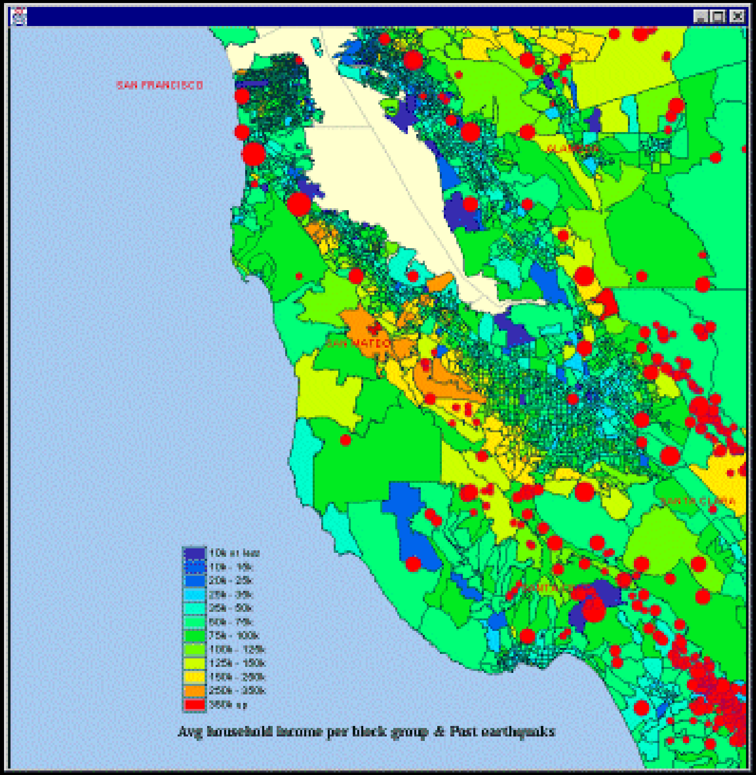
ORACLE FUSION MIDDLEWARE

Oracle Application Server 10*g* MapViewer



Technical Overview
September 2005
Oracle Corporation

Topics

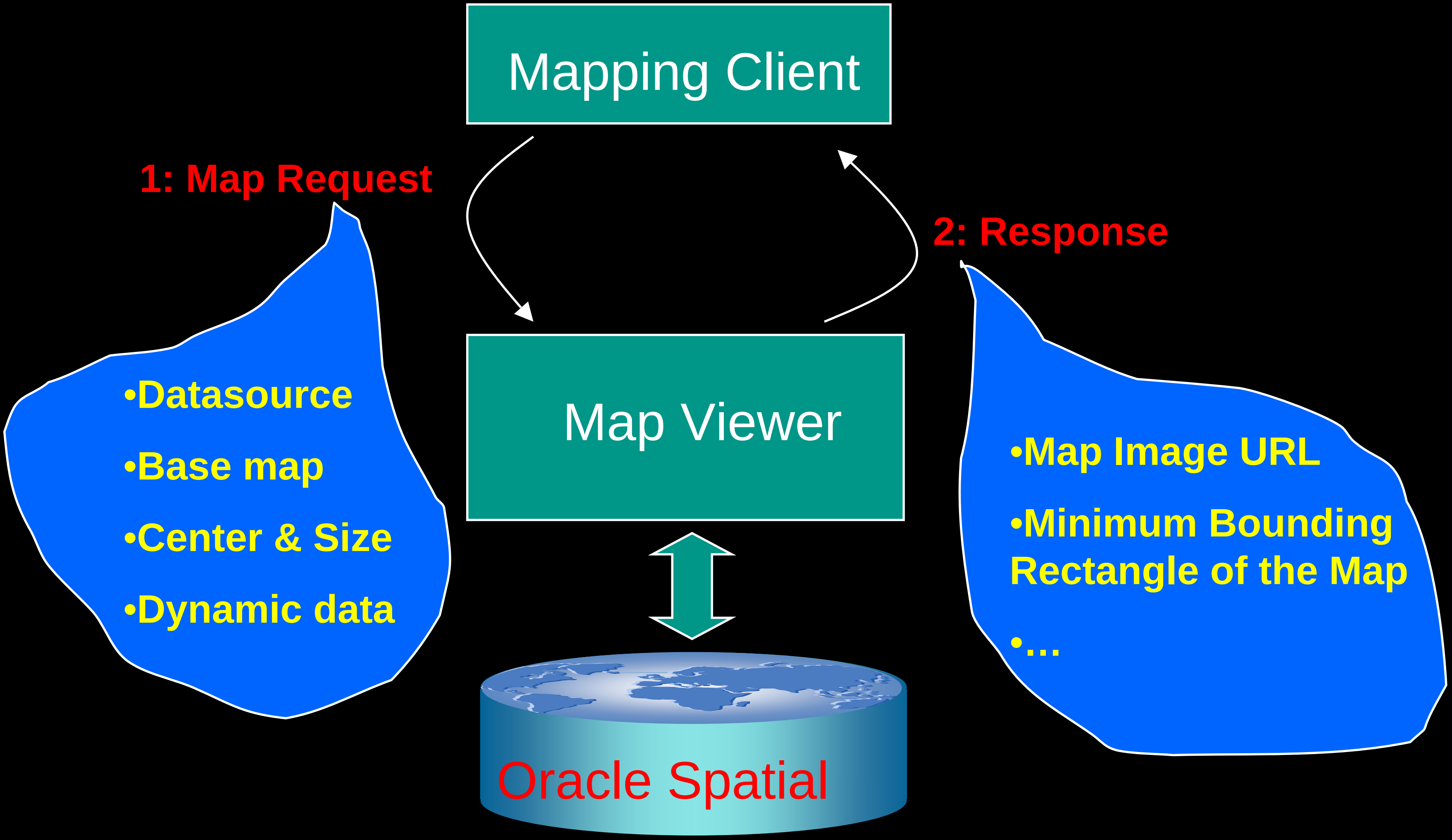
- Overview
- Main features
- Concepts
- APIs
- Fast start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

MapViewer Overview

- Map rendering service in Oracle Application Server 10*g*.
  - Server component; NOT a client viewer
- Visualizes data managed by Oracle Spatial and Locator.
- Provides a comprehensive set of APIs:
  - XML and Java-based
  - Enables easy development of client viewers
- Enterprise-level solution for management of map content and metadata

MapViewer Overview – Basic Flow of Action

Is a J2EE servlet, MapViewer uses a request/response model.



Mapping Client

1: Map Request

2: Response

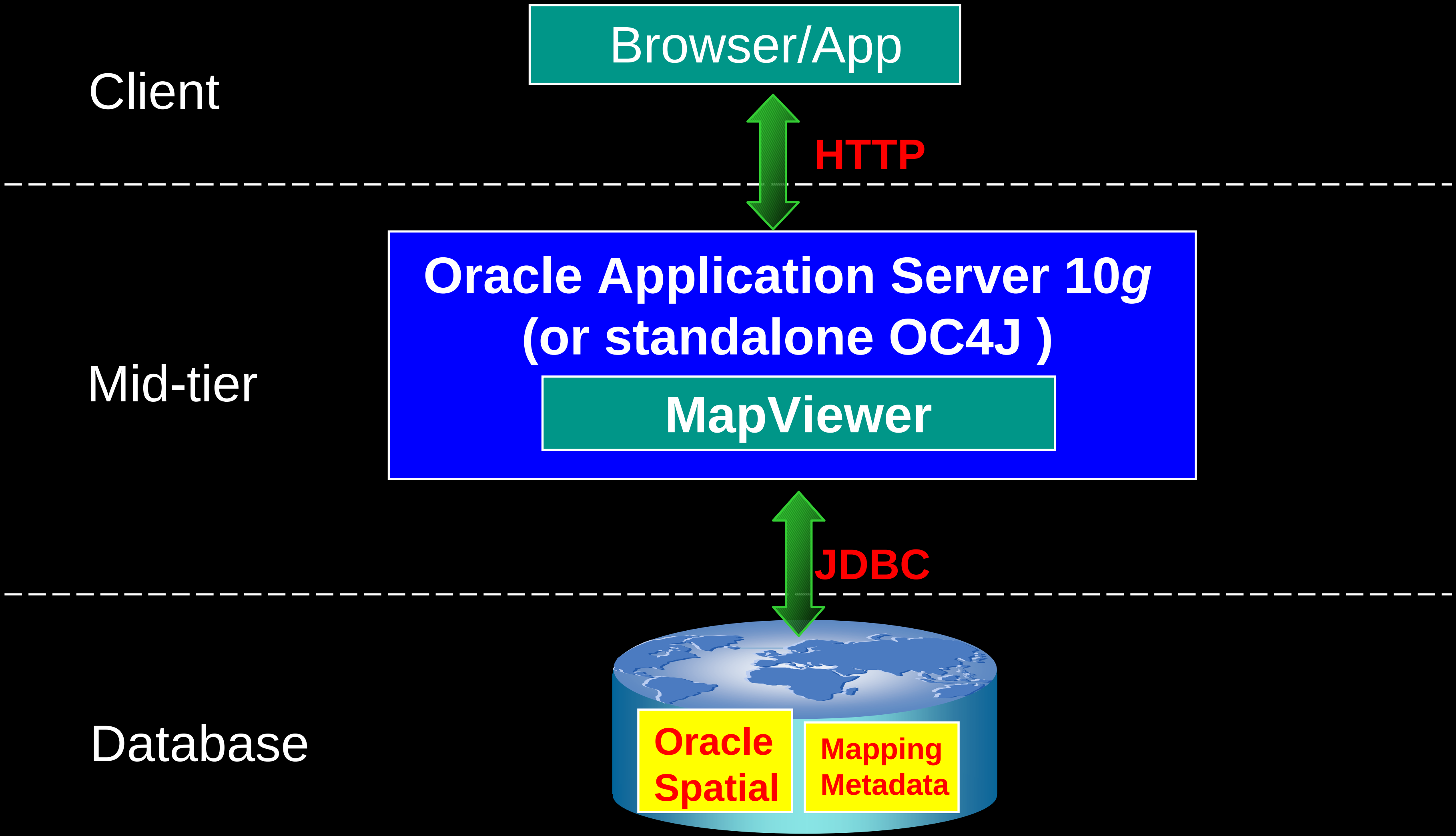
Map Viewer

- Datasource
- Base map
- Center & Size
- Dynamic data

- Map Image URL
- Minimum Bounding Rectangle of the Map
- …

Oracle Spatial

MapViewer Overview – Architecture



Client

Browser/App

HTTP

Mid-tier

Oracle Application Server 10*g* (or standalone OC4J)

