P6 Analytics: Weather Feed in OBIEE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.
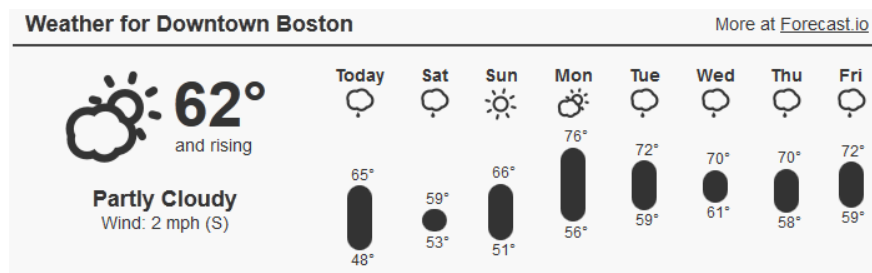
Overview

Oracle Business Intelligence Enterprise Edition provides the ability to embed HTML within an analysis, either as a column in a table or as a static text object. Using this capability along with OBIEE's Master-Detail linking, we can embed an HTML weather feed within an analysis, and make it dynamic based on the location selected on a spatial map.

There are numerous sources for location based weather data, ranging from embeddable HTML widgets to full blown weather APIs that return weather data in multiple formats, including XML and JSON. This white paper will cover the embedded widget approach, as there are several free options available, and the integration with OBIEE is quite simple. The example report uses the embeddable widget provided by Forecast.io (http://blog.forecast.io/forecast-embeds/). Some additional sources will be listed at the end of the white paper.

Forecast.io Forecast Embed

Code:

```
<iframe id="forecast_embed" type="text/html" frameborder="0" height="245"
width="100%" src="http://forecast.io/embed/#lat=42.3583&lon=-
71.0603&name=Downtown Boston&color=#00aaff&font=Georgia&units=uk"> </iframe>
```

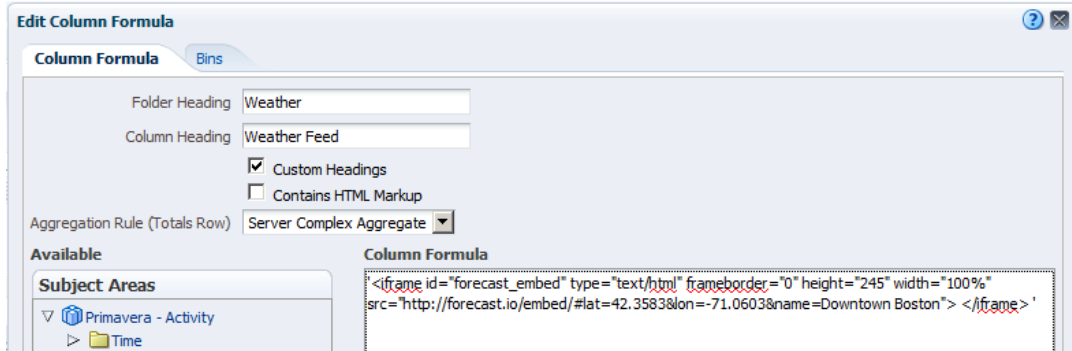*Note: Red parameters are required, blue are optional.*

Creating the Analysis

We'll start with a simple table view that will display the weather feed for a particular location, along with a list of projects in that location.

