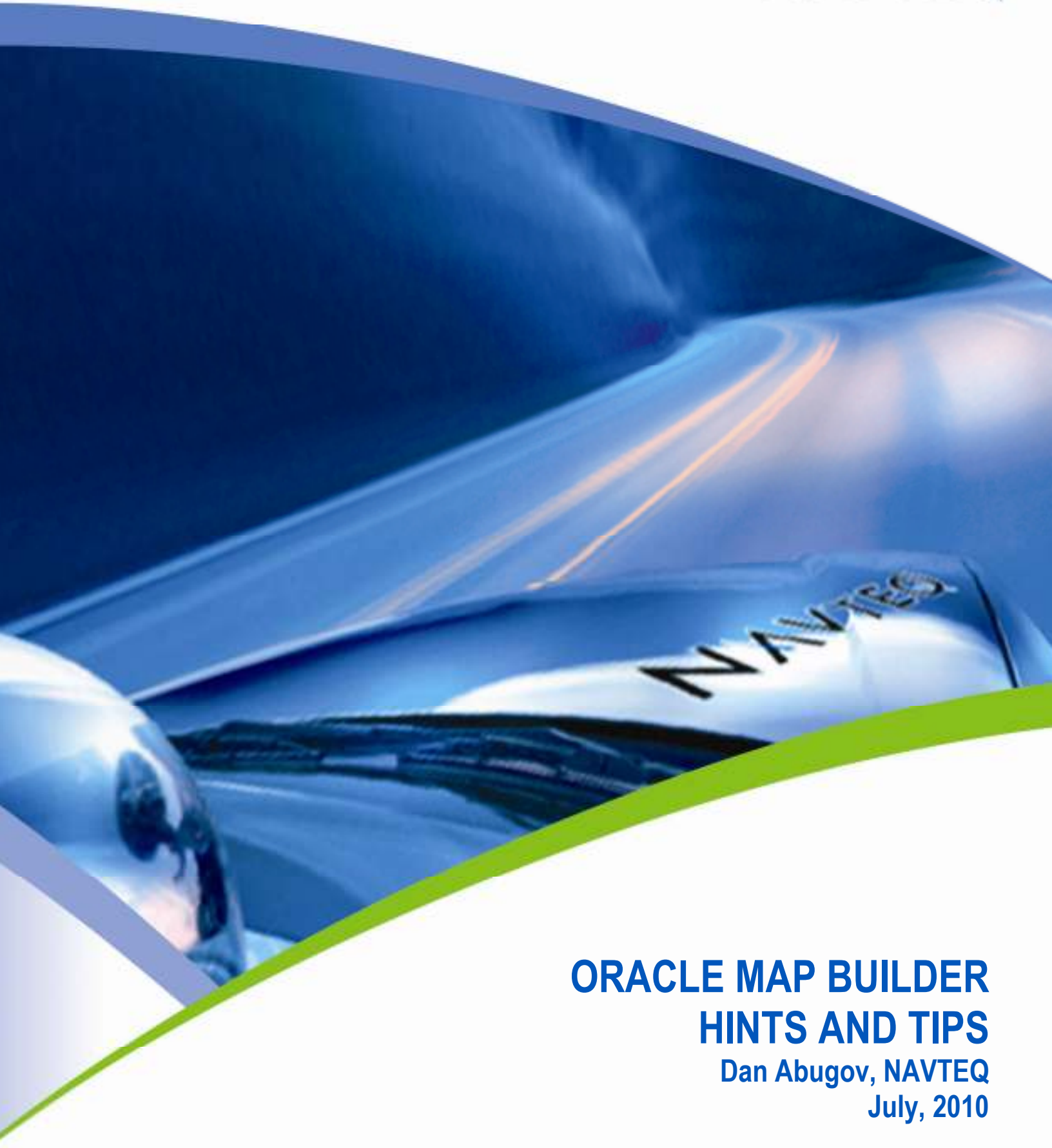
NAVTEQ



ORACLE MAP BUILDER HINTS AND TIPS

Dan Abugov, NAVTEQ

July, 2010

Table of Contents

Disclaimer

This content is provided "as-is" and without warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality and non-infringement. NAVTEQ does not warrant that the content is error-free and NAVTEQ does not warrant or make any representations regarding the quality, correctness, accuracy, or reliability of the content. You should therefore verify any information contained in the content before acting on it.

To the furthest extent permitted by law, under no circumstances, including without limitation NAVTEQ's negligence, shall NAVTEQ be liable for any damages, including, without limitation, direct, special, indirect, punitive, consequential, exemplary and/or incidental damages that result from any content, even if NAVTEQ or an authorised representative has been advised of the possibility of such damages.

Do you have comments regarding this White Paper? If so, please send them to Enterprise@NAVTEQ.com

Introduction

Increasingly, Oracle applications are being shipped with built-in mapping capabilities. Of note, the Oracle Business Intelligence road map includes the ability to integrate mapping and drill-down analysis as a BI component. Oracle Communications, Oracle Utilities, Oracle Transportation Management, and Oracle Field Service all have mapping components. For applications development, Oracle JDeveloper has a GeoMap component for including maps in applications. Similarly, Java script can also be used for delivering interactive mapping components to users.

This paper will briefly discuss the technologies included in Oracle that are used for mapping. Then the paper will focus on the Map Builder component of Oracle Fusion Middleware, showing in a step-by-step fashion how users can create their own maps. Using a layering approach, attendees will see best practices for creating maps, building them layer by layer to get the rich maps end users expect. This paper includes seldom seen but valuable features for making maps look good and perform well.

Overview of Oracle's Location Technologies

Oracle databases (Express, Personal, Standard and Enterprise Editions) all include built-in storage, index, and query capabilities for location data. We define location data as coordinates that represent the location of a feature on the surface of the Earth. Using stored location information from an Oracle database we have the ability to show features on a map as well as analyze features based on interactions and proximity to other features. This feature of the Oracle database is called Oracle Locator.

Every Oracle Application Server (including the WebLogic Application server) includes the ability to render maps for display in a wide variety of web-based applications. The map rendering component of the application server is called Oracle Fusion Middleware MapViewer.

Putting mapping aside momentarily, the built-in location capabilities in all Oracle databases include the ability to analyze information geographically, allowing us to answer questions such as:

How many hotels are within two miles of a proposed conference location?
How many of my customers have average household income of greater than $67,000?
What is the closest ATM to my current location?

In addition to mapping and the abovementioned analysis capabilities, Oracle Spatial is an option with Oracle Enterprise Edition. Oracle Spatial includes advanced capabilities beyond the basic capabilities shown above. Oracle Spatial includes functionality such as Geocoding, Routing, Image Storage and retrieval, Network Analysis, Topology, Web Services, and a lot more.

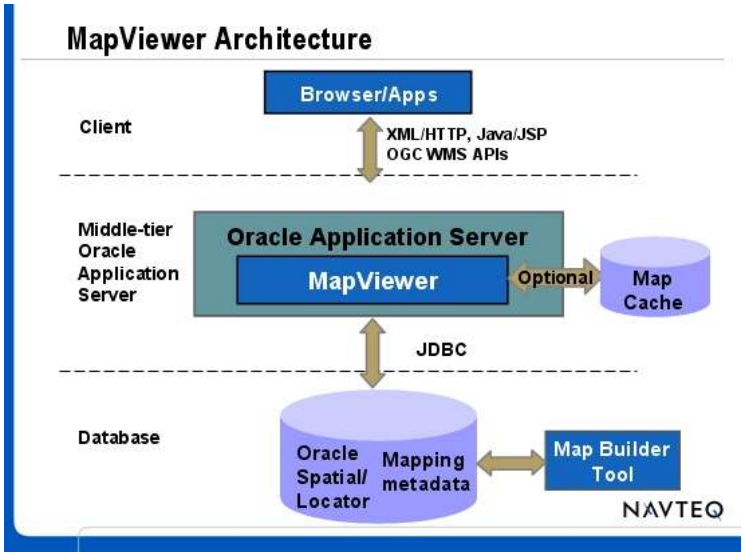
If this paper is the first you have heard of Oracle's Location technologies, please see the Introduction to Mapping and Location Analysis in Oracle white paper at navteq.com/enterpriseamericas.

Oracle Fusion Middleware MapViewer

Every Oracle Application Server includes MapViewer, a middle tier map drawing/rendering component. MapViewer renders location data directly from the database and includes APIs to render data from other sources as well.

In addition MapViewer uses centralized information (map metadata) that includes the rules associated with how maps are styled and displayed to users. This map metadata is easily managed using the map authoring tool which is part of MapViewer, called Map Builder.

MapViewer make is very easy for web-based/web accessible applications to include maps. MapViewer includes JAVA, XML, Open Geospatial Consortium Web Mapping Service (OGC WMS), and JSP APIs.



Below is an example of a map rendered by Oracle Fusion Middleware MapViewer:

In this paper we will discuss building maps using Map Builder, which is the map authoring tool associated with Oracle Fusion Middleware MapViewer.

Oracle Map Builder

Every instance of the Oracle Application Server includes Oracle Fusion Middleware MapViewer, which is Oracle's standard map rendering component. In order to show maps in MapViewer, Oracle users need data to display, and map metadata to describe how to display it. Oracle Map Builder is the Oracle map authoring tool that is used to create the map metadata.

Map Metadata

Oracle map metadata has three important components:

Styles

Styles control fill color, border color, line thickness, line style, transparency, and many other aspects of how features will be rendered in MapViewer. In addition, text styles control the font, size, color, bold, and many other aspects of how text is rendered on the map.

Styles are stored as XML in view `USER_SDO_STYLES`.

Themes

A theme associates features in a table with a style. For instance, park features should be rendered in green.

A text label column can also be associated with a theme. The label column has its own style (size, font, color, and other attributes).

Themes are stored as XML in view `USER_SDO_THEMES`.

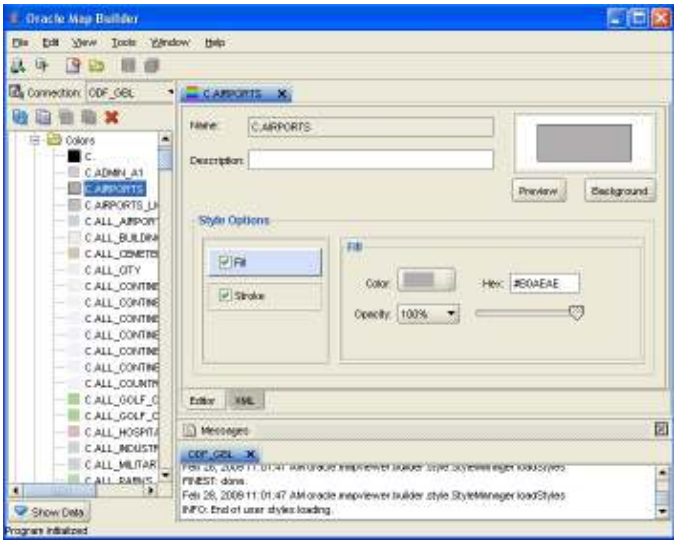
Maps

A map includes one or (nearly always) more themes. A map defines which themes are displayed at which zoom levels, and provides the directions to MapViewer for the display of rich maps.

Maps are stored as XML in the view `USER_SDO_MAPS`.

Oracle Map Builder creates and manages XML metadata associated with styles, themes, and maps (plus a lot of other things) through the use of a simple, easy-to-use GUI.

Compare this GUI:



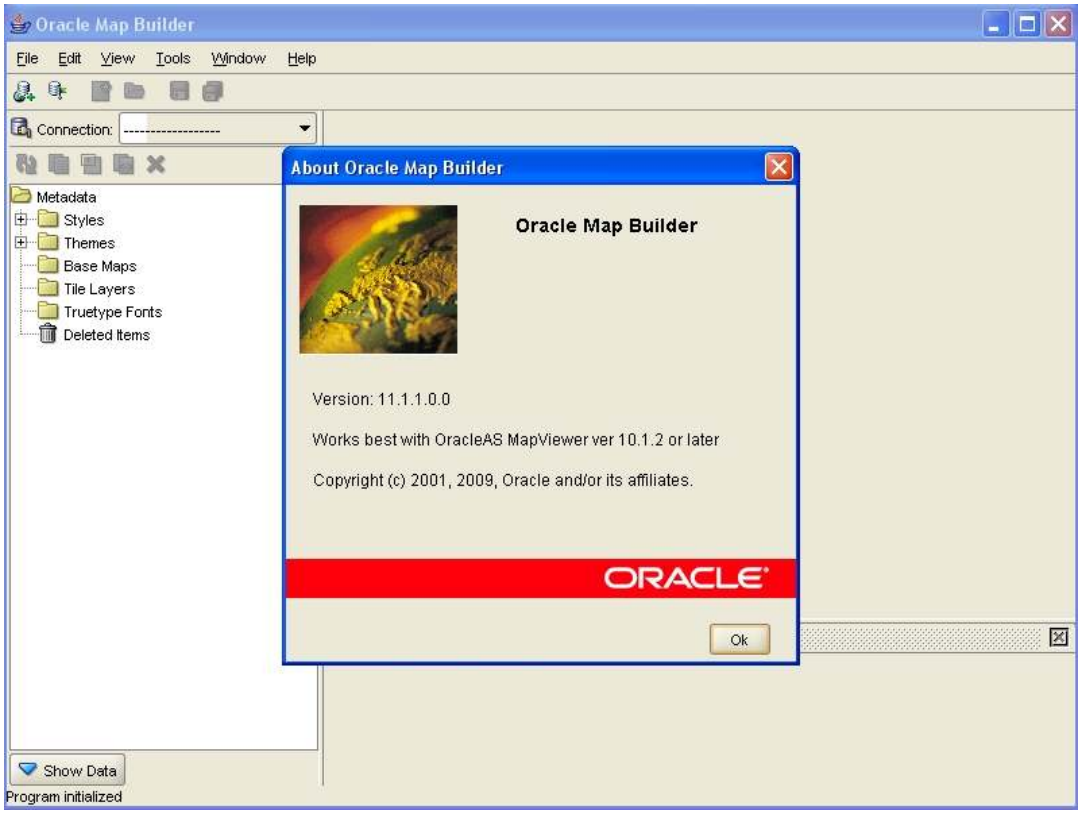
With this text:

```
<?xml version="1.0" standalone="yes"?>
<svg width="1in" height="1in">
  <desc/>
  <g class="color"
     style="stroke:#403E3E;fill:#B0AEAE">
    <rect width="50" height="50"/>
 </g>
</svg>
```

Oracle Map Builder removes the need to understand the XML structure of Styles, Themes, and Maps through the use of its simple, easy-to-use interface.

Software and Content for this White Paper

This paper shows use of Oracle Map Builder Version 11.1.1.0.0, which is available from Oracle.



Data Content Setup

This White Paper shows the use of NAVTEQ map content in Oracle. The data used in this presentation is available at NAVTEQ Network for Developers (www.nn4d.com). To download the data:

- Got to www.nn4d.com
- Mouse over "BUILD" and click on Sample Data on the left of the drop down menu
- Register if you haven't registered, or log in
- Select the "Yes, I want to download sample data clips." radio button
- Click "Continue"
- Scroll to "Oracle Delivery Format (ODF) Data..."
- Click the Download link

After downloading, follow the included instructions for loading and setting up the data.

