

Oracle® Utilities Work and Asset Management

MapBuilder Tutorial

E26186-02

October 2012

Oracle® Utilities Work and Asset Management MapBuilder Tutorial

E26186-02

Copyright © Copyright ©1986, 2012, Oracle and/or its affiliates. All rights reserved.
Oracle and/or its affiliates. All rights reserved.

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

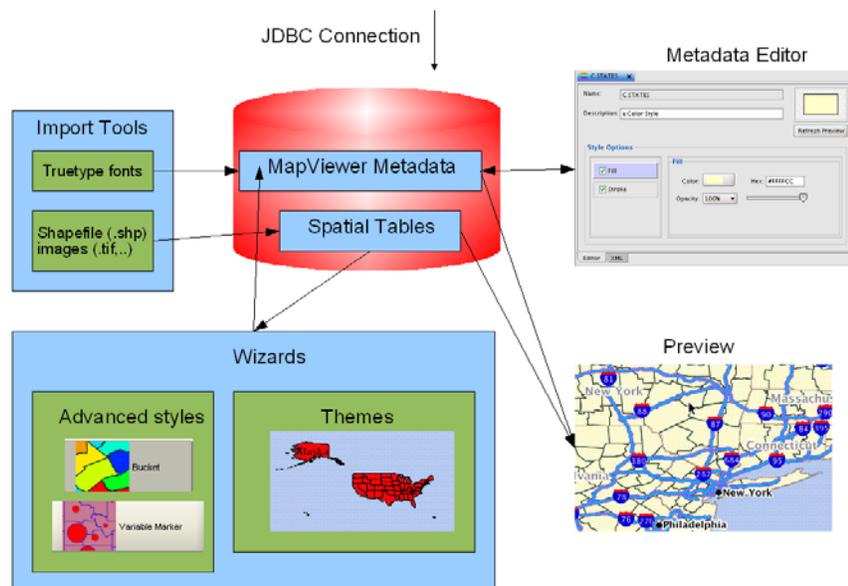
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Oracle GIS Viewer MapBuilder Tutorial

Oracle Map Builder is a standalone application that lets you create and manage the Oracle Application Server MapViewer mapping metadata (styles, themes, and base maps) in the database. For example, you can use this tool to create a style, theme or base map or modify its definition. Besides handling the metadata, the tool provides interfaces to preview the metadata (for example, to see how a line style will appear on a map or how is the theme rendered based on its styling rules), and display the original spatial information from the spatial tables without creating MapViewer metadata. A set of wizards is also available to create metadata based on database table contents. The following diagram illustrates the main tasks that can be performed in MapBuilder. Existing metadata can be edited and previewed, existing spatial tables can be previewed directly, spatial table contents can be used in wizards to generate MapViewer metadata (advanced styles and themes), and import tools can be used to store metadata (TrueType fonts as style information) and also to generate new spatial tables (shapefiles generate tables with SDO_GEOMETRY type, image files generate tables with SDO_GEORASTER type).

Oracle MapBuilder Diagram



This tutorial describes the steps to create MapViewer metadata for some spatial tables containing columns of type SDO_GEOMETRY. The basic steps include the following:

- Import the sample data

- Create styles
- Create themes that use the styles
- Define a base map that can be used in an Oracle Map application.

Running Oracle Map Builder

Oracle Map Builder is shipped as a JAR file (`mapbuilder.jar`). You can run it as a standalone Java application in a Java Development Kit (J2SE SDK) 1.5 or later environment, as follows:

```
% java -jar mapbuilder.jar [Options]
```

Options:

`-help`

displays information about the available options.

`-config <config-file>`

specifies the location of file containing Oracle Map Builder configuration and preference information. If you do not specify this option, Oracle Map Builder looks for a file named `oasmapbuilder.xml` in your Java home directory.

`-noconnect`

disables the automatic opening of previously created database connections. If you specify `-noconnect`, data sources specified in the `oasmapbuilder.xml` preferences file or the file specified with the `-config` option will not be connected; and to use any objects in these data sources, you will need to connect to the data sources. If you do not specify `-noconnect`, MapBuilder will try to connect to the specified data sources in the `oasmapbuilder.xml` preferences file or the file specified with the `-config` option (if the connection is not available a timeout may happen).

`-cache <cache-size>`

specifies the size of the in-memory geometry cache.

For example: `-cache 64M`

The Configuration File

MapBuilder creates a configuration and preferences file named `oasmapbuilder.xml` in the user's home Java directory (defined by the Java System property `user.home`). This file can contain information about the product version, Web proxy settings, custom mappings between EPSG and Oracle spatial reference codes, spatial data providers, and database connections. This file is updated whenever any of its parameter values are changed during application use. If the file becomes corrupted for any reason, you can delete it, and Map Builder will create a new one the next time it starts. This is an example of the configuration file:

```
<?xml version="1.0" standalone="yes"?>
<oas_mapbuilder_preferences>
  <prod_version id="11.0.0.0.0 - Preview" />
  <logging level="finest" />
  <srs_mapping>
    <srs epsg="1111" sdo="82362" />
  </srs_mapping>
  <spatial_provider id="xmlProvider"

class="spatialprovider.samples.CustomSpatialProviderS
ample"

jarfile="/lbs/demo/spatialprovider/spatialprovider.ja
r"
>
```

```

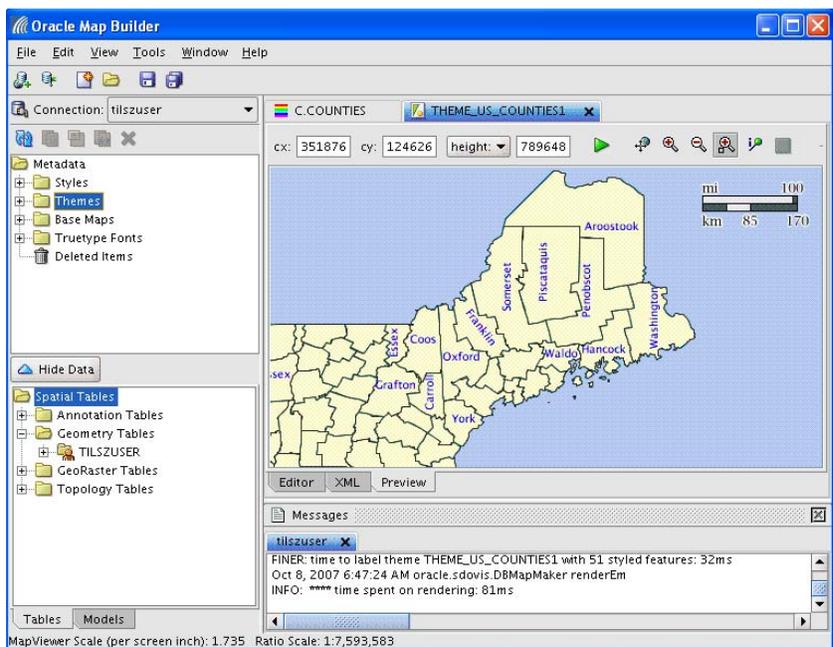
<parameters>
  <parameter name="datadir" value="" />
</parameters>
</spatial_provider>
<db_connection name="mb_tutorial"
  host="10.149.251.219"
  user="mb_tutorial"

  passwd="oDdB8JELY3JMngn5gBawDWTvJVjMyviwwWg0Qt/tRaQ="
  sid="orcl"
  port="1521"
/>
</oas_mapbuilder_preferences>

```

The User Interface

This section briefly describes the various elements of the MapBuilder tool's UI (Figure 2). These are the Menus, the toolbar, the database connection drop-down, the metadata and data navigator trees on the left, and editor/preview and message panels on the right.



Menus

The Menu entries at the top include:

File: contains commands to create/edit database connections and metadata.

Edit: contains some edit commands such as undo/copy/paste that are reserved for future use. The only operation available is the Delete option to erase metadata.

View: contains commands to preview metadata and to refresh the connection contents.

Tools: contains submenus and commands to import external data (images such as tif files, or vector data such as Shapefiles), operations to import/export MapViewer metadata (styles, themes, base maps), operation to register external spatial providers to render data in native format, and operations to update some configuration parameters on the preferences file.

Window: contains submenus and commands to show/hide some user interface components.

Help: submenus to access MapBuilder online help and to get the version information.

Toolbar

The Toolbar entries under the menu options are used for creating, loading or removing database connections; creating or opening metadata information; and to save edited metadata.

Connection List

The Connection drop-down lets you establish a database connection, use a different connection, create a new connection, or edit an existing connection.

Metadata Navigator

The Metadata Navigator tree is a hierarchical display of MapViewer metadata objects (styles, themes, base maps, truetype fonts) read from the currently active database connection. To select an object, expand the appropriate tree node or nodes, then click the object. The nodes refer to:

Styles: style information stored in the USER_SDO_STYLES view.

Themes: theme information stored in the USER_SDO_THEMES view.

Base Maps: base map information stored in the USER_SDO_MAPS view.

Truetype Fonts: true type fonts, stored in the USER_SDO_STYLES view, which can be used as marker styles.

Metadata Icons

Above the **Metadata Navigator** tree there are a set of **icons** related to this tree:

Refresh: refreshes the contents of the navigator tree by loading the current database content for selected connection.

Duplicate: opens a dialog to duplicate the metadata definition.

Preview: opens a preview window on the right side of the application to display the theme or base map contents.

Copy XML: this is currently not supported.

Delete: removes the metadata definition from the database.

Spatial Data Navigators

The **Spatial Tables Navigator** shows the current spatial tables that are accessible for the selected connection. The nodes are organized by Oracle Spatial object types (ST_ANNOTATION_TEXT for annotation text tables, SDO_GEOMETRY for geometry tables, SDO_GEOASTER for GeoRaster tables and SDO_TOPO_GEOMETRY for topology tables). The spatial tables must be registered in corresponding metadata views (e.g. USER_SDO_GEOM_METADATA for geometry tables) in order to be shown on the tree.

The **Spatial Models Navigator** tree shows the current network and topology models that are accessible for the selected connection.

Editor Pages

The right side of the application may contain **Editor** pages. Users can edit/preview the metadata contents for styles, themes and base maps.

