# HANDS-ON TUTORIAL:

## *Using Oracle MapViewer HTML5 JavaScript APIs in Oracle Business Intelligence Enterprise Edition*

## TABLE OF CONTENTS

## DESCRIPTION

This tutorial provides step-by-step instructions to create simple examples that use MapViewer's HTML5-based Javascript API within Oracle Business Intelligence Enterprise Edition (OBIEE).

The new MapViewer V2 API delivers several performance and feature improvements over previous releases, which enable users to:

- Render maps in BI analyses and reports at runtime using computed fields - without needing predefined BI metadata mappings
- Mash up analytical data with external spatial or non-spatial data
- Boost user experience and interactivity within a map, and between maps and other BI components.

## PREREQUISITES

- Download and Deploy OBIEE SampleApp from OTN

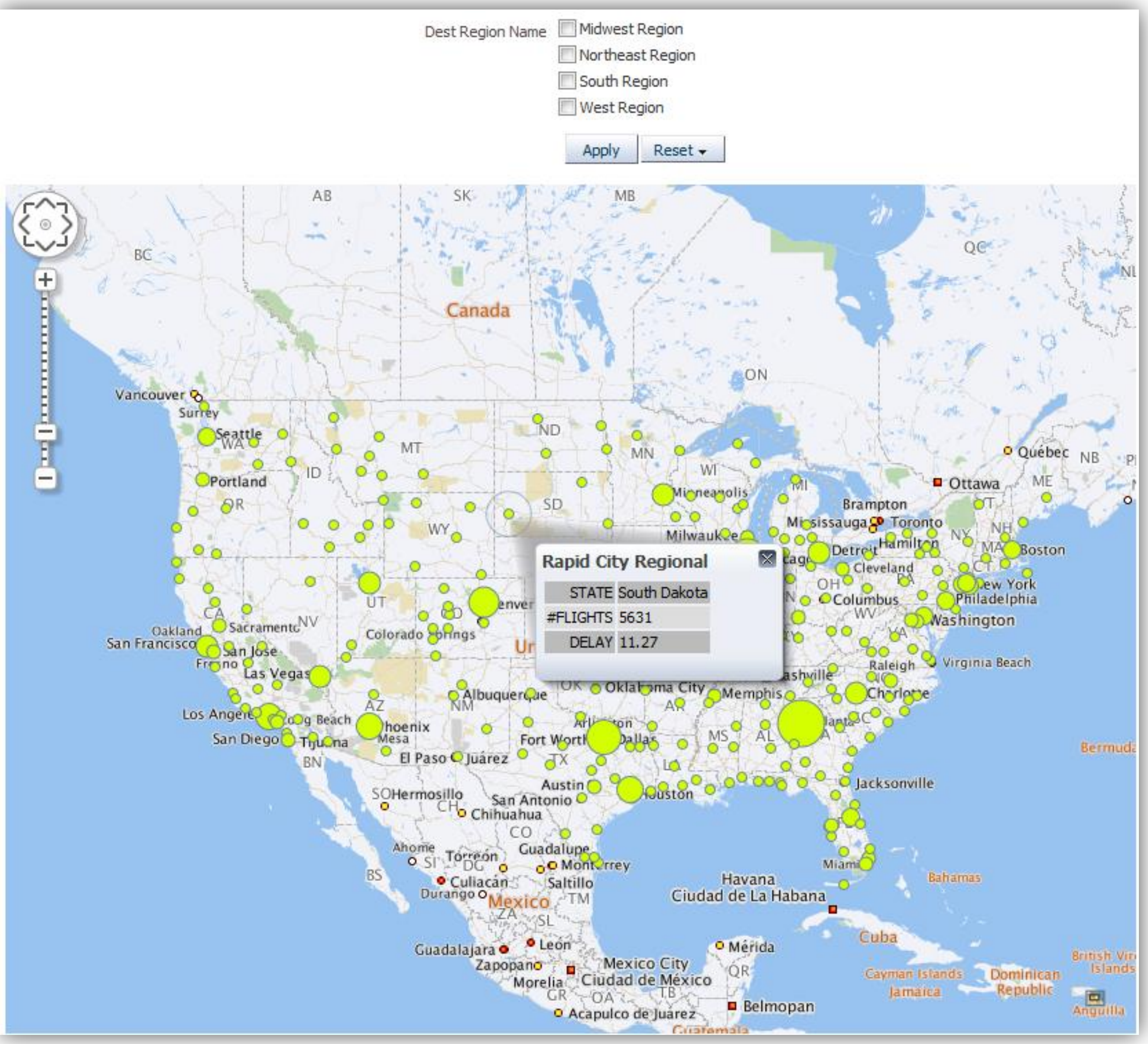
These tutorials are based on OBIEE SampleApp V406. Some knowledge of HTML and Javascript, and familiarity with Oracle Business Intelligence Enterprise Edition (OBIEE) is required to understand these examples.

## HARDWARE AND SOFTWARE REQUIREMENTS

- Laptop/Desktop (quad-core preferred)
- Virtual Box 4.2 or later (www.virtualbox.org)
- V2 API will only work with modern browsers that support HTML5 canvas. Firefox browser is strongly recommended. Few versions of IE/Chrome were observed to show inconsistent issues with certain BI components.
- Minimum 8Gb RAM: The VM requires 6GB and two processors to run
- Minimum 90Gb free disk space (the image is approx 28Gb download that requires an additional 60 Gb of disk during import)

## EXAMPLE 1: DISPLAY US AIRPORT LOCATIONS

Source airport's location and metrics from a BI report and then display them on a US map. Display airport location as bubbles with size indicative of departing flights volume. Once completed the dashboard page should look like the following screenshot:



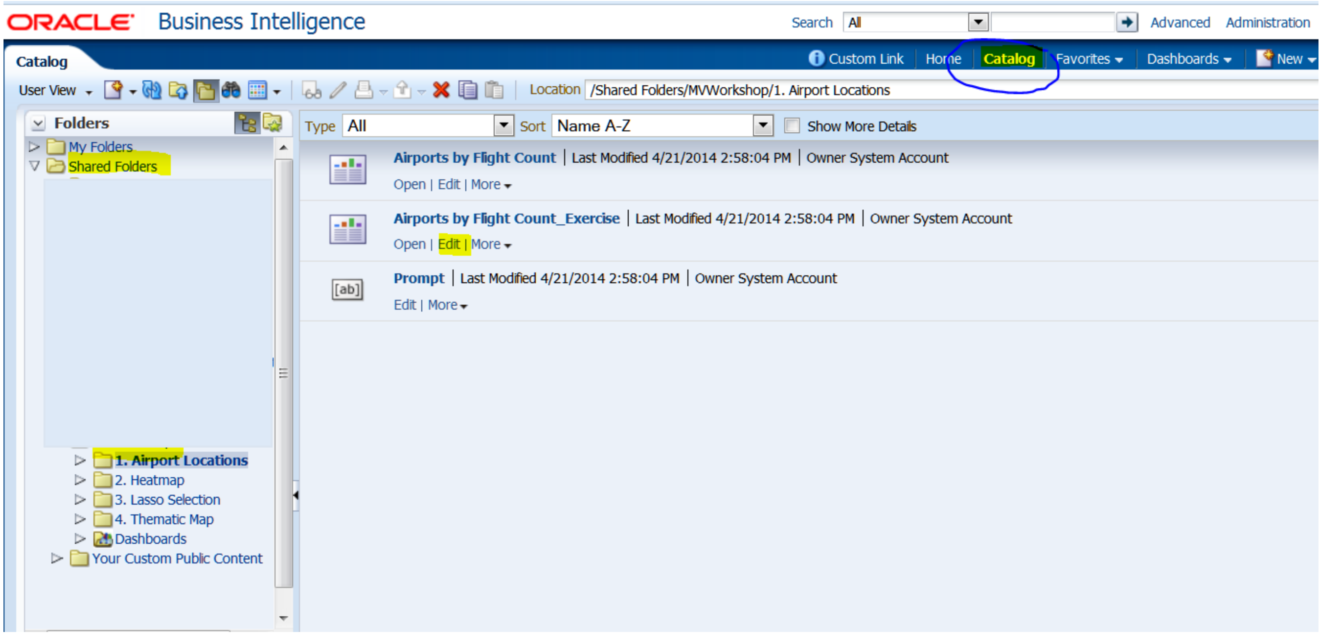
Dest Region Name
□ Midwest Region
□ Northeast Region
□ South Region
□ West Region

Apply Reset ▾

Rapid City Regional

| | |
|---|---|
| STATE | South Dakota |
| #FLIGHTS | 5631 |
| DELAY | 11.27 |

- Edit "Airports by Flight Count_Exercise" report from folder: `BI Catalog > Shared Folders > Workshops > HTML5 Maps > 1. Airport Locations`



Location /Shared Folders/MVWorkshop/1. Airport Locations

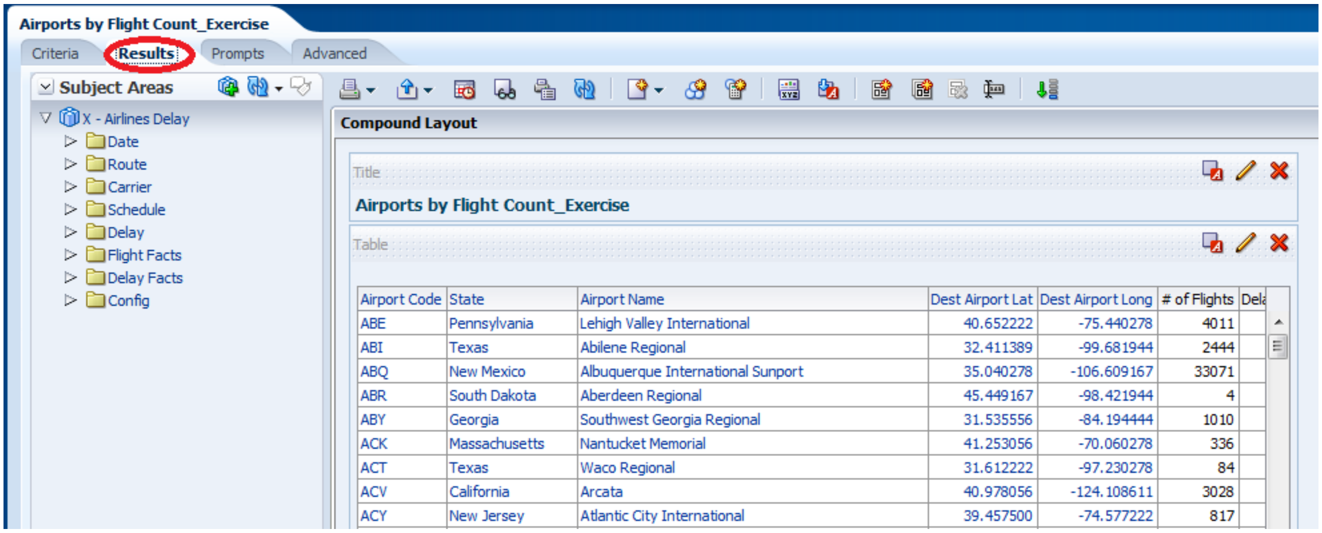
Type All Sort Name A-Z □ Show More Details

Airports by Flight Count | Last Modified 4/21/2014 2:58:04 PM | Owner System Account
Open | Edit | More

Airports by Flight Count_Exercise | Last Modified 4/21/2014 2:58:04 PM | Owner System Account
Open | Edit | More