The sample data is the same data that is used by many Oracle Applications. All of the Oracle applications that require map content use NAVTEQ data. A list of application that require NAVTEQ data and/or that can integrate NAVTEQ content includes:

- Oracle eLocation Services http://elocation.oracle.com/elocation/ajax/
- Oracle E-Business Suite Field Service Advanced Scheduler
- Oracle Utilities Mobile Workforce Management
- Oracle Communications Unified Inventory Management
- Oracle Site Hub
- Oracle Content Management Suite
- Oracle Discoverer
- Oracle Business Intelligence Enterprise Edition (OBIEE)
- Oracle Retail
- Oracle Transportation Management

Also note that many other applications easily integrate with NAVTEQ content using Oracle Location Technologies.

Starting Oracle Map Builder

If you use Oracle Map Builder frequently, put a startup icon on your desktop so you can simply double click to start Map Builder 

. The shortcut associated with the icon is (on windows, assuming the mapbuilder.jar is in c:\oracle):

```
c:
cd \oracle
java -jar -Xmx256M mapbuilder.jar -noconnect -cache 64M
```

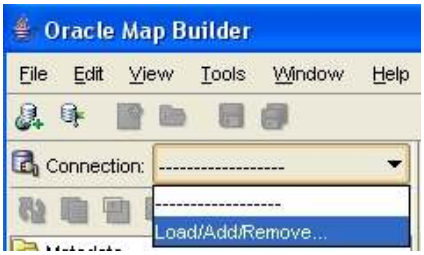
`-Xmx256M`: Allows Map Builder to use up to 256 MB of memory, which is enough memory for very rich map development.

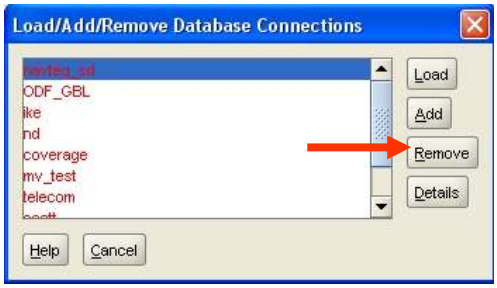
`-noconnect`: Use if you tend to use a lot of different accounts for map development. The default Map Builder behavior logs in to all previously loaded connections and loads all map metadata for all users

`-cache 64M`: Map Builder will use up to 64 MB for a geometry cache. The geometry cache speeds rendering by reusing "unpickled" geometries

Connect to an Oracle Database

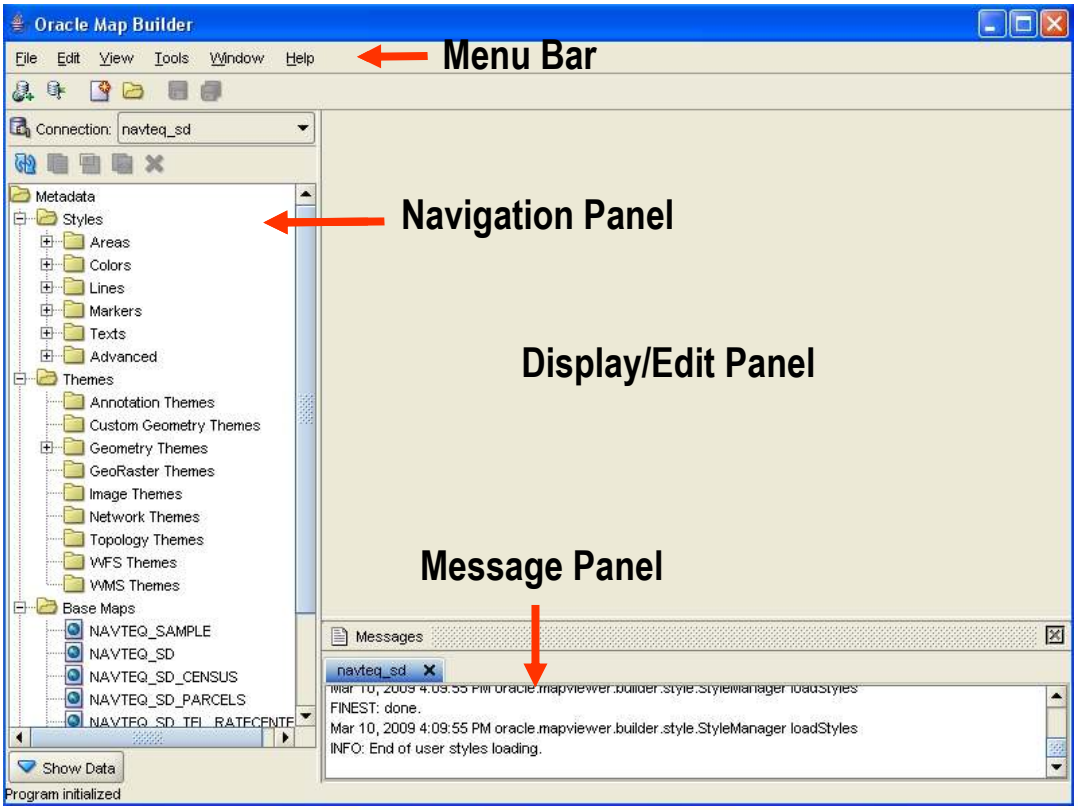
After startup, you need to connect to a database to read existing map metadata, and/or to store new map metadata
Click Connection, choose Load/Add/Remove
Click Add
Enter the connection information, then click OK





Map Builder Screen Layout

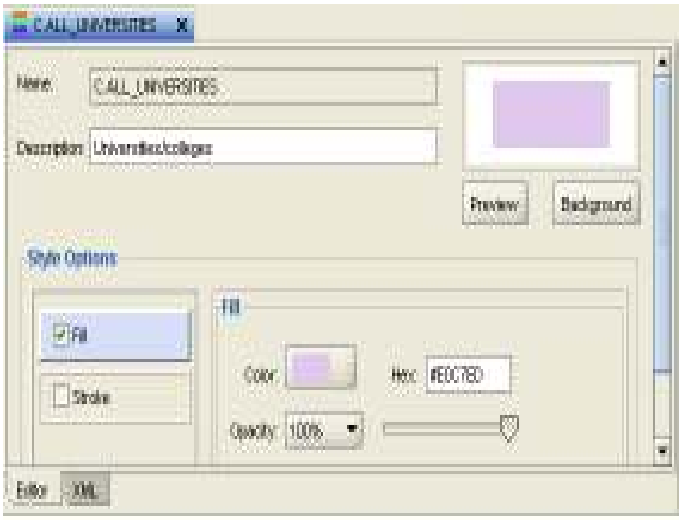
The Map Builder interface is below. Most interactions for building maps are through the Navigation Panel and the display/edit panel.



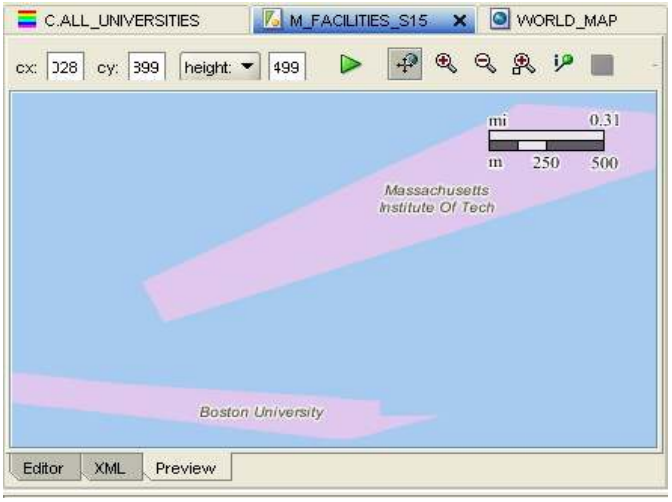
Map Creation in Map Builder

To create a map using Oracle Map Builder:

- Create styles that will define how features will look on the map



- Create themes, which associate features with the style you have created



- Create a map, which is a set of themes and instructions for when to turn themes on and off

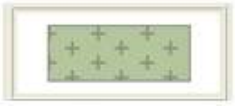
Styles in Map Builder

Styles define how objects will look on the map. When you create a style, it is not associated with a feature. A style is simply a description about how something will look when it is drawn if it is associated with that style.

The following describes styles:

Areas

- A fill pattern for an area



Colors

- The fill color of an area; can also include stroke color and thickness (the line surrounding the area), and transparency

Lines

- The color, transparency, thickness, wing lines, center lines, fencing, and end caps associated with a line

Markers

- Icons usually representing point features or labels on a map

Texts

- The size, font, color, and halo of text

Advanced

- A way of assigning different styles based on column values

NAVTEQ

Organizing your information in Map Builder is very important. Rich maps require a lot of information to define them. Fortunately there are some standards that, when followed, make management of map metadata easier.

Naming convention for styles will help you keep track of how your data is rendering. Later we will see that themes associate features with styles. When looking at a theme definition in Map Builder, or when assigning a style to a theme, if you have used these naming conventions your work will be easier.

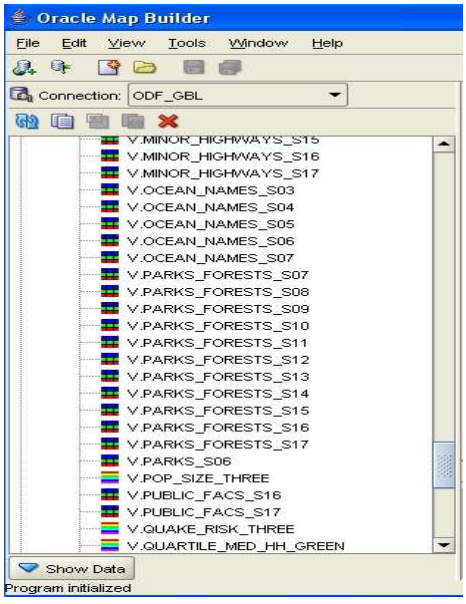
Prefix the style name with a standard character:

```
Areas      A.NAME
Colors     C.NAME
Lines      L.NAME
Markers    M.NAME
Texts      T.NAME
Advanced   V.NAME
```

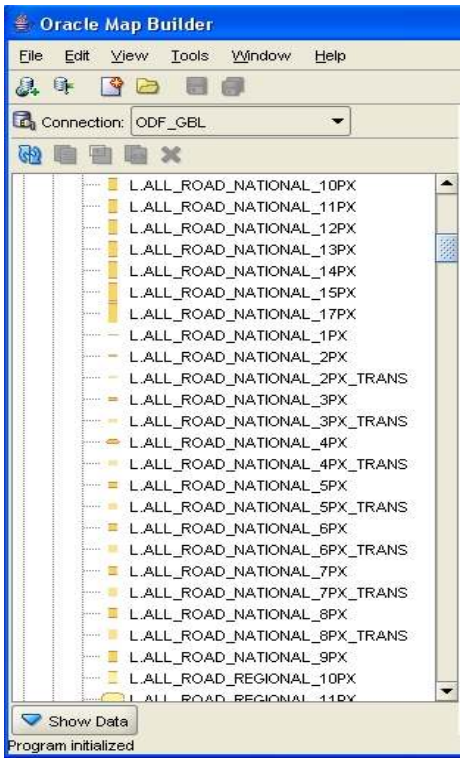
Richly developed maps can require hundreds of styles, useful at different zoom levels. Additional naming conventions are crucial to map maintenance. There are different schools of thought regarding naming conventions, most of which include a combination of one or more of the following:

- Naming by zoom level
- Naming by size
- Naming by scale

My favorites use a descriptive term followed by size and/or zoom level. Due to the constant updating and changing of scales to get the exact look for my maps, I have found using scale in the name requires too much maintenance. Below, some Advanced styles (V.) are named based on zoom level (_Snn):



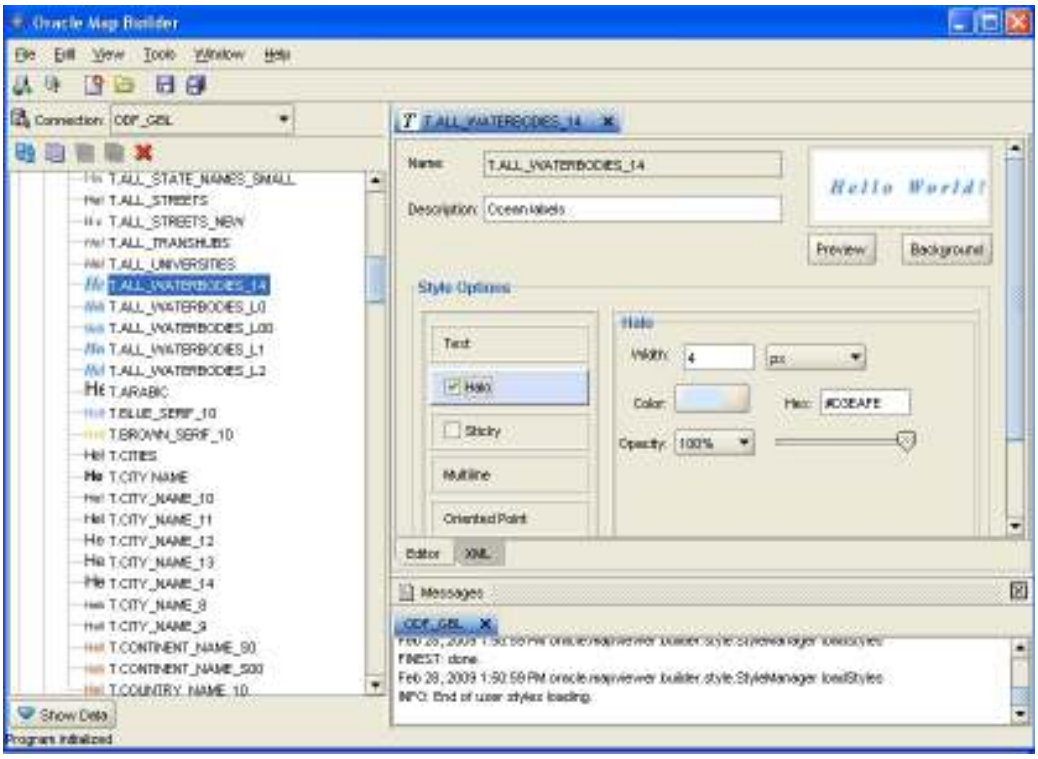
Below some line styles (L.) are named based on size (in this case, the width of the line in pixels, e,g, 4PX, and whether the line is transparent, e.g. TRANS):



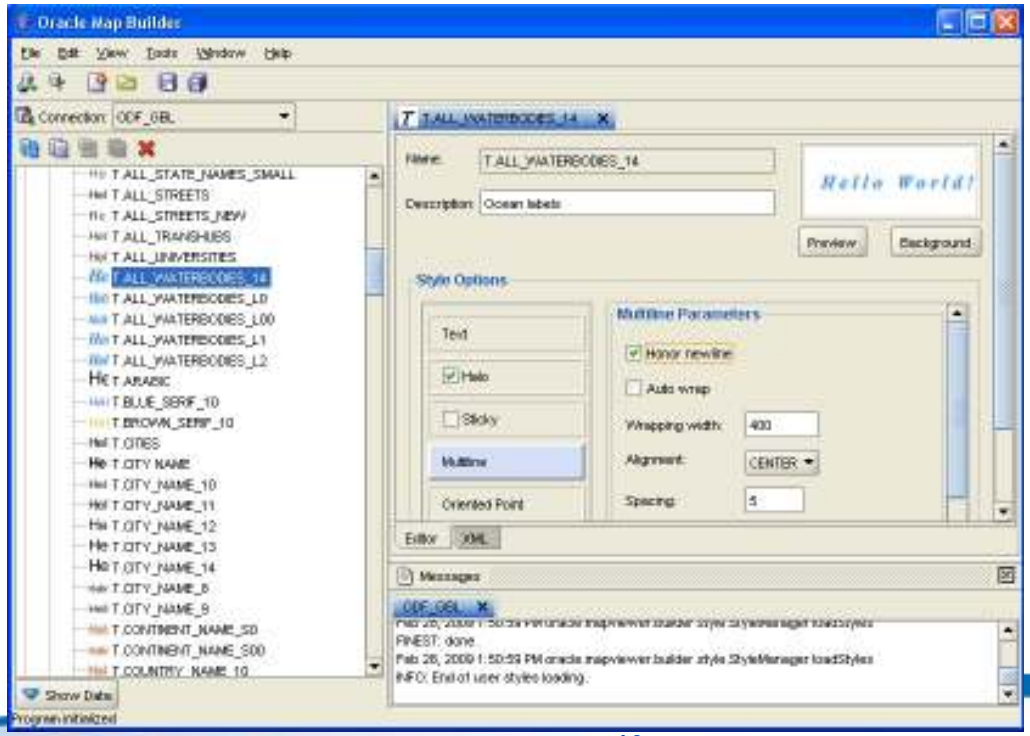
Text Styles

Text styles are often named base on what they are labeling (description) and the font size.