Message Panel

Below the **Editor** page there is a **Message** window that displays log information for actions performed on the current connection.

Load Spatial Data

A database user must be created and spatial tables must be loaded before creating MapViewer metadata. This section assumes that an Oracle database instance (10gR2 or 11g) has been installed on some server and that the `mb_tutorial.dmp` file has also been copied to a directory on that system.

Create User

1. **Start SQLPlus as an administrator:**

```
sqlplus system/<password_4_system>
```

2. **Create a new user *mb_tutorial*:**

```
SQL> create user mb_tutorial identified by mb_
tutorial;
```

3. **Grant some privileges to this new user:**

```
SQL> grant connect, resource, create view to mb_
tutorial;
```

4. **Exit SQLPlus:**

```
SQL> quit
```

Import Geometry Tables

Navigate to the directory where file `mb_geometry.dmp` is and execute:

```
imp mb_tutorial/mb_tutorial file=mb_geometry.dmp
full=y
```

You might see console output similar to the following. The important part is that five tables (cities, counties, customers, interstates, and states) should be successfully imported.

```
import done in US7ASCII character set and AL16UTF16
NCHAR character set
import server uses WE8DEC character set (possible
charset conversion)
. importing MVDEMO's objects into MB_TUTORIAL
. importing MVDEMO's objects into MB_TUTORIAL
.. importing table "CITIES"
195 rows imported
.. importing table "COUNTIES"
3230 rows imported
.. importing table "CUSTOMERS"
205 rows imported
```

```

. . importing table                "INTERSTATES"
239 rows imported
. . importing table                "STATES"
56 rows imported
Import terminated successfully without warnings.

```

Once the import is complete connect to the database as the user mb_tutorial using SQL*Plus. Check that the spatial metadata and indexes exist. That is, execute the queries

```

SQL> select table_name, column_name from user_sdo_
geom_metadata;
and
SQL> select table_name, index_name from user_sdo_
index_info;

```

to verify that there are entries for the cities, counties, customers, interstates, and states tables.

The file mb_metadata.dmp contains a copy of the user_sdo_maps, user_sdo_themes, and user_sdo_styles metadata in tables named mb_tutorial_maps, mb_tutorial_themes, and mb_tutorial_styles. If you wish to use these readymade styles, themes, and map definitions instead of creating them by following the instructions detailed in subsequent sections then import this file too. Once you've imported the file connect as the mb_tutorial user in SQL*Plus. Then insert the contents of these tables into the corresponding views as follows.

```

insert into user_sdo_maps select * from mb_tutorial_
maps;
insert into user_sdo_themes select * from mb_tutorial_
themes;
insert into user_sdo_styles select * from mb_tutorial_
styles;

```

If however you prefer to create all the required styles, themes, and maps with MapBuilder then follow the instructions provided in the remainder of this tutorial.

Connect to Database with MapBuilder

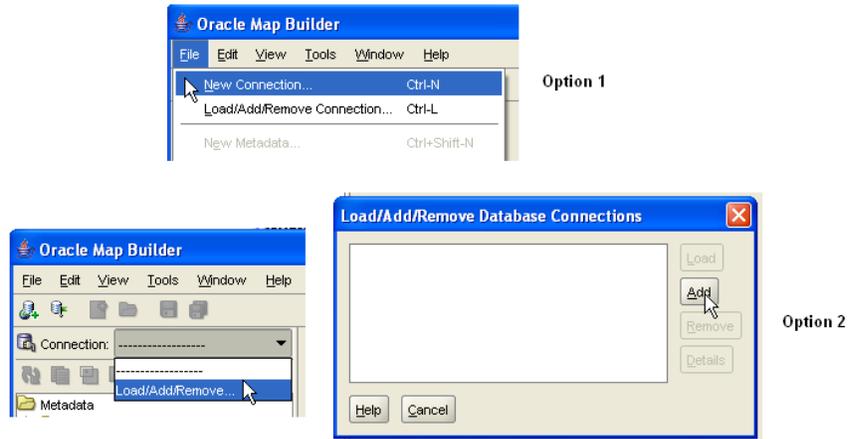
Navigate to the directory where MapBuilder application is and run the following command:

```
java -jar -Xmx512M mapbuilder.jar -noconnect
```

To add a new database connection in MapBuilder, use one of the following options:

Option 1: Menu **File** and then menu item **New Connection...**

Option 2: Click the **Connection** drop-down list and select item **Load/Add/Remove...** A dialog with connections from the preferences file is shown, and click on **Add** button.

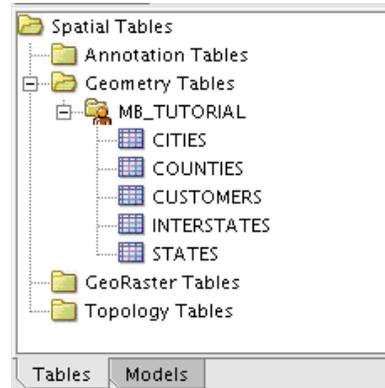


Enter the required details in the Add Connection dialog and click Test Connection followed by OK if the connection test is successful. The connection details will depend on your database installation.



MapBuilder will load the relevant metadata and populate the navigator trees.

At this point the Metadata Navigator tree is empty, as there is no MapViewer metadata for this connection. The Tables Navigator tree will have a list of geometry tables available.

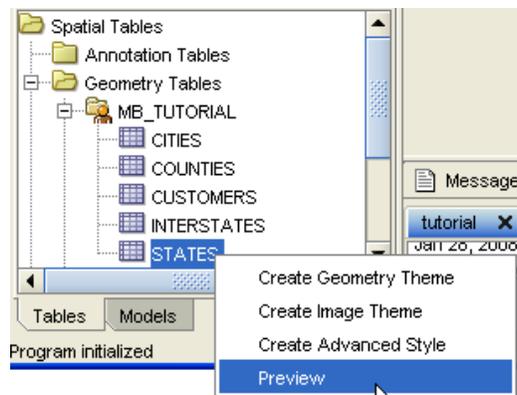


Quick Preview of Spatial Data

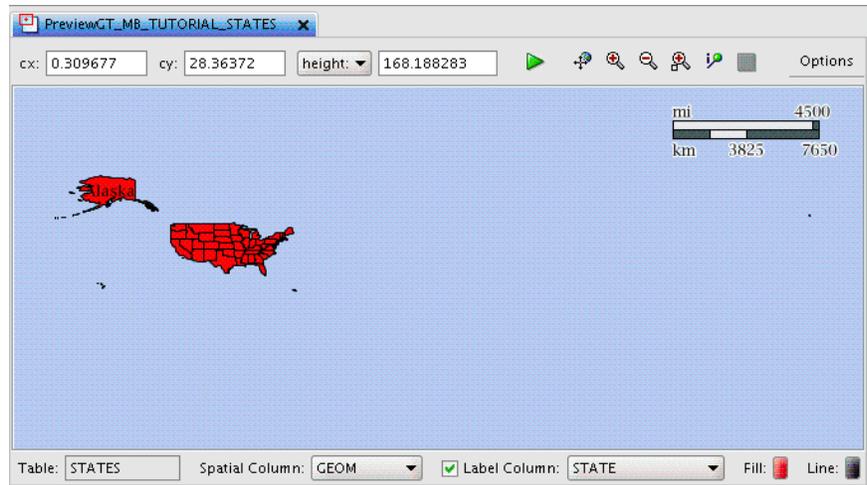
MapBuilder allows users to preview the original spatial data without creating any styles, themes, or map definitions. To quickly preview the STATES table:

Step 1: Click on **STATES** node of the **Spatial Tables Navigator**.

Step 2: Right click to bring up a submenu and select menu item **Preview**.



The quick preview panel is displayed on the right side of the application. It contains three areas: the top area has text fields to define the map data location, some buttons to draw the map and interact with it (pan, zoom in, zoom out, zoom by rectangle, and identify operations), and an Options menu; the middle area is the mapdisplay; and the bottom contains some information about the selected table. To render the feature geometries with labels, check box Label Column and choose a label column on the list (for example STATE column). Press the green arrow to display the full map. Play with the zoom buttons to enlarge or reduce the map scale. As you zoom in you may notice more labels on the map.



Create MapViewer Metadata

This section describes how to create styles, themes and base maps to render spatial data for a map that will display STATES, COUNTIES, INTERSTATES, CITIES, and CUSTOMERS. First define the styles (colors, lines, and markers), then the themes, and finally the base map.

Styles

Styles are used to render and to label spatial features such as interstate highways.

Color Style

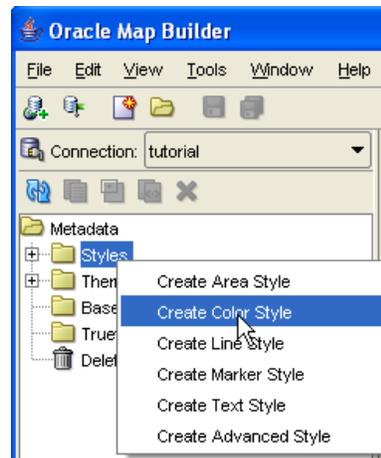
Color styles can be used to render area, linear and point features. Create a style for rendering the STATES table as described below.

Option 1: Click the **Styles** node on the **Metadata Navigator** tree, right click to bring up a

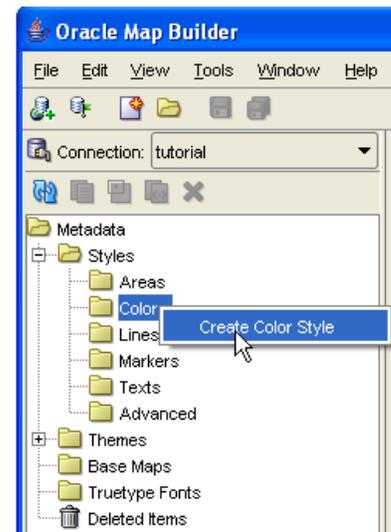
menu, and select **Create Color Style**.

Option 2: Expand the **Styles** node on the **Metadata Navigator** tree and click on **Colors** node, then

right click to bring up a menu and select **Create Color Style**.