MapViewer

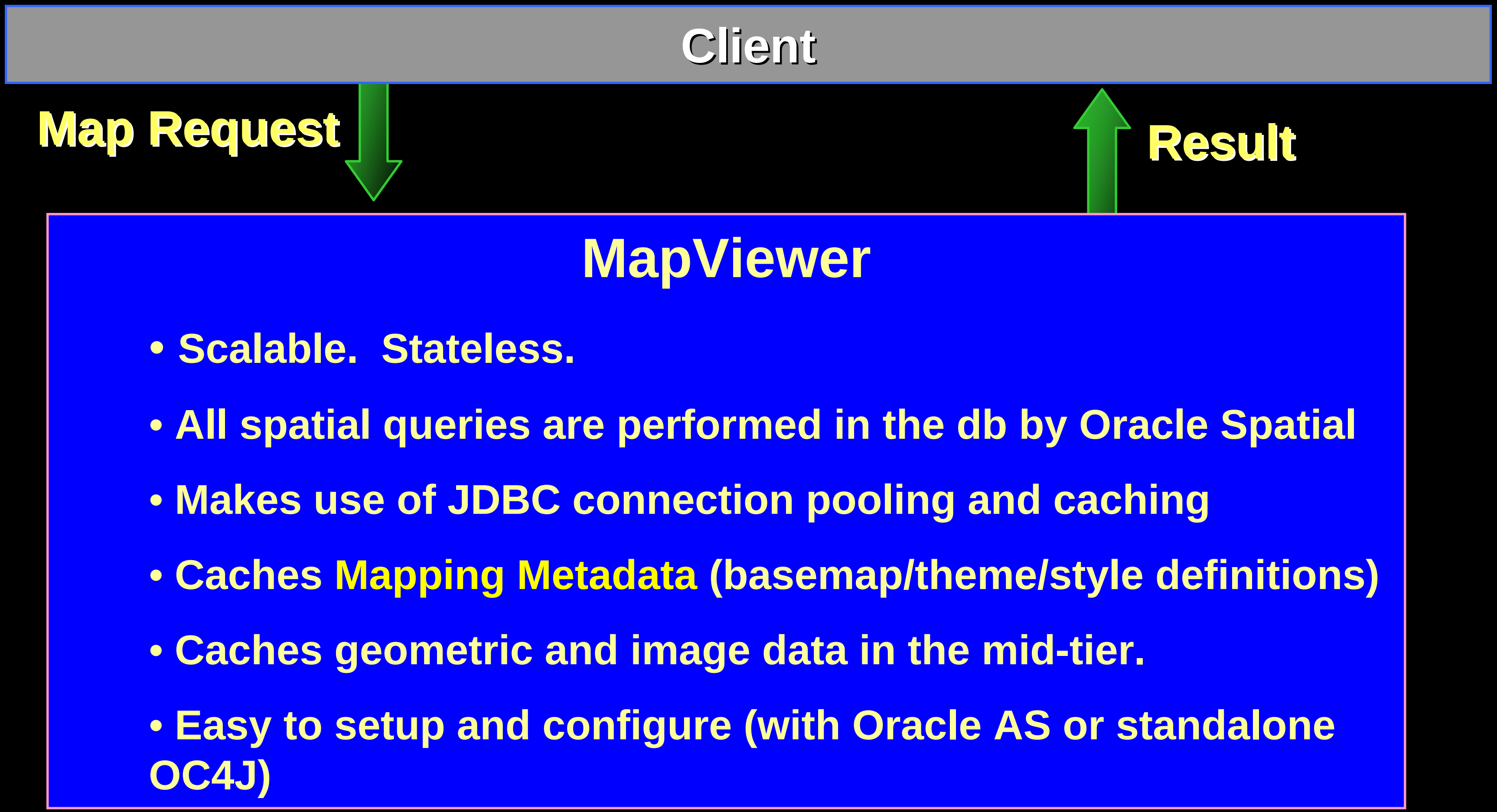
JDBC

Database

Oracle Spatial

Mapping Metadata

MapViewer Overview - Main Characteristics



Client

Map Request

Result

MapViewer

- Scalable. Stateless.
- All spatial queries are performed in the db by Oracle Spatial
- Makes use of JDBC connection pooling and caching
- Caches Mapping Metadata (basemap/theme/style definitions)
- Caches geometric and image data in the mid-tier.
- Easy to setup and configure (with Oracle AS or standalone OC4J)

Topics

- Overview
- Concepts
- Main features
- APIs
- Fast start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

MapViewer Key Concepts

- Datasource
- Map
- Basemap
- Theme
- Style
- Styled GeoFeature

MapViewer Key Concepts

Datasource

- A MapViewer admin defines one or more target databases from which MapViewer will generate maps. These target databases are called Datasources.

- A datasource always references to a database schema that contains some spatial layers or tables.

- Every map request MUST specify a datasource

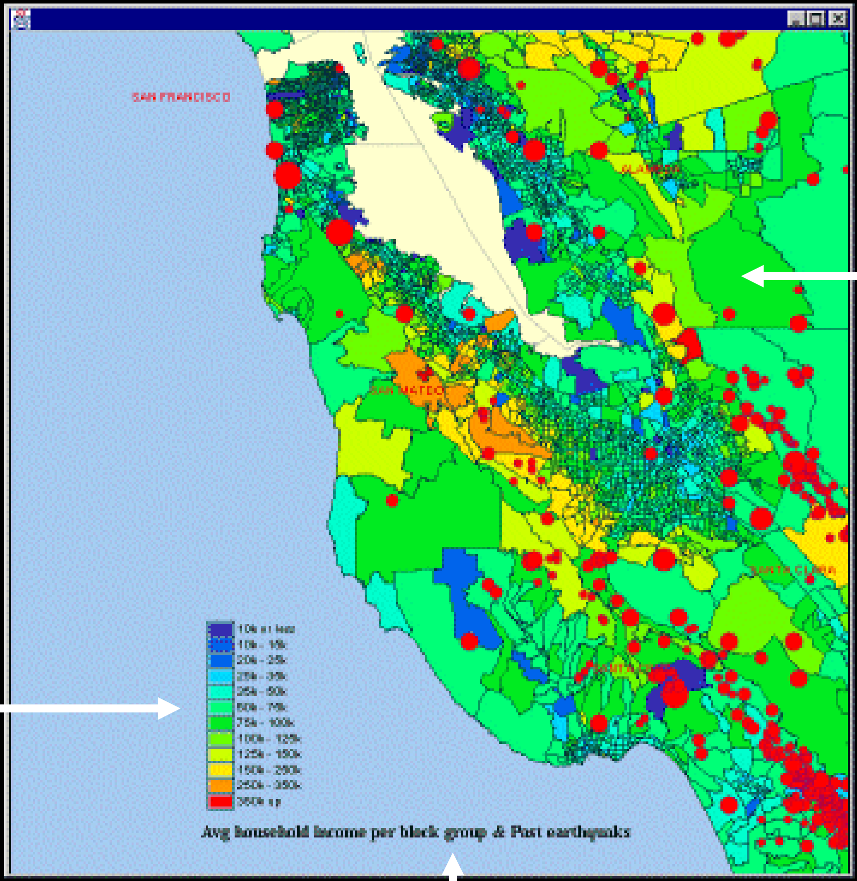
Map Request

- Client sends a **map** request to MapViewer…
  MapViewer returns a map to client…

MapViewer Concepts

What is a Map?