1. Create a new analysis from the **Primavera – Activity** subject area.
2. Choose the following columns:
   - **Project Id**
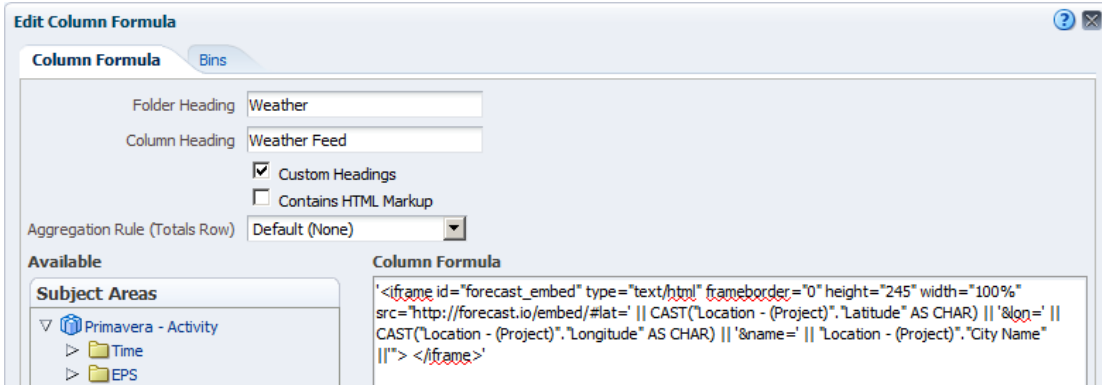   - **Project Name**
   - **City Name**
   - **Location Name**
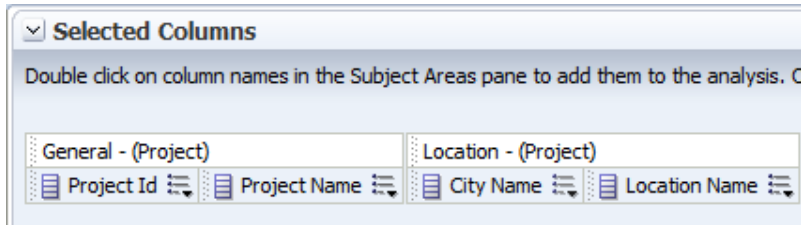
3. Edit the column formula for **Location Name** and select the **Custom Headings** option. Change the folder heading to **Weather,** the column heading to **Weather Feed**, and then paste in the HTML code for the weather feed, omitting the optional parameters.

Formula:

```
<iframe id="forecast_embed" type="text/html" frameborder="0" height="245"
width="100%" src="http://forecast.io/embed/#lat=42.3583&lon=-
71.0603&name=Downtown Boston"> </iframe>
```

4. Edit the column properties, select **Override Default Data Format** and pick **HTML** from the **Treat Text As** drop-down.
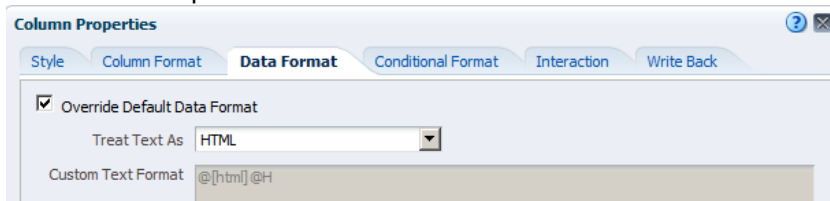
If you were to click on the **Results** tab at this point you would see that the forecast always shows for the same location, which is Boston, Massachusetts in the sample code. We still need to substitute the subject area columns for the required parameters in the HTML.

5. Build a concatenated string using the sample HTML code and the subject area columns for **Latitude**, **Longitude**, and **City Name**. Note that the **Latitude** and **Longitude** columns need to be cast as CHAR type in order for the formula to be valid.

Formula:

```
'<iframe id="forecast_embed" type="text/html" frameborder="0" height="245"
width="100%" src="http://forecast.io/embed/#lat=' || CAST("Location -
(Project)"."Latitude" AS CHAR) || '&lon=' || CAST("Location -
(Project)"."Longitude" AS CHAR) || '&name=' || "Location - (Project)"."City
Name" ||'"> </iframe>'
```

6. Edit the column properties and set the width to 540, this will be wide enough to accommodate the entire weather widget. Now when you click on the **Results** tab you'll see a listing of projects with the weather displayed for each project's location in the **Weather Feed** column. Save the analysis.

7. Now add the following additional columns:
   - **Project Start**
   - **Project Finish**
   - **Project Owner**
   - **Project Risk Exposure**
   - **Location Name**.