Option 1



Option 2

This opens up a Editor panel for a Color Style on the right. Enter basic parameters such as the name and description of the style, and define its fill and stroke attributes. The fill attribute defines how the geometry will be filled, while the stroke defines how the outline (border) will be rendered. Now let's create a style for the STATES table (note that not all steps are required and they need not be done in the order shown below).

Step 1: Select the **Fill** option (check the box if it isn't). Define the fill color by clicking on the

color icon to bring up a dialog with colors, or entering the hexadecimal value (for example

#F2EFE9) and then press <Enter>.

Step 2: Select the **Stroke** option and uncheck the box.

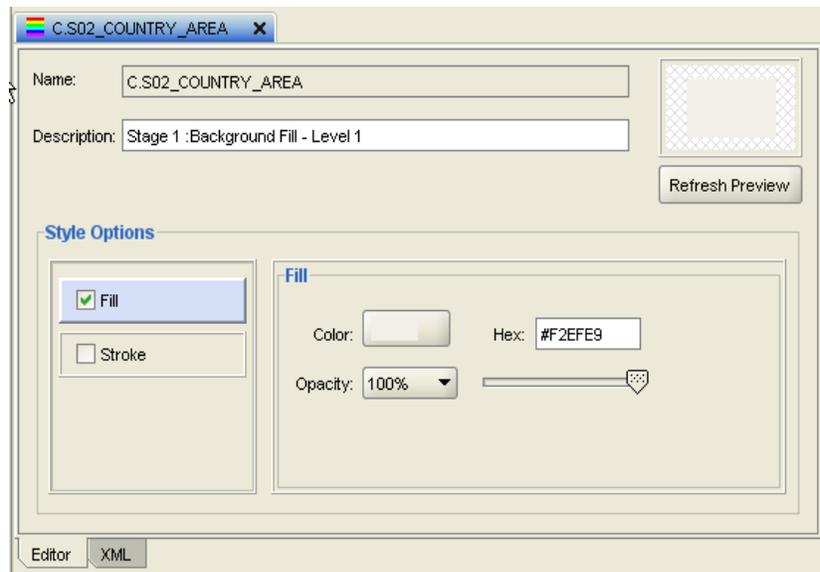
Step 3: Click on **Preview** button to see the current color style representation.

Step 4: Click on **XML** tab to see the current XML definition for this color style.

Step 5: Enter *C.STATES* in the name text field, and an optional description text.

Step 6: Click the **Save** icon on the application tool bar to store the color style definition on

USER_SDO_STYLES database view.



After saving the style, the **Metadata Navigator** tree is updated with this style name under **Colors** node.

Line Style

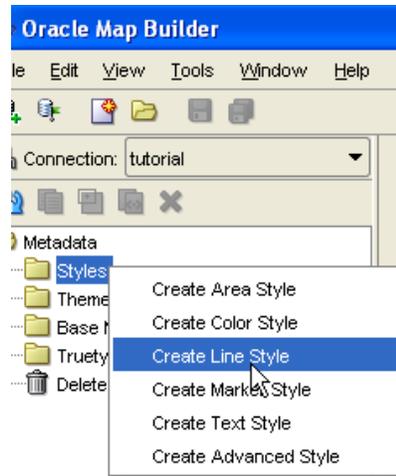
Line styles can be used to render linear features. They can also be used to render the boundary of area features. Create line styles for displaying counties and interstate highways. A line style can be created using either of the following options:

Option 1: Click the **Styles** node on the **Metadata Navigator** tree, right click to bring up a

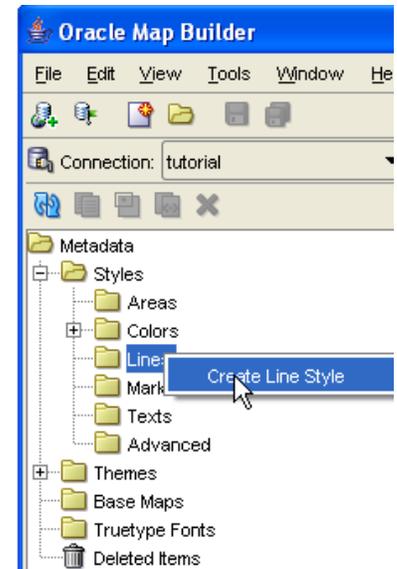
menu, and then select **Create Line Style**.

Option 2: Expand the **Styles** node on the **Metadata Navigator** tree and click on **Lines** node, then right

click to bring up a menu and select **Create Line Style**.



Option 1



Option 2

An editor page for a Line Style is opened on the right. Here we'll define basic parameters such as name and description, and attribute values for the overall base, center, and wing lines. First we define the base line by specifying the attributes of the overall line. The center line attributes define the line that will be drawn over and in the center of the base line. The wing line parameters define the borderlines of the style. To create a Line style to be used with *INTERSTATES* table, perform the following actions to create a line style with an overall line and a wing line as follows:

Step 1: Select the **Overall Line** option and set the Width to **3**, End Cap type to Butt, Joint to Round, and define the color by clicking on the color icon for a dialog box to select a color, or by entering the hexadecimal value (for example, #DB883F) and pressing <Enter>. Leave the Opacity at 100% and make sure the Cased check box is checked.

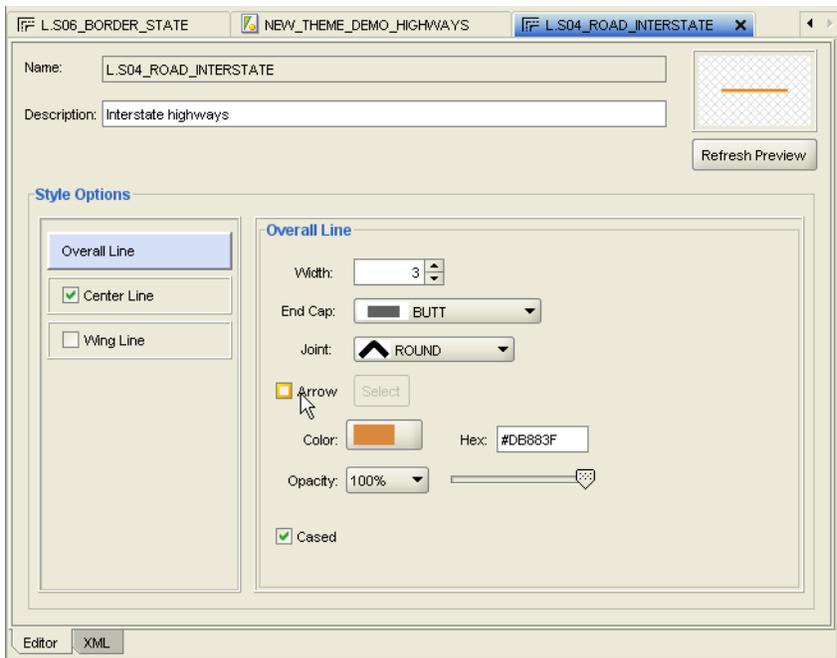
Step 2: Select the **Center Line** option ensuring the check box is checked. Set the **Width** to 1, leave the Dash Pattern text field blank, and define the color as hexadecimal #FCD462.

Step 3: Click on **Preview** button to see the current line style representation.

Step 4: Click on **XML** tab to see the current XML definition for this line style.

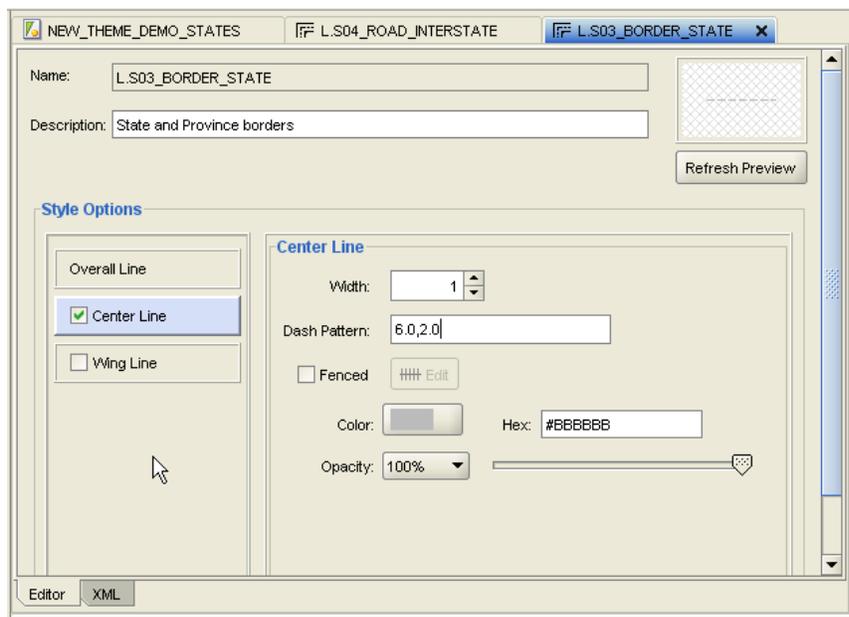
Step 5: Enter *L.INTERSTATES* (not L.S04_ROAD_INTERSTATE) in the name text field, and an optional description text.

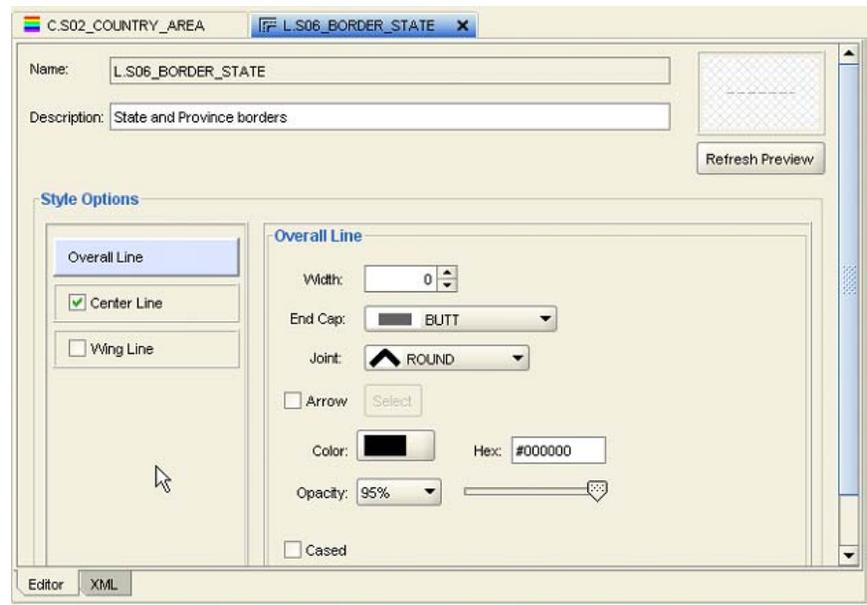
Step 6: Click the **Save** icon on the application tool bar to store the line style definition in the *USER_SDO_STYLES* database view.