HINT: A halo around your text can often enhance readability.



HINT: You can tell MapViewer to use carriage control characters that are embedded in your text for labeling purposes.



You can include carriage control characters in the column that will be used for the label. Here is example SQL that includes character spacing and carriage control to make text render nicely on a map:

```
insert into POI_OCEAN values
  (sdo_geometry(2001,8307,sdo_point_type(-57, 36, NULL),null,null),
  '  N o r t h'||chr(10)||'A t l a n t i c'||chr(10)||'  O c e a n');
```

When the style is associated with location information and a label, the label will render in the map like this:

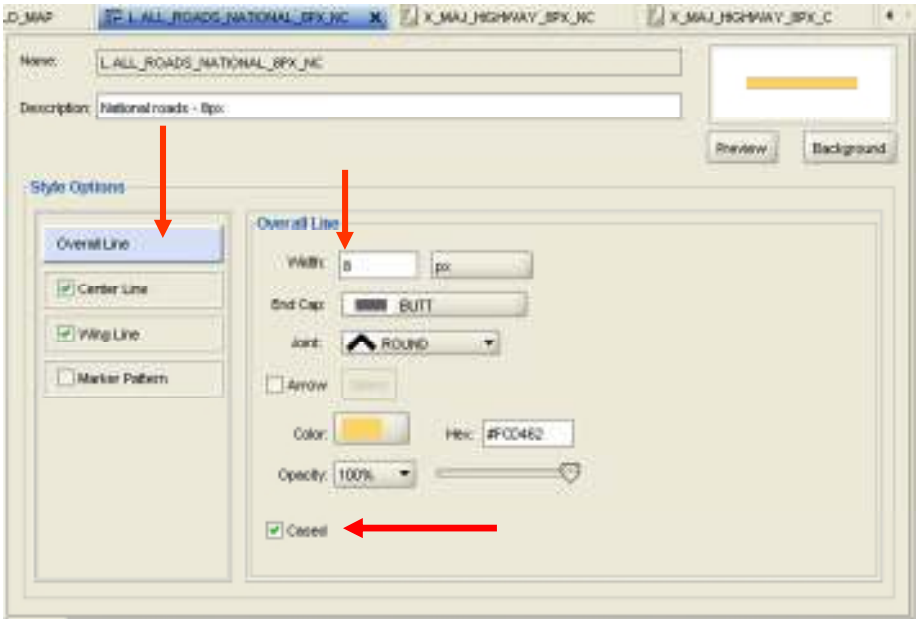
Line Styles

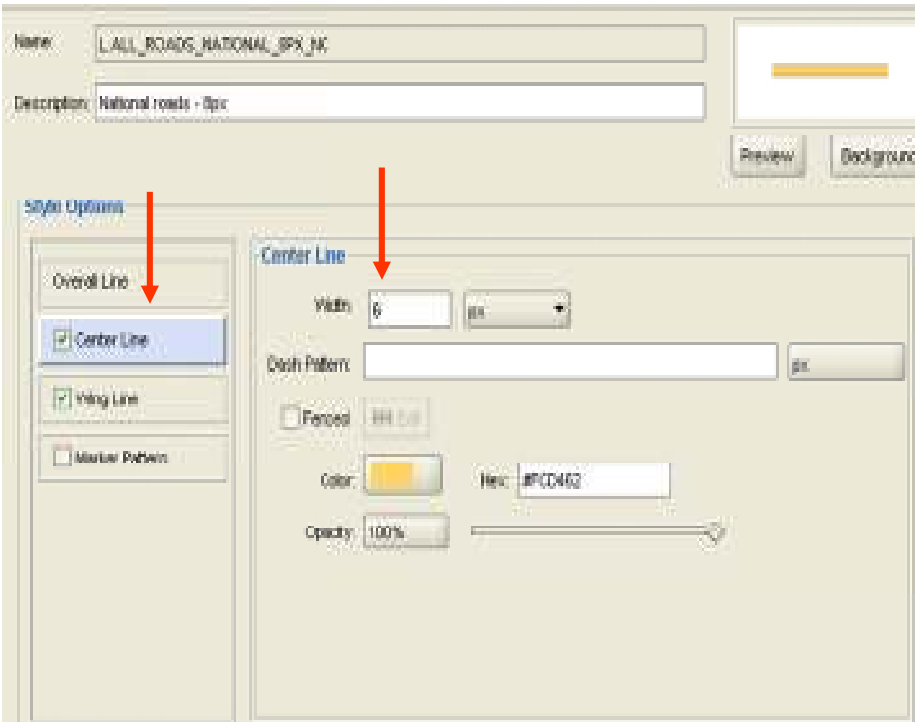
HINT: When rendering line styles, often using Cased line styles makes lines look smooth and professional:

Cased:

Not Cased:

With cased line styles, all overall lines are rendered first, then all wing lines and end caps are rendered, and lastly center lines are rendered. The center line width = overall line width – wing line width

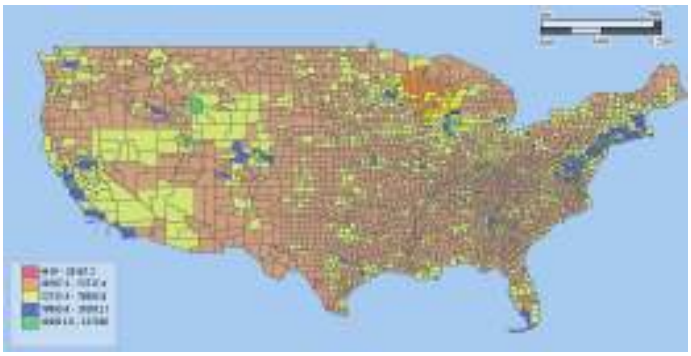




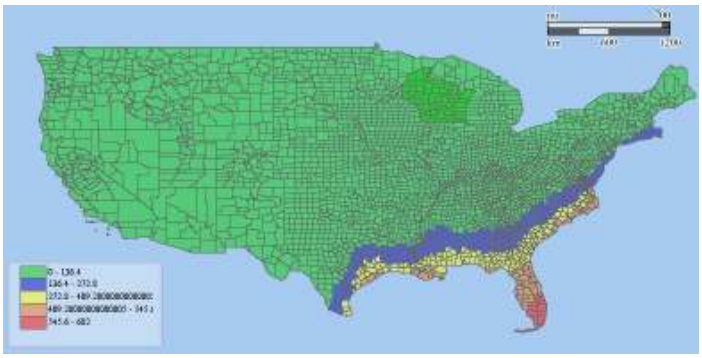
Advanced Styles

Advanced styles give you a very powerful way expressing information beyond simply drawing data on a map. Advanced styles allow you to impart information at a glance by defining styling rules based on the contents of one or more stored values associated with a feature.

Average Household Income



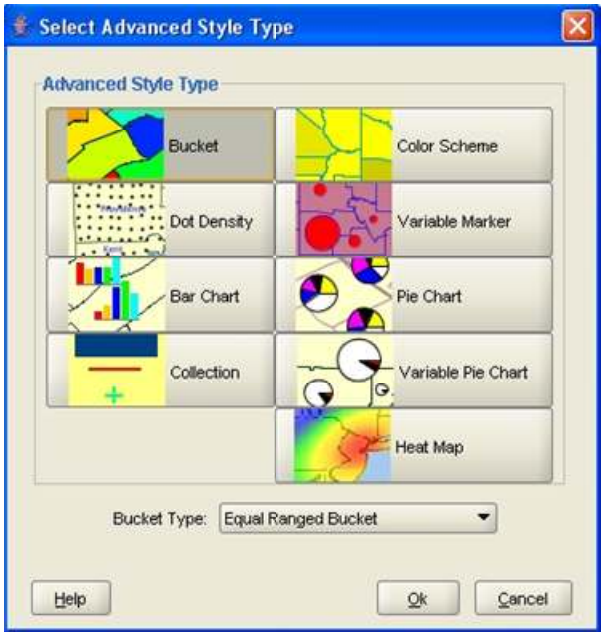
Hurricane Risk



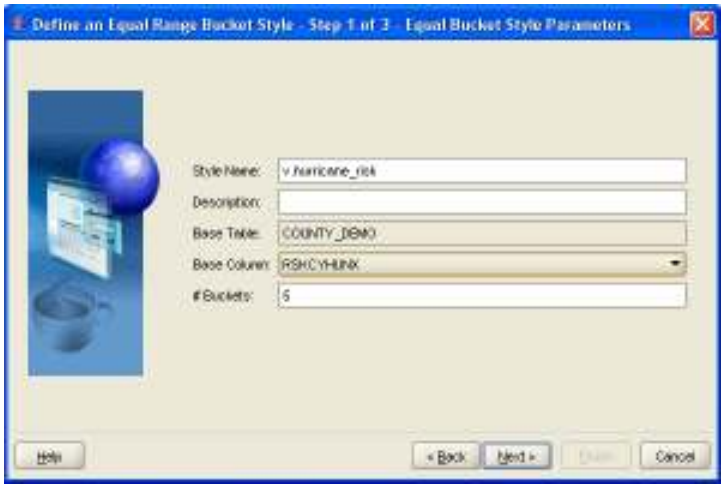
The demographic and risk data used above are courtesy of Primus Geographics, Inc. http://www.navmart.com

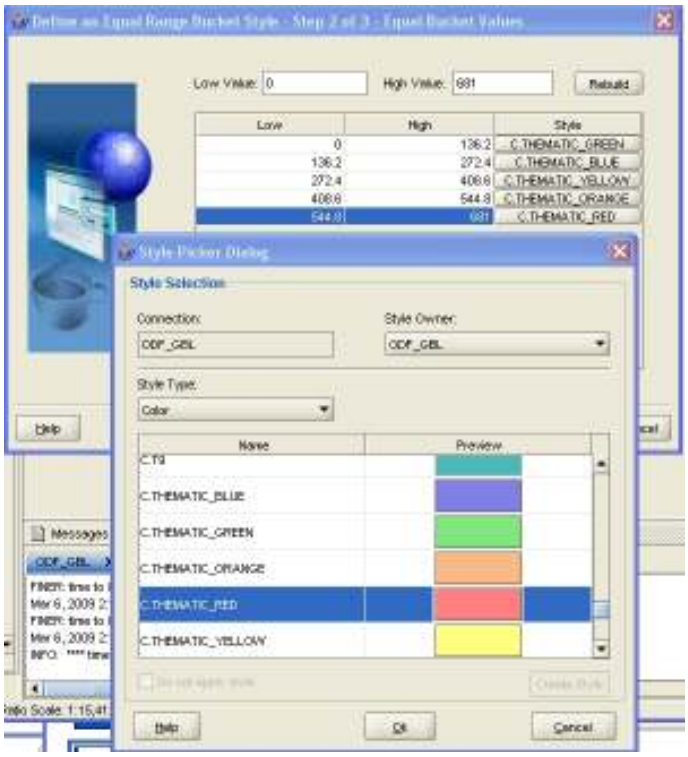
HINT: Use the Advanced Style Wizard to create advanced styles and associate them with a table.

To use the Advanced Style Wizard, click Show Data, navigate to the table, right click and choose Create Advanced Style, then use Equal Ranged Bucket Style

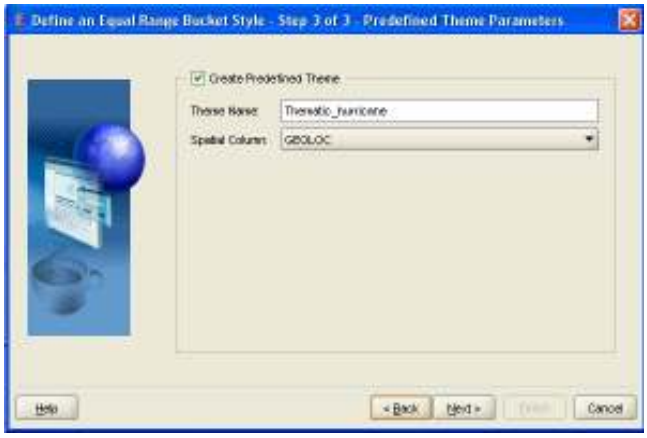


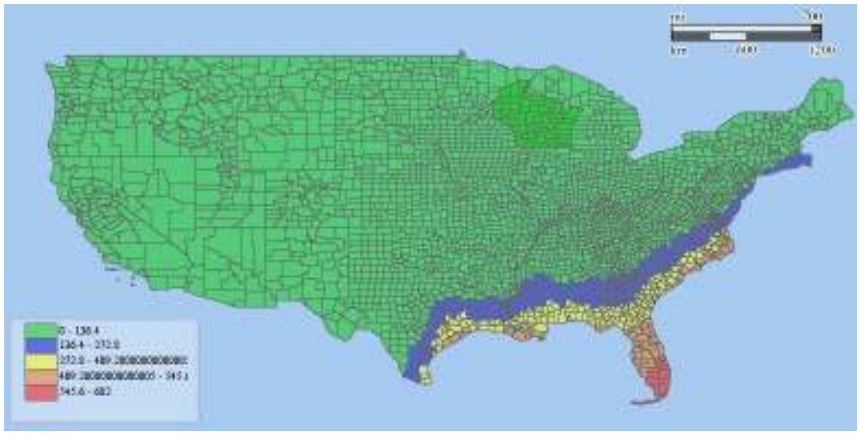
Now, name your advanced style (v.name), choose the column on which to base the style, and the number of buckets (color gradations). Next, assign the color associated with each bucket.



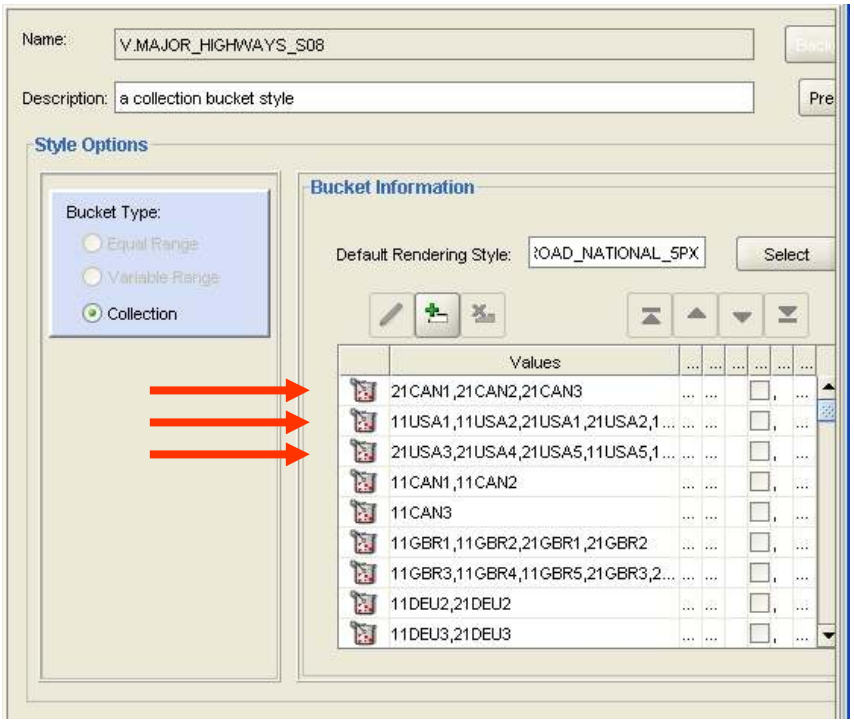


In the next step, the wizard allows you to build a theme (associate the style with the table) by clicking Create Predefined Theme. Choose a name, click Next, then Finish, and you have a theme ready to include in your map.

