Building Map-based Dashboards

Participant Handbook

Abhinav Agarwal, Jayant Sharma

April, 2012
Purpose
The following workshop material is designed to introduce the new Map View feature of Oracle Business Intelligence Suite Enterprise Edition Plus (OBIEE 11g) through a set of simple hands-on exercises.

Topics
This hands-on portion workshop will cover the following topics.

- Creating a simple map view
- Adding charts to a map view
- Master-detail linking via map views
- Action Links and Map Views

Participants who are familiar with these topics, and who prefer to explore additional topics, can jump to the sections towards the end which discuss

- Using spatial functions (queries) in an analysis
  - Prompt for postcode, distance, and business category. Use sdo_within_distance to find and display census blocks within specified distance of the businesses in that postcode. This example uses BI presentation variables, session variables, and opaque views.
  - Find stores of a specified category (e.g. pharmacies, convenience store) within specified distance (e.g. 500 meters) of given address. Use Spatial’s geocoding function (sdo_gcdr.geocode_as_geometry) and sdo_within_distance to find and list the stores. This example uses a direct database request with presentation variables.
  - Compute the distance between a selected sales outlet/office and customers associated with that office. Display both locations and the distance between them. This example uses the OBIEE EVALUATE function and Spatial’s sdo_distance function.
- A start to finish example of importing a table with business information, importing the related map data, setting up maps and metadata, and creating map views.
The presentation and demo portion will cover the following topics.

- Why Map Views
- Map metadata management

If time permits the demo portion may also include using spatial queries in analyses and the fully worked example mentioned above.

**Workshop Environment**

Participants will need a laptop or desktop machine with internet connectivity and a web browser (either Firefox 3+ or Internet Explorer 8+).

Each participant will be working on their own client machine connecting to OBIEE server instance deployed on cloud servers. Details of the URL, username, and password will be provided at the start of the workshop.

Note: You can also use this with the OBIEE 11.1.1.5 Virtual Box image with SampleApp V107 currently (April 2012) available on OTN.

Use an existing user such as ‘abell’ or ‘prodney’ with SampleApp.
You will need MapBuilder11g for the mapview_lab portion. That is available from the MapViewer page on OTN. Use version 11.1.1.5 or later.

**Creating a Map View**

**Objective**

In this section participants will create interactive maps using the new OBIEE 11g "Map Views" feature. They will

- Create a simple Analysis that includes a geographic dimension
- Render it as a Map View
- Preview it and observe map drill behavior

**Overview**

Map views are a new view type in Oracle Business Intelligence Enterprise Edition (OBIEE) 11g for displaying analysis results. This provides a valuable, rich interactive visualization capability since most BI data already contains a geographic dimension such
as store addresses, sales districts or regions. OBIEE 11g allows BI administrators to pre-associates the business model with mapping data so that analysis results are automatically map-ready. Any analysis that includes a column associated with a configured map feature is ready to be rendered in a Map View without any customization or coding whatsoever. In this lab we will create simple map views that demonstrate this capability.

**Hands On:**

1. First, sign in to your hosted instance using the specified workshop username and password.

![Sign In](image)

2. Create a New Analysis using the Sample Sales subject area containing Customers → Cust Geo Codes → Geo Country Code, Base Facts → Revenue, and Base Facts→ Billed Quantity as items. Note: The following and other images shown will contain relevant snippets from multiple screenshots of the steps involved in any portion of the hands-on exercise.
3. Click the Results tab to see a tabular view of the report. Next remove the Table View (click on the X) and click on the New View icon and select Map.
The default map view should look like the following. Click on the pencil icon to edit the map view.
4. Hover the mouse to the right of the OBIEE_COUNTRY label in the map formats pane and edit or remove it (click on the pencil icon or X). If you removed it then next click on Create a New Map Format and select Color Fill and then select the OBIEE_COUNTRY layer.

Select Percentile Binning, 10 colors, select a color ramp and click OK.

Optional: Click Edit again (but click on the edit icon next to the 1-Revenue layer) and then click “Allow Dashboard Users to edit thresholds” in the Color Fill (OBIEE_COUNTRY) dialog pane and click OK.