After saving the style, the **Metadata Navigator** tree is updated with this style name under **Lines** node.

Next we define the line style to display the county borders. The steps are similar to the ones above. Except in this instance the Overall Line width is 0 (i.e. there is no overall line) and the Center Line has a width of 1, a dash pattern (6.0, 2.0 indicating 6 pixel long line followed by 2 blank pixels) and color (#BBBBBB) as shown in the figures below. The first figure shows the Center Line attributes while the second shows the Overall Line attributes. Name the style L.COUNTIES (not L.S03_BORDER_STATE) and click Save.





Marker Style

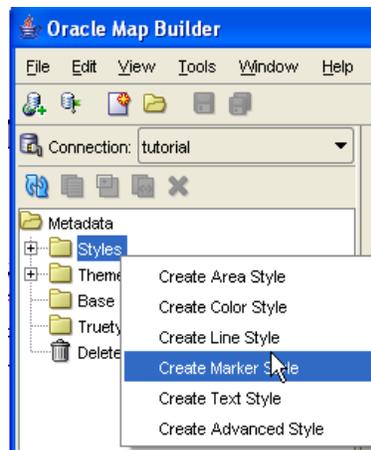
Marker styles can be used to render point features, and to label linear and point features. The base marker can be associated with an image, with a vector representation, or with a true type font. A marker style can be created using either of the following options:

Option 1: Click the **Styles** node on the **Metadata Navigator** tree, right click to bring up a

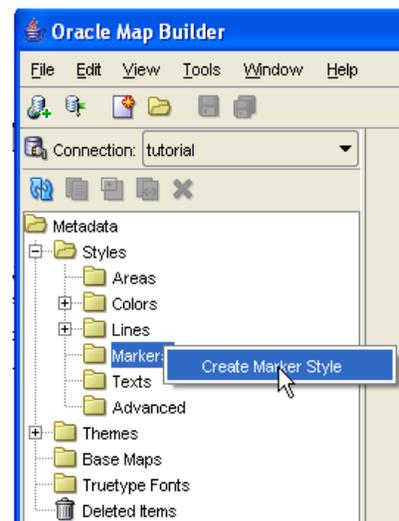
menu, and then select **Create Marker Style**.

Option 2: Expand the **Styles** node on the **Metadata Navigator** tree and click on **Lines** node, then right

click to bring up a menu and select **Create Marker Style**.



Option 1

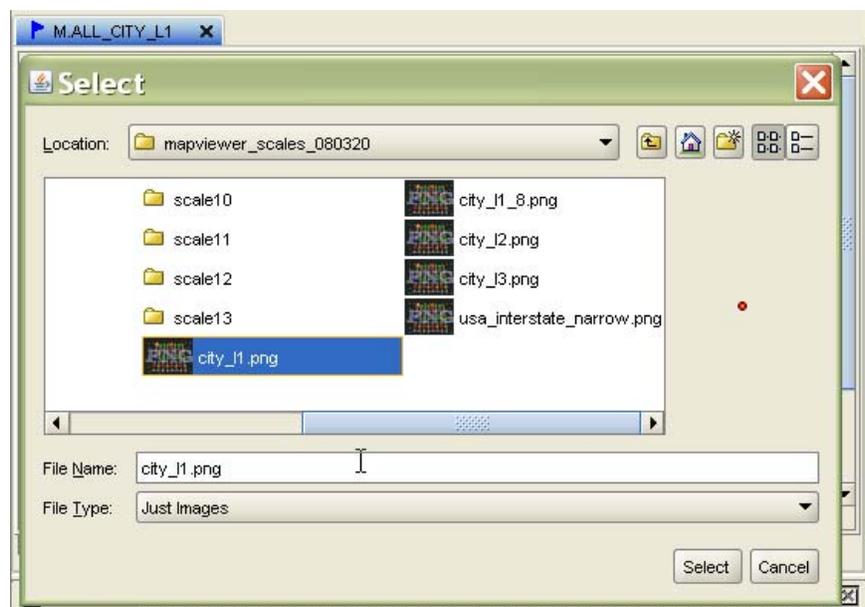


Option 2

An editor page for a Marker Style is opened on the right side. The style options include the marker type (image, vector or true type font), the marker text that is applied inside the marker when labeling linear or point features, and the marker size. To create a Marker style to be used with *CITIES*, *INTERSTATES*, and *CUSTOMERS* tables, perform the following actions:

Step 1: Select the **Marker Type** option and click on the **Image** radio button.

Step 2: Click button **Load Image**. Navigate to the directory where file **cities.png** is and select it.



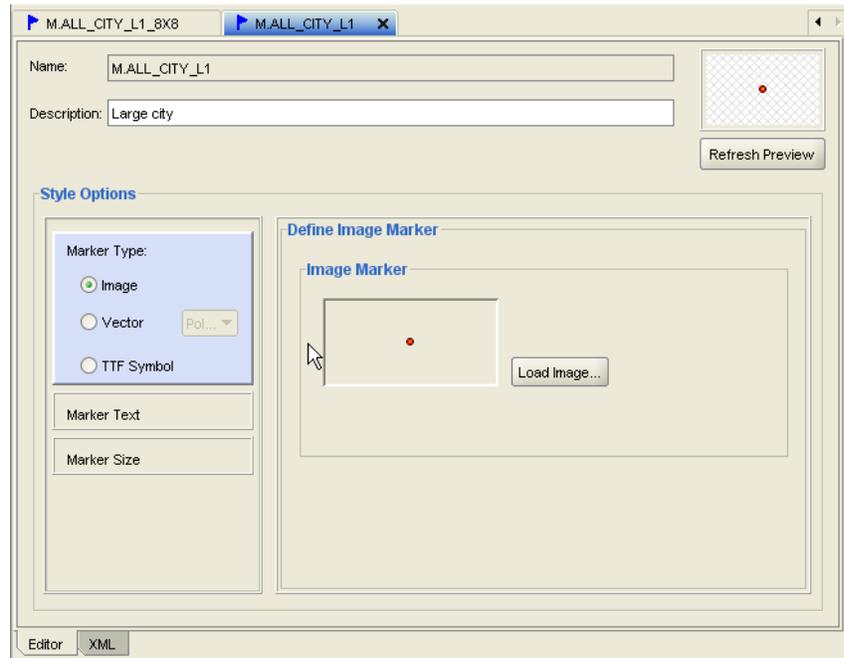
Step 3: Click on **Preview** button to see the current marker style representation.

Step 4: Click on **XML** tab to see the current XML definition for this marker style.

Step 5: Enter *M.CITIES* (not *M.ALL_CITY_L1*) in the name text field, and an optional description text.

Step 6: Click the **Save** icon on the application tool bar to store the marker style definition in the

USER_SDO_STYLES database view.



After saving the style, the **Metadata Navigator** tree is updated with this style name under **Markers** node.

Perform the same steps for the images files *cities_8X8.png* (name the style *M.CITIES_8X8*), *medium_cities.png* (name the style *M.MEDIUM_CITIES*), *small_cities.png* (*M.SMALL_CITIES*), *customers.png* (*M.CUSTOMERS*), and *highway_shield.png* (*M.HIGHWAY_SHIELD*).

7.1.4 Text Style

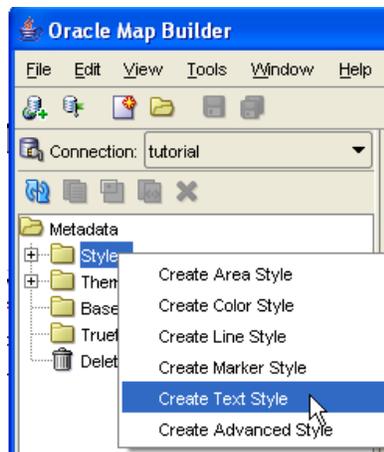
Text styles are used to label features. We will create two text styles, one for displaying city names and the other for state abbreviations.

Option 1: Click the **Styles** node on the **Metadata Navigator** tree, right click to bring up a

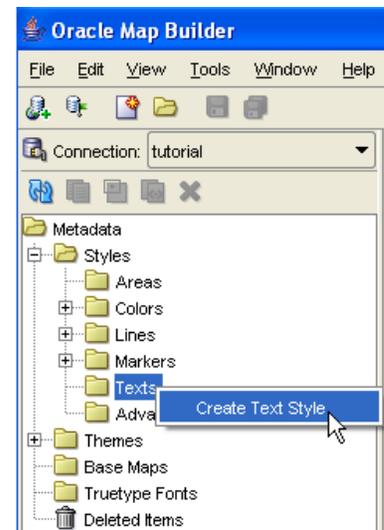
menu, and then select **Create Text Style**.

Option 2: Expand the **Styles** node on the **Metadata Navigator** tree and click on **Texts** node, then right

click to bring up a menu and select **Create Text Style**.



Option 1



Option 2

An editor page for a text Style is opened on the right. Style options for a text style include basic information for text such as font and size to be used, and other attributes related with multi lines, path labeling, and orientation points. To create text styles for labeling STATES and CITIES, perform the following actions:

Step 1: Select the **Text** style option.

Step 2: Change the text size to 11, set the color to #666666, and keep all other defaults.