Map = basemap + additional themes + additional features + legend + title + footnote…



Avg household income per block group & Past earthquaks

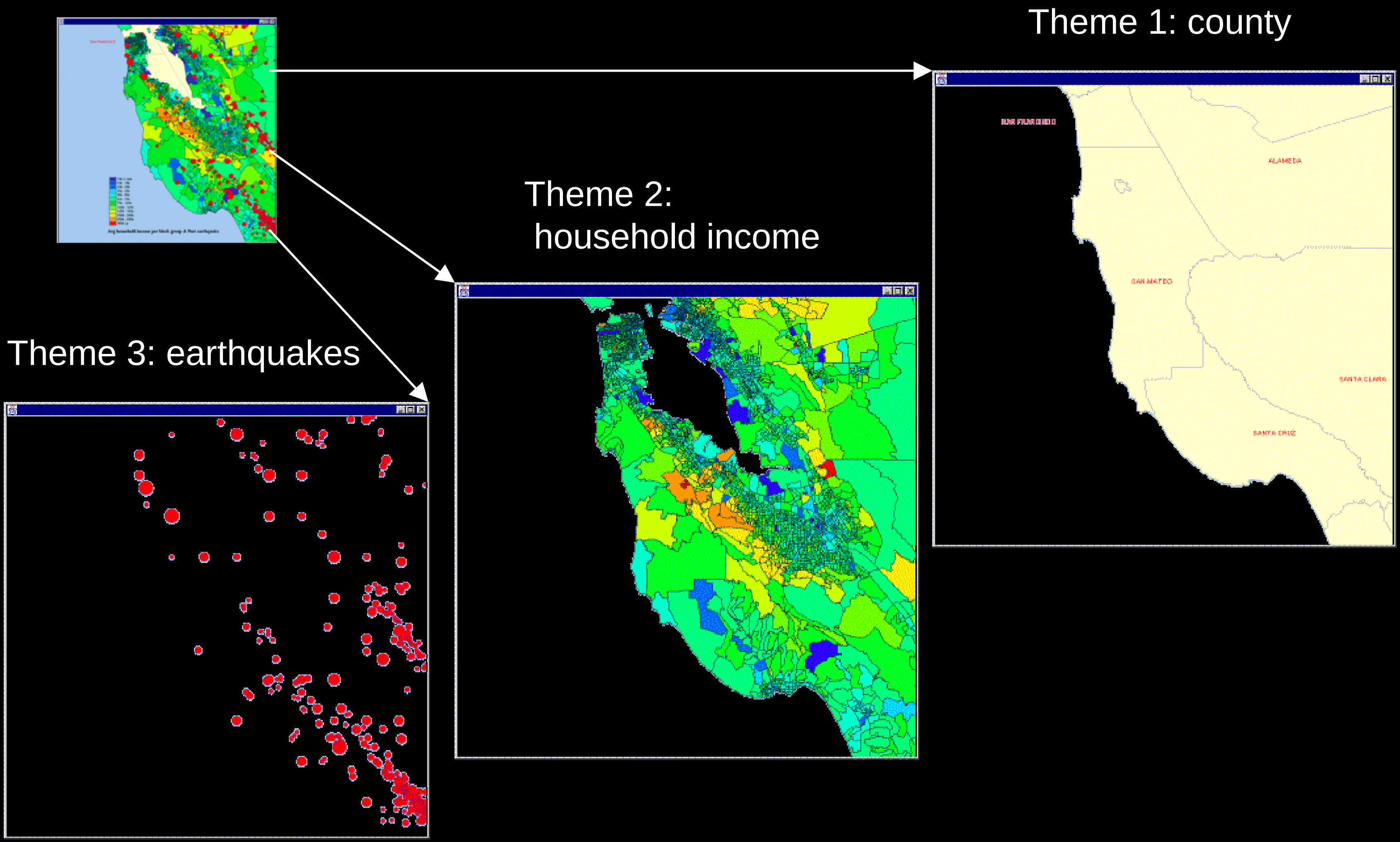
Themes

Legend

Title

MapViewer Concepts: basemap

A collection of predefined themes

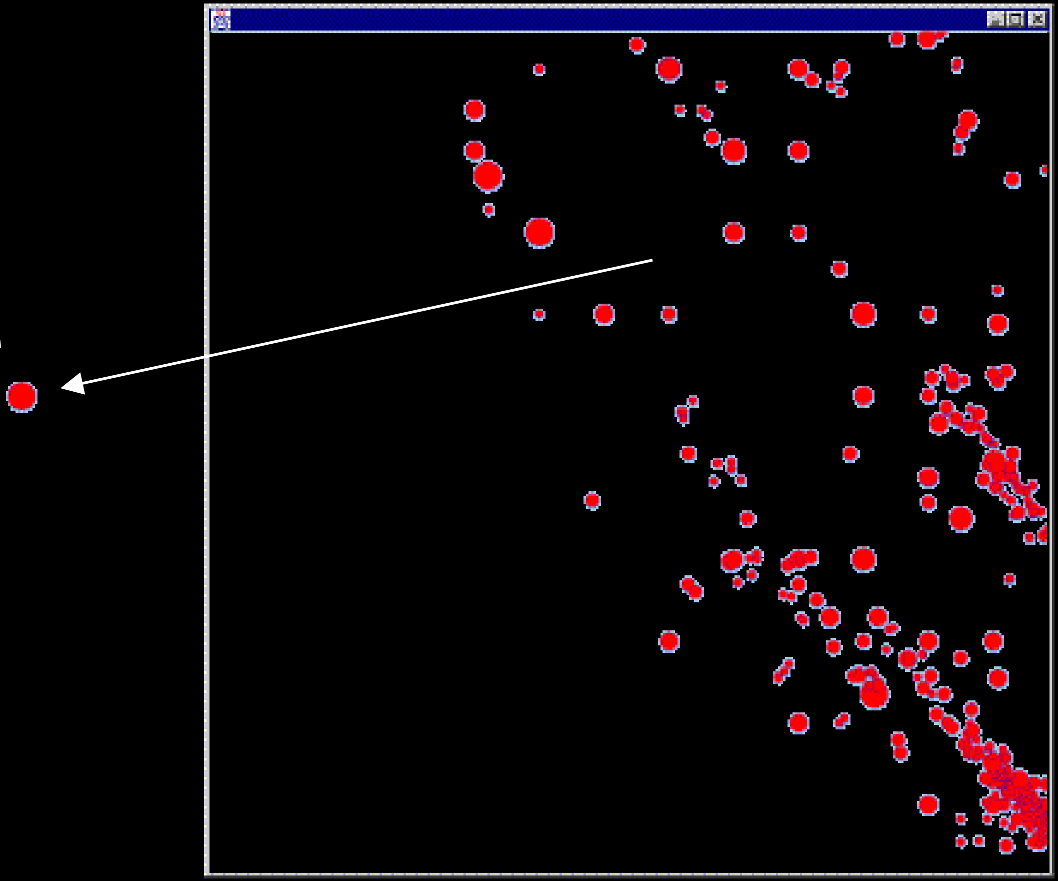


Theme 1: county

Theme 2:
household income

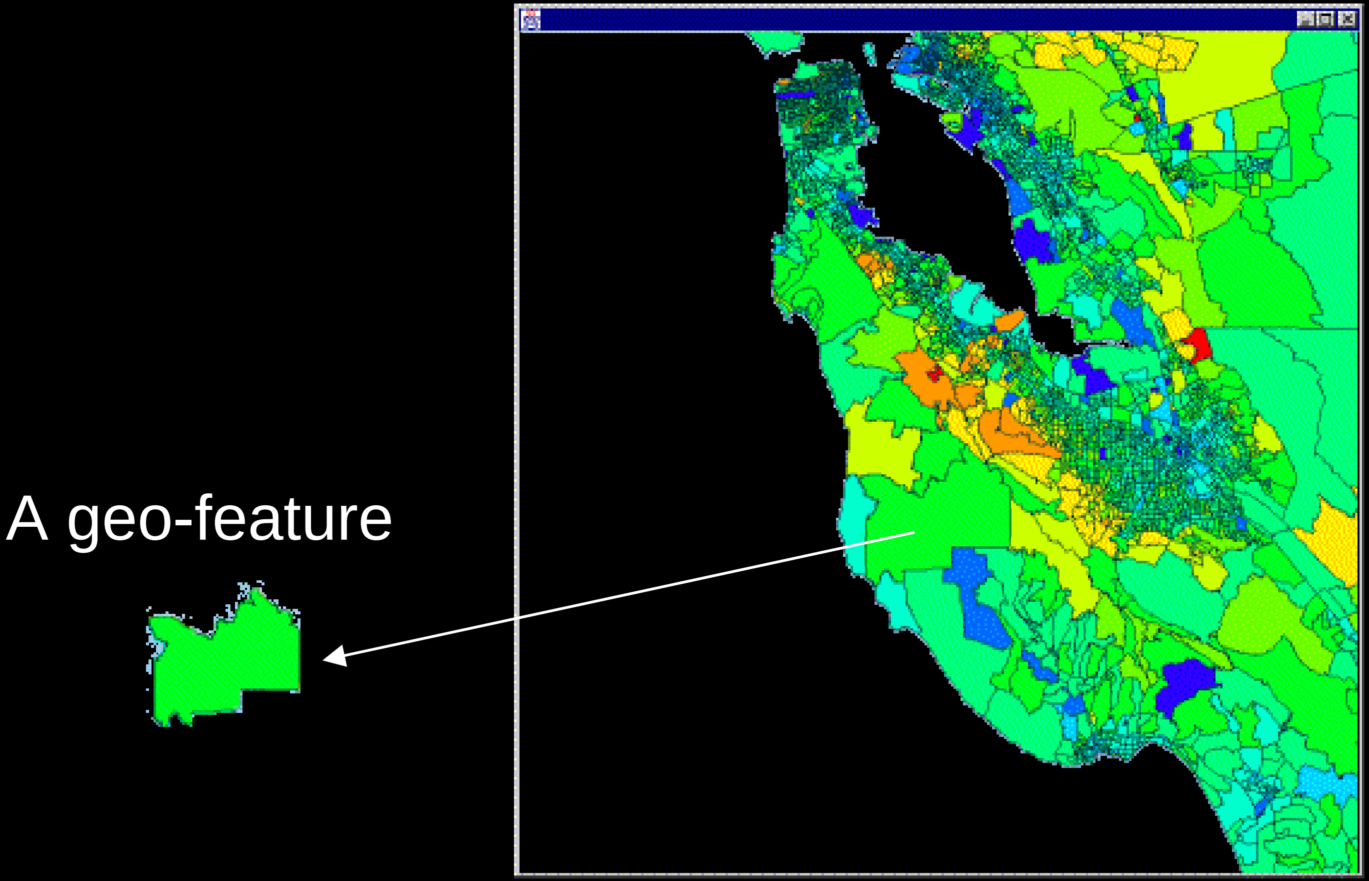
Theme 3: earthquakes

Theme = a collection of Geo-Features



A geo-feature

Theme = a collection of Styled Geo-Features



A geo-feature

Themes Provide the Data for a Map Request

A map request can contain any combination of the following types of themes:

- Predefined themes implicitly included in a basemap.
- JDBC themes that provide dynamic SQL queries.
- Explicitly referenced predefined themes whose definitions are stored in a database.
- User-supplied individual features (grouped into a single theme at the server-side).