5. Next, click the “Preview” button in the Compound View toolbar to see how results would look on a Dashboard.
Hover over a Country, e.g. the USA, and you will see an info-window. Click on the country and then within the info-window click on the hyperlinked label USA to drill to the State level. Uncheck the box next to the map format for the OBIEE_COUNTRY layer and then either double-click someplace near the USA on the map or use the navigation panel on the map to zoom in one level and then pan over to the USA. You should see the states color-filled by Revenue.
6. Close the preview and save the newly created analysis.

Click OK.
Adding a Bar Chart to the Map View

Now that we have a basic map view, let’s add additional formats to it to display other measures. We’ll add a bar chart that shows product revenue by brand by state when a user drills down from USA to US states. In order to do so we have to add Product Brand to the criteria (i.e. set of columns in the analysis), and create a new format associated with the OBIEE_STATES layer.

1. Click on the Criteria tab and add the Products → P4 Brand column to the set of selected columns for the analysis. The following image contains snippets from multiple screenshots.

![Image of adding columns to criteria]

2. Click the Results tab and edit the map view. Click on the Add a new map format icon and select Bar Graph. Then select OBIEE_STATE as the layer for this format, thereby specifying that the bar graphs will be displayed per state. Click OK when prompted that the required column (C62 Geo Ctry State Name) will be added to the analysis and finally select P4 Brand as the column for the Bars.

![Image of adding bar chart format]
Pan over to the USA, zoom in once if you wish, and uncheck the OBIEE_COUNTRY layer to see the bars graphs for the states. Check the OBIEE_COUNTRY layer to turn it back on.

3. Click on Preview to see how it’ll look on a dashboard. Click on USA, and then on the USA link in the info-window to drill down to the State level. Uncheck the OBIEE_COUNTRY layer, click on a bar chart for a state, and you should see something like the following screenshot.

Exit the preview and save the modified analysis.

**Master-Detail Linking via Map Views**

This section describes simple master-detail linking with map views. We’ll add some bar charts to the analysis created above and link them to the map. The map will act as the master and the bar charts will present details for the selected State in the map view. The resulting view, when previewed on a dashboard page, should look something like the screenshot below.
1. To get started open the previous analysis or create a new one with the C62 Geo Ctry State Name and 1-Revenue columns. Add a filter, C61 Geo Country Code is equal to/ is in USA, to the analysis. If you’re creating a new analysis then add a map view and edit it to choose percentile binning (decile) and a preferred color ramp.

2. Click on Criteria again and then on the State Name column and modify the column properties. Click on the Interaction tab and select “Send Master Detail Events” for the Value property. Specify a name for the channel, e.g. M1, and click OK.
3. Next add a bar graph to the analysis (click on results, then new view → graph → bar → default (vertical)). It should add it below the map. Edit the Graph view.

Move the State Ctry Name column to Sections area, check Display as slider, click on section properties and set the maximum number of section values to 50, and click OK.
4. Add P4 Brand to the Bars, Group By (Horizontal Axis) area. Then click on the Edit icon for the graph to change its size (to 400x300 pixels), and check the box “respond to master events” on channel M1. In the Style tab select “Rectangle” and click OK.
The graph should look like the one shown below.

5. Now click duplicate view, and

remove Revenue from the Measures and add Billed Quantity instead, and finally click Done.
6. Add Graph 2 to compound view to the right of the first graph. That is drag the Graph:2 icon from the Views pane and drop it in the Compound Layout just to the right of the existing Graph view.
7. Click Preview and test Master Detail linking.
Action Links via Map Views

The last hands-on exercise uses action links as the Interaction mechanism instead of the master-detail events which were used above. Save the analysis with a different name, i.e. click on Save As and name it something else (e.g. SimpleActionLinkMapExample).

1. Now create a new analysis which will have details by state. Add the C62 Geo Ctry State Name column, add a filter for Country Code = USA AND C62 Geo Ctry State Name is prompted, and add other columns to the analysis. See the screenshot below for an example.
View the results and save the analysis (e.g. name it PromptedStateDetailAnalysis).