Prompt | Last Modified 4/21/2014 2:58:04 PM | Owner System Account
Edit | More

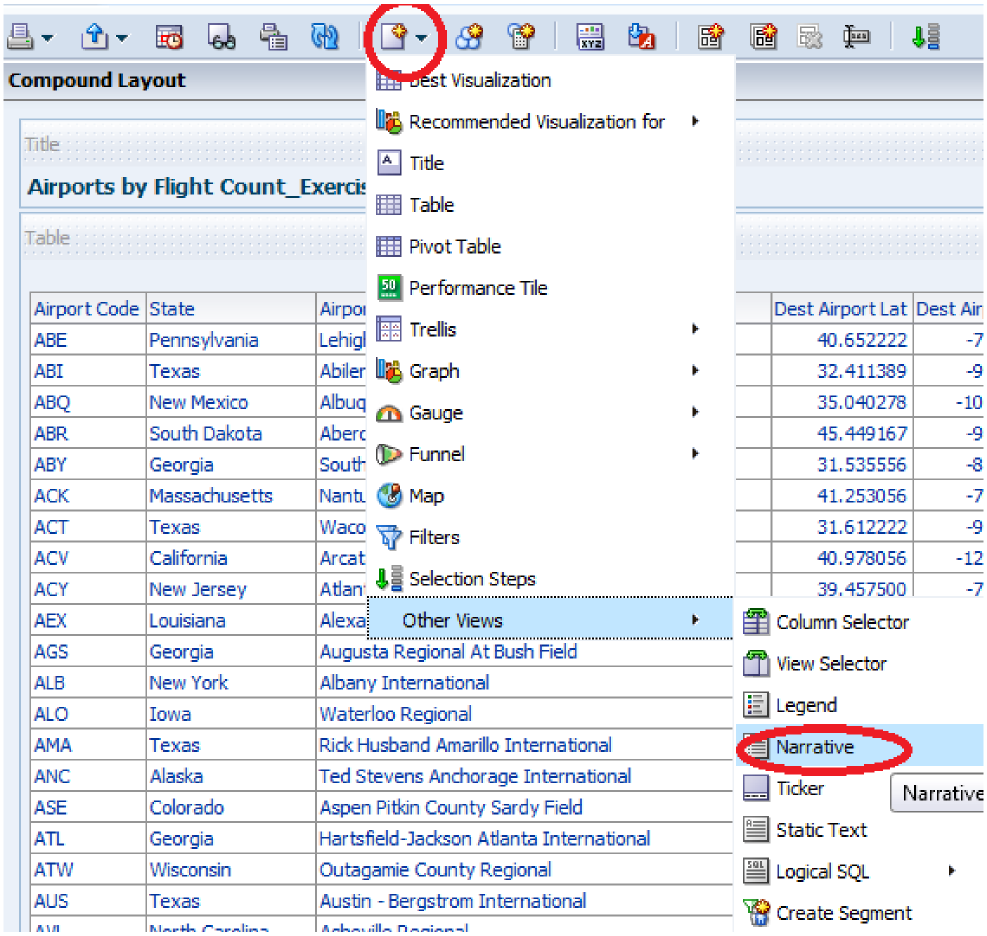
- Open the results tab and observe the BI data presented in a default tabular form. This is data is going to serve as input to our custom map view.



Airports by Flight Count_Exercise

| Airport Code | State | Airport Name | Dest Airport Lat | Dest Airport Long | # of Flights | Dela |
|---|---|---|---|---|---|---|
| ABE | Pennsylvania | Lehigh Valley International | 40.652222 | -75.440278 | 4011 | |
| ABI | Texas | Abilene Regional | 32.411389 | -99.681944 | 2444 | |
| ABQ | New Mexico | Albuquerque International Sunport | 35.040278 | -106.609167 | 33071 | |
| ABR | South Dakota | Aberdeen Regional | 45.449167 | -98.421944 | 4 | |
| ABY | Georgia | Southwest Georgia Regional | 31.535556 | -84.194444 | 1010 | |
| ACK | Massachusetts | Nantucket Memorial | 41.253056 | -70.060278 | 336 | |
| ACT | Texas | Waco Regional | 31.612222 | -97.230278 | 84 | |
| ACV | California | Arcata | 40.978056 | -124.108611 | 3028 | |
| ACY | New Jersey | Atlantic City International | 39.457500 | -74.577222 | 817 | |

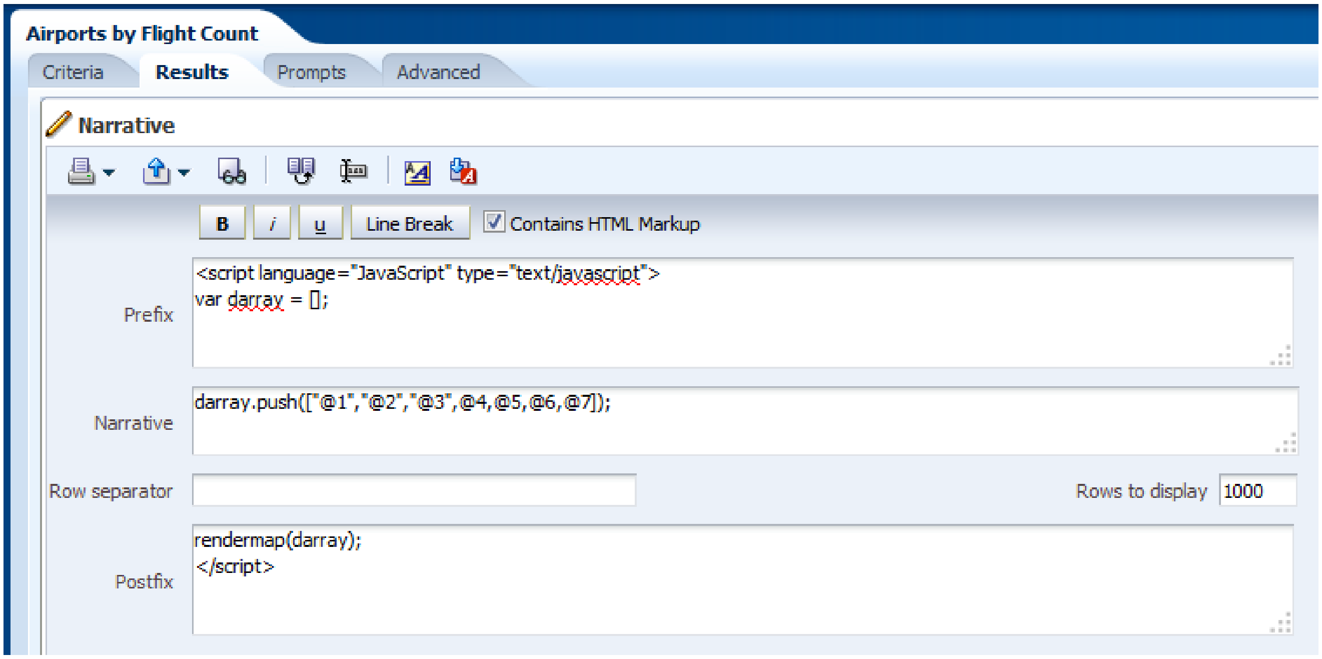
- Add a narrative view to this report.



Airports by Flight Count_Exercis

| Airport Code | State | Airpor | Dest Airport Lat | Dest Air |
|---|---|---|---|---|
| ABE | Pennsylvania | Lehigl | 40.652222 | -7 |
| ABI | Texas | Abiler | 32.411389 | -9 |
| ABQ | New Mexico | Albuq | 35.040278 | -10 |
| ABR | South Dakota | Aberc | 45.449167 | -9 |
| ABY | Georgia | South | 31.535556 | -8 |
| ACK | Massachusetts | Nantu | 41.253056 | -7 |
| ACT | Texas | Waco | 31.612222 | -9 |
| ACV | California | Arcat | 40.978056 | -12 |
| ACY | New Jersey | Atlan | 39.457500 | -7 |
| AEX | Louisiana | Alexa | | |
| AGS | Georgia | Augusta Regional At Bush Field | | |
| ALB | New York | Albany International | | |
| ALO | Iowa | Waterloo Regional | | |
| AMA | Texas | Rick Husband Amarillo International | | |
| ANC | Alaska | Ted Stevens Anchorage International | | |
| ASE | Colorado | Aspen Pitkin County Sardy Field | | |
| ATL | Georgia | Hartsfield-Jackson Atlanta International | | |
| ATW | Wisconsin | Outagamie County Regional | | |
| AUS | Texas | Austin - Bergstrom International | | |
| AVL | North Carolina | Asheville Regional | | |

- Edit the newly created Narrative view (click on the pencil icon on the right). Then refer to the screen-shot below and build the narrative view.



☑ Contains HTML Markup

Prefix

```
<script language="JavaScript" type="text/javascript">
var darray = [];
```

Narrative

```
darray.push(["@1","@2","@3",@4,@5,@6,@7]);
```

Row separator

Rows to display 1000

Postfix

```
rendermap(darray);
</script>
```

This will add a script to the page to render the map. The prefix is text to add to the view before the report content is processed. The Narrative section is executed once per row of the report and builds up the data array which is passed to the rendermap function (which itself is defined next in a Static Text view). The variables @1 through @7 refer to the current content of that column in the report. Columns that contain string values (such as Airport Code, State, and Airport Name) should be enclosed in quotes.

Add Prefix as:

```
    <script language="JavaScript" type="text/javascript">
    var darray = [];
```

Add Narrative as:

```
    darray.push(["@1","@2","@3",@4,@5,@6,@7]);
```

Add Postfix as:

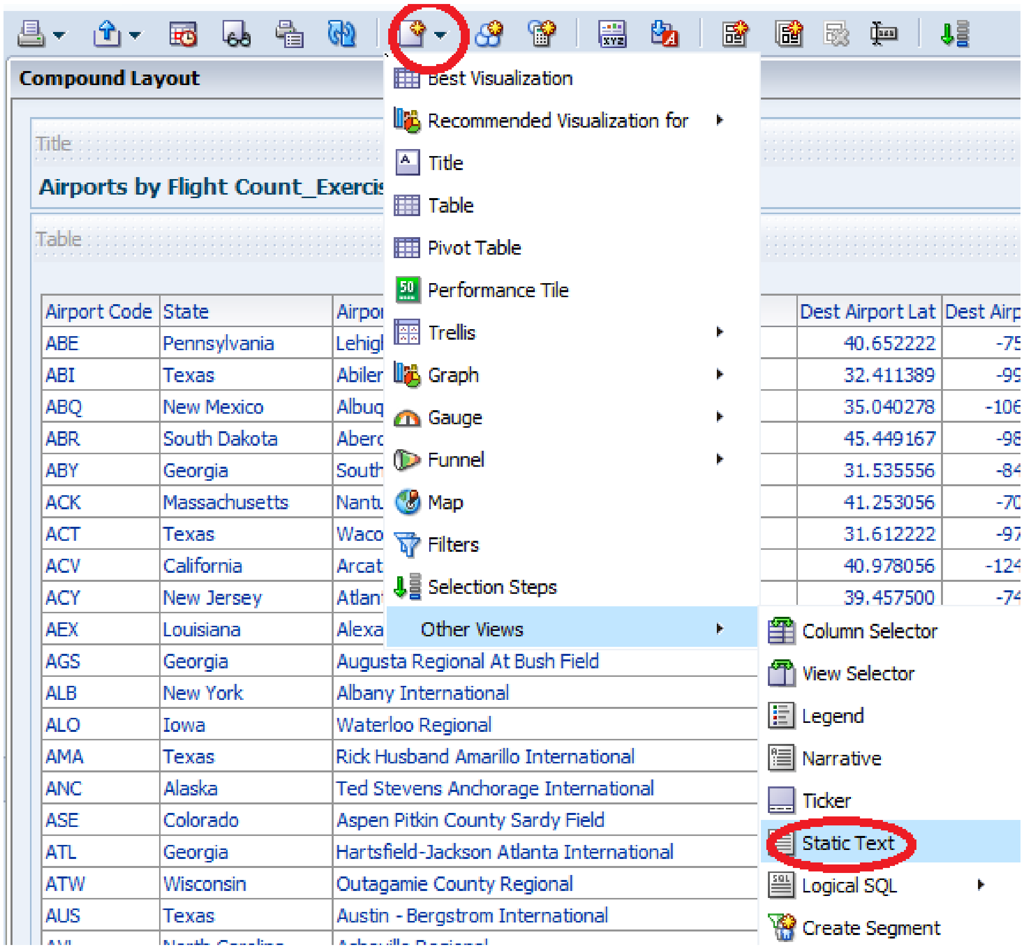
```
    rendermap(darray);
    </script>
```

Set rows to display = 1000

Tick “Contains HTML Markup” checkbox.