Note: you can have at most 1 basemap per map request.

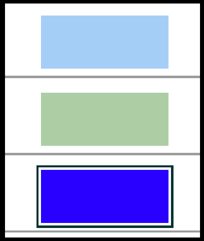
A Geo-feature = geometry + attributes + styling information

population, sales, …

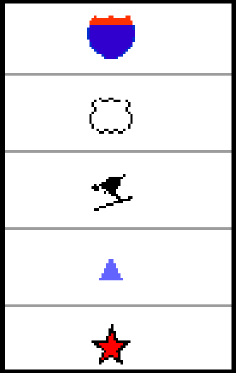
Vector data
from database or map request

- Feature style (color, marker, line/area pattern or advanced)
- Label style (font/color)

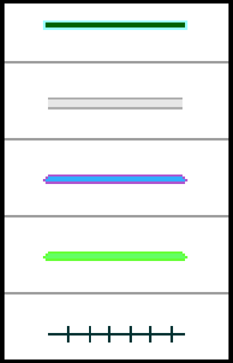
Style : There are 6 types of styles in MapViewer



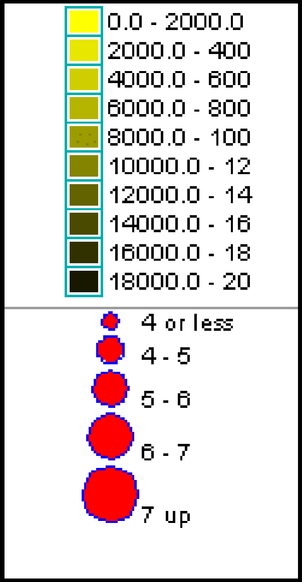
COLOR
(applicable to any geometry)



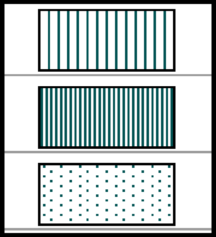
MARKER
(points)



LINE
(linestrings)



ADVANCED
(thematic mapping)



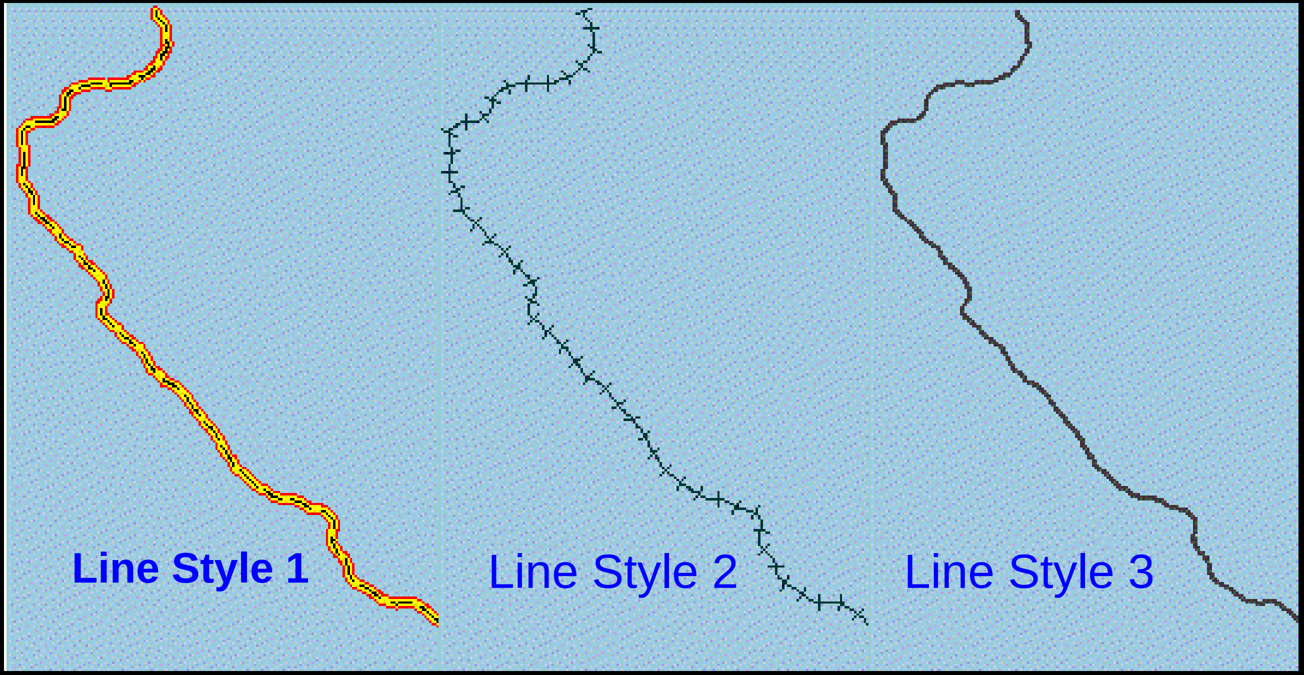
AREA
(polygons)

Hello World
Hello World
Hello Wo

TEXT
(labels)

Style Applies to Individual Geometries

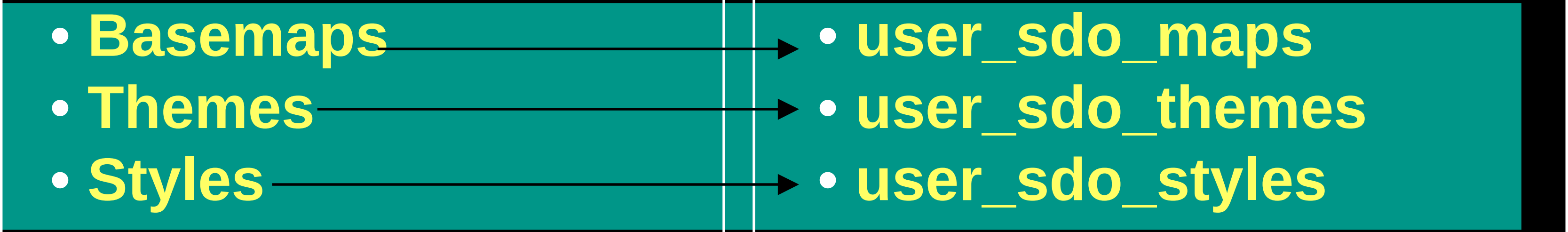
Each geometry can appear differently when rendered using different styles. For instance:



MapViewer Key Concepts - Metadata

The definitions of base maps, predefined themes and styles are the metadata for mapping:

All mapping metadata are stored in the database.
Three views are automatically created for each database user to store them:



- **Basemaps**
- **Themes**
- **Styles**

- **user_sdo_maps**
- **user_sdo_themes**
- **user_sdo_styles**

Topics

- Overview
- Concepts
- Main Features
- APIs
- Fast start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

Key Features in This Release

- Enhanced Mapping and Visualization Capabilities
- Enhanced APIs and JDeveloper Integration
- Enhanced Administrative Functions

Enhanced Mapping and Visualization Capabilities

- Supports complex thematic mapping.
- Integrated visualization of geo-referenced imagery and vector data.
- GeoRaster theme & Customizable Image Renderer.
- Spatial 10g Network and Topology themes
- Generates maps in SVG and JPEG formats.
- Improved area/polygon label placement algorithm.
- Supports seamless browsing of global data using a built-in globular projection (experimental in this release).

Enhanced Mapping and Visualization Capabilities – continued

- Multi-threaded geometry loading from database.
- Automatic caching of geometry and imagery data in MapViewer.
- Pre-caching of entire themes in memory (eliminates geometry fetching from database for fixed geometry layers)
- Map legend support
- Dynamic theme support through JDBC themes.
- Multiple datasource support.
- Sticky label support: force a label to appear regardless of conflicts.

Enhanced Administrative Functions

- Restart MapViewer without restarting the container itself.
- Administrative functions are now secured through a login page.
- Supports invalidation of cached data by theme.
- More options/controls in the mapViewerConfig.xml file.
- Permanent data sources with automatically encrypted passwords can be defined in the mapViewerConfig.xml file.