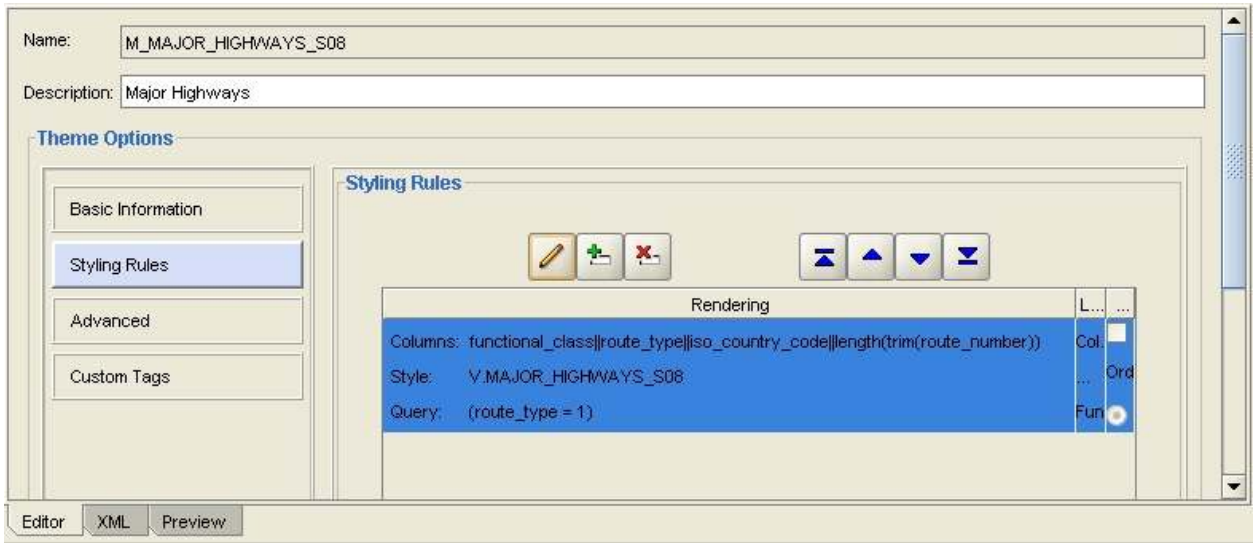




HINT: You can create an advanced style that is based on values of multiple columns in a table by defining the style using a character string of the concatenated columns Use collection bucket style, with values associated with the concatenated data.



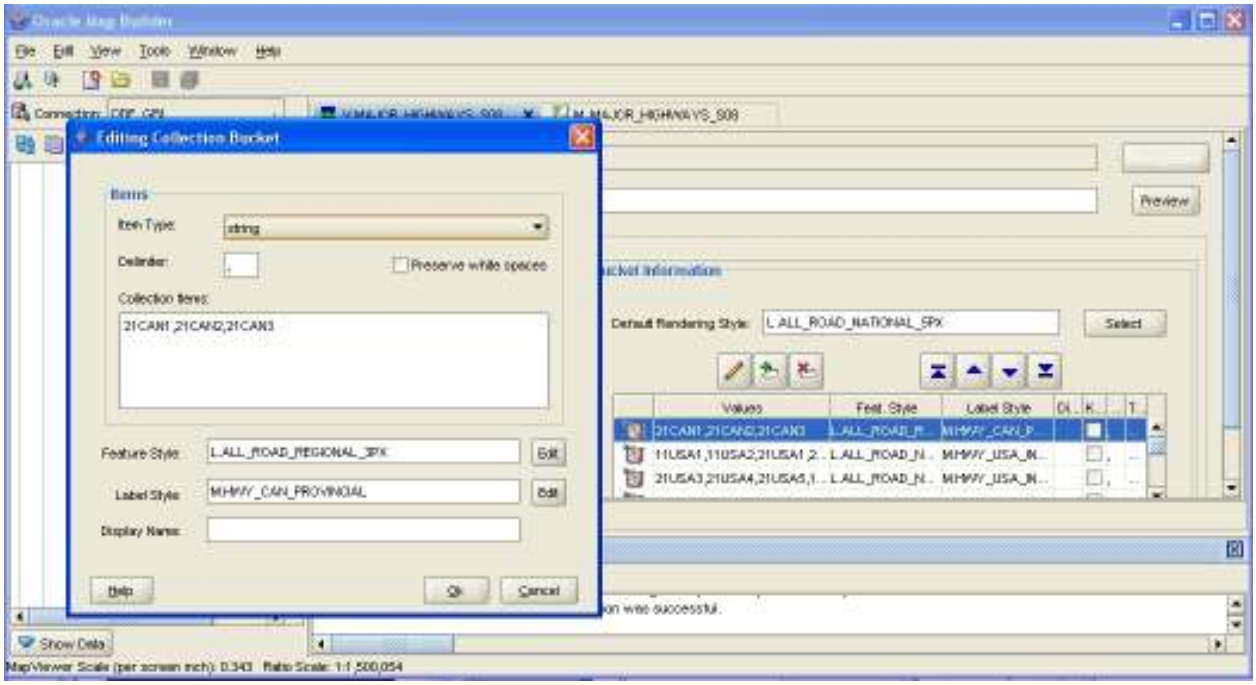
When the *theme* is created, use the concatenated columns in the Columns list:



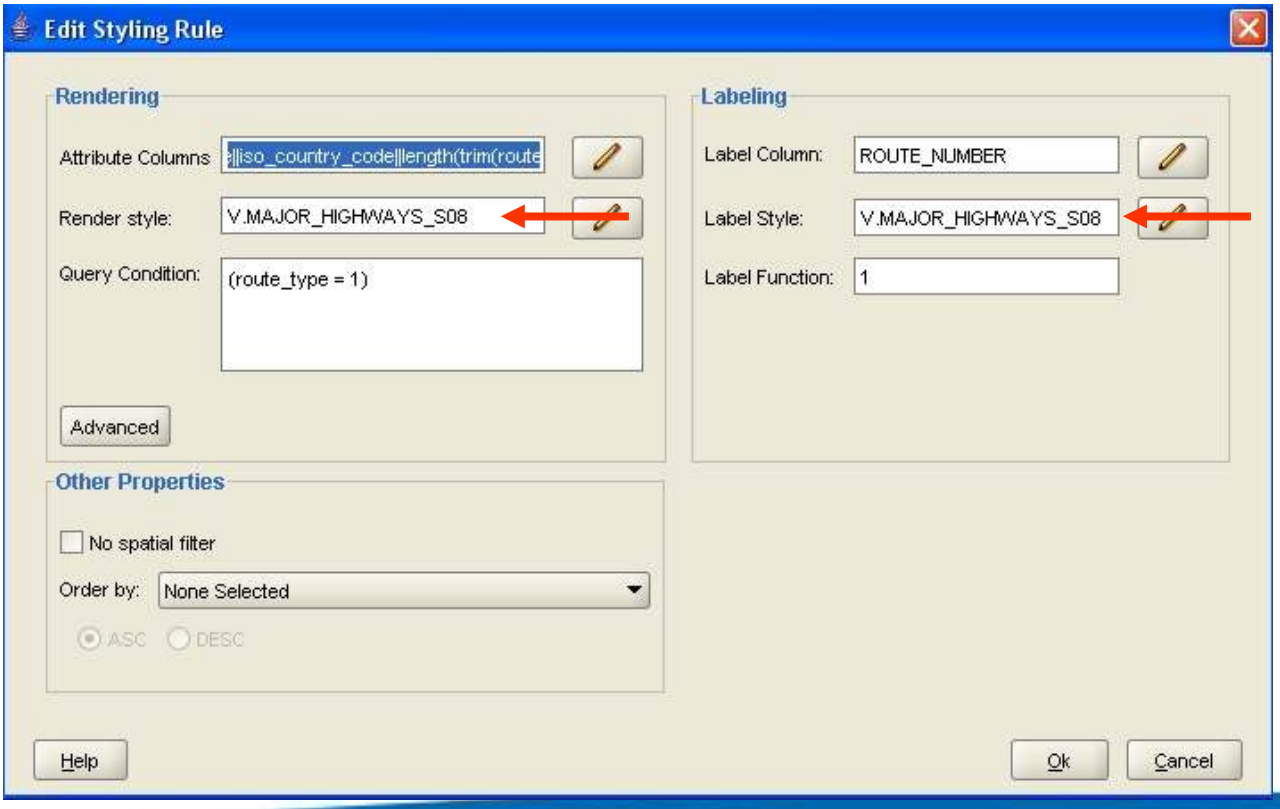
The concatenated columns above are:

```
functional_class|| route_type|| iso_country_code|| length(trim(route_number))
```

HINT: In an advanced style, you can have different drawing and marker styles for each element in the advanced style.



HINT: If your advanced style includes label information, in the theme definition use the advanced style for both Render and Label styles.



NAVTEQ uses the previous three features:

- Concatenated values in advanced styles
- Different marker styles in an advanced style
- Assigning the label to the advanced style

NAVTEQ supports over 78 countries (today)

- Different highways may be styled differently
- Each country has own highway markers
- May be many levels of highway markers (National, US, State)

These features make it possible to create a single maintainable map supporting world-wide highways.

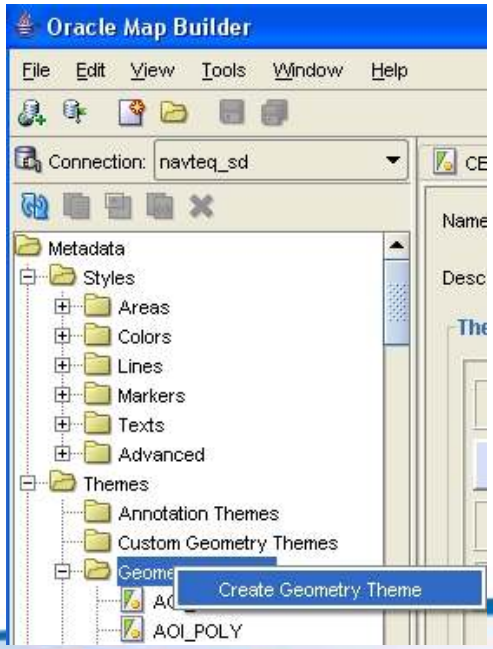
Themes in Map Builder

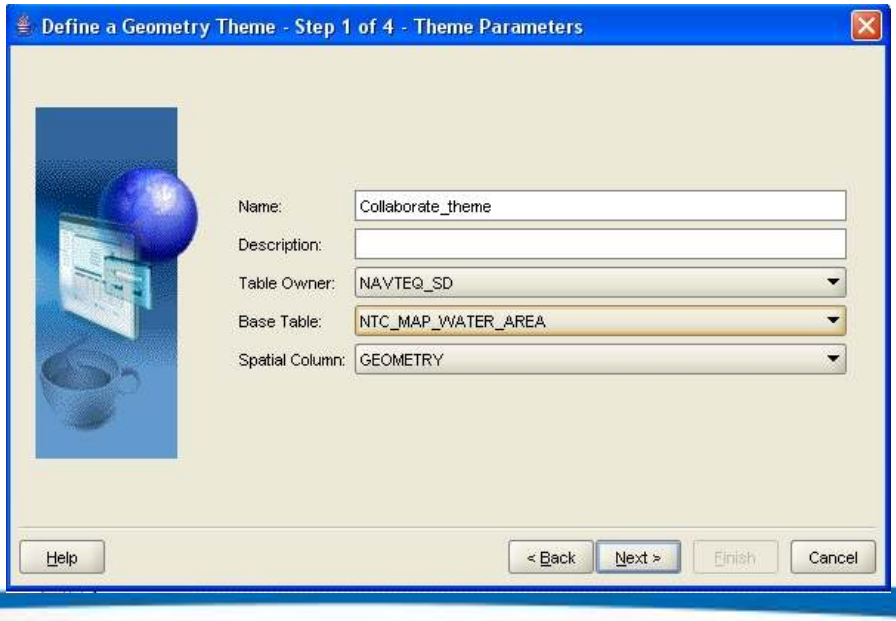
To create a theme in Map Builder, assign style information to features for rendering location data. Color or fill patterns are assigned to area features. Line styles which include attributes such as color, thickness, wing line, and more are assigned to linear features. Markers are assigned to point features, and can also be assigned for labeling purposes (such as highway markers). Text styles are assigned for labeling purposes. Advanced styles are assigned when the feature drawing style is dependent on one or more values in the table.

```
Location data in a table + rendering style = Theme
```

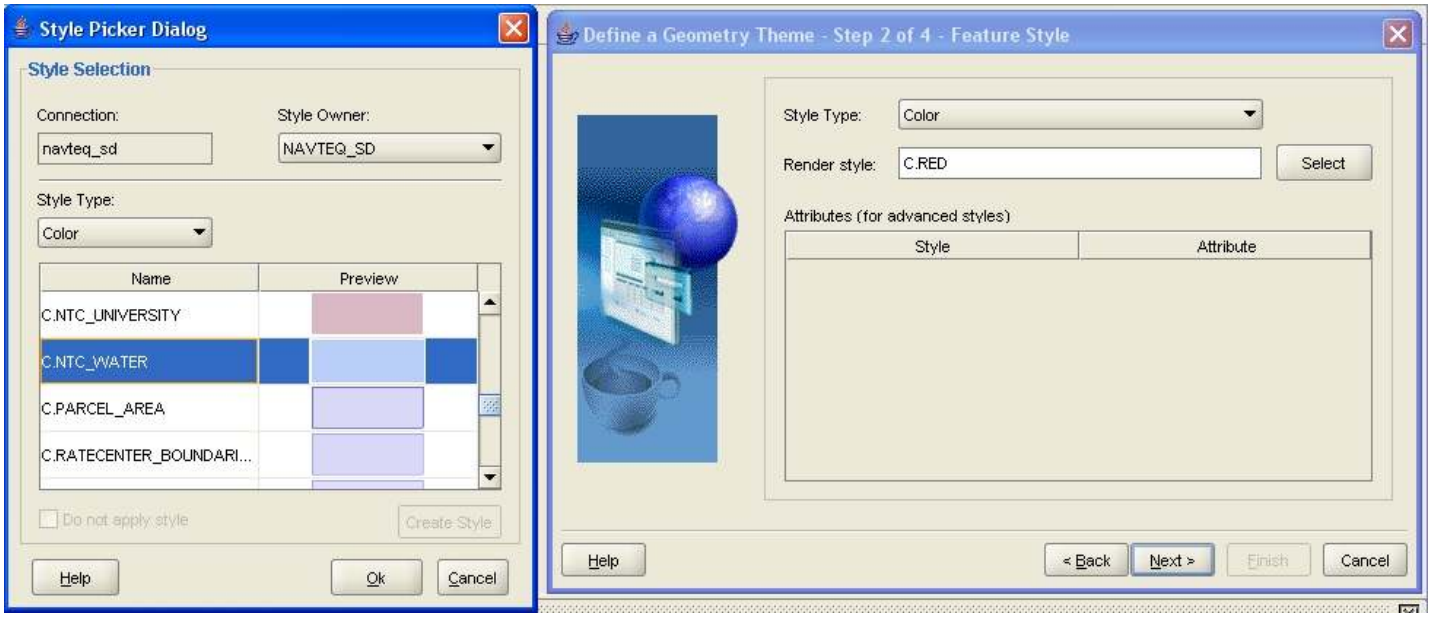
Creating a Theme

To create a theme from scratch, right click the type of theme (Geometry Theme), then click Create Geometry Theme. Next, name the theme, chose a table and an associated geometry column.