Click the “Done” button when completed.

- Add a Static Text view. Scroll down if necessary and edit the view.

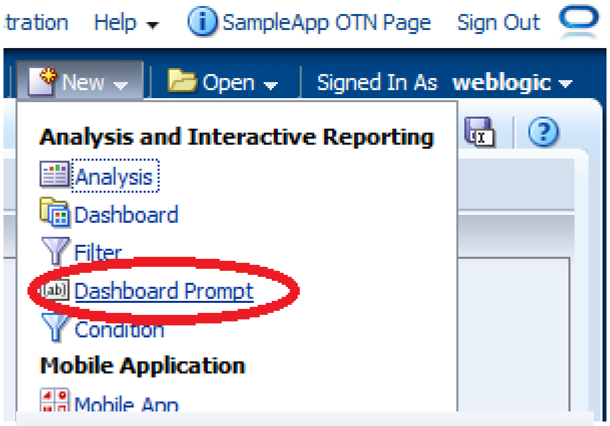


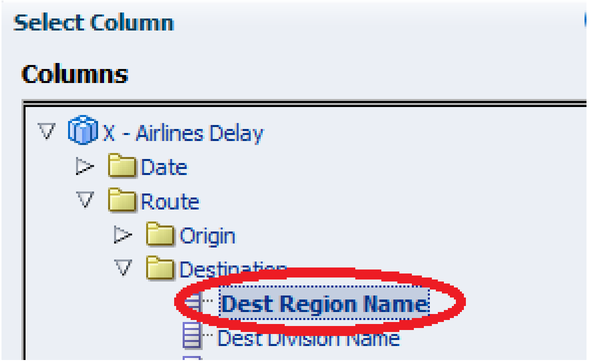
| Airport Code | State | Airpo | Dest Airport Lat | Dest Airp |
|---|---|---|---|---|
| ABE | Pennsylvania | Lehigl | 40.652222 | -75 |
| ABI | Texas | Abiler | 32.411389 | -99 |
| ABQ | New Mexico | Albuq | 35.040278 | -106 |
| ABR | South Dakota | Aberc | 45.449167 | -98 |
| ABY | Georgia | South | 31.535556 | -84 |
| ACK | Massachusetts | Nantu | 41.253056 | -70 |
| ACT | Texas | Waco | 31.612222 | -97 |
| ACV | California | Arcat | 40.978056 | -124 |
| ACY | New Jersey | Atlan | 39.457500 | -74 |
| AEX | Louisiana | Alexa | | |
| AGS | Georgia | Augusta Regional At Bush Field | | |
| ALB | New York | Albany International | | |
| ALO | Iowa | Waterloo Regional | | |
| AMA | Texas | Rick Husband Amarillo International | | |
| ANC | Alaska | Ted Stevens Anchorage International | | |
| ASE | Colorado | Aspen Pitkin County Sardy Field | | |
| ATL | Georgia | Hartsfield-Jackson Atlanta International | | |
| ATW | Wisconsin | Outagamie County Regional | | |
| AUS | Texas | Austin - Bergstrom International | | |
| AVL | North Carolina | Asheville Regional | | |

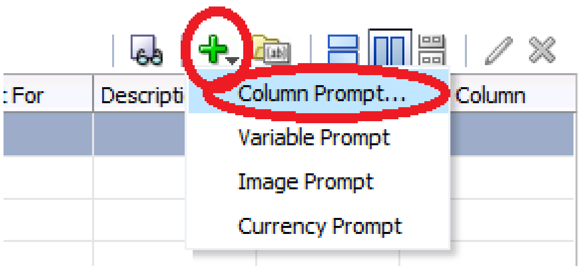
- Insert following JavaScript code (in blue color) into static text field. Remove extraneous CR/LF chars if any are introduced in the copy-n-paste process. This code does the following:
  - Defines a DIV for the map display and for handling user interactions.
  - Includes the Oracle Maps V2 javascript library.
  - Defines the Marker style (variable-size green circles) for the airport locations.
  - Defines the rendermap() function which creates the airport locations layer, sets up event handlers, displays the airport using the marker style.

```
<DIV id=map style="width:800px;height:600px;padding:10px;"></DIV>
<script language='JavaScript'
src='/mapviewer/jslib/v2/oraclemapsv2.js'></script>

<script language="JavaScript" type="text/javascript">
function setupAirportStyles(layer)
{
   var circles = [];
   //var fillColors =
['#34CA16','#79C636','#EAF167','#E9E753','#F9B975','#E6A23C','#EB8075','#F
A5E5E','#EB1919','#B32205'];
   for(var i=0; i<10; i++)
   {
   circles[i] =  new OM.style.Marker({
         vectorDef : [{
             shape: { type:"circle", cx:5, cy:5, width:10, height:10 },
             style: {
             fill:"#D4FF00",
             stroke:"#6784A3"
             }
          }],
          width: 7 + i*3,
          height:7 + i*3
       });
   }

   var bucketStyle = new OM.style.BucketStyle(
   {
   styles: circles,
   numClasses:10,
   classification:'equal',  //creates equal-ranged buckets
   defaultStyle: circles[0]
   });
   layer.setRenderingStyle(bucketStyle, ["#FLIGHTS"]);
}

function rendermap(darray){
   var baseURL  = "http://"+document.location.host+"/mapviewer";
   var center = new OM.geometry.Point(-100,40,8307);
   var map = new OM.Map(document.getElementById('map'),{mapviewerURL:
baseURL/*,disableOverviewMap:true*/}) ;

   var tilelayer = new OM.layer.ElocationTileLayer("oracle.maps");
   map.addLayer(tilelayer) ;

   var layer = new OM.layer.VectorLayer("div", {
   def:{
      type:OM.layer.VectorLayer.TYPE_PREDEFINED,
      dataSource:"OBIEE_NAVTEQ_Sample",
      theme:"OBIEE_US_DIVISIONS"
   }
   });
   layer.setRenderingStyle( new
OM.style.Color({fill:'#A6CEE3',fillOpacity:0.9,strokeThickness:2,stroke:'#
ffffff'}));
   //map.addLayer(layer);

   var airportsLayer = new OM.layer.VectorLayer("airports",
   {
   def:{
      type:OM.layer.VectorLayer.TYPE_LOCAL
   }
   });

   /*"Code","State","Airport","Lat","Long","# Flights","Delay Mins"*/
   for (i=0;i<darray.length; i++){
         var fid = darray[i][0];
         var mpoint = new OM.geometry.Point(darray[i][4],
darray[i][3],8307);
         var f_attr = {attributes :{"_LABEL_":darray[i][2],
"STATE":darray[i][1], "#FLIGHTS":darray[i][5]+"",
"DELAY":darray[i][6]+""}};
         var feature = new OM.Feature(fid,mpoint,f_attr);
         airportsLayer.addFeature(feature);
      }

   setupAirportStyles(airportsLayer);

   /*var pulse = new OM.style.PulseAnimation({
      stroke: "#336699",
      duration: 1,
      beginSize: 8,
      endSize: 50,
      repeatCount:3
   });
   airportsLayer.setSelectStyle(pulse);*/

   //airportsLayer.enableFeatureSelection(true);
   //airportsLayer.enableInfoWindow(true);
   //airportsLayer.setLabelsVisible(true);
   //airportsLayer.setBringToTopOnMouseOver(false);

   //map.addMapDecoration(new OM.control.NavigationPanelBar({ style:1 }));
   map.addLayer(airportsLayer) ;
   map.setMapCenter( center );
   map.setMapZoomLevel(3) ;

   map.init();
}
</script>
```

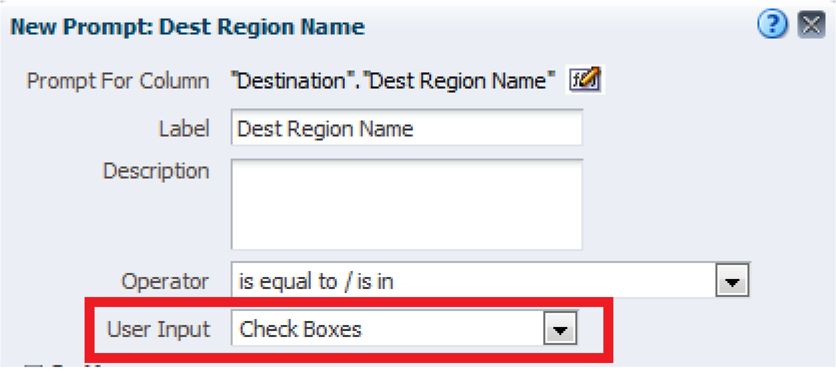
- Tick “Contains HTML Markup” checkbox and click the “Done” button when completed.
- Remove Title and Table views from the compound layout
- Optional: add additional views (charts, tables, etc) to the compound layout
- **Important**: Sequence the views so that Narrative view is the last view in the compound layout. Click on the title bar for the Static Text view and drag it to above the Narrative view. A blue line will show up indicating where the view will be inserted.
- Verify that the compound layout contains the static text and narrative views. Save the Analysis.
- Next, create a dashboard column prompt. Click on New and choose Dashboard Prompt.
- Select Subject Area “X – Airlines Delay”
- Click on the green + to add a column prompt for “Route”.”Destination”.“Dest Region Name”. Expand the Route and then Destination folders and select Dest Region Name.







- Set User Input type as CheckBoxes and select OK


New Prompt: Dest Region Name

Prompt For Column "Destination"."Dest Region Name"

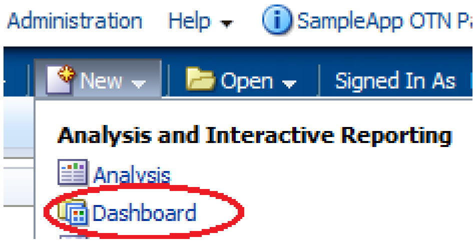
Label: Dest Region Name

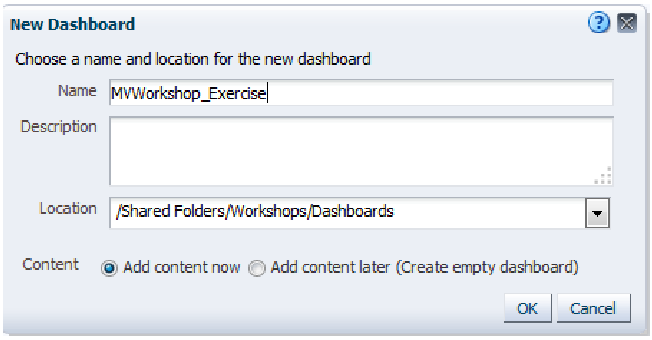
Description:

Operator: is equal to / is in

User Input: Check Boxes

- Save the prompt as “`Prompt_Exercise`” under the folder: `Shared Folders > Workshops > HTML5 Maps > 1. Airport Locations`
- Now click on New to create a new dashboard named “MVWorkshop_Exercise” and choose /Shared Folders/Workshops/Dashboards from the Location drop-down list.