Topics

- Overview
- Main features
- Concepts
- APIs
- Fast start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

MapViewer API

MapViewer supports 3 API flavors:

- XML-based : the ultimate API
- Java thin library : a mapping “bean” (w/o UI)
- JSP custom tags : a subset of functions.

MapViewer XML API

3 types of XML requests:

- Map-Request: requesting a map
- Info-Request: searching for attribute info based on locations
- Non-Map-Request: for administrative or metadata requests

The XML API is the native language to the MapViewer !

The XML DTD, examples and usage notes are detailed in MapViewer User's Guide !

MapViewer XML API

- Map-Request

Partial DTD of Map-Request:

```
<!ELEMENT map_request  ((box | center)?, srs?, legend?, themes?, geoFeature*)>
<!ATTLIST map_request
      datasource    CDATA #REQUIRED
      basemap       CDATA #IMPLIED
      width         CDATA #IMPLIED
      height        CDATA #IMPLIED
      antialiasing  "FALSE" #IMPLIED
      format (GIF_STREAM|GIF_URL|PNG_STREAM|PNG_URL)  #IMPLIED
      title         CDATA #IMPLIED
      bgcolor       CDATA #IMPLIED
      bgimage       CDATA #IMPLIED >
```

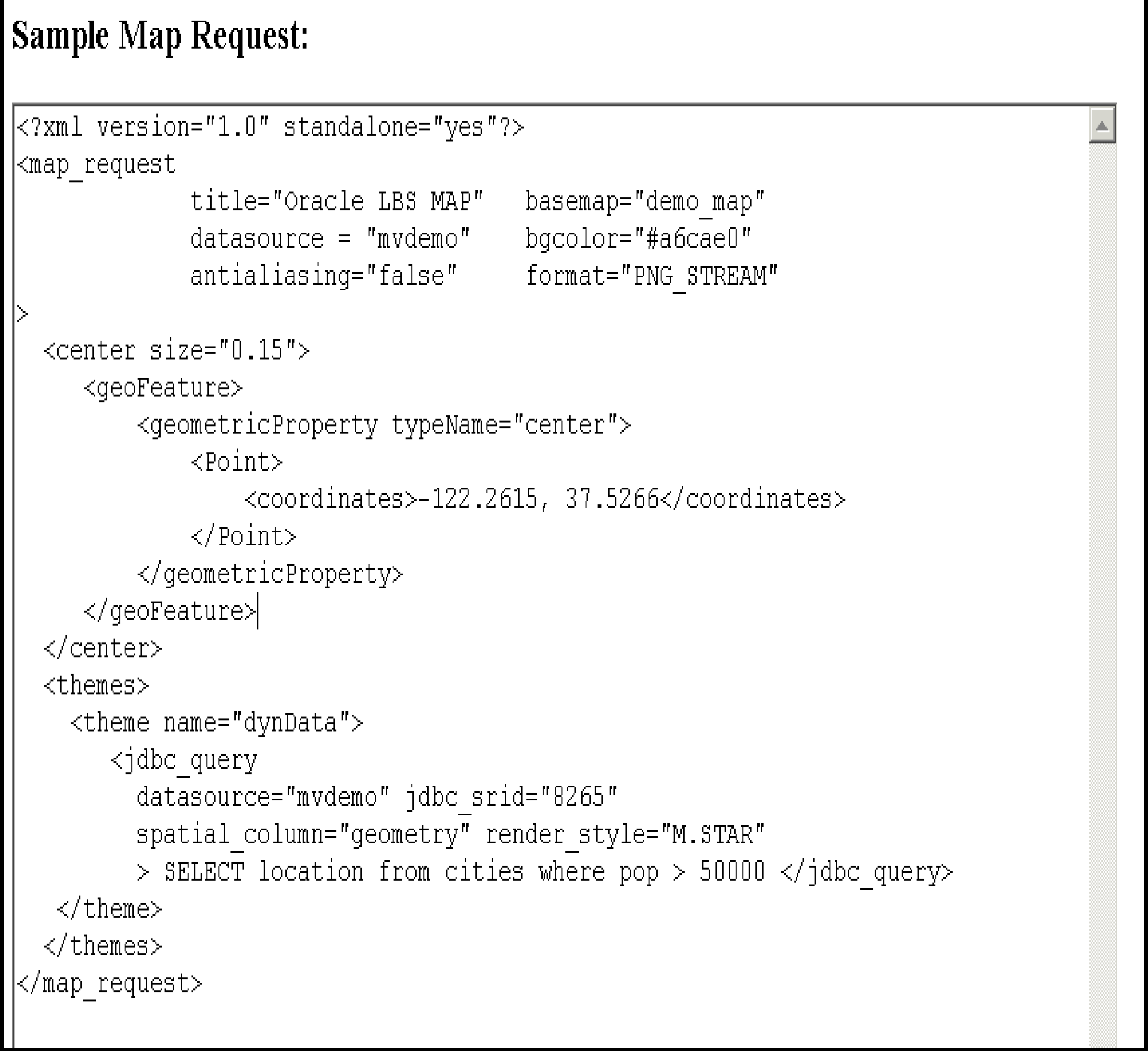
For complete DTD please refer to MapViewer User’s guide.

MapViewer XML API

- Map-Request

In XML jargon:
A <map_request> element must define a datasource as one of its attributes, and can optionally include a <basemap>, a <center>, list of <theme>s, <style>s and individual <GeoFeature>s as its child-nodes. It may also include a <legend> element.

MapViewer XML API

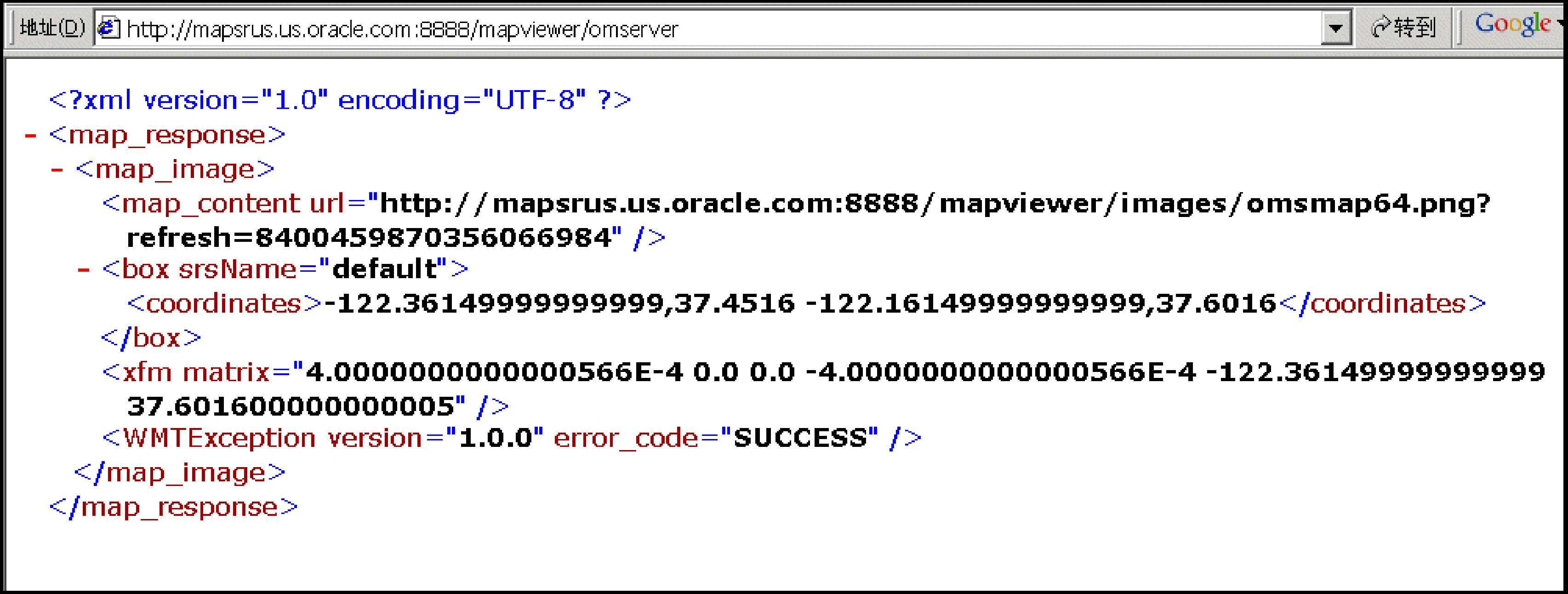


Sample Map Request:

```
<?xml version="1.0" standalone="yes"?>
<map_request
             title="Oracle LBS MAP"   basemap="demo_map"
             datasource = "mvdemo"    bgcolor="#a6cae0"
             antialiasing="false"     format="PNG_STREAM"
>
  <center size="0.15">
     <geoFeature>
         <geometricProperty typeName="center">
             <Point>
                 <coordinates>-122.2615, 37.5266</coordinates>
             </Point>
         </geometricProperty>
     </geoFeature>
  </center>
  <themes>
    <theme name="dynData">
       <jdbc_query
         datasource="mvdemo" jdbc_srid="8265"
         spatial_column="geometry" render_style="M.STAR"
         > SELECT location from cities where pop > 50000 </jdbc_query>
   </theme>
  </themes>
</map_request>
```