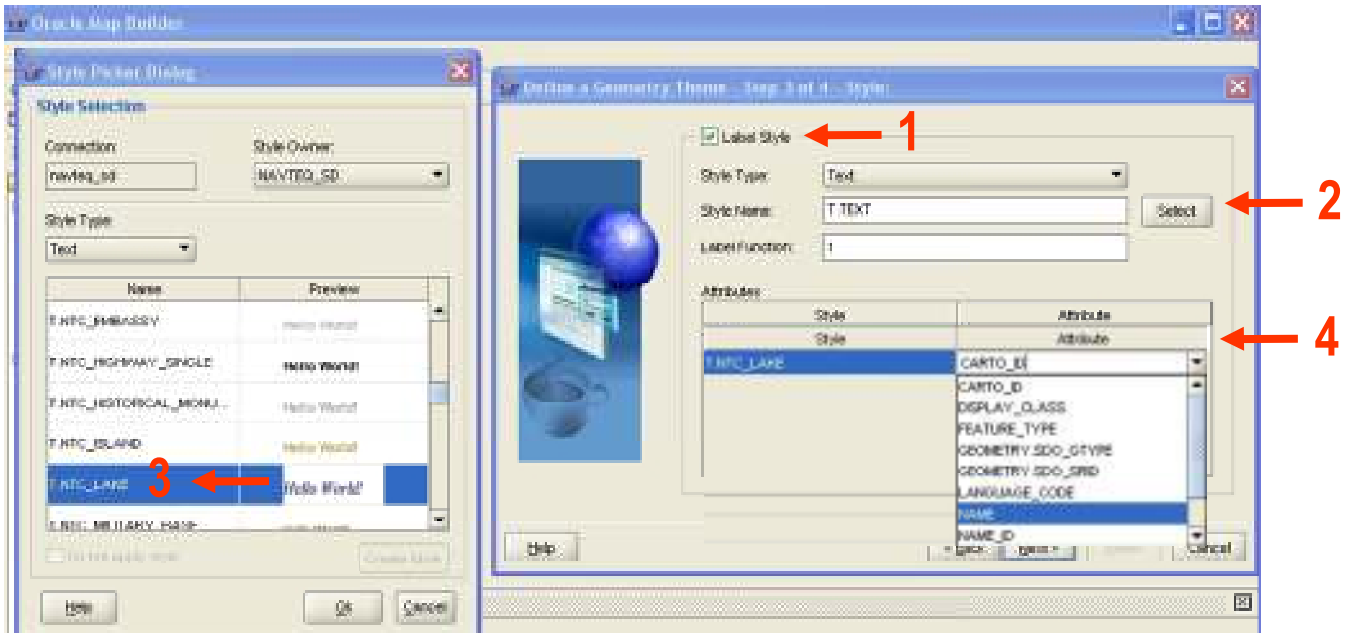




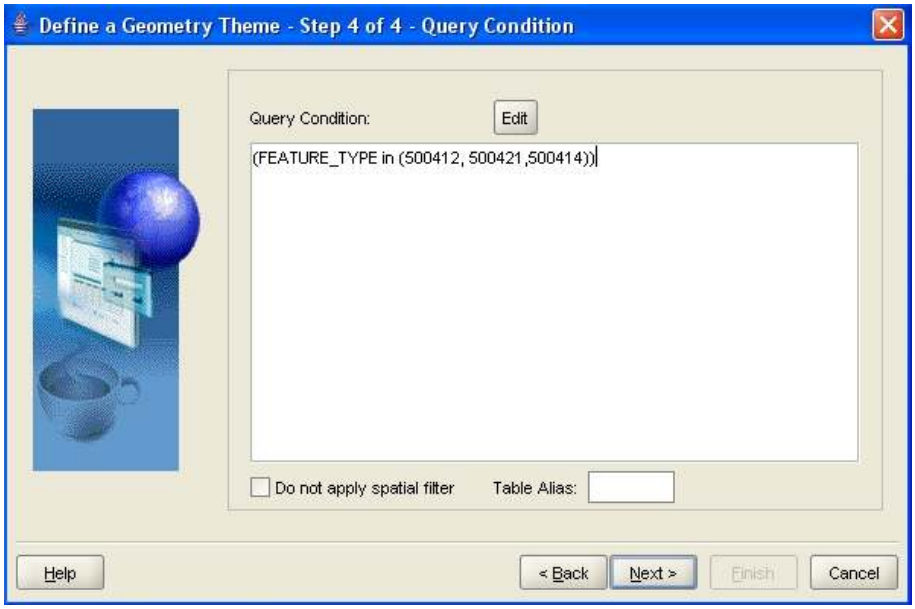
Choose a color style from the colors style picker (given the theme we are creating is based on an area feature). If you assign an Advanced style, in the lower box choose the column whose value defines the style.



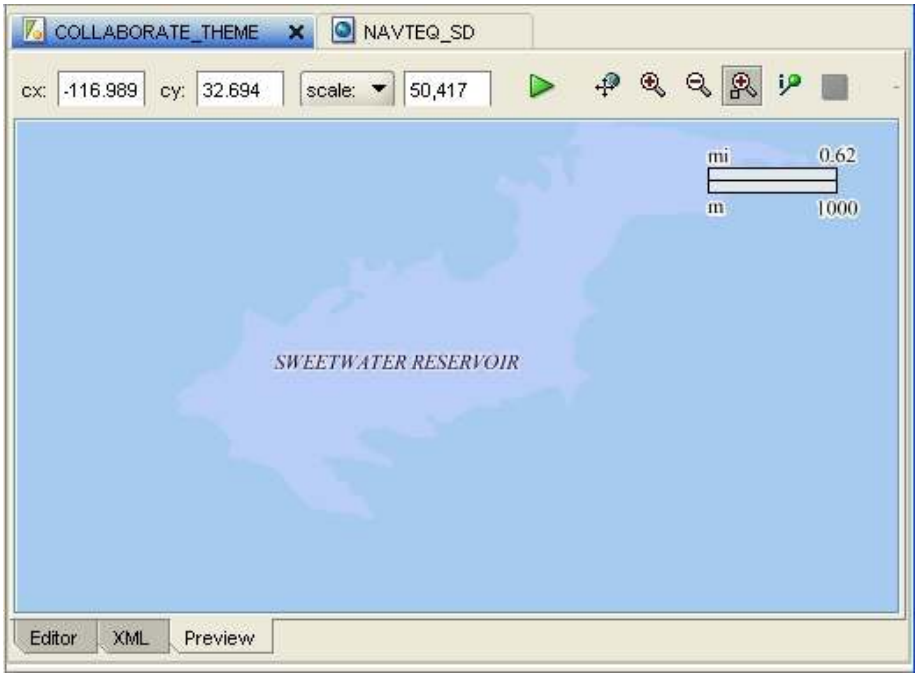
Check the Label Style check box if you want to label this theme, and then choose the text style. Click Attributes to choose the column with the label text.



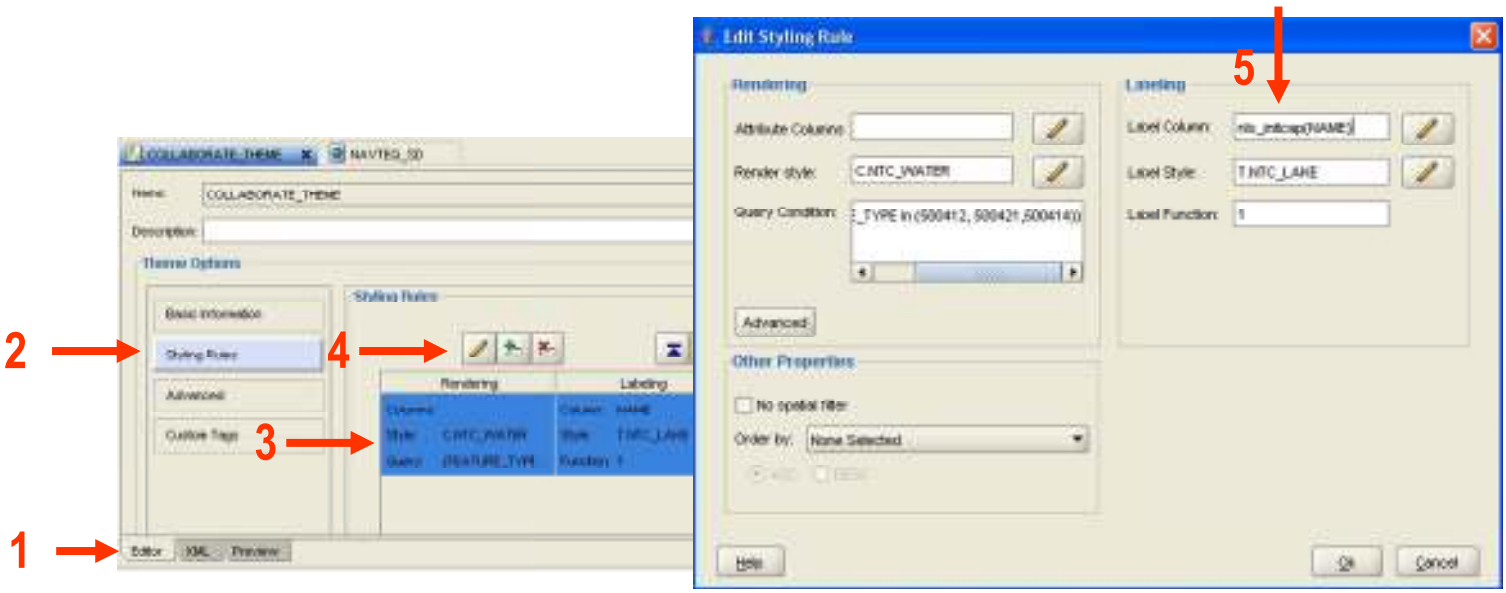
Most often there will be no Query Condition. In this case there is; we choose fresh water features (lakes, rivers, canals), but not bays:



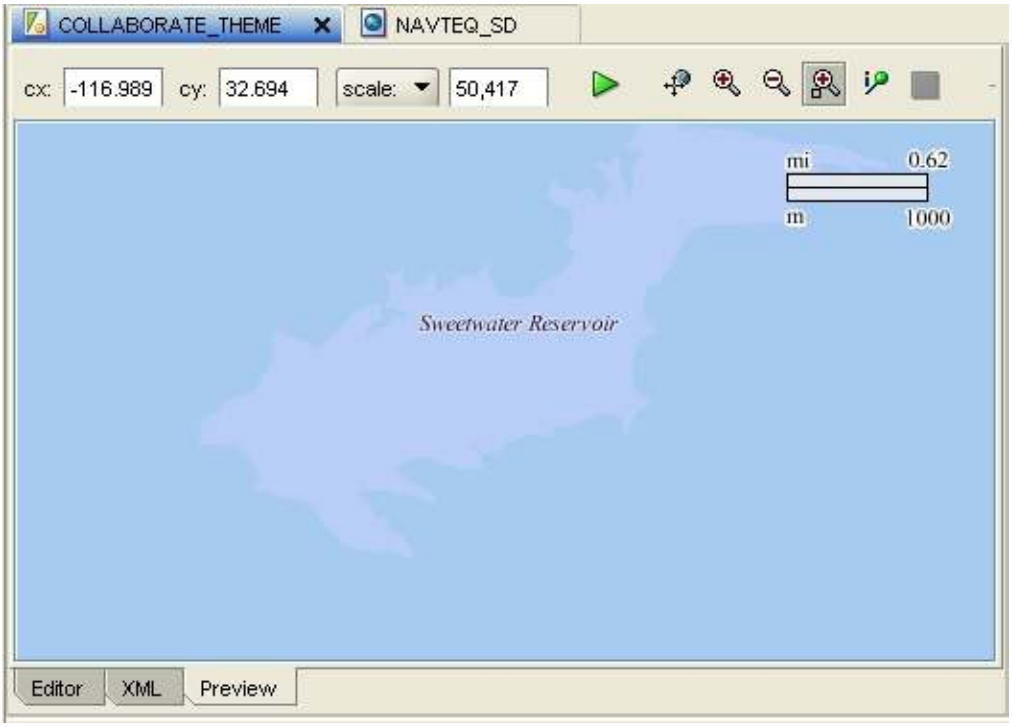
When we are done creating a theme we can preview it:



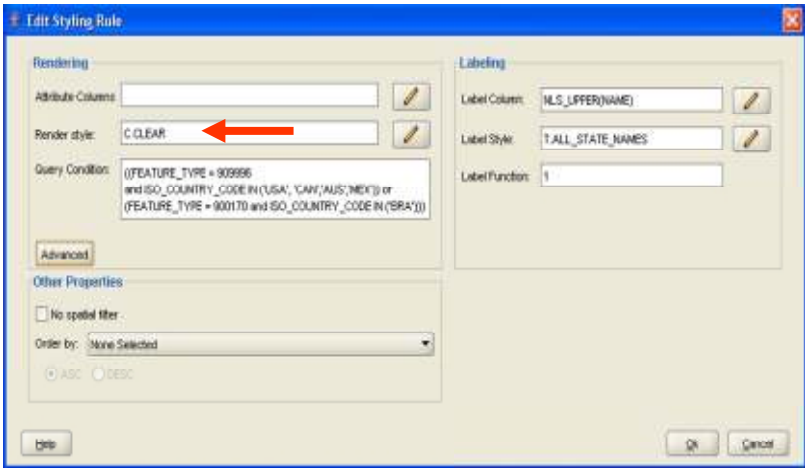
HINT: We can use Oracle SQL functions to define how our labels will look on the map. To force labels to have the first character of each word in the label uppercased, use the NLS_INITCAP function. To do this, click Editor, Styling Rules, select the style, then click edit. Next, add the NLS_INITCAP function around NAME.



Now the preview shows the label in the desired format:



HINT: You can label a theme without having a visual representation of it. For area features use a transparent style with no fill or stroke, then assign that style for rendering.



Note the use of complicated Query Condition above, and the NLS_UPPER function applied to the Label Column.

HINT: In some instances, it may improve performance to alter the fetch size. MapViewer, deployed in an Application Server, fetches data from the database via JDBC. MapViewer's default fetch size from the database is 100 rows. If the network between the database and the application server is slow or restricted, and there is a lot of data fetched in a theme (many hundreds or thousands), this can be increased.