8. Edit the **Location Name** column formula. Select the **Custom Headings** option, change the **Column Heading** to **Location**, and paste in the following formula. This will build a concatenated string in the format 'Country Name – City Name'.
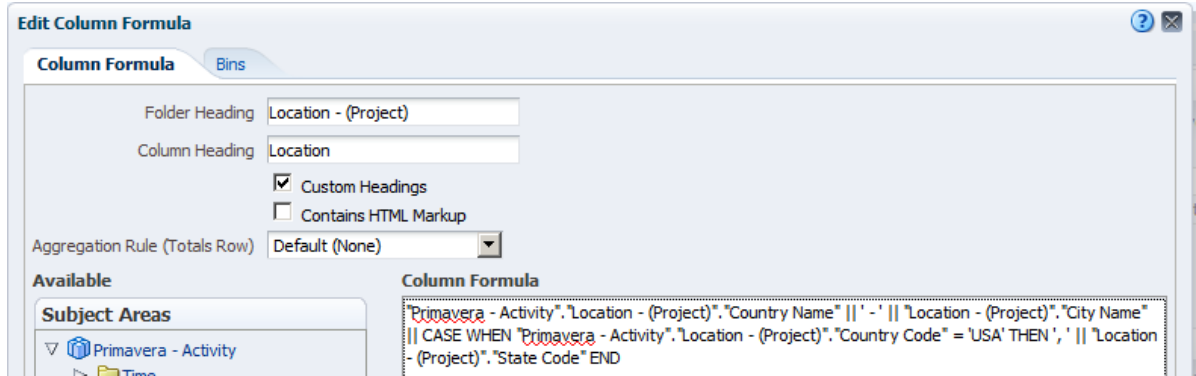
Formula:
```
"Primavera - Activity"."Location - (Project)"."Country Name" || ' - ' ||
"Location - (Project)"."City Name" || CASE WHEN "Primavera -
Activity"."Location - (Project)"."Country Code" = 'USA' THEN ', ' || "Location
- (Project)"."State Code" END
```

9. Edit the table and configure the table columns as shown below. The **Location** field we created will be used as the table prompt, and the **Weather Feed** column will be displayed as a section. Click the **Done** button and save the analysis.
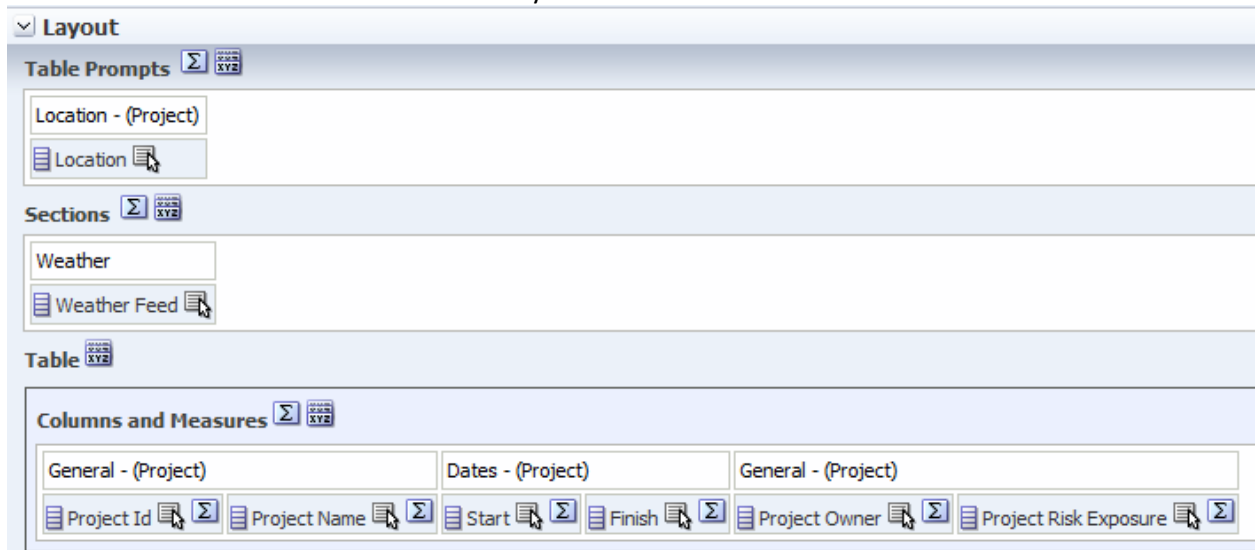
Now from the **Results** tab you'll see the weather for the location selected in the drop-down, with the table below listing out any projects in that location.