2. Next open the previously saved analysis (SimpleActionLinkMapExample), click on Results if necessary, and edit the Compound Layout. Remove the two graph views (Graph and Graph (2)).

Add a Table view below the map view (Not sure if it’s a bug or a feature but a table view is needed for the action link to work in the map view).
3. Now click on Criteria and edit the column properties for the Ctry State Name column to change the Interaction from Send Master-Detail Events to Action Link. In the Action Links dialog click on add Action Link, select Navigate to BI Content, choose the saved detail analysis (PromptedStateDetailAnalysis), and click OK where prompted.
4. Preview the analysis (i.e. click on show how results will look on a dashboard) and test the action link.
### Collaborate 12: Building map-based dashboards

#### Map Formats

- **OBIEE_STATE**
  - First Tenth
  - Second Tenth
  - Third Tenth
  - Fourth Tenth
  - Fifth Tenth
  - Sixth Tenth
  - Seventh Tenth
  - Eighth Tenth
  - Ninth Tenth
  - Last Tenth

#### Prompted State Detail Table

<table>
<thead>
<tr>
<th>CS2 Geo City State Name</th>
<th>Avg Brand</th>
<th>P1 Type</th>
<th>P1 Product</th>
<th>46 Avg Order Process Time</th>
<th>24 Avg # of Orders by Sales Rep</th>
<th>20 Avg # of Orders by Customer</th>
<th>15 Avg Order Size</th>
<th>11-Variable Costs</th>
<th>10-Variable Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA, Nevada</td>
<td>BioTech</td>
<td>Communication</td>
<td>Cell Phone</td>
<td>7.00</td>
<td>1.00</td>
<td>1.00</td>
<td>184.96</td>
<td>55.00</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>Electronics</td>
<td>Accessories</td>
<td>Bluetooth Adapter</td>
<td>13.00</td>
<td>1.00</td>
<td>1.00</td>
<td>3142.91</td>
<td>943.00</td>
<td>2177</td>
</tr>
<tr>
<td></td>
<td>FunPod</td>
<td>Digital</td>
<td>Camera</td>
<td>6.00</td>
<td>1.00</td>
<td>2.00</td>
<td>2096.37</td>
<td>673.00</td>
<td>1668</td>
</tr>
<tr>
<td></td>
<td>Games</td>
<td>Portable</td>
<td>iPod/iPod</td>
<td>11.00</td>
<td>1.00</td>
<td>2.00</td>
<td>1242.48</td>
<td>582.00</td>
<td>770</td>
</tr>
<tr>
<td></td>
<td>HomeView</td>
<td>Services</td>
<td>Maintenance</td>
<td>9.00</td>
<td>1.00</td>
<td>2.00</td>
<td>1975.35</td>
<td>881.00</td>
<td>1425</td>
</tr>
<tr>
<td></td>
<td>TV</td>
<td>LCD</td>
<td>Standard</td>
<td>8.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1038.77</td>
<td>426.00</td>
<td>815</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCD 36X</td>
<td>Standard</td>
<td>3.00</td>
<td>1.00</td>
<td>1.00</td>
<td>3555.25</td>
<td>1082.00</td>
<td>1545</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tungsten E</td>
<td>Plasma TV</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>4355.56</td>
<td>1301.00</td>
<td>3016</td>
</tr>
</tbody>
</table>
Exercises

1. Publish your analyses containing maps to a dashboard page. Optionally create a dashboard with three pages. The first one containing the master-detail linking example, the second containing the action link example, and the third containing the prompted state detail analysis which is linked to from the action link map example.

2. Modify the simple map action link analysis to add a section action link to the C62 Geo Ctry State Name column value. This one should navigate to the third dashboard page (the one with the prompted state detail analysis) you created above.
Collaborate 12: Building map-based dashboards
Using spatial queries in analyses

Example 1: Demographics info for census blocks near selected stores

In this example we’ll recreate portions of the “Blocks in Distance Range” page of the “Source Specific Features”. “8.5 Oracle Spatial” dashboard in SampleApp. We will reuse the existing prompts, opaque views, and presentation and session variables. That is, we will not define them again. The rationale for, and steps involved in, defining them will be outlined however.