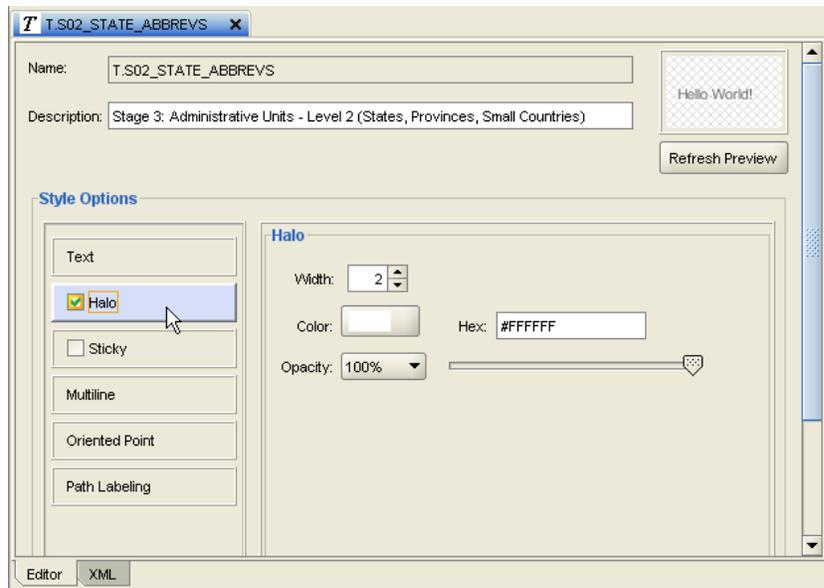
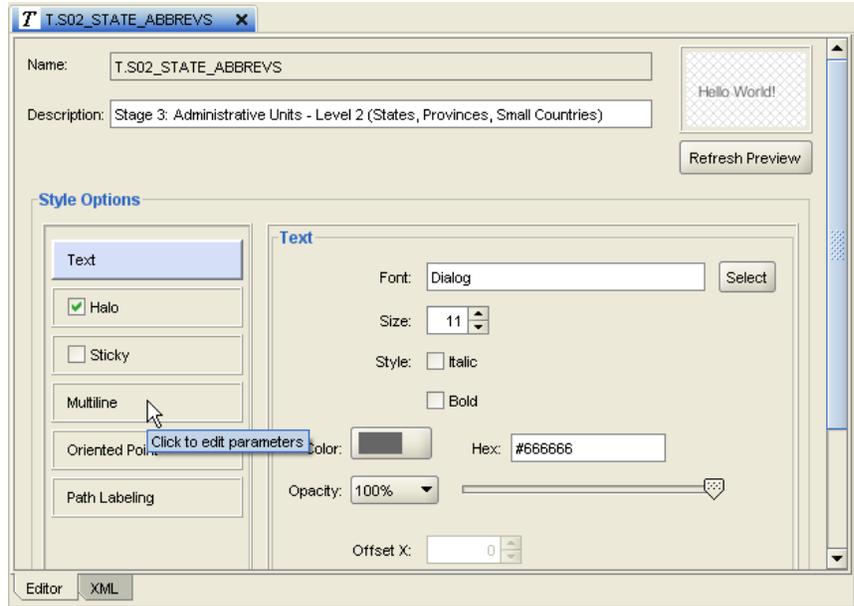
Step 3: Click on the Halo option and set Width to 2 and Color to #ffffff (white).

Step 4: Click on **Preview** button to see the current text style representation.

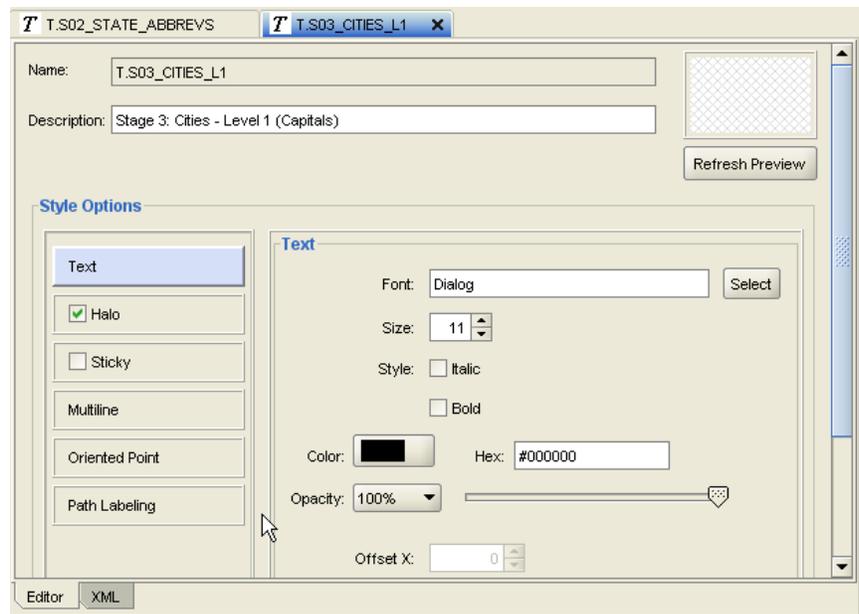
Step 5: Enter `T.STATE_ABBRV` in the name text field, and an optional description text.

Step 6: Click the **Save** icon on the application tool bar to store the text style definition in

`USER_SDO_STYLES` database view.



Repeat above steps to create a text style (*T.CITIES*) for *CITIES* table, setting the Font to Dialog, size as 11, and Color to #000000 (black). Select the Halo option and set width to 2 and color to #ffffff (white). Click on Save.



Repeat the above to create text label styles for medium (T.MEDIUM_CITIES) and small (T.SMALL_CITIES) cities. The Halo is the same in both cases (width=2 color=#FFFFFF). For T.MEDIUM_CITIES set the font (in the Text options) to Dialog 10 and Color to #3F3F3F (dark grey). For T.SMALL_CITIES set the font to Dialog 9 and Color to #333333.

Advanced Style

Advanced styles are constructed from simple styles and a condition that determines which simple style is used to render the current geospatial feature. They are generally used to render features, but may also be used to label them. The base advanced style is the bucket style. A bucket style is a one-to-one mapping between a set of primitive styles and a set of buckets. Each bucket contains one or more attribute values for features to be plotted. For each feature, one of its attributes is used to determine which bucket it falls into and then the style assigned to that bucket is applied to the feature. We will create an Advanced Style to render cities based on their population. An advanced style can be created with the following options:

Option 1: Click the **Styles** node on the **Metadata Navigator** tree, right click to bring up a menu and select **Create Advanced Style**.

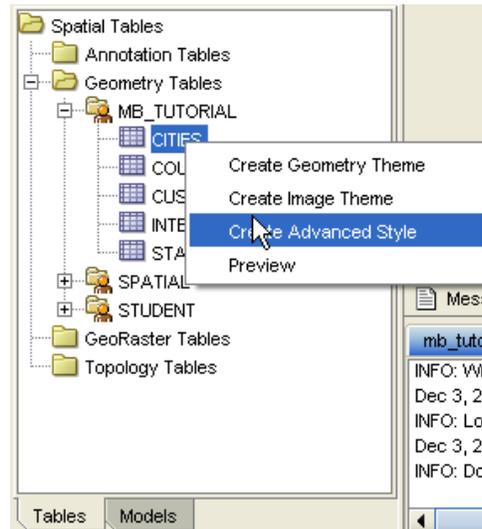
Option 2: Expand the **Styles** node on the **Metadata Navigator** tree and click on **Advanced** node, then right click to bring up a menu and select **Create Advanced Style**.

Option 3: Advanced styles are based on attribute values, and users can create an advanced styles using database table content. Expand the **Spatial Tables** tree; select the spatial table node, right click to bring up a menu and select **Create Advanced Style**.

A dialog will display the available advanced styles. A style creation wizard is invoked when you select one of the available style choices.

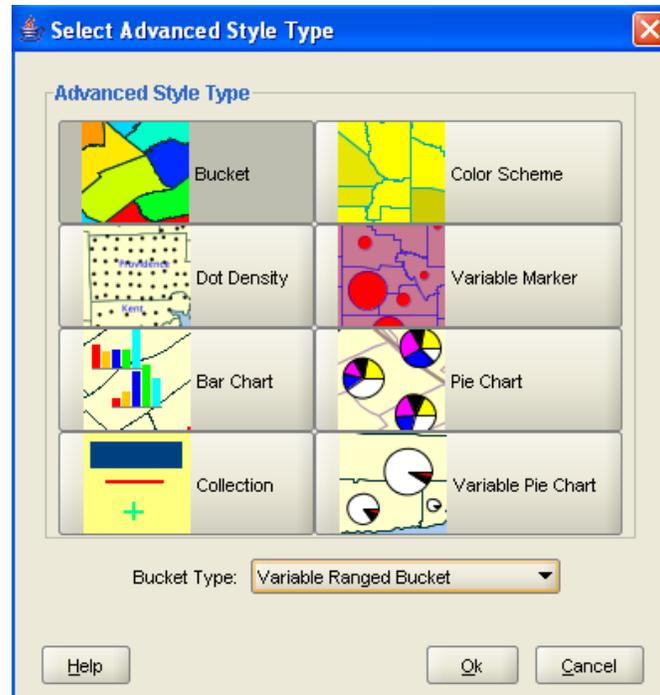
Using *Option 3*, let's create an advanced style to represent different population ranges for the *CITIES* table. The wizard to do that, which will help us define a theme based on the advanced style being created.

Step 1: Select the *CITIES* node in the Spatial Tables/Geometry Themes tree under MB_TUTORIAL (or MVDEMO or whatever you named the user schema where you imported the tutorial data). Right click and select **Create Advanced Style**.



Step 2: A dialog is shown with the advanced style types. Keep the default selection type **Bucket** but

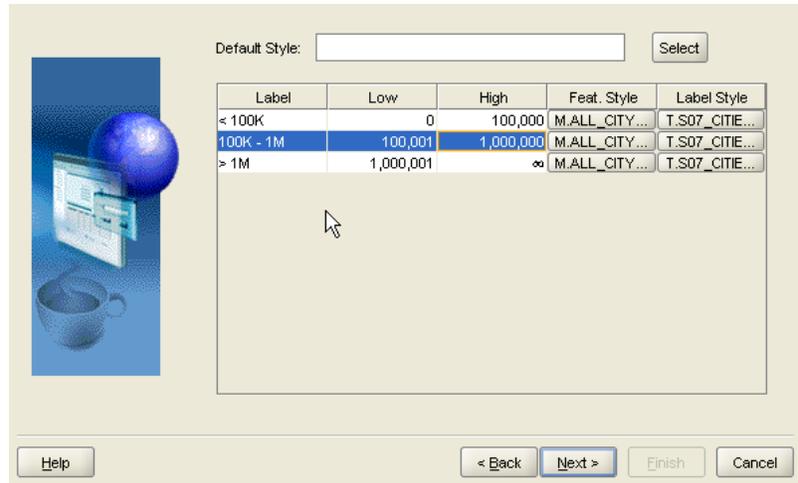
choose **Variable Ranged Bucket** as the Bucket type. Click OK to start the wizard.



