Integrating Oracle Application Express with Oracle MapViewer

An Oracle White Paper
April 2008

Integrating Oracle Application Express with Oracle MapViewer

INTRODUCTION

Oracle Application Express – a feature of the Oracle Database – is a powerful and easy to use web application development platform. With Oracle Application Express, you can quickly develop and deploy applications in a matter of hours, often without writing a single line of code. Application Express can be easily integrated with the Oracle Fusion Middleware MapViewer's JavaScript API (also known as Oracle Maps API). The Oracle Maps API allows you to create online mapping applications and visualize enterprise information stored in your Oracle database on a map.

Embedding Oracle Maps into your Application Express applications is accomplished using JavaScript to invoke the Oracle Maps API. It requires that you have a separate MapViewer server instance running either in a standalone OC4J or in an Oracle middleware installation. For starters, you can download the latest MapViewer 11g R1 Preview 2 from OTN and deploy to a standalone OC4J 11g Preview kit. In this paper we will be using the MVDEMO sample data set that comes with MapViewer distribution as the source of our geographic data. It is assumed that the MapViewer instance already has a data source named "mvdemo" connecting to the MVDEMO database schema.

Create an Application

To create an Application Express application:

1. Login to your Application Express instance.
Please note that this tutorial cannot be demonstrated on apex.oracle.com because it does not support external network call-outs. You will need to use your own instance to be able to execute the resulting application.

2. Select **Application Builder**

3. Click **Create >**

4. Supply information as required by the Create Application wizard

5. Add one blank page on the Pages step of the wizard

6. Once you have created the application, if your environment requires a proxy server to reach pages on the Internet, supply the proxy server in the Proxy Server field of the Application Definition (Shared Components > Definition - under Application.) This is only necessary if your MapViewer server must be accessed through a proxy server.

Create an HTML Region on the Page for the Embedded Map

To create an HTML region on the page:

1. Click the create icon in the Regions section

2. Select **HTML** for the region type and click **Next** >

3. Select **HTML** from the region container list and click **Next** >

4. Enter **Map** in the Title field

5. Choose **No Template** from the Region Template list

6. Click **Next** >

7. Enter the following in the HTML Text Region Source text area:

```
<div id="map" style="width: 600px; height: 400px"></div>
```

8. Click **Create Region**

Call Oracle Maps API With JavaScript

You now call the Oracle Maps API with JavaScript embedded in this page. The JavaScript is added to the HTML Header of the page and then the map is initialized in the onload event of the page.
To embed JavaScript in this page:

1. Click the edit icon  in the Page section

2. Add the following JavaScript code in the HTML Header text area:

```
<script language="Javascript"
src="http://sdolnx2.us.oracle.com:7777/mapviewer/fsmc/jslib/oraclemaps.js">
</script>

<script type="text/javascript">
  var mapview;
  var themebasedfoi;
  function initMap()
  {
    var baseURL  = "http://sdolnx2.us.oracle.com:7777/mapviewer";
    var mapCenterLon = -122.5;
    var mapCenterLat =  36.5;
    var mapZoom      =  2;
    var mpoint = MVSdoGeometry.createPoint(mapCenterLon,mapCenterLat,8307);
    mapview = new MVMapView(document.getElementById("map"), baseURL);
    mapview.addMapTileLayer(new MVMapTileLayer("mvdemo.demo_map"));
    mapview.setCenter(mpoint);
    mapview.setZoomLevel(mapZoom);

    themebasedfoi = new MVThemeBasedFOI('themebasedfoi1','mvdemo.customers');

    themebasedfoi.setBringToTopOnMouseOver(true);
    mapview.addThemeBasedFOI(themebasedfoi);

    var navPan = new MVMapDecoration(new MVNavigationPanel(),
                                     0,0,null,null,4,4) ;
    mapview.addMapDecoration(navPan) ;
    mapview.display();
  }
</script>
```

Note that in the above code the bold texts are things you can or should change. For instance the MapViewer URL (`http://sdolnx2.us.oracle.com:7777/mapviewer`) should be changed to that of your particular MapViewer instance.

Note also the above code is based on the Oracle Maps API of MapViewer version 11g R1 Preview 2. If your MapViewer server is of an earlier version, you may need to modify the code accordingly.