Incorporating a Map View

Now that we have a table displaying the weather feed, the next step is to incorporate the spatial map.

> **Note:**
> Mapviewer must be configured before moving on to these steps. See: http://docs.oracle.com /cd/E51723_01/English/Install_and_Config/Admin_PDF_Library/p6_analytics_post_install_ administrators_guide.pdf

For a spatial analysis, OBIEE requires you to have at least one measure column selected. However, in this example we're using **Project Risk Level**, which is a dimensional value, so we need to get creative about how to handle that. Measures contain the data that you want to analyze, typically numerical values such as cost data, while dimensions are columns that are used to identify and categorize data. Measure columns can be found in the **Fact** folder within the P6 Analytics subject areas. For more details about facts and dimensions, refer to http://docs.oracle.com/cd/B28359_01/server.111/b28313/dimen.htm
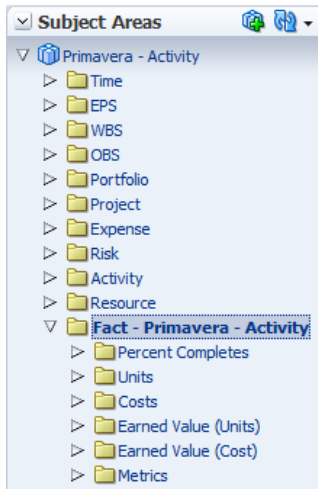
1. Select **Project Risk Level** as a column, and then edit the column formula. Select the **Custom Headings** option, and then create a case statement that will evaluate to a number based on the value in the **Project Risk Level** column. In this way we can get OBIEE to treat the dimension column as a measure that can be used in the spatial analysis.
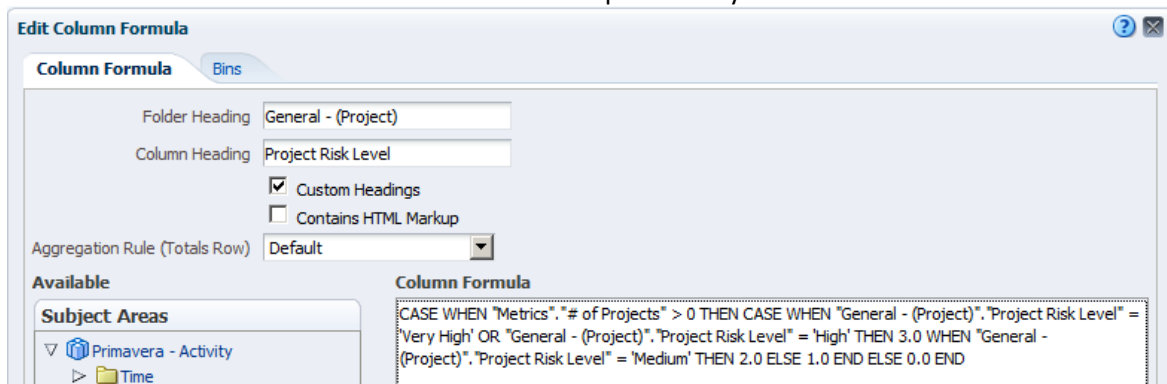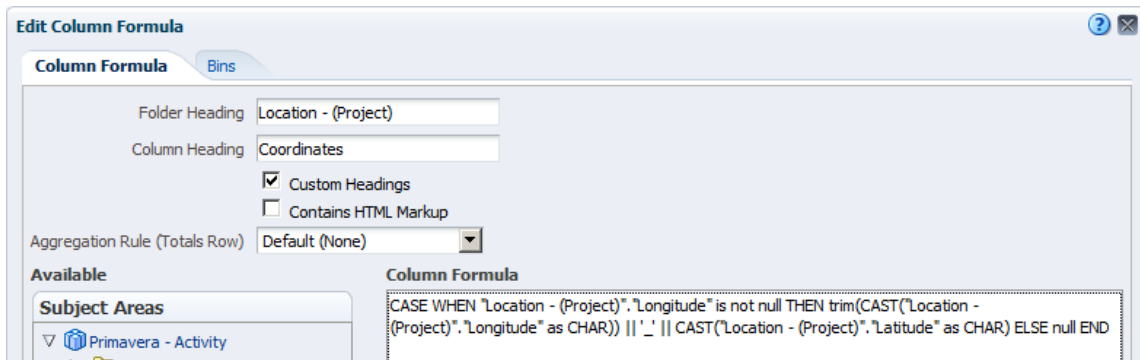