Step 3: Press **Next**, on the wizard's introductory page, to continue.

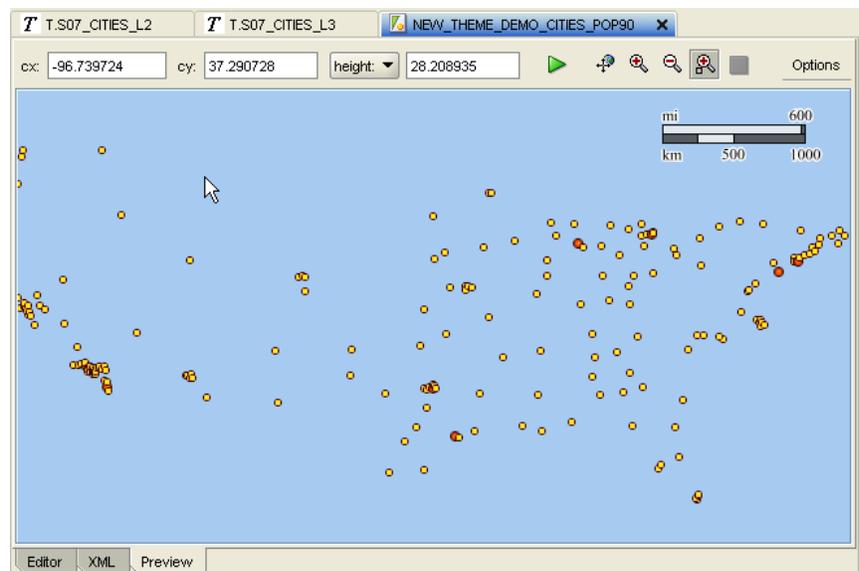
Step 4: This page is already populated with some information from *CITIES* table. Enter V.CITIES for the style name and set the number of buckets to 3. The POP90 column (attribute) is already selected. We will use that value as the bucket selector.

Step 5: The bucket ranges are already defined with some default values obtained from *CITIES* table. Edit these values and set the Label, Low, High, Feature Style, and Label Style values to the following. Double-click on each row's entry to edit the values. The first bucket's values should be {< 200K, 0, 200000, M.SMALL_CITIES, T.SMALL_CITIES} assuming you used those names for the corresponding marker and text styles. The second bucket's values should be {200K – 1M, 200001, 1000000, M.MEDIUM_CITIES, T.MEDIUM_CITIES}. Enter the values {> 1M, 1000001, Infinity, M.CITIES_8X8, T.CITIES} for the third bucket.



Step 6: Next create the predefined theme based on this advanced style. Check the Create Predefined Theme option, and enter the theme name `THEME_CITIES_POP90`. Click Next.

Step 7: This last page contains the summary information for the advanced style and for the theme that will be stored on `USER_SDO_STYLES` and `USER_SDO_THEMES` database view. Press **Finish** to end the wizard. The metadata trees will be updated, and the theme editor will be opened. Press the **Preview** tab and the **green arrow** to render the theme. Note that at this point you just have the markers representing the cities. Later we are going to add labeling information to this theme.

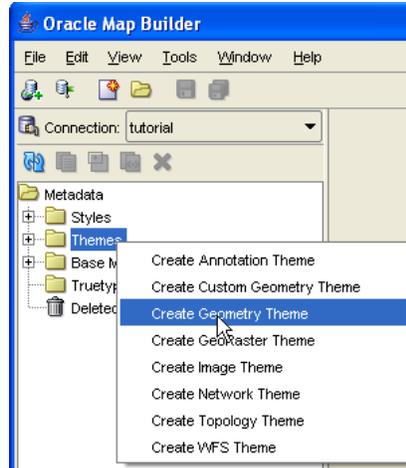


Geometry Themes

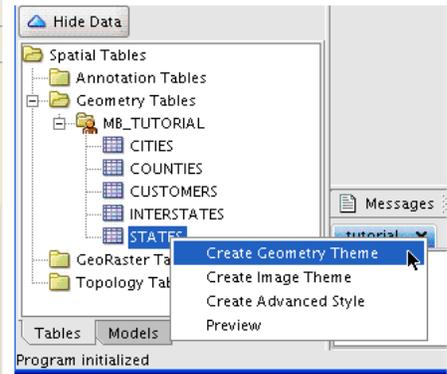
Geometry themes are based on spatial tables with column of type `SDO_GEOMETRY`. Now that we've created all the styles we'd like to use in rendering the data on a map we will create the themes. They are created using one of the following options:

Option 1: Click the **Themes** node on the **Metadata Navigator** tree, right click and select **Create Geometry Theme**.

Option 2: If the **Tables** and **Models** data navigators are not visible click on button **Show Data** located on the bottom left part of the application to display them. Select **Tables** tab to display the **Spatial Tables Navigator** tree. Expand the **Geometry Tables** node up to get the table names for node MB_TUTORIAL. Select the table node, right click and select **Create Geometry Theme**. The geometry theme definition wizard will start.



Option 1

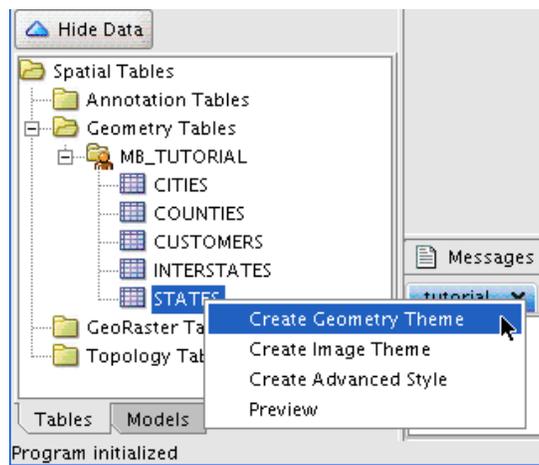


Option 2

Using *Option 2*, let's create some geometry themes based on the STATES, INTERSTATES, CITIES, COUNTIES, and CUSTOMERS tables.

STATES Theme

Step 1: Click the **STATES** node and right click to select **Create Geometry Theme**.



Step 2: A wizard will start with an introductory page. Press **Next** to continue.

Step 3: This page is already populated with information obtained from the *STATES* table. Define the theme name as *THEME_STATES* and keep the other field values. Press **Next** to continue.

Define a Geometry Theme - Step 1 of 4 - Theme Parameters

Name:

Description:

Table Owner:

Base Table:

Spatial Column:

Buttons: Help, < Back, Next >, Finish, Cancel

Step 4: Here you define the feature rendering style. Keep the style type as Color, and type *C.STATES* in the render style field or press the **Select** button to choose the style.

Define a Geometry Theme - Step 2 of 4 - Feature Style

Style Type:

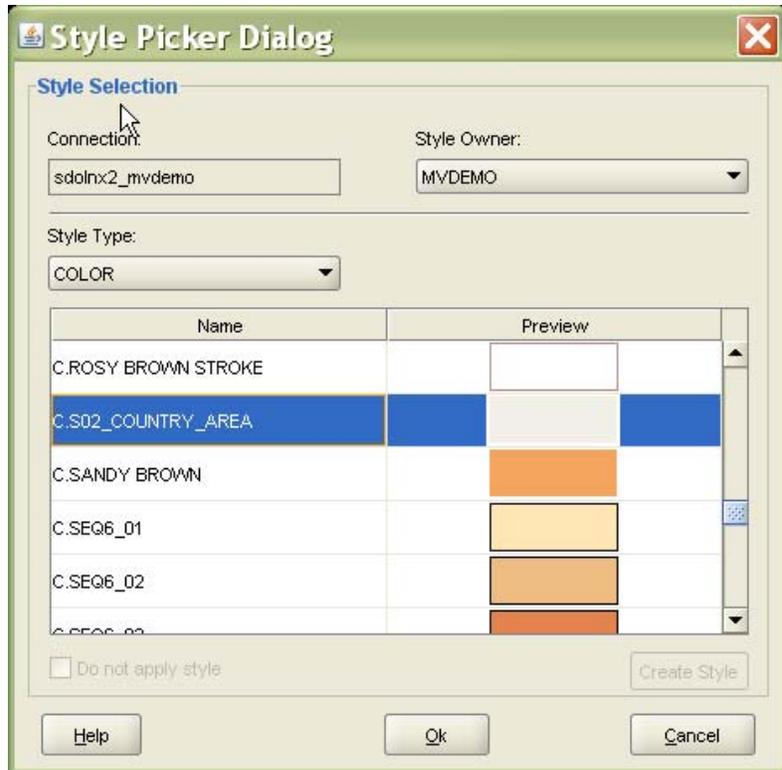
Render style:

Attributes (for advanced styles)

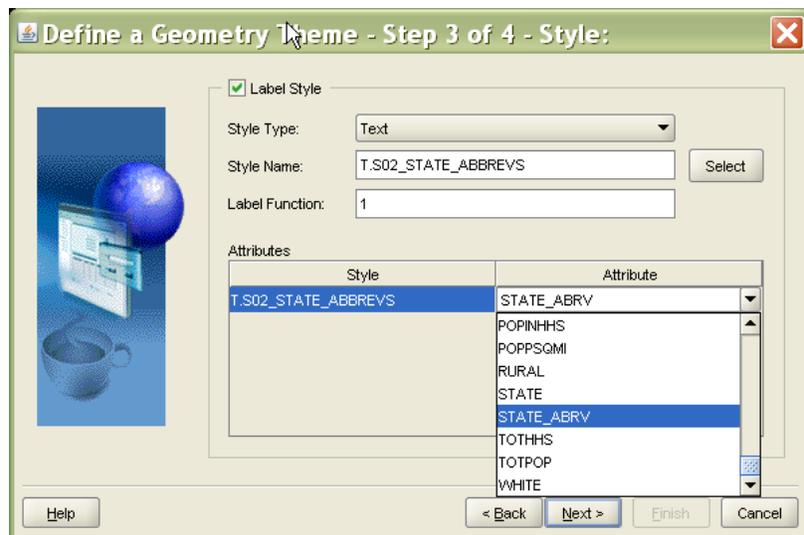
Style	Attribute

Buttons: Help, < Back, Next >, Finish, Cancel

This brings up the Style picker Dialog as shown below.