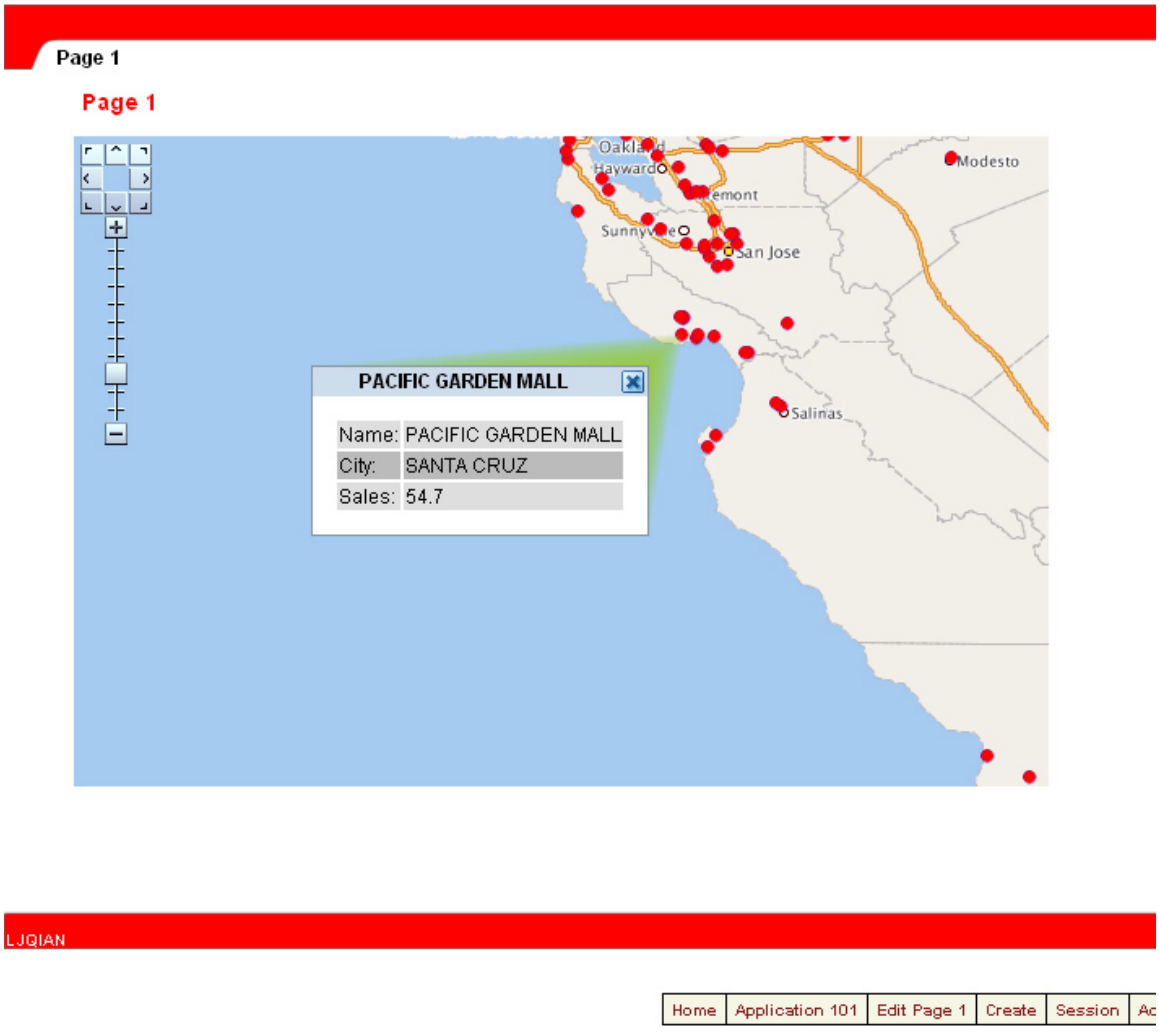
3. Add the following code to the Page HTML Body Attribute text area:

```
onload="initMap()"
```

Note that if you get an error about page focus, you may want to disable the page focus by selecting "Do not focus cursor" for Cusor Focus in the "Display Attributes".
4. Click **Apply Changes**

Test the Application

You can now test the mapping application by clicking the Run page button . You should see a page like the following:



Resources

Oracle Application Express Home http://apex.oracle.com/otn
Oracle Fusion Middleware MapViewer Home: MapViewer Home

Integrating Oracle Application Express with Oracle Maps
April 2008
Author: Liujian Qian
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com