**New Dashboard**

Choose a name and location for the new dashboard

Name: MVWorkshop_Exercise

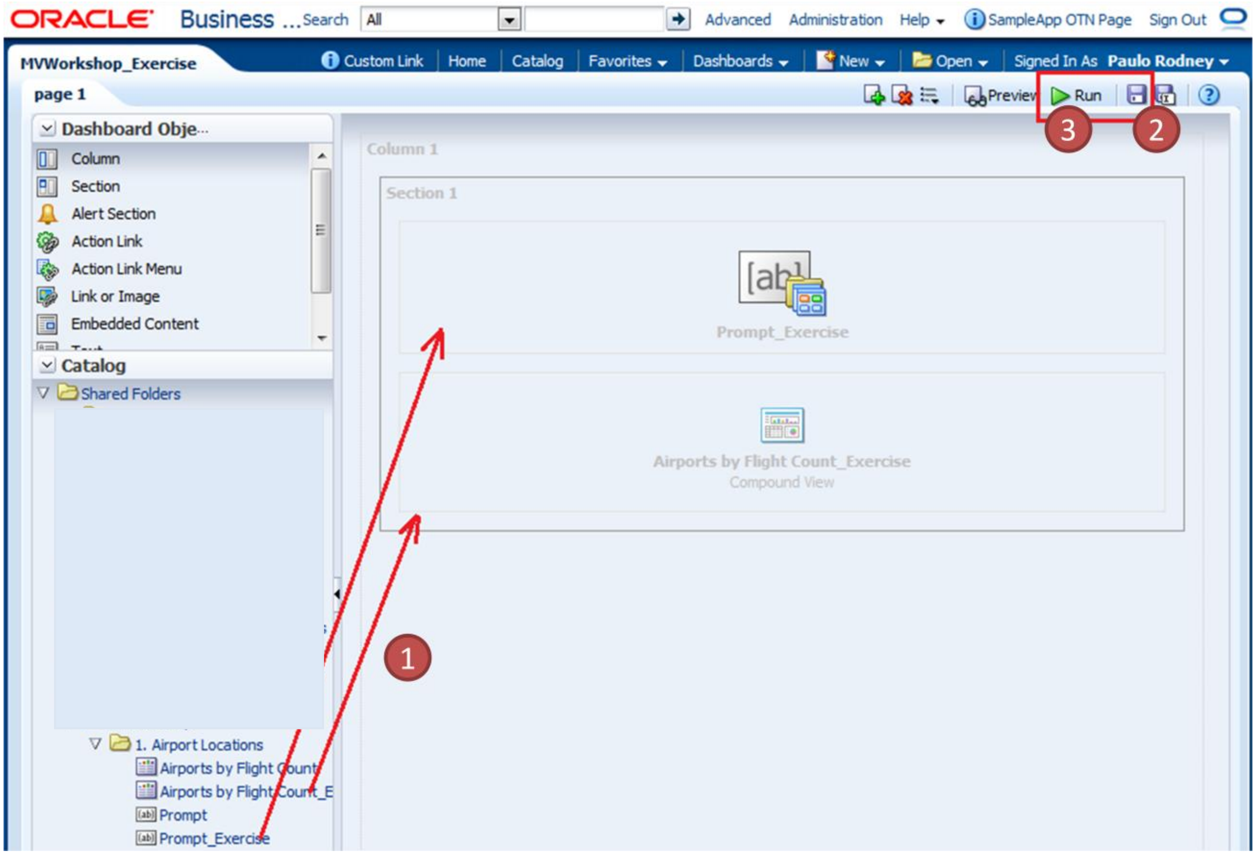
Description:

Location: /Shared Folders/Workshops/Dashboards

Content: ● Add content now ○ Add content later (Create empty dashboard)

OK Cancel

- Expand the Shared Folders > Workshops > HTML5 Maps > Airport Locations folders in the Catalog panel. Drag and drop the Prompt and Analysis that we just created on to dashboard.
- Save the page and run it.



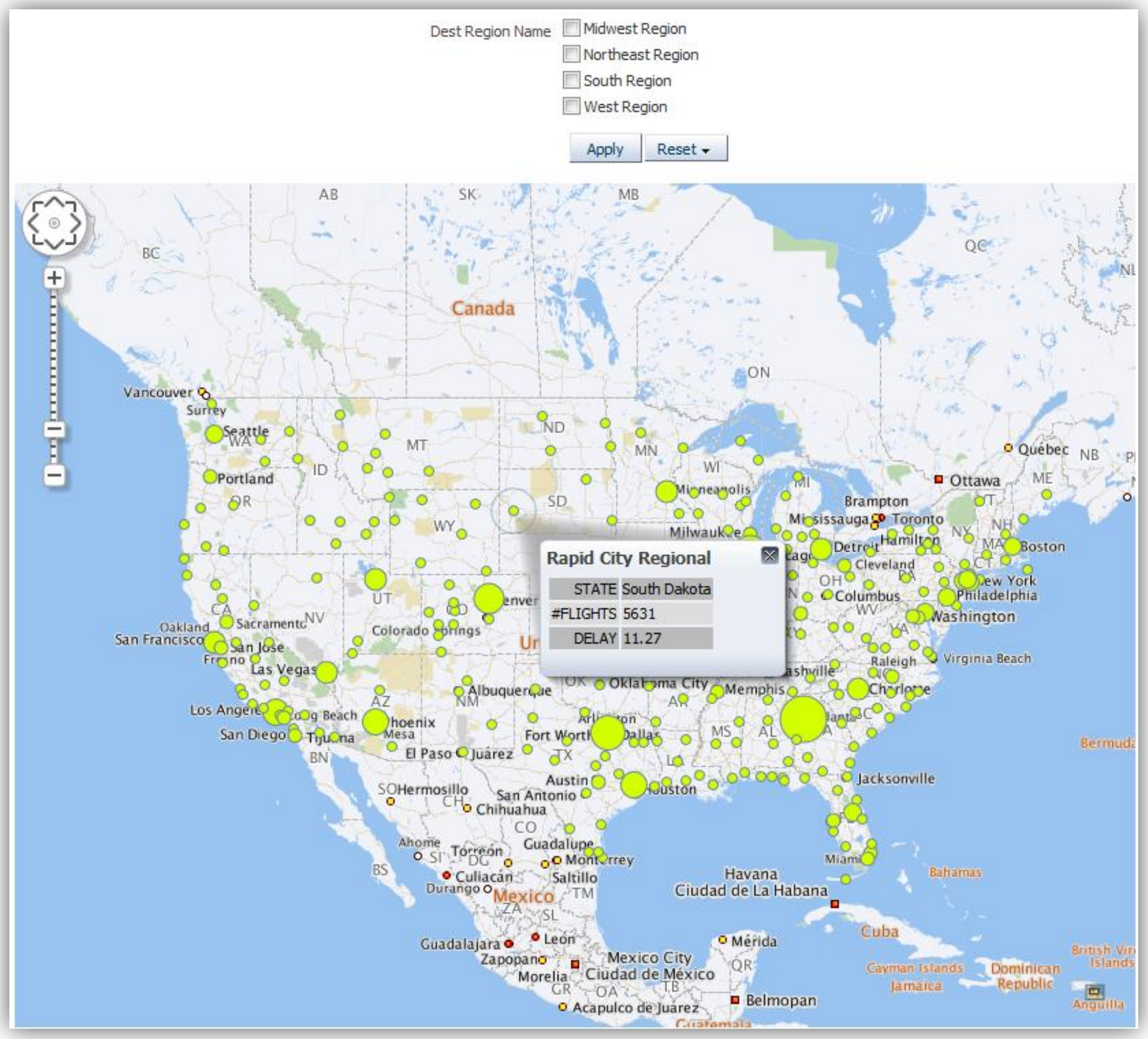
- You should now be able to see the map and the airport locations indicated as bubbles on that map
- Next, you may optionally add a Navigation Panel (zoom, pan controls) to the map by uncommenting the following line within the static text view in the analysis:
`map.addMapDecoration(new OM.control.NavigationPanelBar({ style:1 }));`

- Open a separate browser window, edit the analysis we created earlier, and then edit the static text view to uncomment the above line from the script.
- Go to the previous browser window where the dashboard page is already open. Refresh the window to view the changes.

- Optional: Similarly, uncomment the following lines of code to enable pulse animation at airport locations.

```
var pulse = new OM.style.PulseAnimation({
      stroke: "#336699",
      duration: 1,
      beginSize: 8,
      endSize: 50,
      repeatCount:3
   });
   airportsLayer.setSelectStyle(pulse);
```

- Screenshot of the resulting dashboard page:



Dest Region Name
□ Midwest Region
□ Northeast Region
□ South Region
□ West Region