This example is intended to illustrate functionality and is not necessarily a realistic use case. It shows the following:

- How to use spatial functions in database queries from OBIEE
- How do use opaque views, presentation and session variables, and dashboard prompts to pass parameters to the above spatial queries

The NAVTEQ sample data contains street, postcode, and points-of-interest or business listing details, and some census block level demographics, for San Francisco, London, and Sydney.

The screenshot below shows the relevant SampleApp dashboard page.
First let’s look at the RPD side of things, i.e. the database query, bind variables, and corresponding business and presentation layers.

The physical layer consists of the tables and opaque views shown below. The Block_Group table contains the census block level demographics (household income, population, and education). The GC_POI_NA (points-of-interest North America) contains a sample, limited to San Francisco, of NAVTEQ’s POI content. It contains information on the establishment name (e.g. Flight 001), the category ID, address, phone number and other details. The NTC_META_POI_CAT_REF table is a metadata table containing category_ids and names (e.g. 5400 = GROCERY STORE or 9567 = SPECIALTY STORE).

The view V_POI_Block_Group is the query to find block groups that are within a specified distance of selected stores. For example, block groups within 500 meters of Specialty Stores in zip 94102. The postal code, category id, and distance (in meters) values are passed in as bind variables.
The view is defined as a SELECT table in the BI Admin tool as shown in the screenshot below. Note the use of Database specific SQL.

The query text is

```
select distinct
b.id as block_group_id,
s.poi_id as poi_id,
d.name as category_name
from
block_group b,
ntc_map.poi_shop s,
ntc_meta.poi_cat_ref d
where
s.iso_country_code='USA' and
s.cat_id = d.cat_id and
d.name = 'VALUEOF(NQ_SESSION.OGS_CATEGORY)' and
s.poi_postcode = 'VALUEOF(NQ_SESSION.OGS_POSTCODE)' and
sdo_within_distance(b.geometry, s.geometry, 'distance=VALUEOF(NQ_SESSION.OGS_DIST_M)')='TRUE'
```

OGS_CATEGORY contains the category name (e.g. SPECIALTY STORE), OGC_POSTCODE is the zipcode (e.g. 94102), and OGC_DIST_M is the distance in meters. The spatial operator `sdo_within_distance(blocks, stores, 'distance=500')` returns TRUE for blocks (b.geometry) that are within 500m of a SPECIALTY STORE in 94102 and FALSE otherwise.
The session variables are defined as non-system with default values.

The dashboard prompts are defined as shown with each setting a request variable.
Hands-on portion of the example

Next we’ll create an analysis and a dashboard page using the elements described above.

Click on New Analysis and select the Geoloc subject area. Expand the Blocks in range of POIs folder and add the Block Group Id, POI Category Name, and Households columns to the analysis. Expand the POI Details folder and add the POI Name, Loc Long, Loc Lat, # of POIs, and Full Address columns to the analysis.

Then click on the Results tab. After the query is performed (may take a while depending on the HOL environment) edit the table properties and change them as shown in the screenshot below. Move (drag and drop) the POI category Name to the Section header in the Layout Panel. Move Block Group ID, Loc long, and Loc lat columns to the Excluded list. Rename # of POIs to be # of Blocks (since we’re grouping by POI name in the table). And order the remaining columns as shown.

Then click on Done and add a map view to the analysis. Like the table, the map view may take a while to render.
Once the map view has rendered click on Edit. You may have to navigate to San Francisco if the map view initially centers somewhere south near Sunnyvale/Cupertino or elsewhere and shows no color-fill. Use the navigation panel or simply pan down (hold down the mouse button and scroll down). Zoom in if necessary once the map is centered on SFO.
Change the color fill palette to the one shown below.