MapViewer XML API

- Sample Map Response



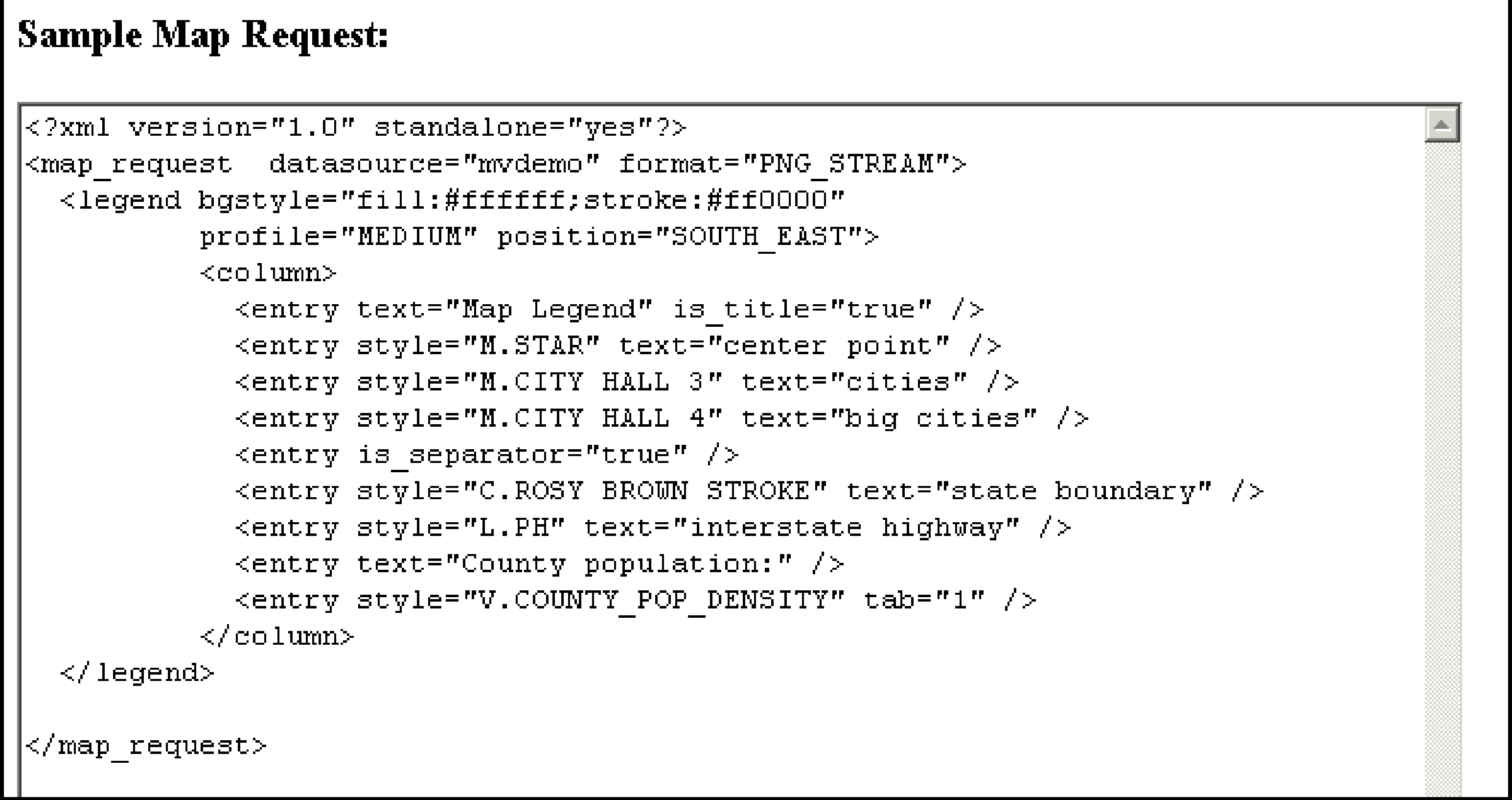
```
  <?xml version="1.0" encoding="UTF-8" ?>
- <map_response>
  - <map_image>
      <map_content url="http://mapsrus.us.oracle.com:8888/mapviewer/images/omsmap64.png?
        refresh=8400459870356066984" />
    - <box srsName="default">
        <coordinates>-122.36149999999999,37.4516 -122.16149999999999,37.6016</coordinates>
      </box>
      <xfm matrix="4.0000000000000566E-4 0.0 0.0 -4.0000000000000566E-4 -122.36149999999999
        37.601600000000005" />
      <WMTException version="1.0.0" error_code="SUCCESS" />
    </map_image>
  </map_response>
```

Note that a map named omsmap64.png has been created as result of the map request, and it can be viewed using the URL returned in the map response above.

MapViewer XML API

- Map-Request for legend only

You can request just a legend image using a Map Request that embeds a legend specification only.



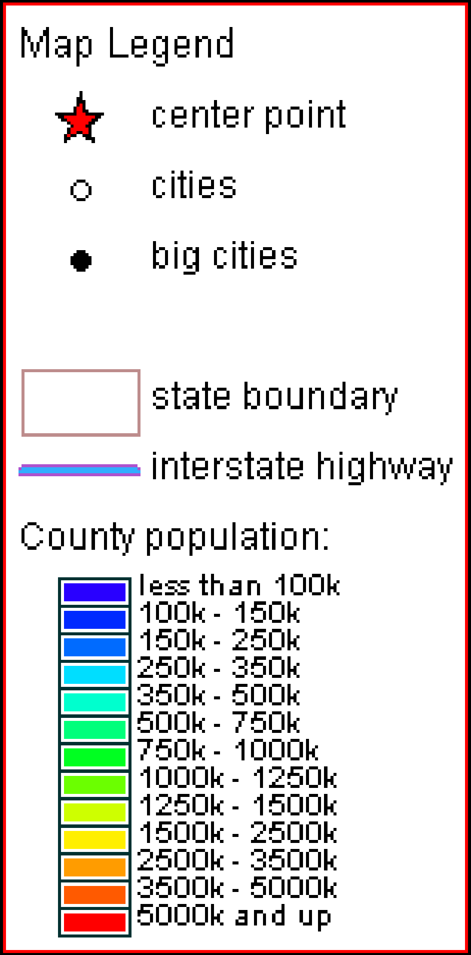
Sample Map Request:

```
<?xml version="1.0" standalone="yes"?>
<map_request  datasource="mvdemo" format="PNG_STREAM">
  <legend bgstyle="fill:#ffffff;stroke:#ff0000"
          profile="MEDIUM" position="SOUTH_EAST">
          <column>
            <entry text="Map Legend" is_title="true" />
            <entry style="M.STAR" text="center point" />
            <entry style="M.CITY HALL 3" text="cities" />
            <entry style="M.CITY HALL 4" text="big cities" />
            <entry is_separator="true" />
            <entry style="C.ROSY BROWN STROKE" text="state boundary" />
            <entry style="L.PH" text="interstate highway" />
            <entry text="County population:" />
            <entry style="V.COUNTY_POP_DENSITY" tab="1" />
          </column>
  </legend>

</map_request>
```

MapViewer XML API

- Response for legend-only map request



Map Legend

★ center point

○ cities

● big cities

□ state boundary

interstate highway

County population:

less than 100k
100k - 150k
150k - 250k
250k - 350k
350k - 500k
500k - 750k
750k - 1000k
1000k - 1250k
1250k - 1500k
1500k - 2500k
2500k - 3500k
3500k - 5000k
5000k and up

MapViewer XML API

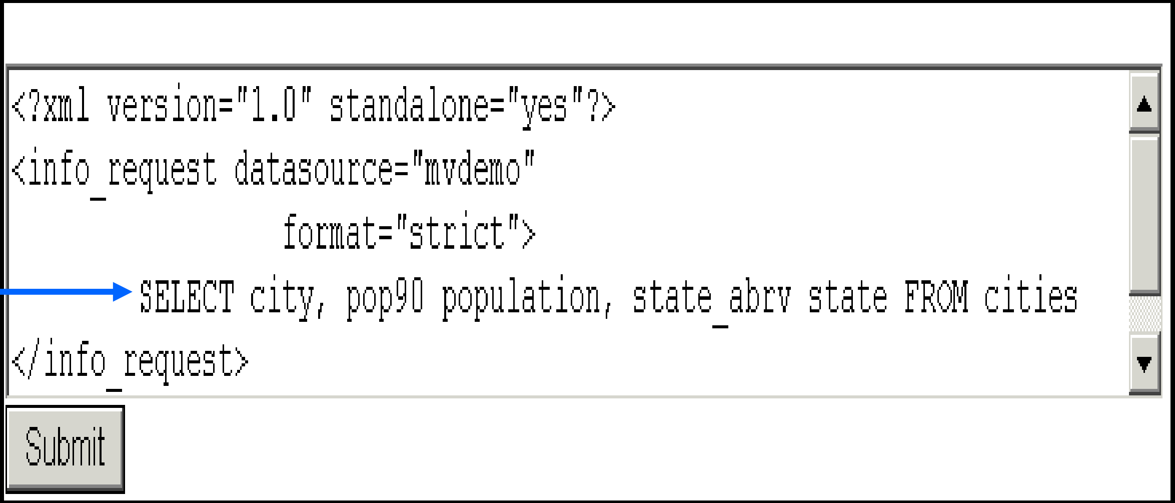
- Info-Request

Search/identify non-spatial attributes

- Search using direct SQL query
- Search Within Radius
- Search At Point
- Search Within Area
- Search Within Rectangle
- Search Nearest Neighbor