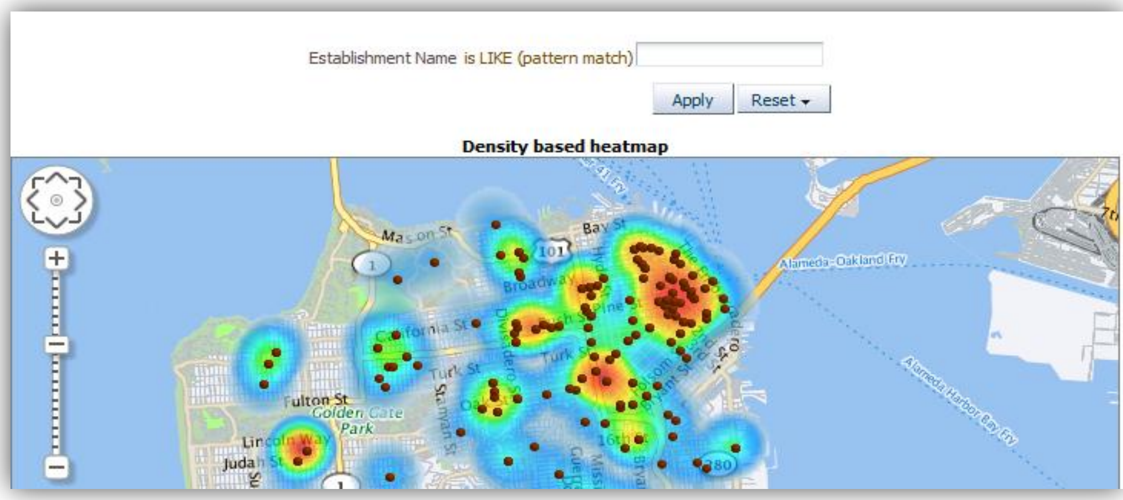
Apply | Reset ▾

Rapid City Regional

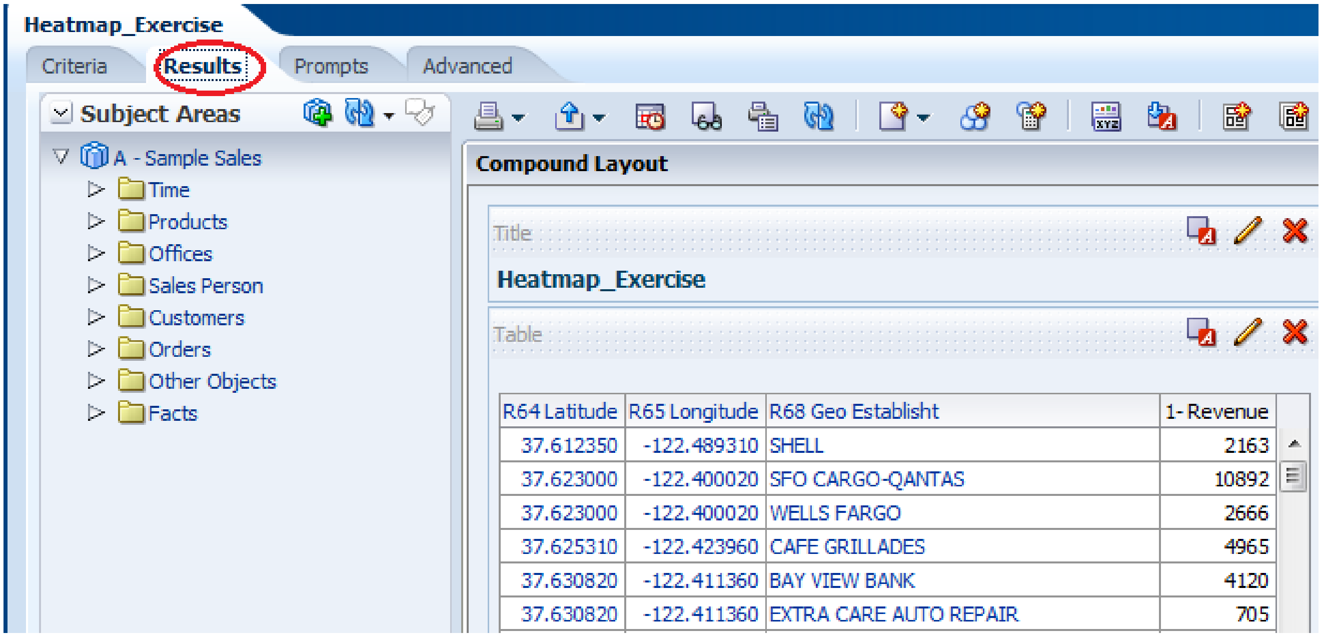
| | |
|---|---|
| STATE | South Dakota |
| #FLIGHTS | 5631 |
| DELAY | 11.27 |

## EXAMPLE 2: HEATMAP BASED ON FEATURE DENSITY

This example shows how to render a vector layer using the heatmap style. There are two ways to apply the style: one is based on the feature density, the other is based on an attribute value (such as Revenue). In this exercise we will build a heatmap based on feature density. The result should look like:



- Click on Catalog and open the Heatmap folder. Edit "`Heatmap_Exercise`" analysis from the folder: `/shared/Workshops/HTML5 Maps/2. Heatmap`
- Click the Results tab to show revenue by ship to locations and its location details



| R64 Latitude | R65 Longitude | R68 Geo Establisht | 1- Revenue |
|---|---|---|---|
| 37.612350 | -122.489310 | SHELL | 2163 |
| 37.623000 | -122.400020 | SFO CARGO-QANTAS | 10892 |
| 37.623000 | -122.400020 | WELLS FARGO | 2666 |
| 37.625310 | -122.423960 | CAFE GRILLADES | 4965 |
| 37.630820 | -122.411360 | BAY VIEW BANK | 4120 |
| 37.630820 | -122.411360 | EXTRA CARE AUTO REPAIR | 705 |

- Add a narrative view as follows:

Add Prefix as:

```
<script language="JavaScript" type="text/javascript">
var darray = [];
```

Add Narrative as:

```
 darray.push([@1,@2,"@3",@4]);
```

Add Postfix as:

```
renderHeatMap(darray);
</script>
```

Set rows to display = 1000

Tick “Contains HTML Markup” checkbox.

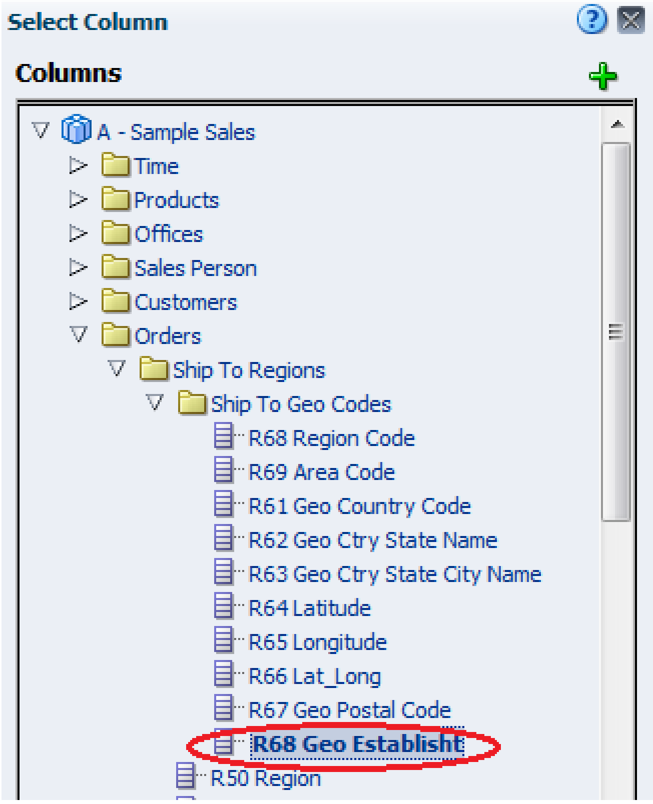
Select the “Done” button when completed.

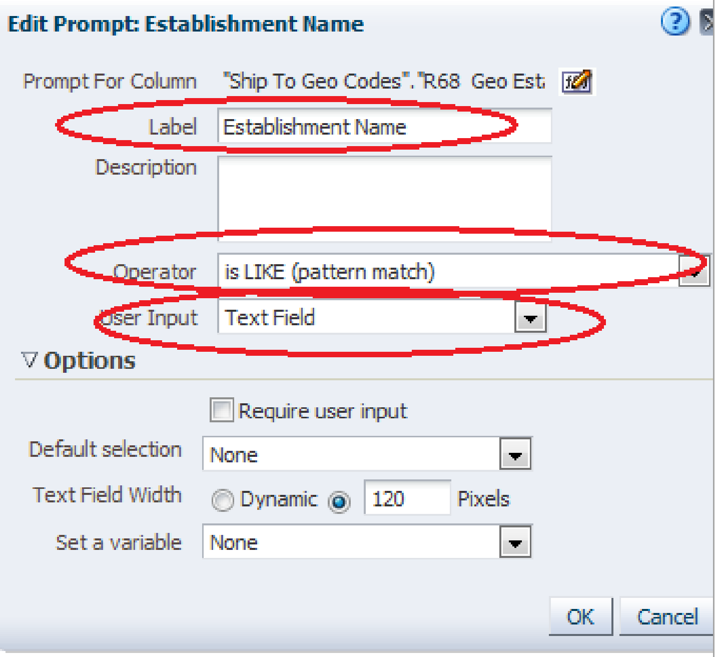
- Add a static text view. Insert following JavaScript code (in blue color) into static text field.

```
<b>Density based heatmap</b>
<DIV id=map style="width:700px;height:750px;padding:10px;border:1px solid
grey;"></DIV>

<script src='/mapviewer/jslib/v2/oraclemapsv2.js'></script>

<script>
function addPoints(darray,layer){
  /*lat,long,estb,metric*/
  for (i=0;i<darray.length; i++){
       var fid = i;
       var mpoint = new OM.geometry.Point(darray[i][1], darray[i][0],8307);
       var f_attr = {attributes :{"_LABEL_":darray[i][2],
"Sales":darray[i][3]+""}};
       var feature = new OM.Feature(fid,mpoint,f_attr);
       layer.addFeature(feature);
   }
}

function renderHeatMap(darray){
  var baseURL= "http://"+document.location.host+"/mapviewer";
  var map = new OM.Map(document.getElementById('map'), {mapviewerURL:
baseURL}) ;
  map.addLayer(new OM.layer.ElocationTileLayer("background")) ;

  var colors =
["#C6DBEF","#6BAED6","#008fff","#00abff","#00d5ff","#00ffff","#00ff7f","#0
0ff00","#7fff00","#ffff00","#ffd500","#ffab00","#ff7f00","#ff5600","#ff2b0
0", "#ff0000", "#A50F15"];
  var config = {
    spotlightRadius:25,
    lengthUnit:'pixel',
    colorStops: colors,
    opacity:0.65,
    sampleFactor: 4
  };
  var style = new OM.style.HeatMap(config);

  var heatLayer = new OM.layer.VectorLayer("heatlayer",{
      def:
          {
          type: OM.layer.VectorLayer.TYPE_LOCAL
      },
      renderingStyle :style
     //styleAttributes:["METRIC"]
  });
  addPoints(darray,heatLayer);
  map.addLayer(heatLayer);

  var marker = new OM.style.Marker({src: "/mapviewer/icons/BALL_sel.png",
width:7, height:7});
  var pLayer = new OM.layer.VectorLayer("pointlayer",{
      def:
      {
          type: OM.layer.VectorLayer.TYPE_LOCAL
      },
      renderingStyle : marker
  });
  addPoints(darray,pLayer);
  map.addLayer(pLayer);
  pLayer.setVisible(true);
  map.addMapDecoration(new OM.control.NavigationPanelBar({ style:1 }));
  var mpoint = new OM.geometry.Point(-122.43,37.716,8307);
  map.setMapCenter(mpoint);
  map.setMapZoomLevel(11) ;
  map.init();
}
</script>
```

- Tick “Contains HTML Markup” checkbox and select the “Done” button when completed.
- Remove Title and Table views from the compound layout
- Sequence the views so that Narrative view is the last view in the compound layout.
- Verify that the compound layout contains the static text and narrative views. Save the Analysis.

- Next, create a new dashboard prompt on "A – Sample Sales" subject area
- Change highlighted values and save the prompt




Edit Prompt: Establishment Name

Prompt For Column "Ship To Geo Codes"."R68 Geo Est:

Label Establishment Name

Description

Operator is LIKE (pattern match)