![Color Fill Palette](image)

Click on the new layer icon (next to the BI Data Layers) to add a new map format to the view. Select Image and then Custom Point Layer.

When the custom point layer dialog shows up edit it as shown below. Uncheck Name automatically and name it Stores. Select Loc Long as the Longitude column and Loc lat as the Latitude column. Click on the drop down list for Vary Image By and choose “# of POIs”. Select 2 and the number of Bins and change Bin Type to Value Binning. Once the default bin value ranges and labels show up edit them to the values shown. Set the ranges to 0-50 and 50-500 and enter Stores for both labels. Click on the default image, scroll down and click on the globe with pin and then choose the green star as the image. Do this for both bins.
Click on Done.

Save the analysis to your preferred folder.
Now create a new Dashboard and add a page and then add the Blocks to Shops Distance Prompt (in the Catalog area expand the folders 8. Source Specific Features, 8.5 and then POIs in range of) and the newly created analysis to the page.
Collaborate 12: Building map-based dashboards
Save and then run the dashboard. You should see something like the following screenshot.

Example 2: Stores near an address

This example replicates the Geocode SQL page in the dashboard named “8.5 Oracle Spatial”. It uses a Direct Database request, prompts, presentation variables, and user-defined functions in the database. We will reuse existing functions, prompts and variables.

The SampleApp dashboard screenshot is shown below. The presentation variables are used to pass in the store category (CATEGORY), the distance value (DISTANCE), and the input address (ADDRESS). The Direct database request SQL is

```sql
SELECT s.poi_id store_id, s.name, s.poi_house_number || ' ' || s.poi_street_name street, s.poi_phonenumber phone, d.name as category_name
FROM ntc_map_poi_shop s, ntc_meta_poi_cat_ref d
WHERE s.cat_id = d.cat_id AND d.name = '{@{CATEGORY}{PHARMACY}}' AND
sdw_within_distance(geometry, geocode_address '{@{ADDRESS}{747 Howard St, San Francisco, CA 94013, US}}','distance=' '{@{DISTANCE}{2000}}{unit=m}')='TRUE'
```
The presentation variables are defined as follows.

The address and distance prompt definitions are
The shop category prompt is
The user-defined function geocode_address takes a comma delimited address string as input and in turn calls the SDO_GCDR.GEOCODE_AS_GEOMETRY function.

```sql
function geocode_address(address varchar2) return sdo_geometry deterministic as
    addr_lines    sdo_keywordarray;
an_addr_line   varchar2(128);
country       varchar2(64);
um_lines      number;
input_addr    varchar2(256);
theGeom       sdo_geometry;
begin
    -- assumes address is comma delimited and is
    -- house no. street,city,state and/or postalcode,country(e.g.US,UK,GB, or AU)
    -- parse out the address pieces. country = from last comma onwards
    -- everything except country goes into a sdo_keywordarray
    -- add a comma at the end so num_commas = num sections
    input_addr := address || '',';
    addr_lines := sdo_keywordarray();
    -- assume database version 11g
    num_lines  := regexp_count(input_addr, ',', 1, 'i');
    addr_lines.extend(num_lines);
    for i in 1 .. (num_lines-1) loop
        an_addr_line := regexp_substr(input_addr, '[^,]+', 1,i);
        addr_lines(i) := an_addr_line ;
    end loop;
    country := regexp_substr(input_addr, '[^,]+', 1,num_lines);
    theGeom := sdo_gcdr.geocode_as_geometry('obiee_navteq', addr_lines, country);
    return theGeom;
end;
```

Geocode-as_geometry is an Oracle Spatial function. It requires reference data from a data vendor such as NAVTEQ.

**Hands-on portion of the example**

Next we’ll create an analysis and a dashboard page using the elements described above.

Click on new Analysis and then Create Direct Database request. Enter obiee_navteq for the Connection Pool and the SQL text given above in the SQL Statement area. Check the Bypass Oracle BI Presentation Service cache and then on the button titled Validate SQL and Retrieve Columns.

Your page should look like the one shown below.
Click on Results.