Formula:

```
CASE WHEN "Metrics"."# of Projects" > 0 THEN CASE WHEN "General -
(Project)"."Project Risk Level" = 'Very High' OR "General - (Project)"."Project
Risk Level" = 'High' THEN 3.0 WHEN "General - (Project)"."Project Risk Level" =
'Medium' THEN 2.0 ELSE 1.0 END ELSE 0.0 END
```

So now we have our measure column, but we still need to add columns that Mapviewer can use to plot points based on location. The keys for the different map layers are defined during the Mapviewer configuration. OBIEE also allows you to define a custom layer key based on the longitude and latitude coordinates, which is what we'll use in this example. We'll also concatenate those two values, separated by a '_' into one column to make the Master-Detail linking a little more straightforward.

2. Add the **Longitude** column, and then edit the column formula. Select the **Custom Headings** option and enter **Coordinates** as the column heading. Then paste in this formula which concatenates the **Longitude** and **Latitude**, separated by '_'.

Formula:

```
CASE WHEN "Location - (Project)"."Longitude" is not null THEN
trim(CAST("Location - (Project)"."Longitude" as CHAR)) || '_' || CAST("Location
- (Project)"."Latitude" as CHAR) ELSE null END
```

We now have a measure column and a column with location coordinates we can use to create a custom point layer. At this point we can create the spatial analysis that will be used to select the location for the weather feed.