MapViewer XML API

- Info-Request : sample



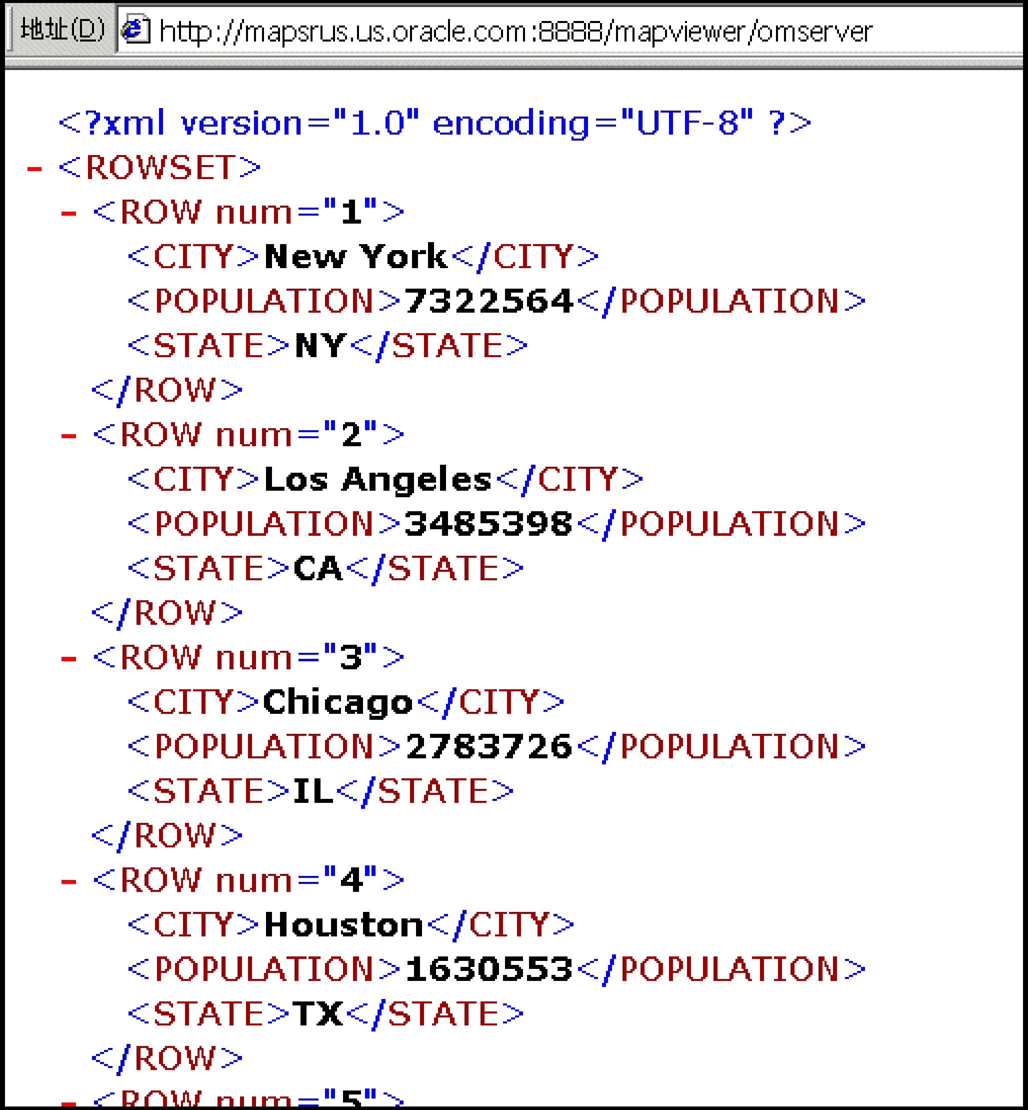
```
<?xml version="1.0" standalone="yes"?>
<info_request datasource="mvdemo"
                  format="strict">
        SELECT city, pop90 population, state_abrv state FROM cities
</info_request>
```

Direct SQL query

MapViewer XML API

- Info-Response

MapViewer returns XSQL-like response.



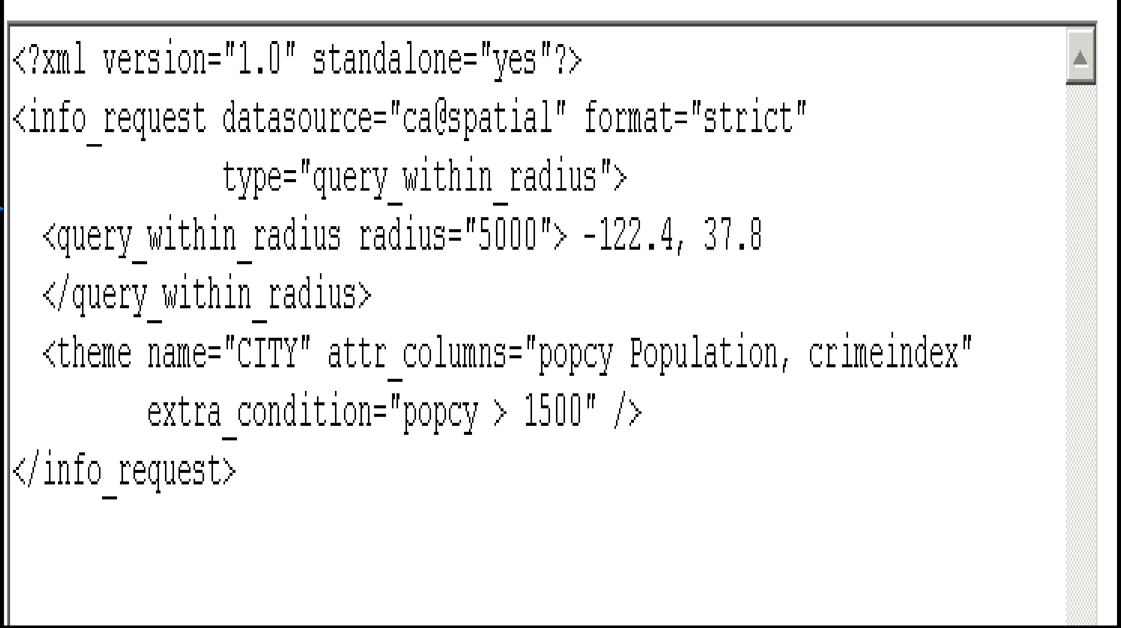
http://mapsrus.us.oracle.com:8888/mapviewer/omserver

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <ROWSET>
  - <ROW num="1">
      <CITY>New York</CITY>
      <POPULATION>7322564</POPULATION>
      <STATE>NY</STATE>
    </ROW>
  - <ROW num="2">
      <CITY>Los Angeles</CITY>
      <POPULATION>3485398</POPULATION>
      <STATE>CA</STATE>
    </ROW>
  - <ROW num="3">
      <CITY>Chicago</CITY>
      <POPULATION>2783726</POPULATION>
      <STATE>IL</STATE>
    </ROW>
  - <ROW num="4">
      <CITY>Houston</CITY>
      <POPULATION>1630553</POPULATION>
      <STATE>TX</STATE>
    </ROW>
  - <ROW num="5">
```

MapViewer XML API

- Info-Request : sample

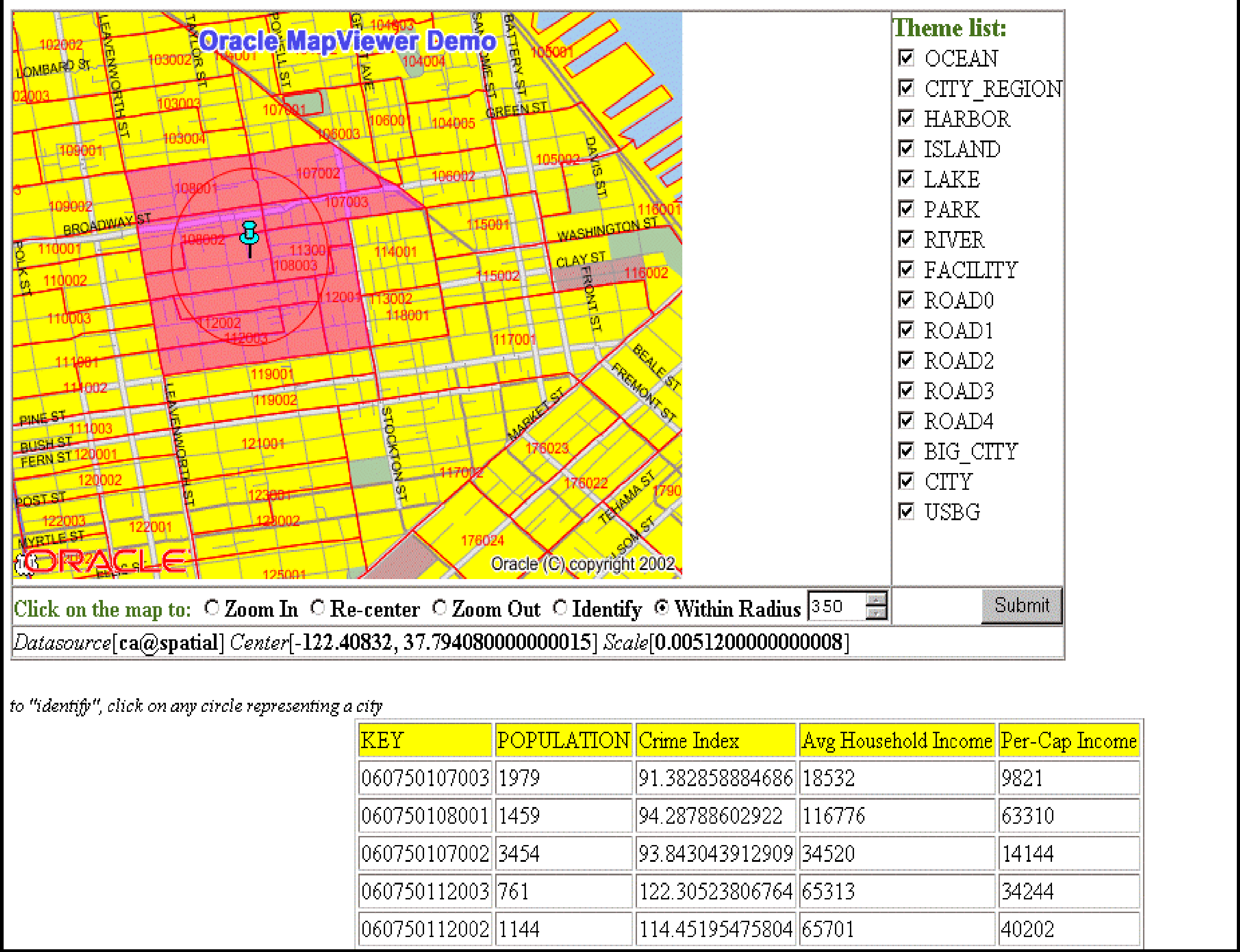
Search Within Radius