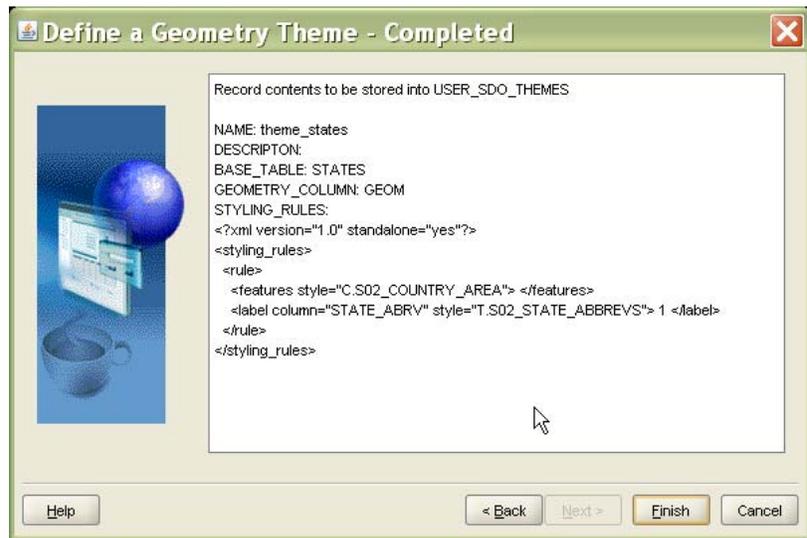


Step 5: Now define the label parameters. Check the **Label Style** box to enable the fields. Define *T.STATES* as the text style and select *STATE_ABRV* as the label column. Click Next.

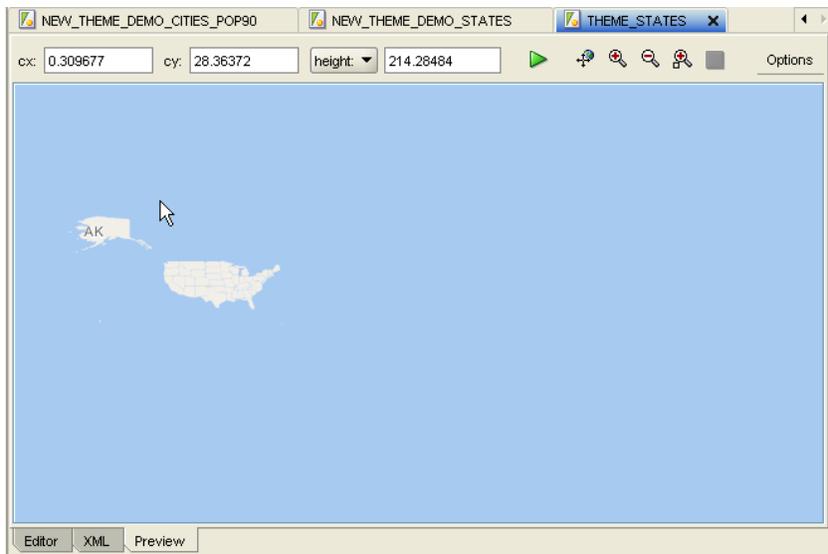


Step 6: Next we define query conditions to be applied. Leave it blank, which means that all features within the current map extent will be selected. Press **Next** to continue.

Step 7: This last page contains the summary information for the theme that will be stored in *USER_SDO_THEMES* database view. Press **Finish** to end the wizard.



Step 8: After ending the wizard the theme editor page is opened on the right side of the application. Select the **Preview** tab and click the **green arrow** to display all data for this theme.



COUNTIES Theme

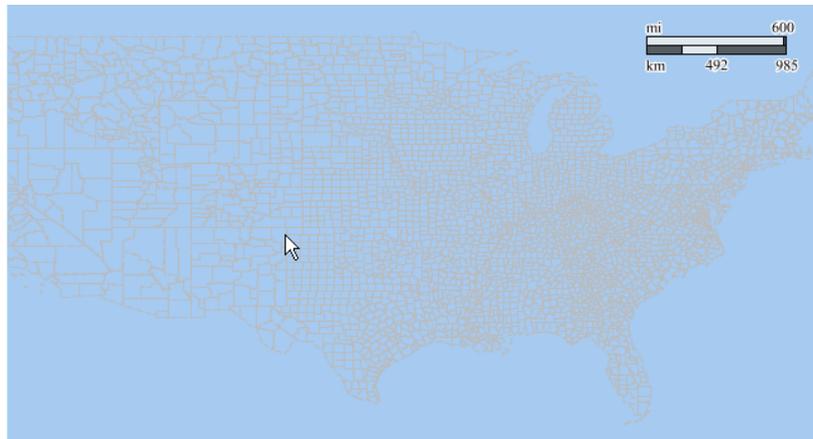
Select the *COUNTIES* node and repeat the previous steps entering the following information in the wizard pages (keep the other parameters as they are):

Theme Name: *THEME_COUNTIES*

Style Type: *Color*

Render Style: *L.COUNTIES*

With the theme editor opened, select the **Preview** tab and click the **green arrow** to display all data for this theme. Depending on the display area, you may get an image with missing counties. This is because these counties are very small for the current display resolution, and the anti alias parameter is currently set to true (see Options menu on the top right corner of this preview panel). Using the zoom area button select the continental US area. The preview image is something like:



HIGHWAYS Theme

Select the *INTERSTATES* node and repeat the above steps entering the following information on the wizard pages (keep the other parameters as they are):

Theme Name: *THEME_HIGHWAYS*

Style Type: *Line*

Render Style: *L.INTERSTATES*

Label Style Type: *Marker*

Label Style Name: *M.HIGHWAY_SHIELD*

Label Attribute: *ROUTEN*

Label Function: *(3-length(routen))*

The label function determines when a feature is labeled. So only those interstates with 2-digit numbers will be labeled since the label function returns 1 (true) for those highways.

With the theme editor opened, select the **Preview** tab and click the **green arrow** to display all data for this theme.



HIGHWAYS_LINE Theme

We'll create another highways theme that just displays the interstates without any interstate highway shield markers. Select the *INTERSTATES* node and repeat the above steps entering the following information on the wizard pages (keep the other parameters as they are):

Theme Name: *THEME_HIGHWAYS_LINE*

Style Type: *Line*

Render Style: *L.INTERSTATES*

Label Style Type: *Marker*

Label Style Name: *M.HIGHWAY_SHIELD*

Click on Save to store the theme definition in *USER_SDO_THEMES*.

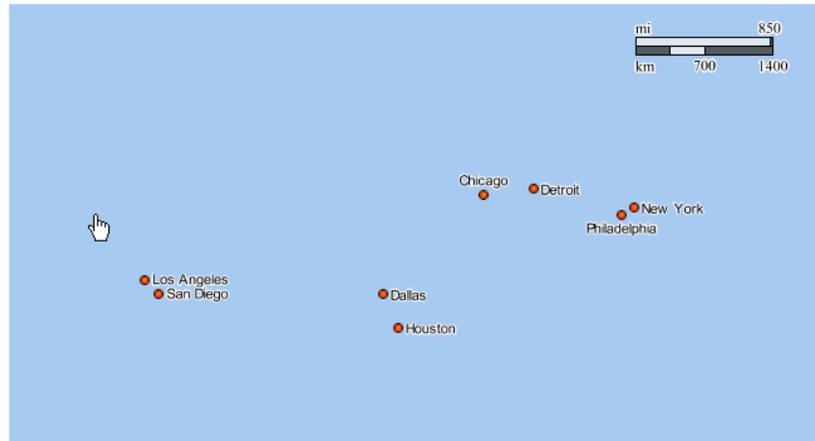
CITIES Theme

We defined a *CITIES* theme when defining the *V.CITIES* advanced style. To add the label style click on the Styling Rules option in the theme's editor pane. Then select the first row and click on the Edit icon (the pencil). Next select *V.CITIES* as the Label Style. However that treats all cities as equal when they're considered for labeling. We wish to label large cities whenever possible and medium or small cities using a different text and marker style. So we'll create two cities themes, one named *theme_cities* (for small and medium sized cities) and one name *theme_bigcities*. The large cities theme will be added to the base map before the small and medium cities theme.

Step 1: Select the **CITIES** node in metadata tree and right click. Name it *THEME_CITIES*.

Step 2: Select Style Type as Marker and Render Style as *M.SMALL_CITIES*.

Step 3: Check the Label Style check box and select the Text Style *T.SMALL_CITIES*. Leave the attribute as *City*. This means the values contained in this column (the city name in this case) will be used to label the feature. The label function can be any valid SQL expression and is used to filter labels. The label will be shown if the label function results in value greater than 0 (just define it as 1, and the labels will be rendered).