3. Create a new map view in the analysis.
4. Select **Country State City Map** from the map selection drop-down, and then click on **New Map Format** and select **Image**. From the list of map layers, choose the **Custom Point Layer** option.
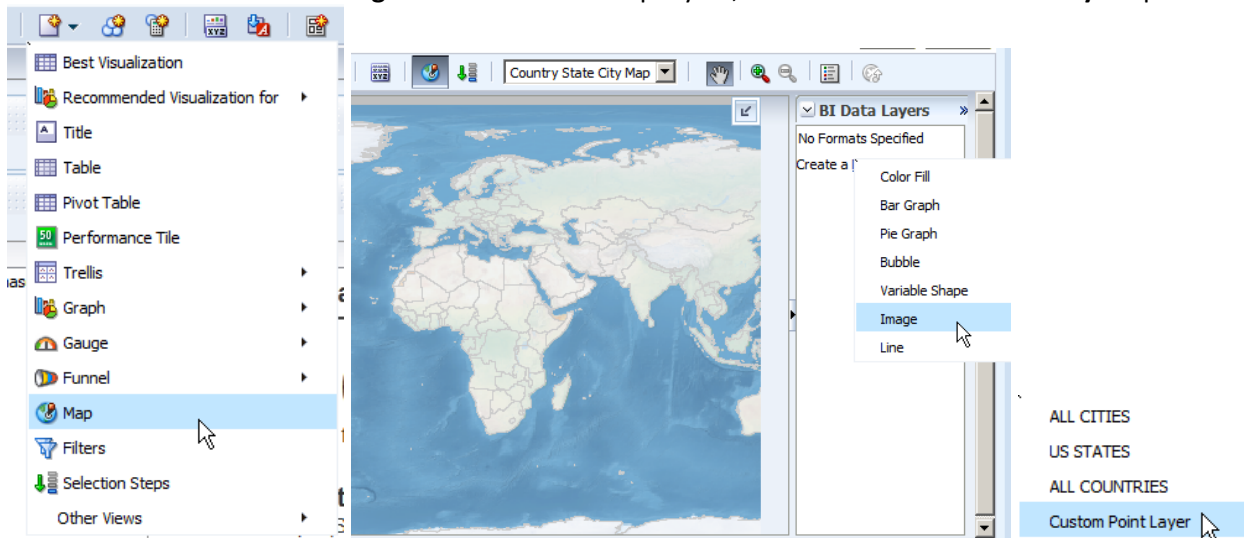
5. Uncheck the **Name Automatically** option, and enter **Project Risk Level** in the **Name** field. Then select the **Single Column** option, type '_' in the **Delimiter** field, and select the column you created for the coordinates from the **Location** drop-down. The full column formula will be displayed here, not the