User Input Text Field

▽ Options

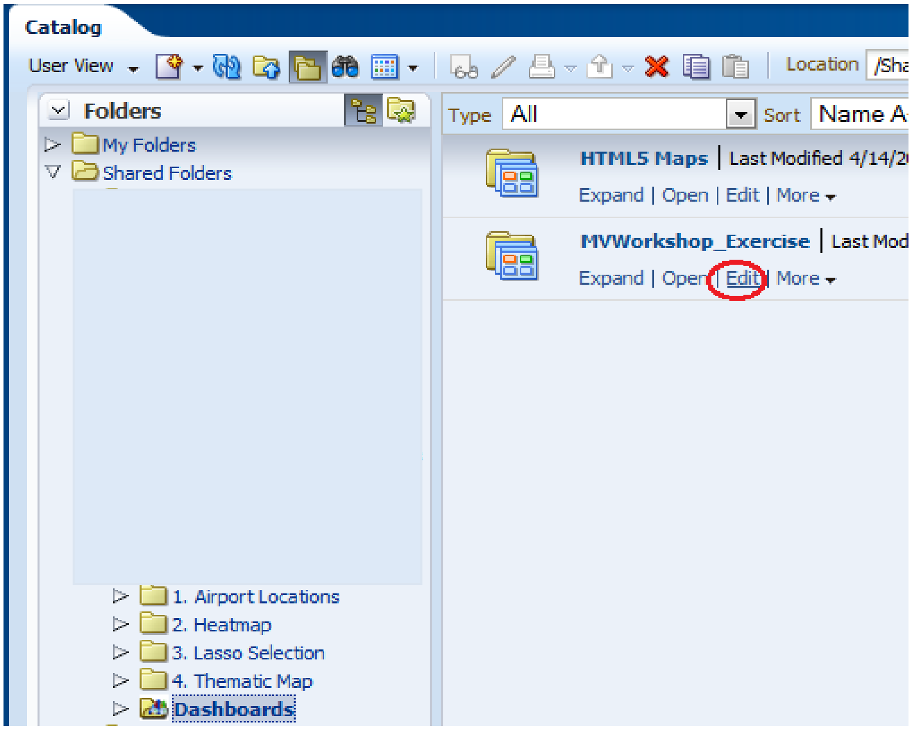
□ Require user input

Default selection None

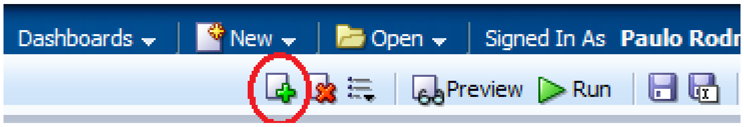
Text Field Width ○ Dynamic ● 120 Pixels

Set a variable None

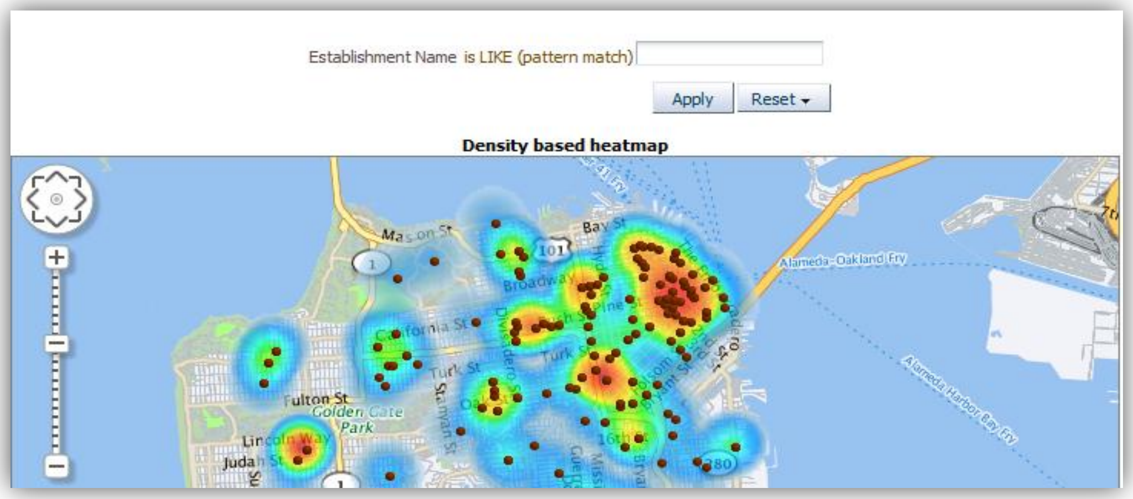
- Save the prompt as "`Prompt_Exercise`" under folder: `/shared/Workshops/HTML5 Maps/2. Heatmap`
- Edit the dashboard "`MVWorkshop_Exercise`" that we created earlier.



- Add a new dashboard page named “`Heatmap`” and drag drop the prompt and analysis that we just created.

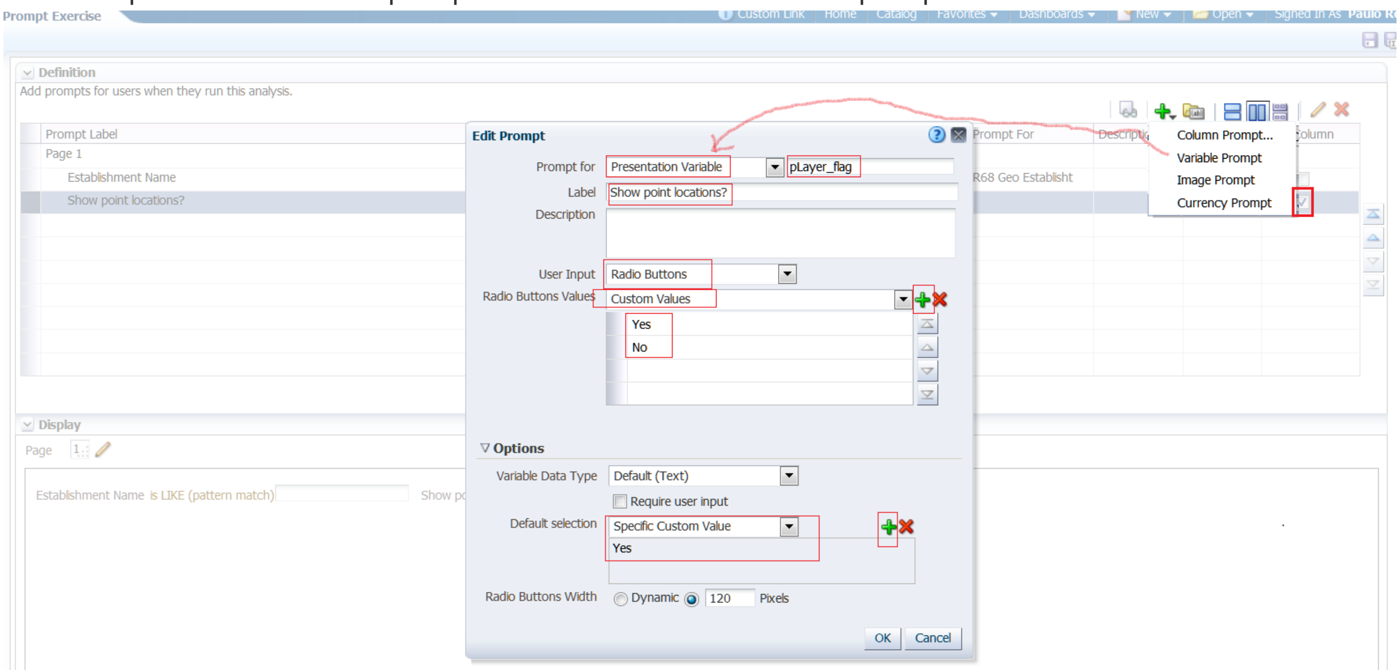


- Save and run the page. You should now be able to see the heatmap displayed on your dashboard



- We can control the point layer visibility from a dashboard prompt. To enable this, edit the “`Prompt_Exercise`” prompt from the “`2. Heatmap`” folder.

- Add a presentation column prompt as shown below and save the prompt.



Edit Prompt

Prompt for: Presentation Variable | pLayer_flag

Label: Show point locations?

Description:

User Input: Radio Buttons

Radio Buttons Values: Custom Values

Yes
No

Options

Variable Data Type: Default (Text)

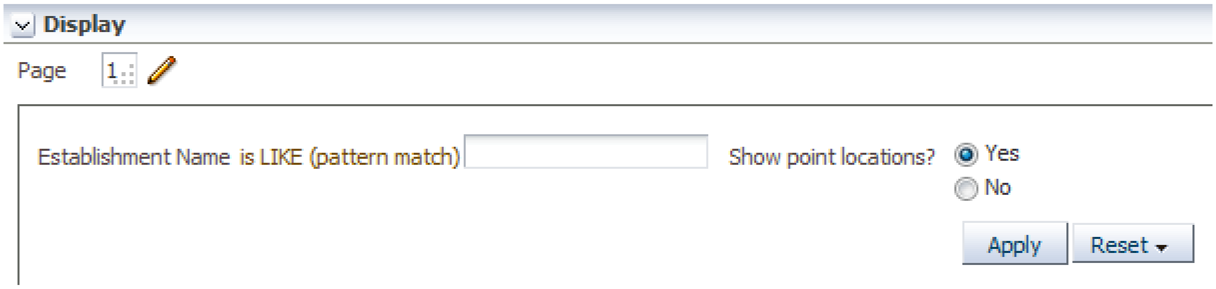
□ Require user input

Default selection: Specific Custom Value

Yes

Radio Buttons Width: ○ Dynamic ● 120 Pixels

Make sure the New Column box is checked for this prompt so that the two of them show up in one line as shown below.



Display

Page 1

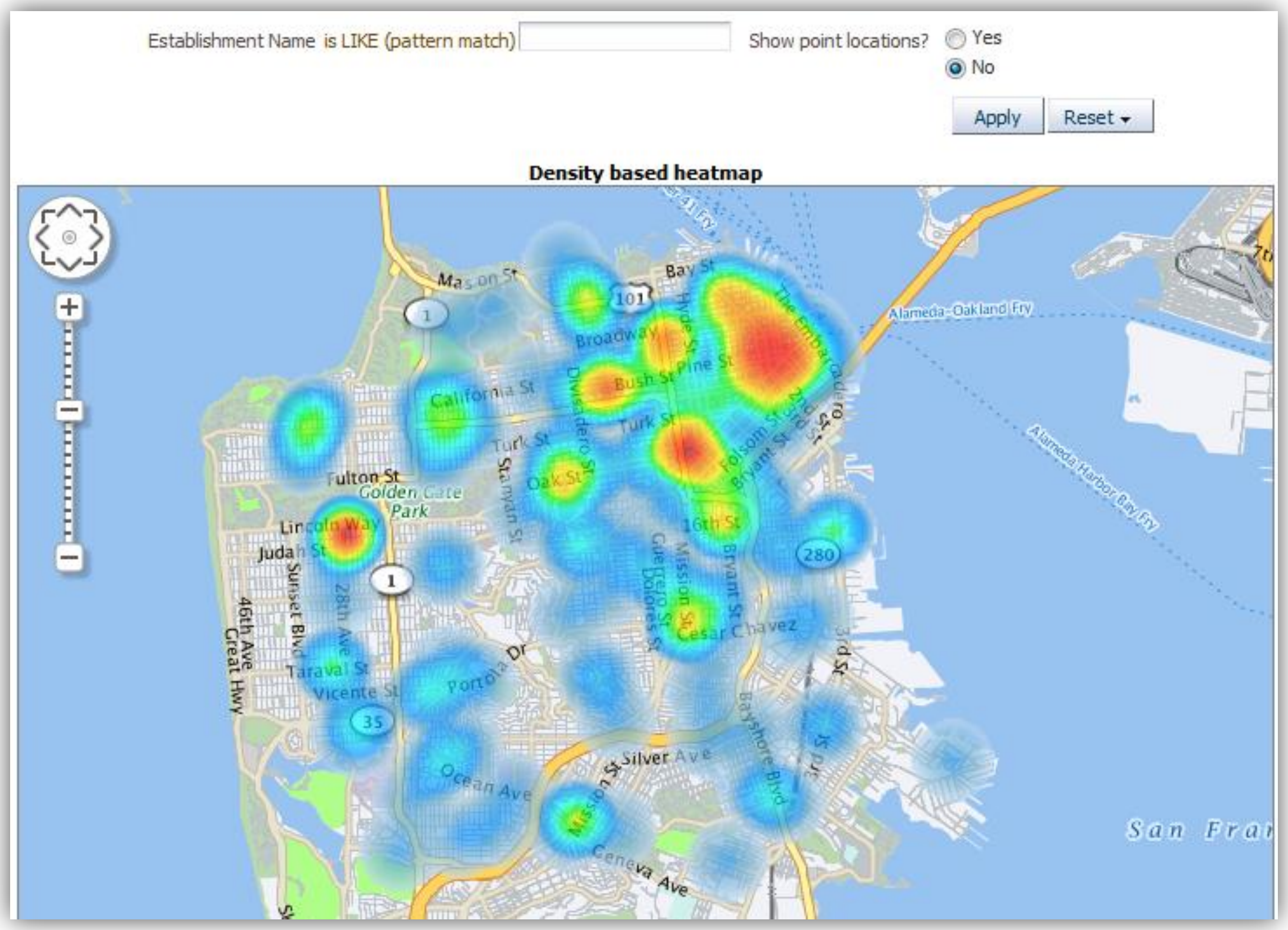
Establishment Name is LIKE (pattern match)