Create an Advanced Style and Associate it with a Theme in Map Builder

Let's look at the block groups table:

```
SQL> desc block_group
 Name
 ----------------------------------------

 ID
 BGP_ID
 FIPSST
 FIPSCO
 FIPSSTCO
 TRACT_ID
 ST_ABRV
 CO_NAME
 ST_NAME
 BGP_LABEL
 GEOMETRY
 TOTAL_POP
 POP_MALE
…
 POP_FEMALE
…
 POP_GE25
 POP_MALE_GE25
 POP_FEMALE_GE25
…
 HOUSEHOLD
 MED_HOUSEHOLD_INCOME
…
```

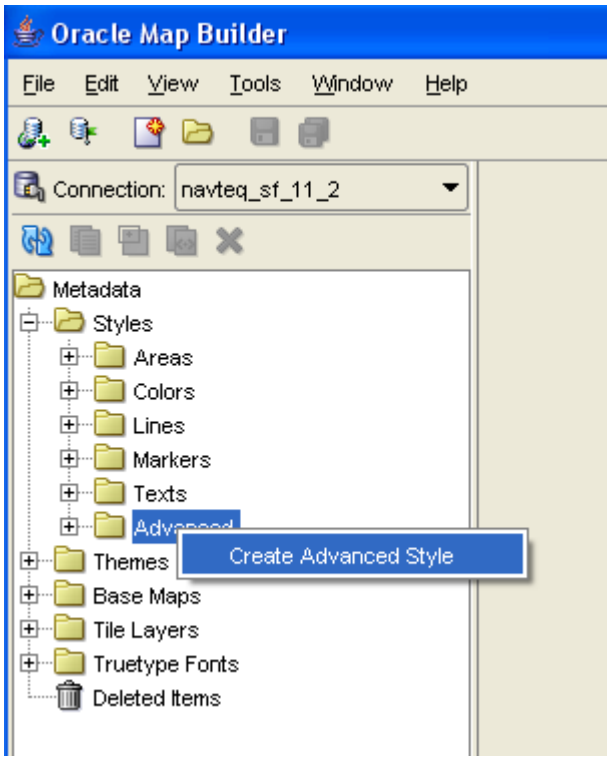
We can thematically style our layer based on many of the attributes. Let's choose median household income (`MED_HOUSEHOLD_INCOME`).

If we want to style the block group boundaries using 5 different colors based on income, and we want each color to represent the same number of block groups, we can determine the minimum and maximum income values for each of the colors (or buckets) by using the following SQL:

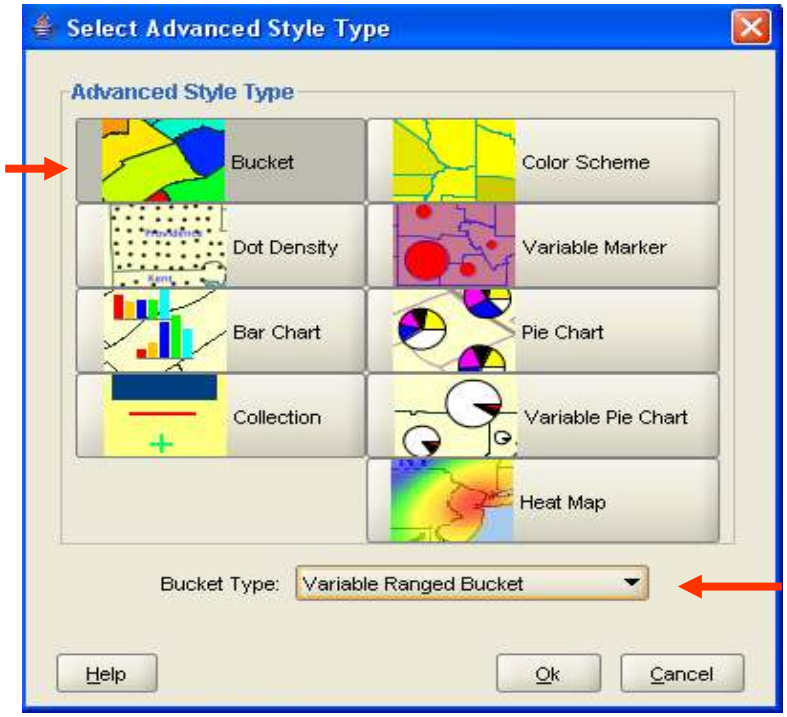
```
select
    bucket,
    min(MED_HOUSEHOLD_INCOME) min_bucket_value,
    max(MED_HOUSEHOLD_INCOME) max_bucket_value,
    count(*)
from ( select MED_HOUSEHOLD_INCOME, ntile(5) over (order by
MED_HOUSEHOLD_INCOME) bucket
       from block_group )
group by bucket
order by bucket;

BUCKET MIN_BUCKET_VALUE MAX_BUCKET_VALUE   COUNT(*)
------ ---------------- ---------------- ----------
     1                0            45112        115
     2            45167            56343        115
     3            56400            65147        115
     4            65227            77287        115
     5            77550           200001        115
```

Note ntile(5) is what calculated the values for each of the 5 buckets.

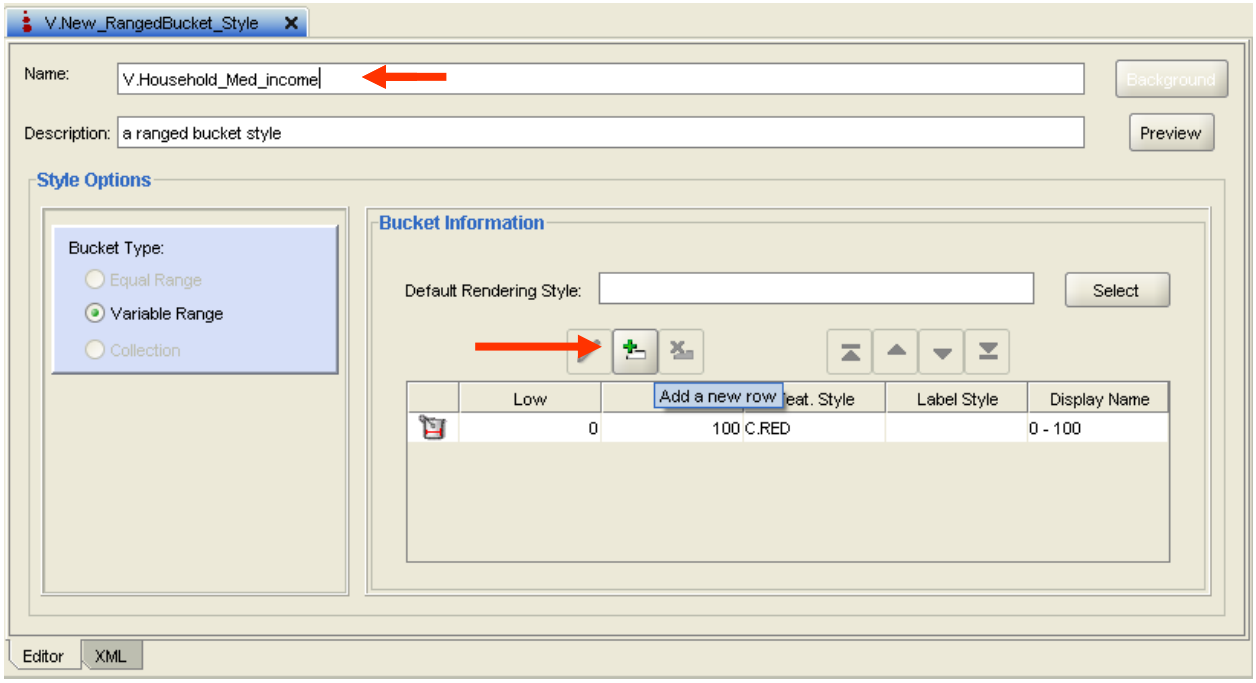
Now let's go to Map Builder and create the Advanced Style by right clicking on Advanced Style and choosing create:



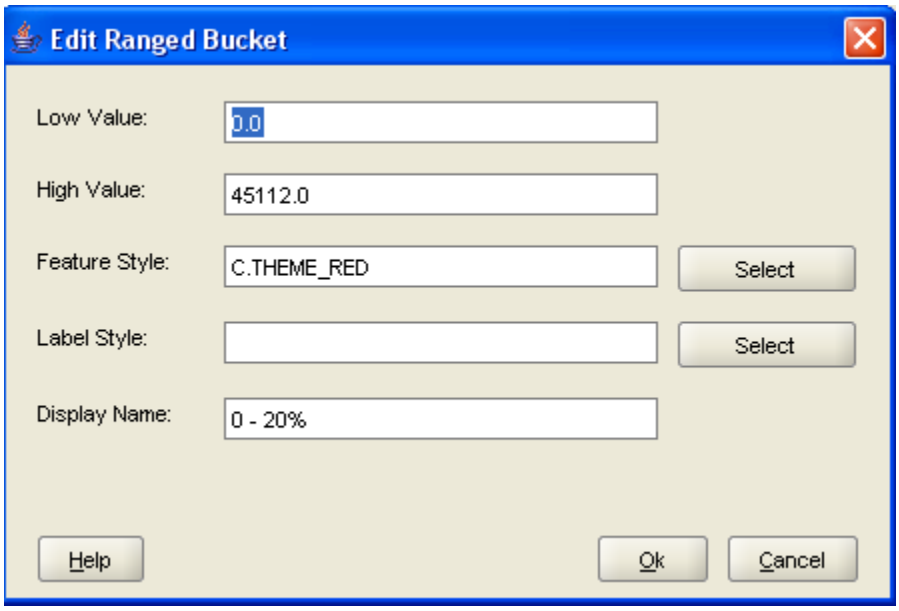
Next, choose a variable ranged bucket style (we choose variable ranged bucket because we want to provide the values to be used for each bucket, rather than let Oracle calculate the values.



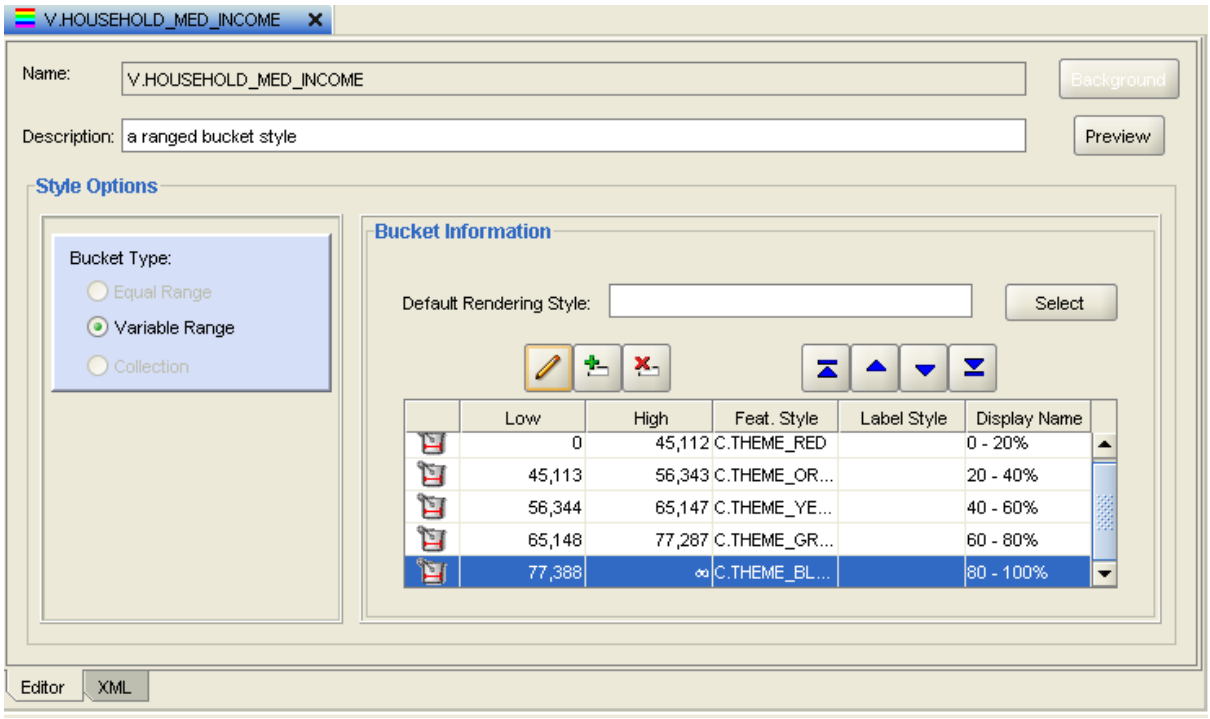
Now we name the Style, and we add 4 more rows by clicking on the green plus sign 4 times



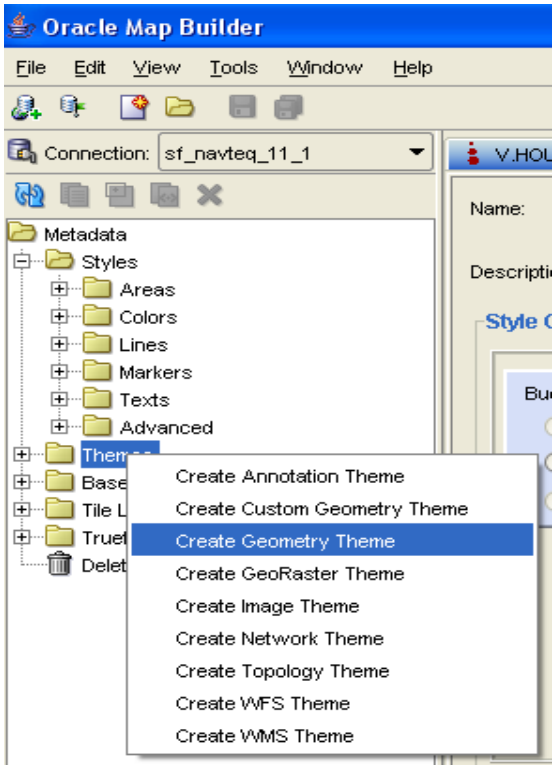
Next, for each of the five rows, highlight the row then click the pencil to edit (next to the + sign). Here is an example of the first bucket values:



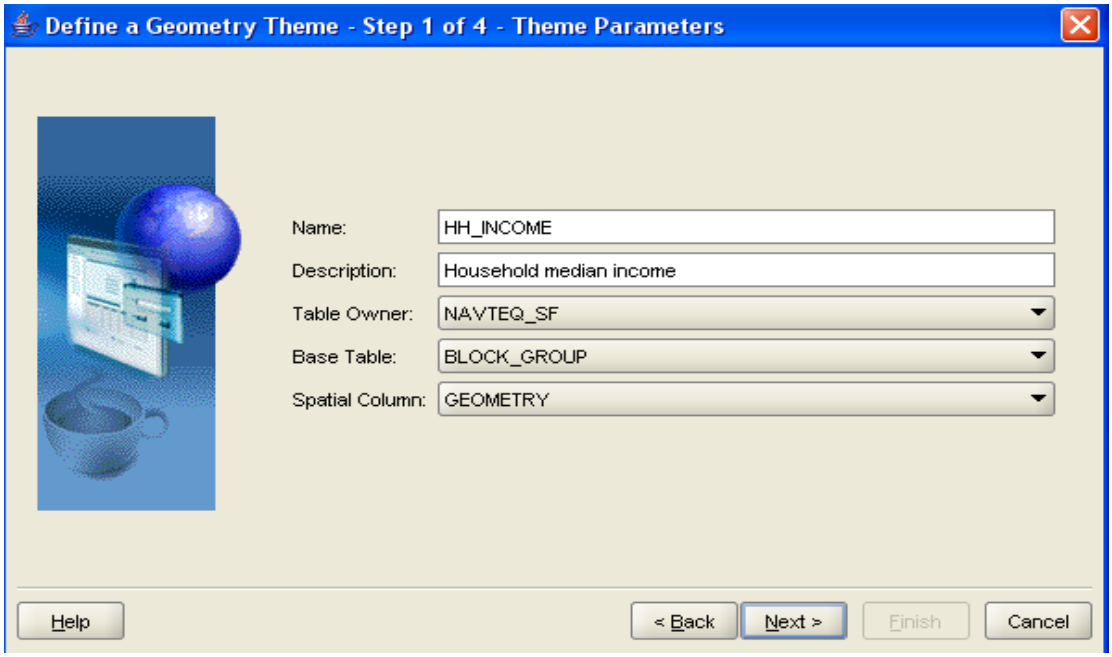
After filling in all values the style that will be associated with the theme looks like this:



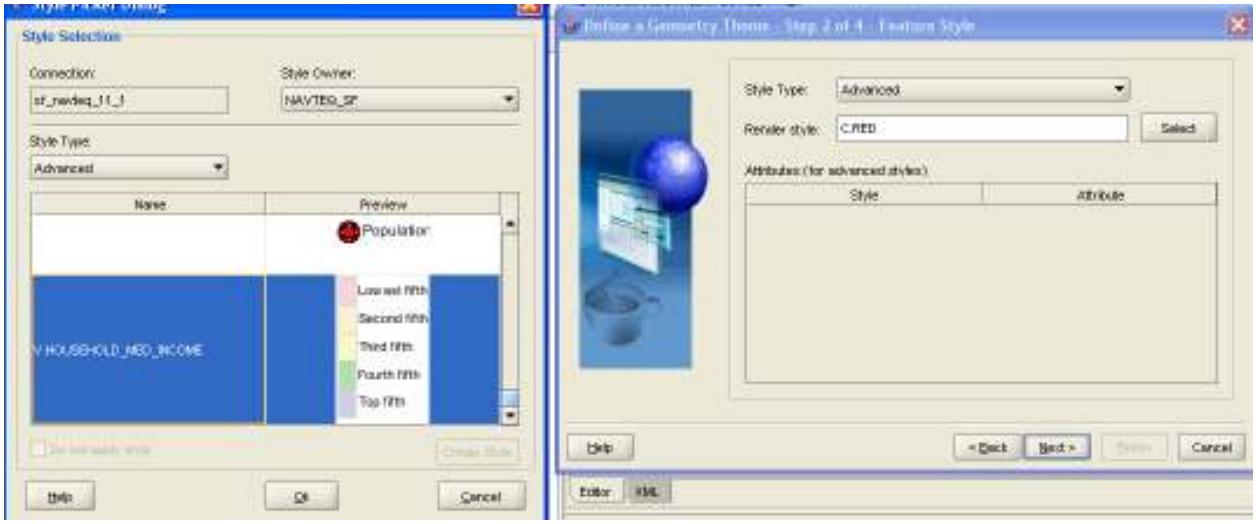
Now that we have the Advanced Style, we can assign the Block Group table to the style to create the theme. To do this, right click Themes, then choose Create Geometry Theme:



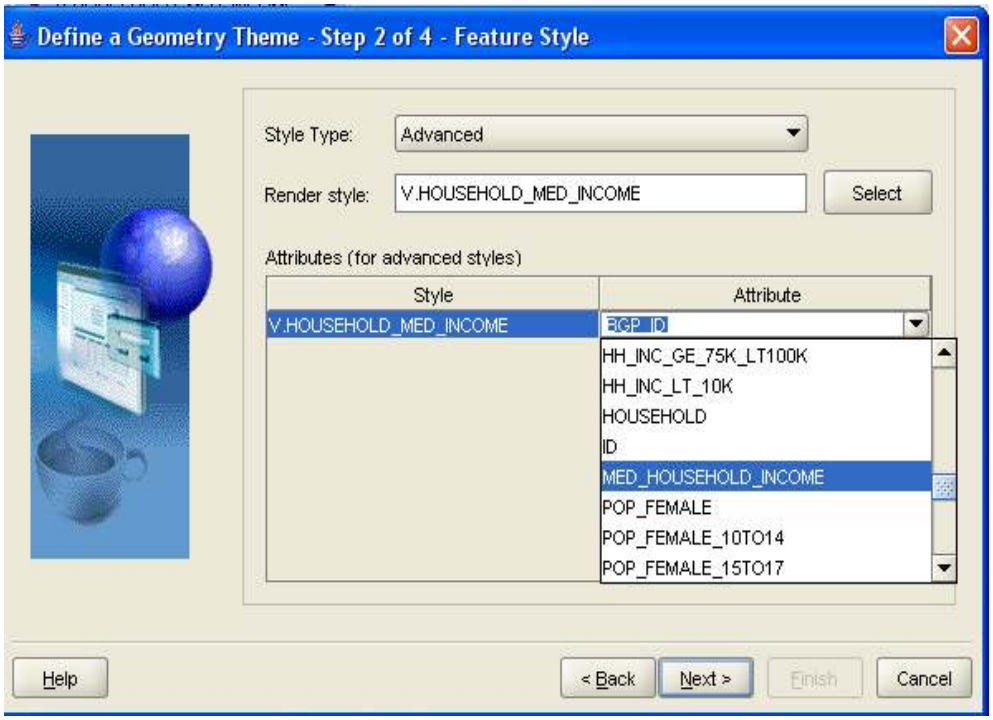
Name the theme and assign BLOCK_GROUP table:



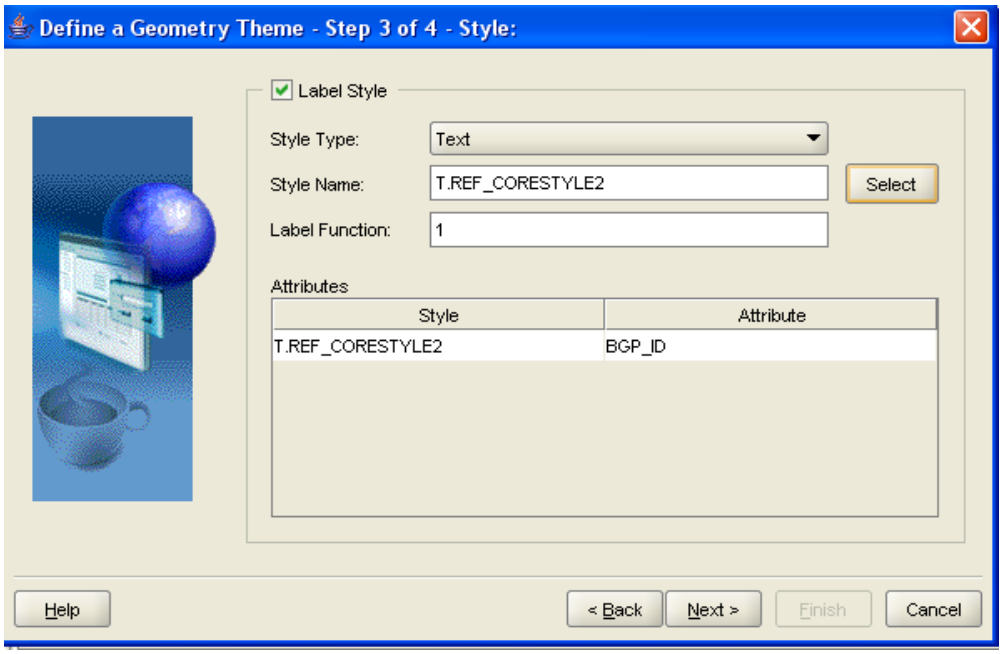
Choose Advanced Style then pick the style we created above:



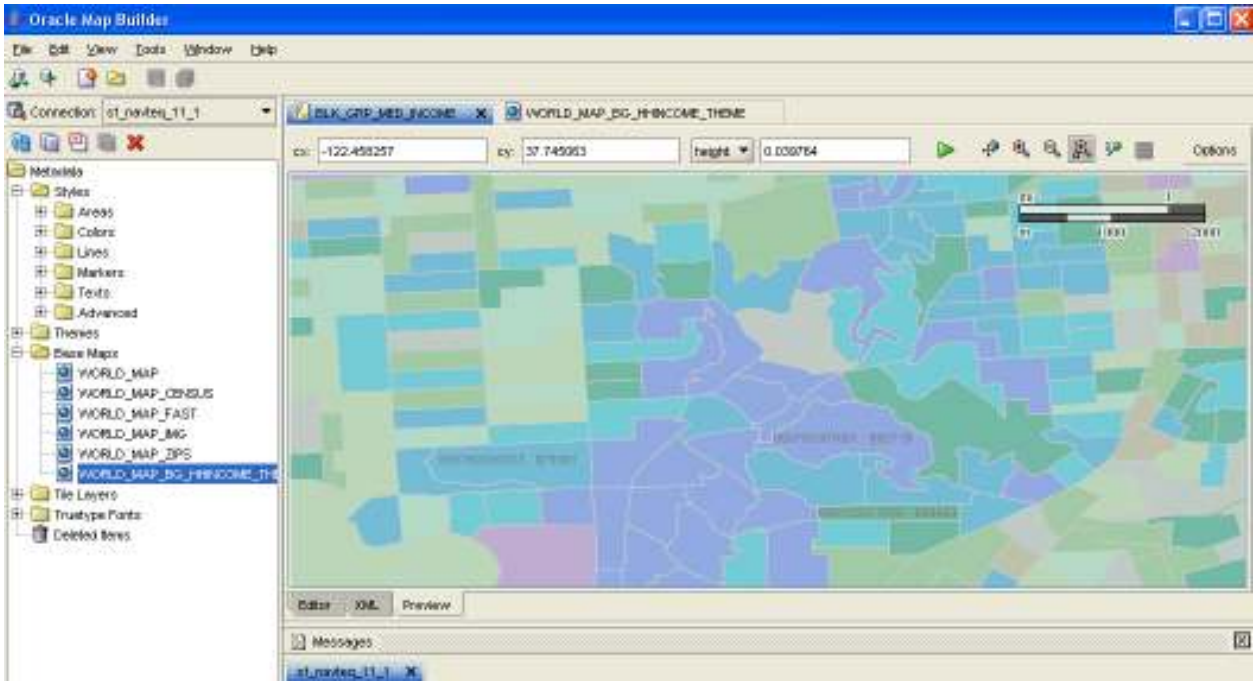
Assign the median household income value as the value to use for the advanced style:



Choose a label style:



Click Next, then Next again and finally Finish. Then we can preview the layer:



Maps in Map Builder

A map is a collection on one or more themes.

When associating a theme with a map, Oracle Map Builder gives the map author a lot of flexibility beyond what has already been discussed. The author can:

- Define the order in which themes render
- Define the map scale at which themes are turned on and off
- Define the map scale at which labels for a theme are turned on and off
- Allow MapViewer to simplify data when rendering
- Force labels to always appear
- Eliminate duplicate labels
- Stop insignificant features from being fetched from the database
- …and much more

Map Creation Hints

When creating map, this order for drawing themes:

Render polygon data first

- Country boundaries, oceans, parks

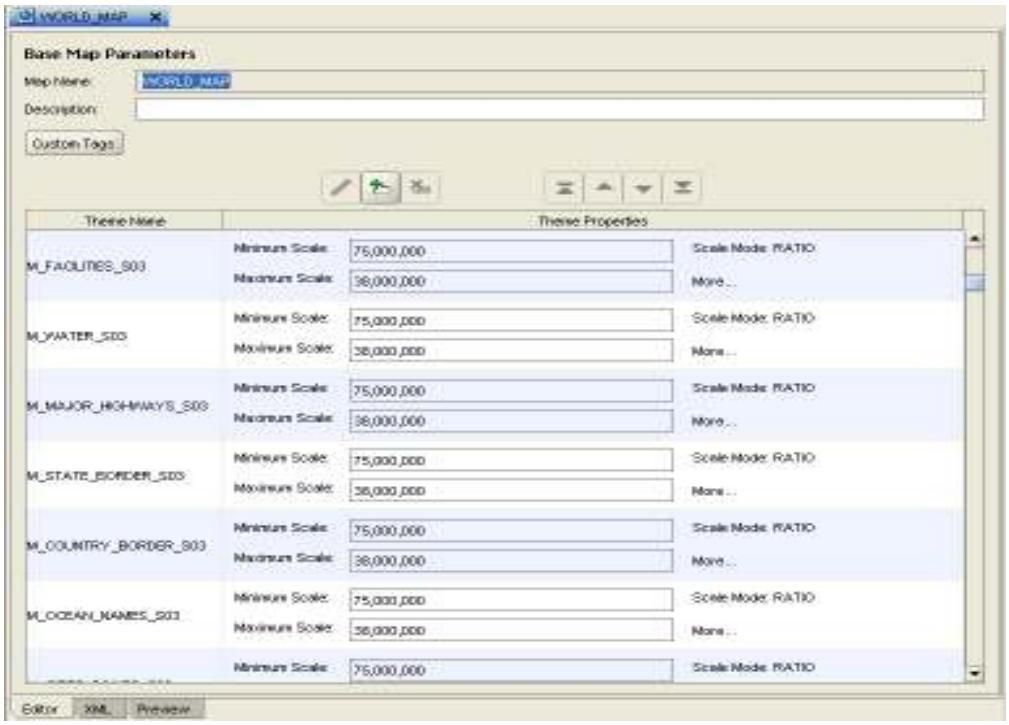
Render linear features next

- Roads, railways, rivers, cartographic lines

Then render point features

The order in which themes are defined in Map Builder is the order in which themes are rendered

- Top to bottom



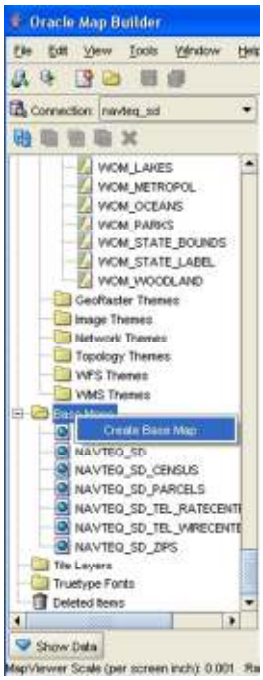
NAVTEQ has a preferred feature rendering order. Importantly: For *area* features render detailed water layers last.