column name. Next, from the **ToolTips** drop-down choose **Project Id** and **Project Name**, these will be displayed in a pop-up along with the **Project Risk Level** and **Coordinates** when you select a location on the map. Then, change the **Bin Type** from **Percentile Binning** to **Value Binning**, and leave the **Bins** field set to 3. Enter the range and labels for the bins as shown below and select images for each value.
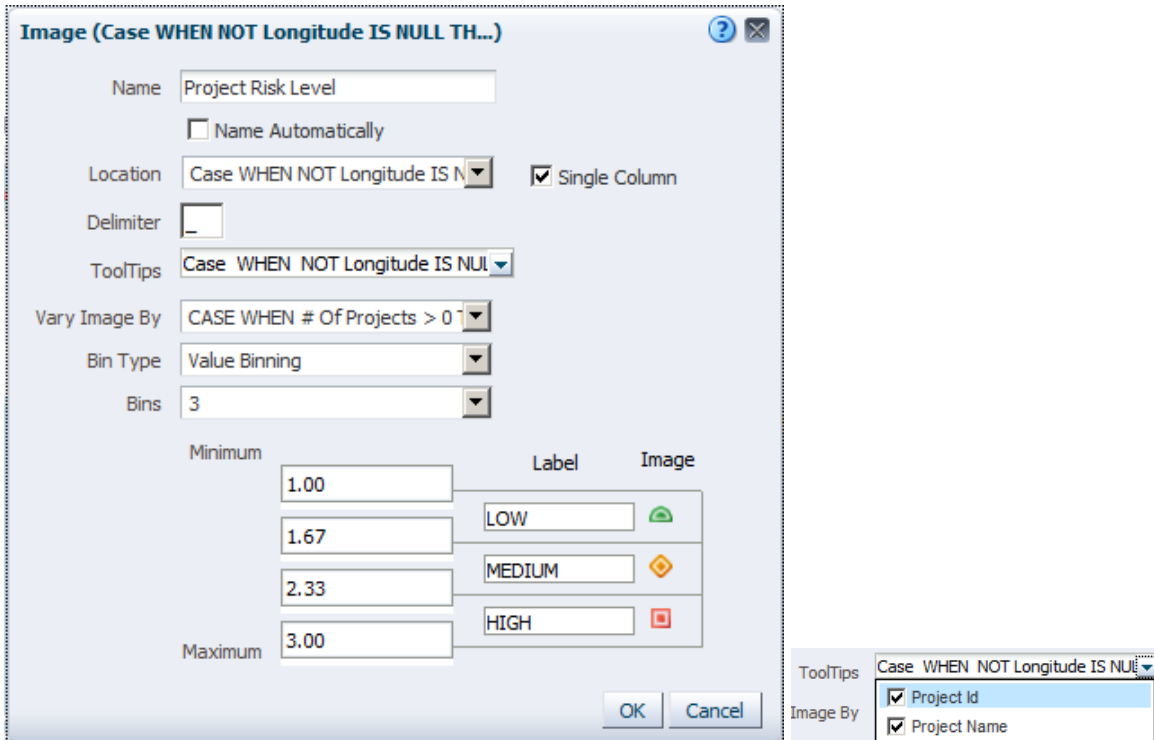
6. Click OK and save the analysis. At this point your map will look something like this.
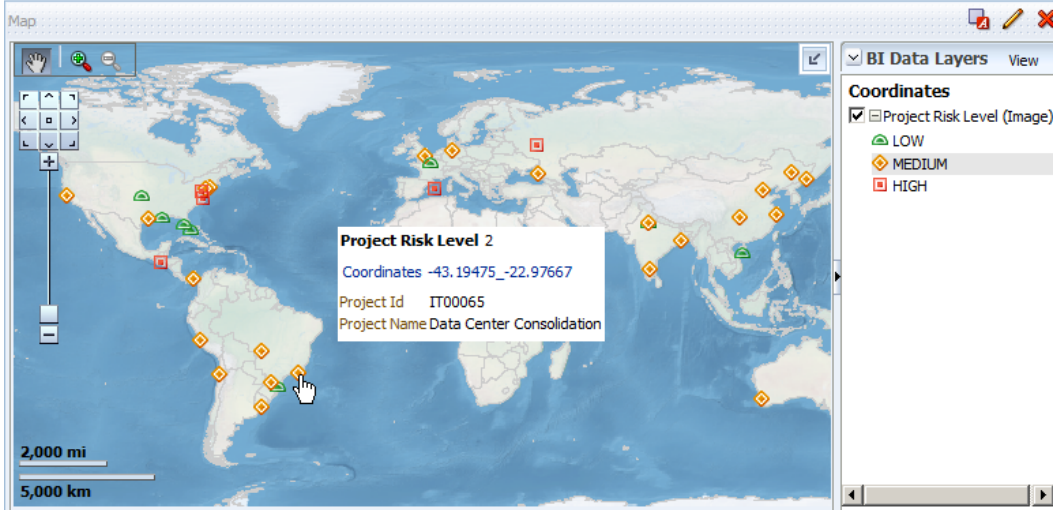