Show point locations? ● Yes ○ No

- Edit the "`Heatmap_Exercise`" analysis and modify the script within the static text view. Replace following line "`pLayer.setVisible(true);`" with below:

```
if ('@{pLayer_flag}'=='No')  pLayer.setVisible(false);
```

- Save the analysis and view the Heatmap dashboard page that we created earlier. It should look like:



Establishment Name is LIKE (pattern match)

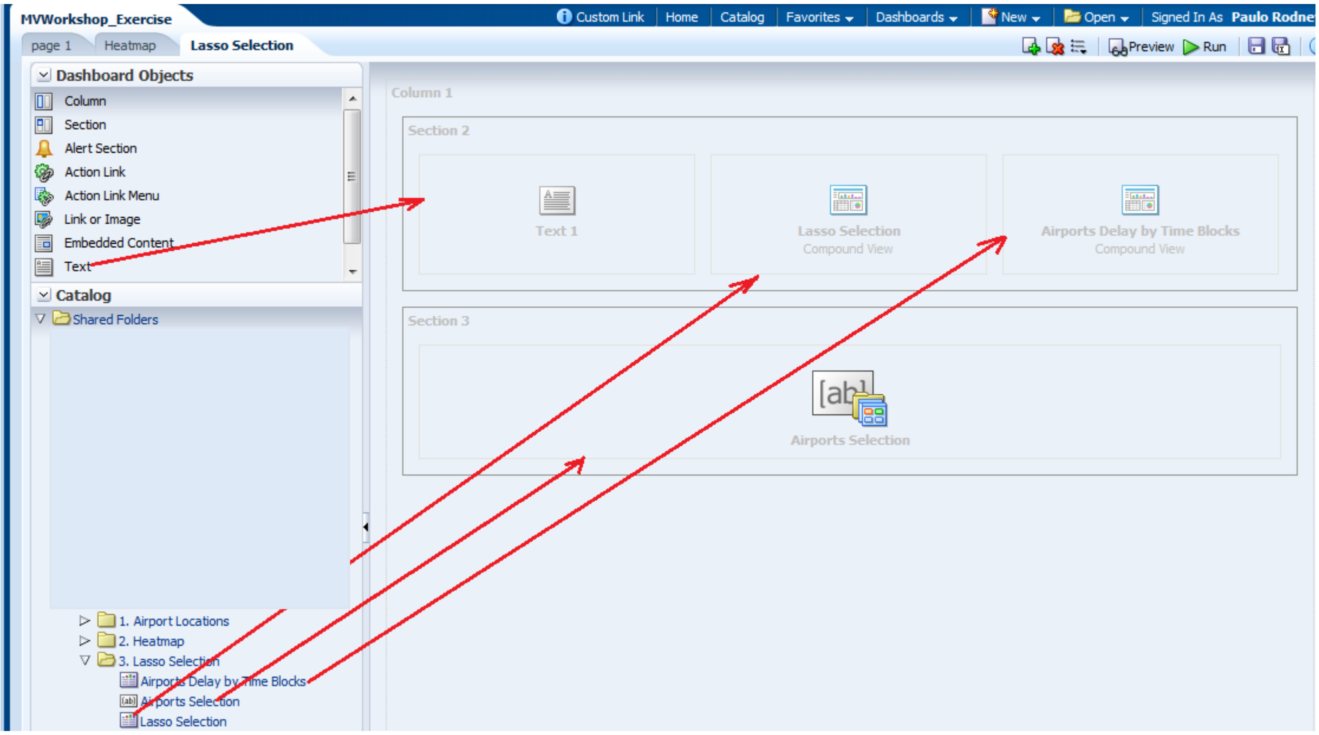
Show point locations? ○ Yes ● No

## EXAMPLE 3: LASSO SELECTION ON MAPS

This example showcases a draggable map toolbar with multiple control buttons. Each button is an Oracle Maps built-in tool (such as the Circle, Rectangle, Redline/Polygon, Distance and Marquee zoom tool).

- Create a new dashboard page under "`MVWorkshop_Exercise`" named "`Lasso Selection`"

- Add a dashboard Text object, and the "`Lasso Selection`" report, "`Airports Delay by Time Blocks`" report and "`Airports Selection`" prompt from the Lasso Selection folder.



- Edit the text object and insert the following code. Tick "Contains HTML Markup" checkbox.

```
<font size=2 color=grey>[b]Airports by Flight Volume[/b]</font><br/><br/>
<table><tr>
<td><DIV id='map' style="width:800px;height:600px;padding:10px;"></DIV></td>
<td valign='top'>&nbsp;&nbsp;<br/><b>Selected Airports:</b><br/>
<DIV id='selectedairports'
style="width:200px;height:600px;padding:10px;overflow-y:
scroll;"></DIV></td>
</tr></table>

<script src='/mapviewer/jslib/v2/oraclemapsv2.js'></script>
<script src='/analyticsRes/SampleApp/customjs/SAutilsv1.3.js'></script>

<script>
var map, airportsLayer, highlightLayer;
var currMapData = "";
var highlights=[];
function setupAirportStyles(layer){
    var circles = [];

    for(var i=0; i<10; i++)
    {
      circles[i] =  new OM.style.Marker({
     name:"circle_"+i,
        vectorDef : [{
            shape: { type:"circle", cx:5, cy:5, width:10, height:10 },
            style: {
              fill:"#AFB6BD",
              stroke:"#6784A3"
            }
          }],
        width: 7 + i*3,
        height:7 + i*3
      });
      highlights[i] =  new OM.style.Marker({
     name:"highlight_"+i,
        vectorDef : [{
            shape: { type:"circle", cx:5, cy:5, width:10, height:10 },
            style: {
              fill:"#FFFF44",
              stroke:"#6784A3"
            }
          }],
        width: 7 + i*3,
        height:7 + i*3
      });
    }
    var bucketStyle = new OM.style.BucketStyle(
    {
      styles: circles,
      numClasses:10,
      classification:'equal',  //creates equal-ranged buckets
      defaultStyle: circles[0]
    });
    layer.setRenderingStyle(bucketStyle, ["#FLIGHTS"]);
}

function clearHighlight(e)
{
  if (highlightLayer != null)
  {
    map.removeLayer(highlightLayer);
    highlightLayer = null;
 document.getElementById('selectedairports').innerHTML = "";
  }
```

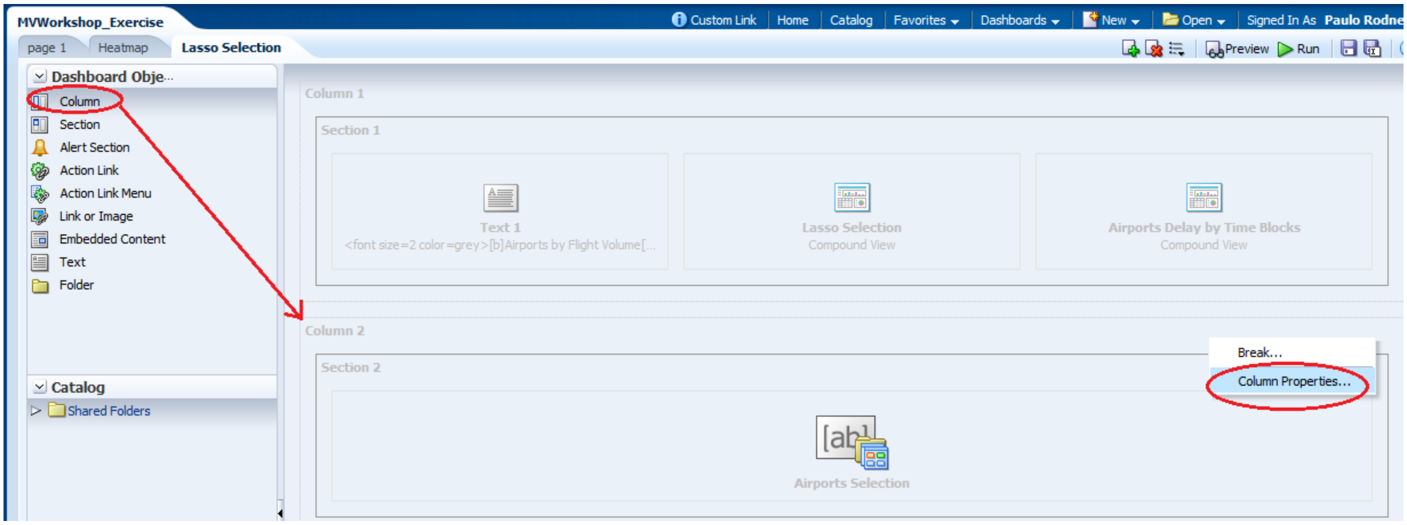
```
}

function addHighlight(e)
{
  clearHighlight();
  var tool = e.target;
  var geom = tool.getGeometry();
  //to select all the features that interact with a custom drawn geometry
  //(using the circle, rectangle or polygon tool), we simply apply the
  //AnyInteract filter on all the current layer.
  var filter1 = new OM.filter.AnyInteract(geom);
  var selectStyle = new OM.style.Color({  fill:"#ffff44", stroke:"#44ff22"
});
  //The result of applying the filter is returned as a new layer.
  highlightLayer = airportsLayer.applyFilter(filter1,false);
  var features=highlightLayer.getAllFeatures();
  var selStr = "", selStr1 = "";
  var hlayerStyle = airportsLayer.getRenderingStyle();
  hlayerStyle = hlayerStyle.clone();
  hlayerStyle.styles = highlights;
  highlightLayer.setRenderingStyle(hlayerStyle);
  for(i=0;i<features.length;i++){
    selStr += features[i].getAttributeValue("AIRPORT") + "<br/>";
    selStr1+="'"+features[i].id + "',";
  }
  document.getElementById('selectedairports').innerHTML = selStr;

  //following is used to set dashboard variable prompt
  if(selStr1.length>2){
   selStr1 = selStr1.substring(0,selStr1.length-1);
   SAutils.setPSVarPrompt('airport_selection',selStr1);
   }
  map.addLayer(highlightLayer);
}

function renderMap(repData){
  if (currMapData == repData)     return;

  currMapData = repData;
  var darray=eval('['+repData+']');
  var baseURL  = "http://"+document.location.host+"/mapviewer";
  var center = new OM.geometry.Point(-100,36,8307);
  map = new OM.Map(document.getElementById('map'),{mapviewerURL: baseURL}) ;
  var tilelayer = new OM.layer.ElocationTileLayer("oracle.maps");
  map.addLayer(tilelayer) ;
  airportsLayer = new OM.layer.VectorLayer("airports", { def:{
 type:OM.layer.VectorLayer.TYPE_LOCAL } });

  /*"Code","State","Airport","Lat","Long","# Flights","Delay Mins"*/
  for (i=0;i<darray.length; i++){
          var fid = darray[i][0];
          var mpoint = new OM.geometry.Point(darray[i][4], darray[i][3],8307);
          var f_attr = {attributes
      :{"_LABEL_":darray[i][0],"AIRPORT":darray[i][2], "STATE":darray[i][1],
      "#FLIGHTS":darray[i][5]+"", "DELAY":darray[i][6]+""}};
          var feature = new OM.Feature(fid,mpoint,f_attr);
          airportsLayer.addFeature(feature);
        }

        airportsLayer.setBringToTopOnMouseOver(true);
        setupAirportStyles(airportsLayer);

        map.addLayer(airportsLayer) ;
        map.setMapCenter( center );
        map.setMapZoomLevel(3) ;

        var navigationPanelBar = new OM.control.NavigationPanelBar( {style:2 } );
        map.addMapDecoration(navigationPanelBar);

        toolbar = new OM.control.ToolBar("toolbar1",{
      builtInButtons:[OM.control.ToolBar.BUILTIN_ALL] });
        toolbar.setPosition(0.4,0.05);
        map.addToolBar(toolbar);

        var circleTool =
      toolbar.getBuiltInTool(OM.control.ToolBar.BUILTIN_CIRCLE);
        circleTool.on(OM.event.ToolEvent.TOOL_END, addHighlight);
        circleTool.on(OM.event.ToolEvent.TOOL_CLEAR, clearHighlight);
        var rectTool =
      toolbar.getBuiltInTool(OM.control.ToolBar.BUILTIN_RECTANGLE);
        rectTool.on(OM.event.ToolEvent.TOOL_END, addHighlight);
        rectTool.on(OM.event.ToolEvent.TOOL_CLEAR, clearHighlight)
        var polyTool = toolbar.getBuiltInTool(OM.control.ToolBar.BUILTIN_REDLINE);
        polyTool.on(OM.event.ToolEvent.TOOL_END, addHighlight);
        polyTool.on(OM.event.ToolEvent.TOOL_CLEAR, clearHighlight);
        polyTool.on(OM.event.ToolEvent.REDLINE_EDITED, addHighlight);
        map.init() ;
      }
</script>
```