And save the analysis. Open the previous dashboard (the Spatial Query Examples created in the example above) for edit and add a new page. Add the saved analysis and the Geocode SQL Prompt (from the folder named POIs in rage of) to the page. Save and run the dashboard.
Example 3: Distance between locations

This example replicates the Cust Distance SF page in the dashboard named “8.5 Oracle Spatial”. It uses the EVALUATE function in a column formula, prompts, presentation variables, and database functions. We will reuse existing functions, prompts and variables.

The example here will only describe the use of EVALUATE in a column formula to determine the distance between two points specified using Longitude, Latitude coordinates. It won’t replicate the analysis or dashboard page shown below.
Hands-on portion of the example
Click on new Analysis and select the Sample Sales subject area. Add the columns shown below (except the Distance in Meters one) to the analysis.

Add a Filter for D55 City and C55 City is equal to/is in San Francisco click on Results.
Modify the data format to include 3 decimal values for the D65 Longitude and D64 Latitude columns so that the Longitude/Latitude values are not truncated. That is, click on Column Properties, Data Format, Override default number format, set decimal places to 3, and uncheck Use 1000’s separator.

Add filters for C65 Longitude is not null and D65 Longitude is not null. Add another column for the distance. Drag and drop another copy of Base facts 1 – Revenue for example. Edit its formula and check Custom Headings. Set Folder name to Distance and column name to Distance in Meters.

Click OK and then edit the column again to set its formula.

Set the column formula to

```
EVALUATE('sdo_geom.sdo_distance(sdo_geometry(2001,8307,sdo_point_type(%1, %2,null),null,null), sdo_geometry(2001,8307, sdo_point_type(%3, %4, null), null, null), 0.5, "unit=meter")' as FLOAT, "Cust Geo Codes"."C65  Longitude", "Cust Geo Codes"."C64  Latitude", "Office Geo Codes"."D65 Longitude","Office Geo Codes"."D64  Latitude")
```
This uses the database function sdo_geom.sdo_distance() to compute the distance in meters between the two points. The SDO_GEOMETRY object constructor used in the sdo_distance function creates a point geometry instance. It takes 5 parameters. The first (2001) specifies that the geometry is a 2-D point. The second (8307) identifies the spatial reference system (WGS84 Longitude/Latitude in this case. AKA GPS coordinates). The third creates a point using the supplied Longitude and Latitude values. The fourth and fifth parameters are null here. They're used when defining lines, polygons, or collections.

Click OK and then on the Results tab.
You can also choose to use some other function to compute the distance between two Longitude/Latitude pairs. For example, the Pythagoras theorem on an equiangular projection as described at the website

http://www.movable-type.co.uk/scripts/latlong.html

The formula is

\[ x = \Delta \text{lon} \cdot \cos(\text{lat}) \]
\[ y = \Delta \text{lat} \]
\[ d = R \cdot \sqrt{x^2 + y^2} \]

where \( R = 6371 \) Km, the approximate radius of the earth at the equator. The latitude.longitude values must be in radians (e.g. D55 Longitude * 180/\pi in this case). The column formula then becomes

\[
\text{SQRT}((("Office Geo Codes"."D65 Longitude" *(180/\pi())) - ("Cust Geo Codes"."C65 Longitude" *(180/\pi())))^2)\cos(("Office Geo Codes"."D64 Latitude" * (180/\pi())) + ("Cust Geo Codes"."C64 Latitude" *(180/\pi())))^2) * (("Office Geo Codes"."D65 Longitude" *(180/\pi())) - ("Cust Geo Codes"."C65 Longitude" *(180/\pi()))) * \text{COS}(("Office Geo Codes"."D64 Latitude" * (180/\pi())) + ("Cust Geo Codes"."C64 Latitude" *(180/\pi()))) / 2) + (("Cust Geo Codes"."C64 Latitude" * (180/\pi())) - "Office Geo Codes"."D64 Latitude" * (180/\pi()))) * ("Cust Geo Codes"."C64 Latitude" * (180/\pi())) - "Office Geo Codes"."D64 Latitude" * (180/\pi())))) * 6371