All that's left is to set up the Master-Detail linking between the map and the table that contains the weather feed column, and some final adjustments to the layout of the analysis.

7. From the **Criteria** tab, edit the column properties for the **Coordinates** column, go to the **Interaction** tab, and set the **Primary Interaction** to **Send Master-Detail Events**. Specify **C1** as the channel. Click **OK** to return the **Criteria** tab.
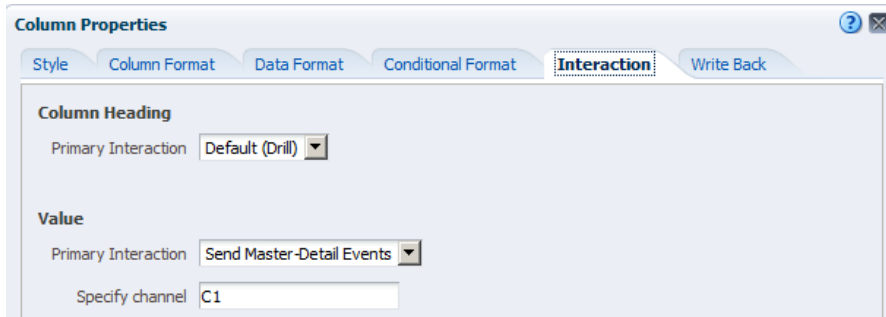
8. Go to the **Results** tab, and edit the table we created earlier that contains the weather feed. Add the **Coordinates** column to the **Table Prompts** section.
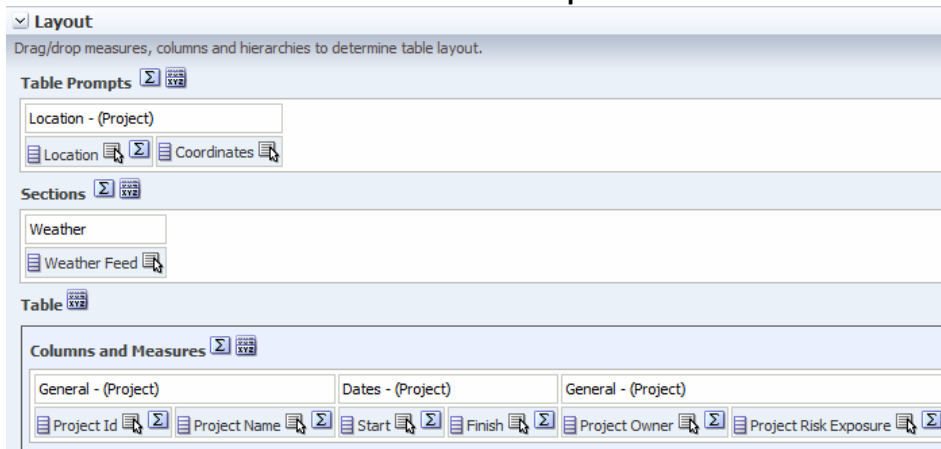
9. Edit the table properties, select the **Fixed headers with scrolling content** option, set the maximum width to 1100 Pixels, select the **Listen to Master-Detail Events** option, enter **C1** the **Event Channels** field, and then click **OK**. Click the **Done** button and save the analysis.
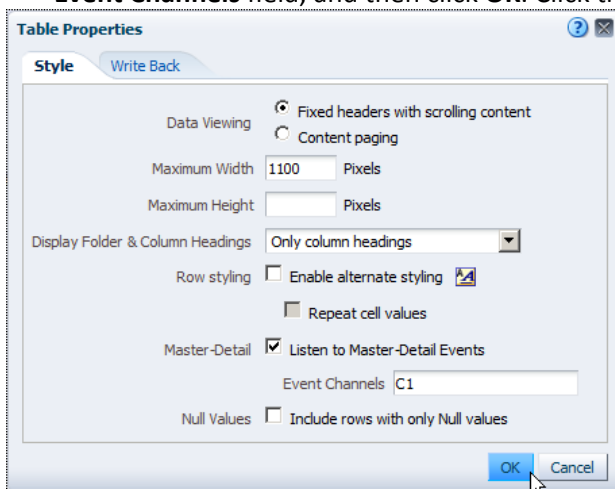
10. From the **Results** tab, arrange the two views so that the map is to the left of the table. To test the Master-Detail linking, click on one of the map locations, and then click on the **Coordinates** for that location in the pop-up. Clicking the coordinates triggers the table to update and display the weather feed and projects for that location. Save the analysis.
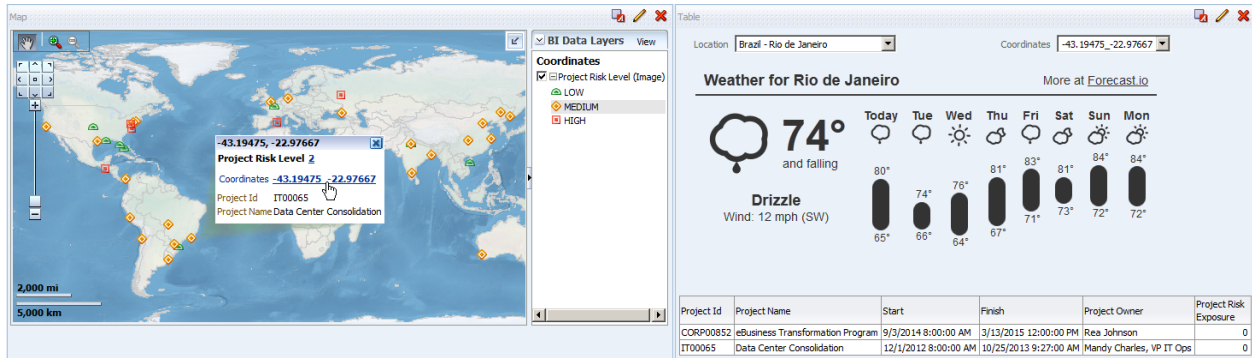
Additional Sources

Here are some additional sources that can also be used. Depending on the type of location data you have, these may provide a better fit than the Forecast.io widget that was used in the example report.

Weather Underground

http://www.wunderground.com/stickers/classic.html

The HTML format takes City Name, State Code, and Country Code (two characters) as parameters for U.S. cities. For international cities it takes City Name and Country Name as parameters.

Weatherbug

http://weather.weatherbug.com/desktop-weather/web-widgets.html

This source only supports U.S. cities. The HTML format takes City Name, State Code, and Postal Code as parameters for U.S. cities. For international cities it uses an internal ID which does not map to any of the location columns in the P6 Reporting Database.

ORACLE®

Hardware and Software, Engineered to Work Together