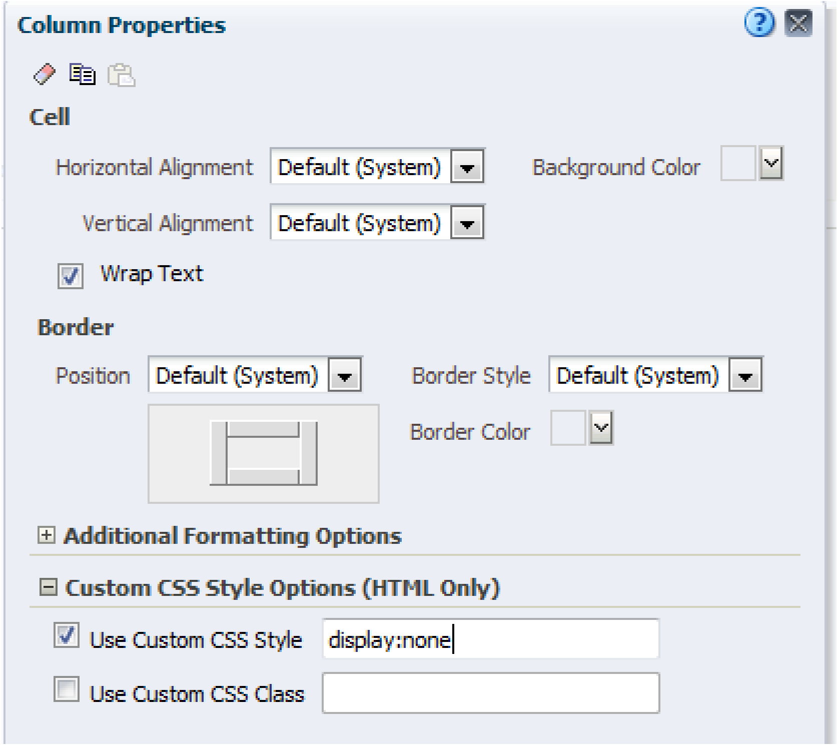
- Save and Run the page.

- Page should render with a draggable map toolbar with multiple control buttons is displayed (highlighted in red above). Each button controls is an Oracle Maps built-in tool (such as the Circle, Rectangle, Redline/Polygon, Distance and Marquee zoom tool). Airport info including its name, location, number of flights and delay details are sourced from BI analysis.

    - Lasso selection of airports on the map is triggering a custom function that highlights the selection and also displaying the names of the selected airports on the right. It also invokes `SAutils.setPSVarPrompt()` function to set a Presentation Server variable prompt to set the selected airport codes. This presentation variable is used in `Airports Delay by Time Blocks` report to filter the airports.

- Next, we can hide Airports Selection prompt as it is only meant to be programmatically set from the map.
  - Edit the dashboard page and add a new column to the bottom of the page.
  - Drag and drop the prompt section into the new column as shown below and edit column properties
  - Set “`display:none`” as custom CSS style for the column





**Column Properties**

**Cell**

Horizontal Alignment: Default (System)
Background Color
Vertical Alignment: Default (System)
☑ Wrap Text

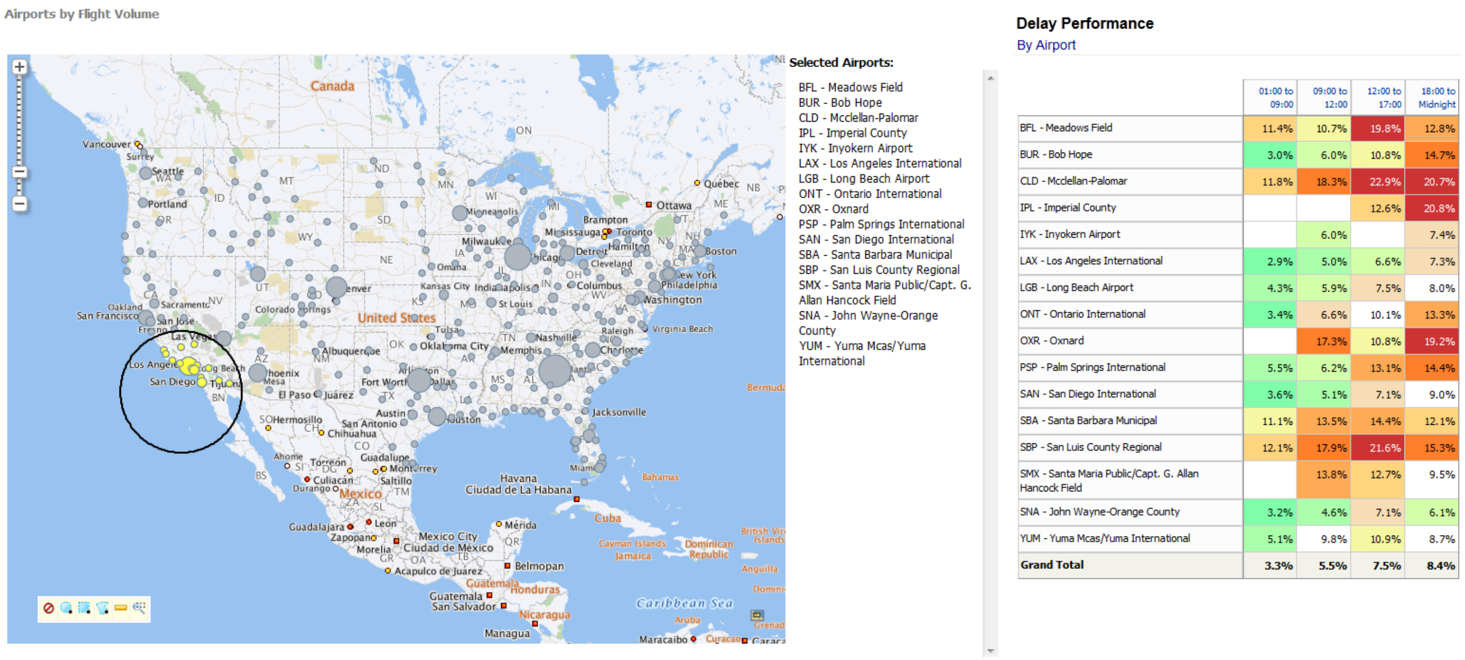
**Border**

Position: Default (System)
Border Style: Default (System)
Border Color

⊞ Additional Formatting Options

⊟ Custom CSS Style Options (HTML Only)

☑ Use Custom CSS Style: display:none
☐ Use Custom CSS Class:



Airports by Flight Volume

Selected Airports:

- BFL - Meadows Field
- BUR - Bob Hope
- CLD - Mcclellan-Palomar
- IPL - Imperial County
- IYK - Inyokern Airport
- LAX - Los Angeles International
- LGB - Long Beach Airport
- ONT - Ontario International
- OXR - Oxnard
- PSP - Palm Springs International
- SAN - San Diego International
- SBA - Santa Barbara Municipal
- SBP - San Luis County Regional
- SMX - Santa Maria Public/Capt. G. Allan Hancock Field
- SNA - John Wayne-Orange County
- YUM - Yuma Mcas/Yuma International

Delay Performance

By Airport

| | 01:00 to 09:00 | 09:00 to 12:00 | 12:00 to 17:00 | 18:00 to Midnight |
|---|---|---|---|---|
| BFL - Meadows Field | 11.4% | 10.7% | 19.8% | 12.8% |
| BUR - Bob Hope | 3.0% | 6.0% | 10.8% | 14.7% |
| CLD - Mcclellan-Palomar | 11.8% | 18.3% | 22.9% | 20.7% |
| IPL - Imperial County | | | 12.6% | 20.8% |
| IYK - Inyokern Airport | | 6.0% | | 7.4% |
| LAX - Los Angeles International | 2.9% | 5.0% | 6.6% | 7.3% |
| LGB - Long Beach Airport | 4.3% | 5.9% | 7.5% | 8.0% |
| ONT - Ontario International | 3.4% | 6.6% | 10.1% | 13.3% |
| OXR - Oxnard | | 17.3% | 10.8% | 19.2% |
| PSP - Palm Springs International | 5.5% | 6.2% | 13.1% | 14.4% |
| SAN - San Diego International | 3.6% | 5.1% | 7.1% | 9.0% |
| SBA - Santa Barbara Municipal | 11.1% | 13.5% | 14.4% | 12.1% |
| SBP - San Luis County Regional | 12.1% | 17.9% | 21.6% | 15.3% |
| SMX - Santa Maria Public/Capt. G. Allan Hancock Field | | 13.8% | 12.7% | 9.5% |
| SNA - John Wayne-Orange County | 3.2% | 4.6% | 7.1% | 6.1% |
| YUM - Yuma Mcas/Yuma International | 5.1% | 9.8% | 10.9% | 8.7% |
| **Grand Total** | **3.3%** | **5.5%** | **7.5%** | **8.4%** |

- Resulting page should look as shown above.

**NOTE:**

1. This example references `SAutilsv1.3.js`, which is a custom built JavaScript utility code that is freely distributed for demonstrative purposes. It is not included and not supported as part of any Oracle Business Intelligence product.

# RESOURCES

## Oracle MapViewer

- Oracle Fusion Middleware MapViewer - Manuals, Guides & More
  - Oracle Maps JavaScript API
- Software Downloads for Oracle Fusion Middleware MapViewer
- Mapviewer HTML5 API Documentation

## Oracle Business Intelligence

- Oracle Business Intelligence Enterprise Edition Documentation
- OBIEE Samples Application OTN Page
- Oracle BI Tech Demos YouTube Channel

ORACLE

Using MapViewer Javascript APIs with OBIEE
April 2014
Authors:Joseph Kuttikat, Jayant Sharma
Contributing Authors: LJ Qian, David Lapp

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

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.