CUSTOMERS Theme

Select *CUSTOMERS* and repeat above steps entering the following information (keep the other parameters as they are):

Theme Name: *THEME_CUSTOMERS*

Style Type: *Marker*

Render Style: *M.CUSTOMERS*

Label Style Type: *Text*

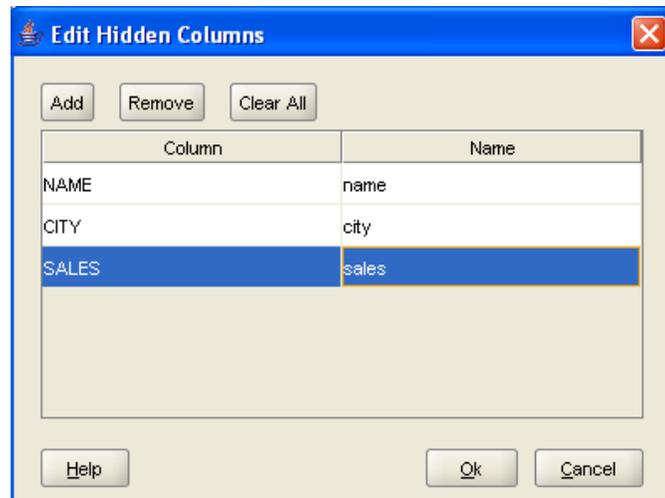
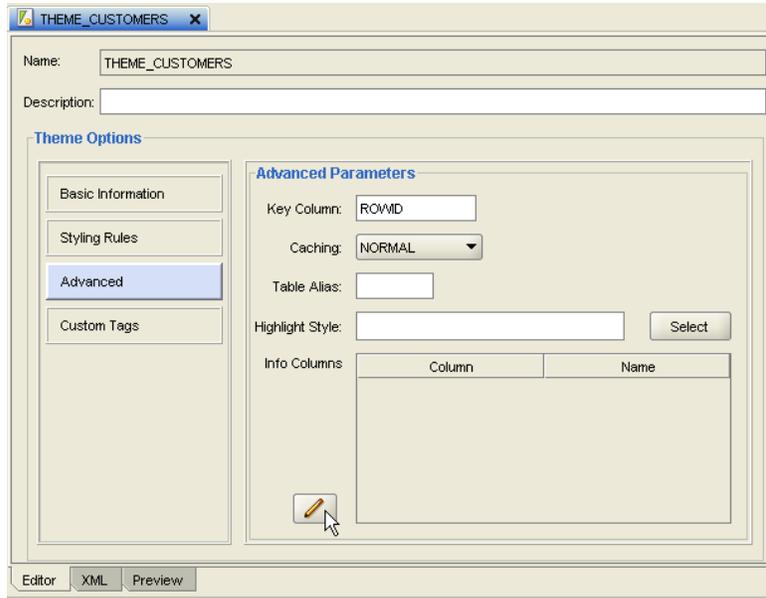
Label Style Name: *T.SMALL_CITIES*

Label Attribute: *NAME*

In Oracle Maps applications you can get information about features of interest (FOIs) on passing the mouse over the feature, or clicking on the feature. This is known as the Hidden Info columns and defined in MapBuilder as follows. The *THEME_CUSTOMERS* theme editor must be opened.

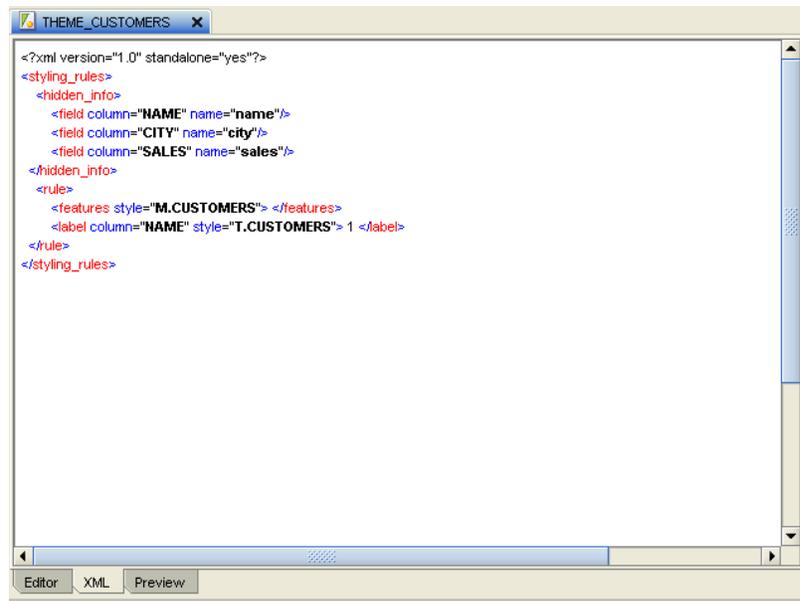
Step 1: Select the Advanced Option, and there will be a panel with Info Columns table at the bottom.

Click the edit button to define info columns.



Step 2: Define the info columns using the Add button.

Step 3: Select the *XML* tab to see the current theme definition, and press the *Save* button on *Toolbar* to save the current definition.



Base Map

Base map definitions, stored in USER_SDO_MAPS, contain one or more themes. A set of parameters can be assigned to each theme that is part of a base map. In this tutorial we use scale ranges. Scale ranges for themes are used to define which theme are displayed at which zoom levels during map visualization. If scale ranges are not defined the theme is always rendered independent of the map zoom level. Base maps are created in MapBuilder using a wizard. The base map creation wizard is invoked in either of the following ways:

Option 1: Menu **File**, then menu option **New Metadata**. Select radio button **Base Map** and click **OK**.

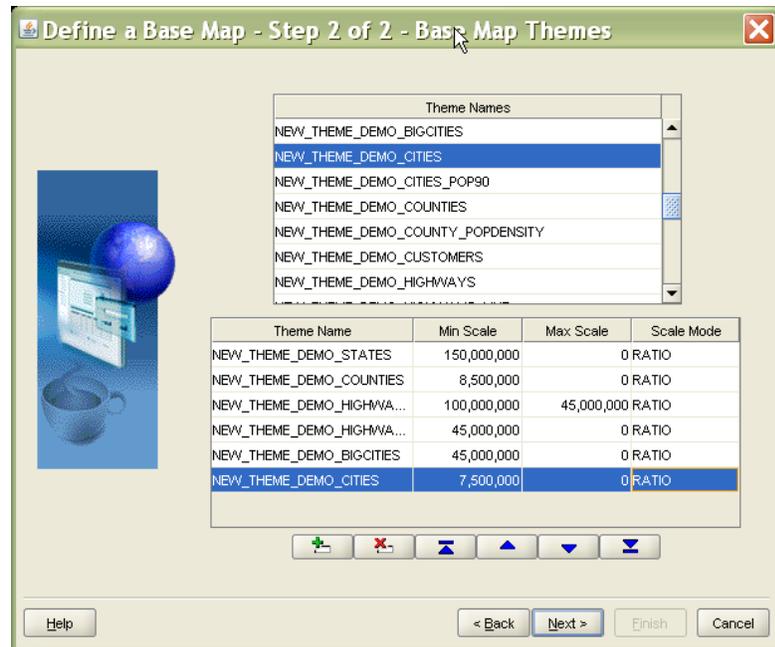
Option 2: Click the **Base Maps** node in the **Metadata Navigator**, right click and then select **Create Base Map**.

Now let's define a base map that uses previously created themes.

Step 1: Using option 2 above, starts the base map wizard. Press **OK** to continue.

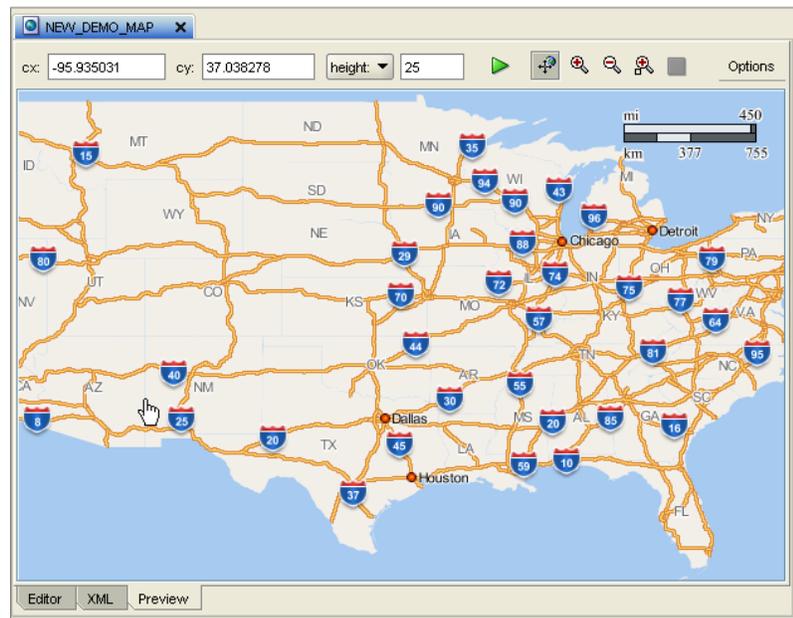
Step 2: Enter the base map name and an optional description. Press **OK** to continue.

Step 3: Add themes to the base map by selecting the theme on the table and pressing the **Add** icon. The scale ranges can be defined using the MapViewer native mode (Coordinate reference system units per screen inch), or the (preferred) ratio mode that is normally used in maps. Ratio scales are generally presented as 1:XXXX, which means 1 unit on the map represents XXXX units on the ground. When defining a scale value for ratio mode, do not enter for example the pattern 1:100000 but just the value 100000. Add the themes and set the scale values based on figure below. Then press **OK** to continue.



Step 4: A summary page is shown with the base map definition. Press **Finish** to store the base map definition and an editor page will be shown. Select the **Preview** tab and click on the green arrow to display the map. Note at the bottom of MapBuilder application the scale values for current visualization. The resulting map should contain just the themes that are in the scale range. Play with the zoom in and zoom out options to see the map results. With the editor you can also modify the initial scale ranges defined.

```
<?xml version="1.0" standalone="yes"?>
<map_definition>
  <theme name="THEME_STATES" min_scale="1.5E8" max_scale="0.0"
scale_mode="RATIO"/>
  <theme name="THEME_COUNTIES" min_scale="8500000.0" max_
scale="0.0" scale_mode="RATIO"/>
  <theme name="THEME_HIGHWAYS_LINE" min_scale="1.0E8" max_
scale="4.5E7" scale_mode="RATIO"/>
  <theme name="THEME_HIGHWAYS" min_scale="4.5E7" max_scale="0.0"
scale_mode="RATIO"/>
  <theme name="THEME_BIGCITIES" min_scale="4.5E7" max_scale="0.0"
scale_mode="RATIO"/>
  <theme name="THEME_CITIES" min_scale="7500000.0" max_
scale="0.0" scale_mode="RATIO"/>
</map_definition>
```



This concludes the tutorial. With the MapViewer metadata stored, you can now develop web applications using Oracle Maps or MapViewer Java API. Refer to the specific documentation, or forthcoming tutorials, for more details.