

function GetToolbar()
{
    var html="<table cellpadding=\"0\" cellspacing=\"0\" ";
    html+="border=\"0\" align=\"left\">";
    html+="<tr><td valign=\"top\" align=\"center\">";
    html+="<a href=\"javascript:ClearPanel();\" ";
    html+="oncontextmenu=\"return
false;\">Clear</a>&nbsp;|&nbsp;";
    html+="<a href=\"javascript:AddPoint('Custom');\" ";

```

```

        html+="oncontextmenu=\"return
false;\">Add</a>&nbsp;|&nbsp;";
        html+="<a href=\"javascript:EmailPanel();\" ";
        html+="oncontextmenu=\"return false;\">Email</a>&nbsp;";
        html+="</td></tr><tr><td>&nbsp;</td></tr></table>";
        return html;
    }

    function OnPageLoad()
    {

        var params = new Object();
        params.latitude = 51.567;
        params.longitude = -0.026;
        params.zoomlevel = 10;
        params.mapstyle = Msn.VE.MapStyle.Road;
        params.showScaleBar = true;
        params.showDashboard = true;
        params.dashboardSize = Msn.VE.DashboardSize.Normal;
        params.dashboardX = 5;
        params.dashboardY = 5;

        map =
            new Msn.VE.MapControl(
                document.getElementById("myMap"),
                params);

        map.Init();

        map.AttachEvent("onendcontinuouspan",
            UpdateInfo);

        map.AttachEvent("onendzoom",
            UpdateInfo);

        map.AttachEvent("onclick",
            MouseClick);

        IOSec=new OutputEncoder();

        p=new VE_Panel("MyPanel",10,200,
            180,200,
            "blue",
            31,
            "My Panel",
            "Nothing here yet!",
            "footer",
            "Mouse Over!",true);

        p.onCloseClick=function(e)
        {
            p.Hide();
        };

        p.SetFooter("Here <a href='http://ViaVirtualEarth.com'>Via
VirtualEarth</a> ");
    }

```

```

        p.SetToolBar(GetToolBar());
    }

function ShowPanel()
{
    if (p!=null)
    {
        p.Show();
    }
}

function DoPanUp()
{
    map.ContinuousPan(0, -10, 20);
}
function DoPanDown()
{
    map.ContinuousPan(0, 10, 20);
}

function DoPanLeft()
{
    map.ContinuousPan(-10, 0, 20);
}
function DoPanRight()
{
    map.ContinuousPan(10, 0, 20);
}

function DoZoomIn()
{
    map.ZoomIn();
}

function DoZoomOut()
{
    map.ZoomOut();
}

function DoBasicFind()
{
    var where = document.getElementById("WhereText").value;
    VE_SearchManager._ResetPaging();
    VE_SearchManager._CancelAllRequests();

    VE_SearchManager.searchPage="http://local.live.com/search.ashx";
    VE_SearchManager._DoSearch(where, where);
}

function DoFind()
{
    var where = document.getElementById("WhereText").value;
    VE_SearchManager._ResetPaging();
    VE_SearchManager._CancelAllRequests();

    VE_SearchManager.searchPage="http://local.live.com/search.ashx";

```

```

VE_SearchManager._DoSearch( where, where);

var where = document.getElementById("WhereText").value;
var a="";
var b=escape( where);
var c=map.GetLatitude(0);
var d=map.GetLongitude(windowWidth);
var e=map.GetLatitude(windowHeight);
var f=map.GetLongitude(0);
var g="";
var i="";
var r=0;
var url="http://local.live.com/search.ashx"+
      "?a="+a+
      "&b="+b+
      "&c="+c+
      "&d="+d+
      "&e="+e+
      "&f="+f+
      "&g="+g+
      "&i="+i+
      "&r="+r;

if(!VE_SearchManager.xmlHttp)
{
    VE_SearchManager.xmlHttp=GetXmlHttp();
}
var xmlHttp=VE_SearchManager.xmlHttp;
if(xmlHttp)
{
    xmlHttp.open("POST",url,true);
    xmlHttp.onreadystatechange=FindResponseHandler;
    VE_SearchManager.searching=true;
    xmlHttp.send("");
}
}

FindResponseHandler=function()
{
    var x=VE_SearchManager.xmlHttp;
    if(x.readyState==4)
    {
        VE_SearchManager.searching=false;
        var code=x.responseText;
        try
        {
            eval( code);
        }
        catch(ex)
        {
            //alert(ex);
            //alert("Return:" + code);
        }
        var where =
            document.getElementById("WhereText").value;

```

```

        AddPoint (where);
    }
}

</script>
</head>

<body onLoad="OnPageLoad()" >
<div id="info" style="HEIGHT: 50px;font-size:10pt">
</div>
<div id="myMap"
    style="WIDTH: 600px; HEIGHT: 400px; OVERFLOW:hidden">
</div>

<input type="button" value="Pan Up"
onclick="DoPanUp()"
ID="PanUpButton" NAME="PanUpButton"
style="position:absolute;left:60px;top:500"/>

<input type="button" value="Pan Left"
onclick="DoPanLeft()"
ID="PanLeftButton" NAME="PanLeftButton"
style="position:absolute;left:10px;top:530"/>

<input type="button" value="Pan Right"
onclick="DoPanRight()"
ID="PanRightButton" NAME="PanRightButton"
style="position:absolute;left:100px;top:530"/>

<input type="button" value="Pan Down"
onclick="DoPanDown()"
ID="PanDownButton" NAME="PanDownButton"
style="position:absolute;left:45px;top:560"/>

<input type="button" value="Zoom In" onclick="DoZoomIn()"
ID="ZoomInButton" NAME="ZoomInButton"
style="position:absolute;left:250px;top:530"/>

<input type="button" value="Zoom Out" onclick="DoZoomOut()"
ID="ZoomOutButton" NAME="ZoomOutButton"
style="position:absolute;left:340px;top:530"/>

<input type="button" value="Show Panel" onclick="ShowPanel()"
ID="ShowPanelButton" NAME="ShowPanelButton"
style="position:absolute;left:480px;top:465;"/>

<input type="button" value="Find" onclick="DoFind()"
ID="FindButton" NAME="FindButton"
style="position:absolute;left:10px;top:600"/>
<input type="text" name="WhereText" size="20" ID="WhereText"
style="position:absolute;left:60px;top:600"/>

</body>

```

```
</html>
```