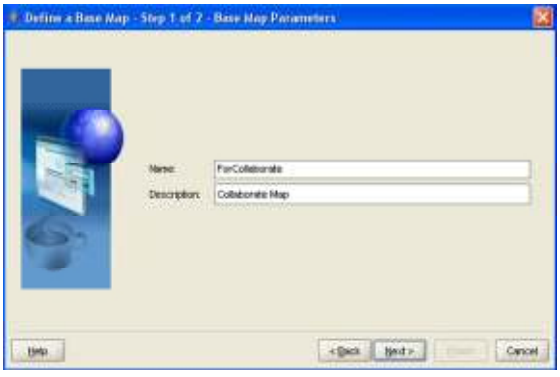
- At this zoom scale(below), the background *water* layer is a simple rectangle (optimized rectangle) covering the entire Earth
- Generalized country layers and parks provide the land features
- Ocean, bay, and lake features provide the final area feature detail
- A point layer provides the label locations for Oceans and Continents

- At this zoom scale (below), the background *land* layer is a simple rectangle (optimized rectangle) covering the entire earth
  - It renders quite fast
- Ocean, bay, lake, and river area features provide all the detail
- Roads, then point features for city/town locations finish the map

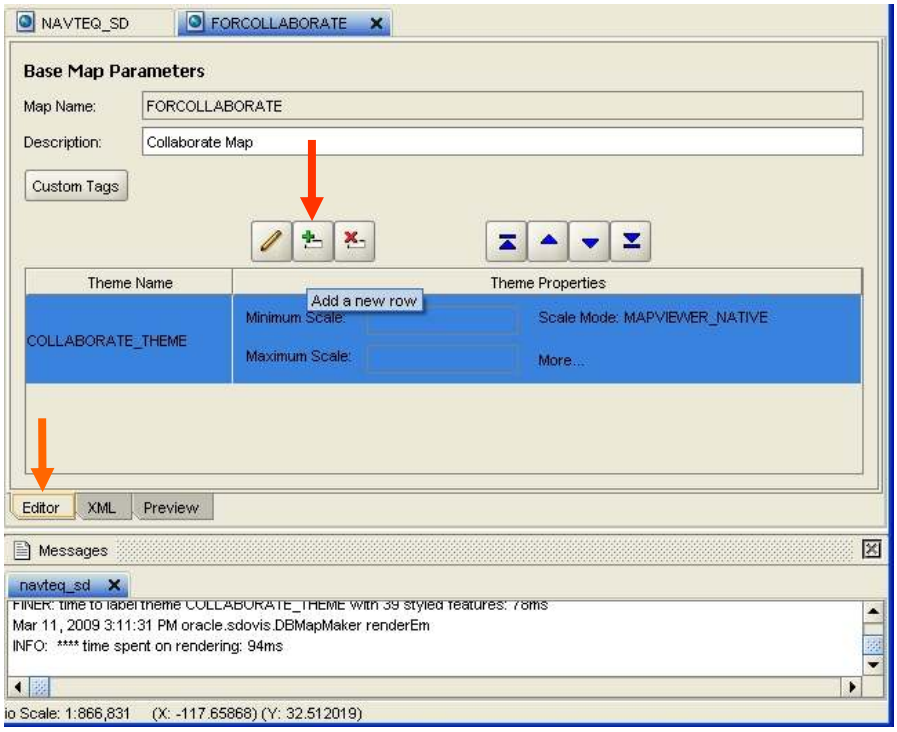
Creating a Map

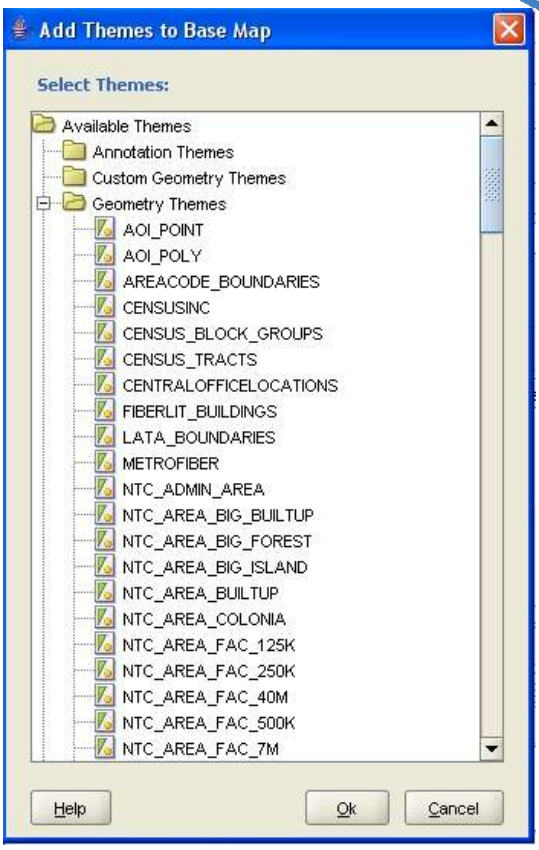
To create a map, right click Base Map in the Navigation Panel on the left, and choose Create Base Map. Next, name the map. Then add one or more themes to the map:



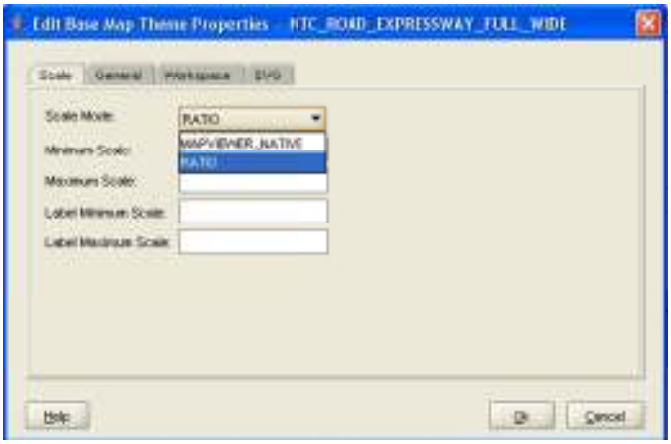


To add themes to the map, click on the Editor tab, then click the plus sign to add a new row. Next, click on a theme to add.

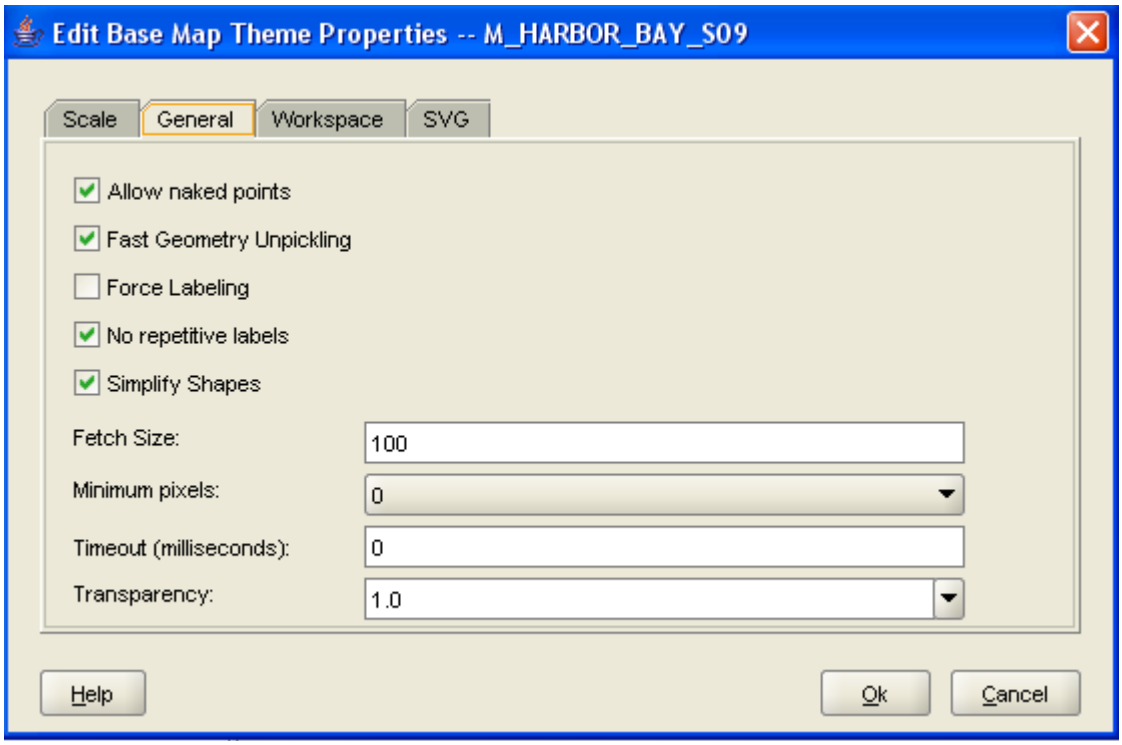




Click on the theme you just added, then click the pencil (to edit). In the scale section of the properties tab, change scale mode from MAPVIEWER_NATIVE to RATIO. The scale is the denominator in map scale, i.e. 1/200000 is 200000 (1 inch on the screen = 200000 inches on the Earth). The minimum scale is the scale at which the theme turns on when zooming in. The maximum scale is the scale at which the theme turns off when further zooming in. The minimum scale will always be greater than maximum scale. In MapViewer, empty scales mean infinity (minimum scale) and 0 (maximum scale).



Click the general properties tab:

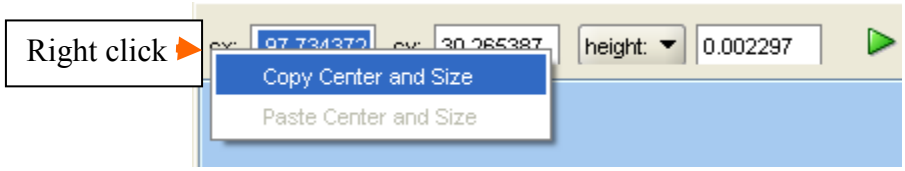


The definitions of the items in this screen include:
`Allow naked points` – Points can be rendered even if the label or marker they are associated with cannot be rendered (due to label or marker conflict)
`Fast Geometry Unpickling` – Uses special MapViewer code to unpickle data. Incurs very slight loss of precision, seldom a problem for read applications
`Force labeling` – MapViewer will always label the theme when map rendering
`No repetitive labels` – Ensures label will only be rendered once in a theme
`Simplify Shapes` – MapViewer will simplify data to improve rendering speed
`Fetch size` – The number of rows to fetch from the database via JDBC, as discussed previously
`Minimum pixels` – MapViewer asks the database not to return data that will not render over at least the given number of display pixels. *This uses the index minimum bounding rectangle, and is very powerful*
`Timeout` – Used for WMS/WFS
`Transparency` – set the transparency for the theme here if desired (can also set transparency in the style definition)

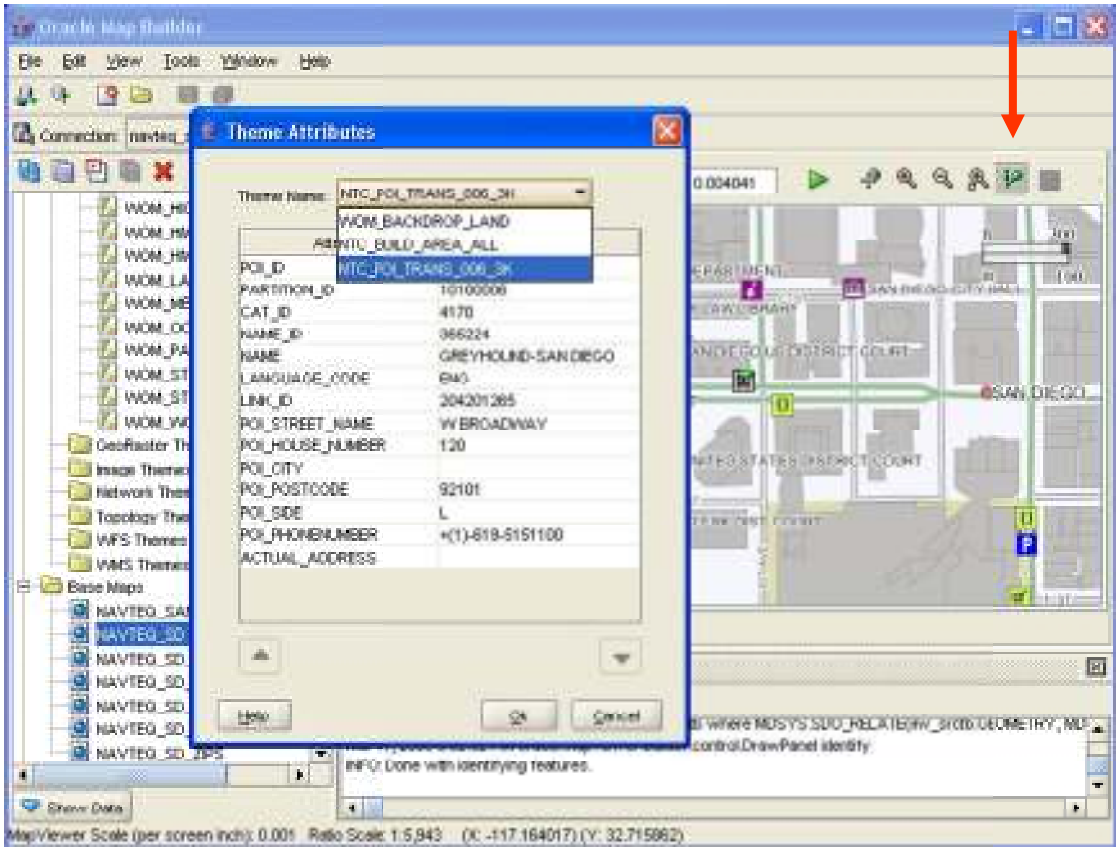
HINT: Use cartographic lines for land boundaries between countries and states. This helps to create an aesthetically pleasing map. There is no line representation where land and water meet. In the case below, the country is rendered, and the states labeled, but not drawn (see Label a Theme Without Rendering). Note there are no outlines around the country or states. The cartographic state lines appear as dashed lines between the states. The cartographic country lines appear as the smooth, light gray lines between the US and Canada and the US and Mexico.

Other Hints and Tips

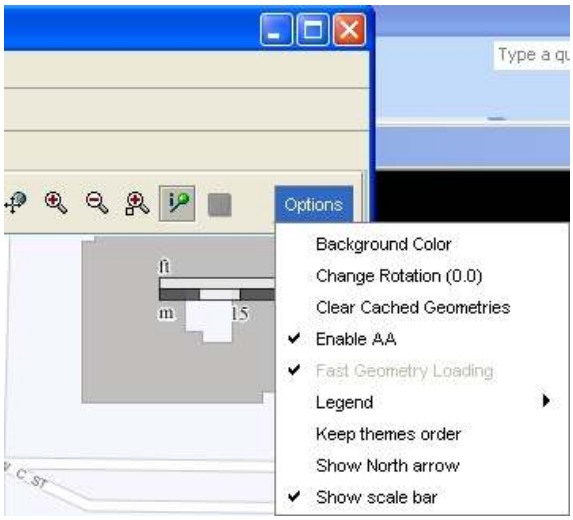
You can copy and paste the Center X, Y, and Height/Scale in the preview window of a theme or map:



You can see all of the attributes associated with a feature by clicking the "Identify" pin:

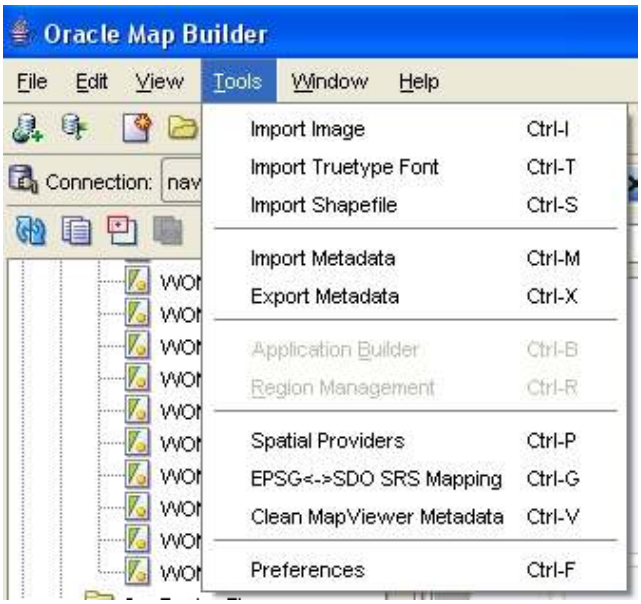


The options tab allows you to configure/alter Map Builder:



Background Color – allows you to change the map background
Change Rotation – allows you to rotate the map (degrees)
Clear Cached Geometries – clears the Map Builder geometry cache
Enable AA – Anti-Aliasing
Legend – allows you to size and place the legend for advanced styles
Keep themes order – Use as defined
Show North arrow – Adds a North Arrow
Show scale bar – Adds a scale bar

There are a set of tools available in Map Builder:



The three most common ones are:

- Import Shapefile – Imports a shapefile and optionally creates a theme on the new table
- Import Metadata – Imports a previously created Map Builder export
- Export Metadata – Creates a Map Builder map metadata file contains one or more of the set of styles, themes, and maps

For More Information

For more information about the concepts or details in this paper, or to find out more about NAVTEQ data and Oracle, contact: **Enterprise@NAVTEQ.com** and mention Oracle.