```
<?xml version="1.0" standalone="yes"?>
<info_request datasource="ca@spatial" format="strict"
              type="query_within_radius">
  <query_within_radius radius="5000"> -122.4, 37.8
  </query_within_radius>
  <theme name="CITY" attr_columns="popcy Population, crimeindex"
         extra_condition="popcy > 1500" />
</info_request>
```

MapViewer XML API

- When Info Request and Map Request are combined:



Theme list:

☑ OCEAN
☑ CITY_REGION
☑ HARBOR
☑ ISLAND
☑ LAKE
☑ PARK
☑ RIVER
☑ FACILITY
☑ ROAD0
☑ ROAD1
☑ ROAD2
☑ ROAD3
☑ ROAD4
☑ BIG_CITY
☑ CITY
☑ USBG

Click on the map to: ○ Zoom In ○ Re-center ○ Zoom Out ○ Identify ● Within Radius 350 Submit

Datasource[ca@spatial] *Center*[-122.40832, 37.794080000000015] *Scale*[0.005120000000000008]

to "identify", click on any circle representing a city

KEY	POPULATION	Crime Index	Avg Household Income	Per-Cap Income
060750107003	1979	91.382858884686	18532	9821
060750108001	1459	94.28788602922	116776	63310
060750107002	3454	93.843043912909	34520	14144
060750112003	761	122.30523806764	65313	34244
060750112002	1144	114.45195475804	65701	40202

MapViewer XML API

- Non-Map-Requests

MapViewer supports following types of admin/metadata requests:

- Managing Data Sources (add/remove/list/redefine datasources)
- Listing All predefined base Maps in a datasource
- Listing predefined Themes (in a datasource, and are part of a base Map)
- Managing Styles (adding/listing styles)
- Clearing Metadata Cache for a Data Source
- Clearing Cache for a predefined theme
- Restarting the MapViewer instance

For details and DTDs please refer to the MapViewer User’s Guide.

MapViewer XML API

The XML API is the most powerful and flexible.

You can manually construct a XML map request, and send it to MapViewer using the HTTP protocol from any programming language that supports HTTP connection and messaging. For instance you can even use PL/SQL from inside an Oracle Database to send a map request outside to the MapViewer and have it generate a map.

But it is often difficult to directly manipulate XML documents in an application. That's why there is a Java API!

MapViewer Java API

- Java API

An easier to use API than XML. Provides a “bean”-like interface.

Eventually the request is quietly converted into an XML doc.

The library is named mvclient.jar, and can be found in the unpacked directory of a MapViewer deployment: mapviewer/web/WEB-INF/lib

A code segment using the lib:

```
import oracle.spatial.mapclient.MapViewer;  // this class is ALL you need.

MapViewer bean = new MapViewer(“http://mapsrus.com:8888/mapviewer/omserver”);

// sets up a request
bean.setDataSource(“ca@spatial”);
bean.setBaseMapName(“us_base”);
bean.setBackgroundColor(Color.white);
bean.setCenterAndSize(-122.4, 37.8, 0.5);
bean.addJDBCTheme(“ca@spatial”, “dynData”, “select location from cities where “+
                              “pop > 50000”, “location”, “M.STAR”, NULL, NULL, true);
bean.run();  //issues the request

String mapURL = bean.getGeneratedMapImageURL();

// display the map image

bean.zoomIn(100,200, 2.0);
mapURL = bean.getGeneratedMapImageURL();

// display the zoomed in map image.
```

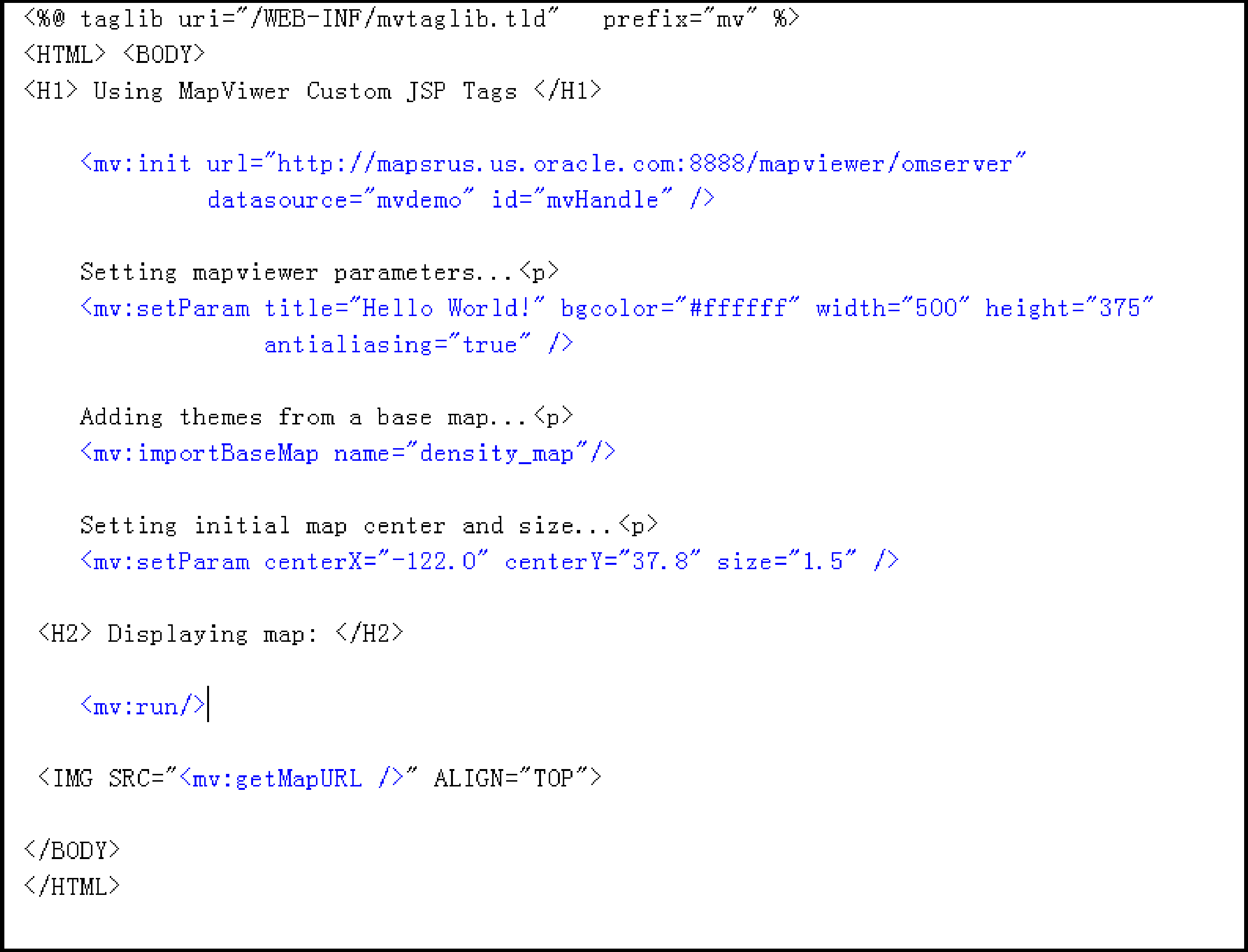
JavaDoc is @: http://<host>:<port>/mapviewer/mapclient

MapViewer API – JSP taglib

- Provides a simple set of custom tags

Does not include all functions in the Java API or the XML API. Meant as a fast start for beginners.

A code segment using the custom tags:



```
<%@ taglib uri="/WEB-INF/mvtaglib.tld"   prefix="mv" %>
<HTML> <BODY>
<H1> Using MapViwer Custom JSP Tags </H1>

    <mv:init url="http://mapsrus.us.oracle.com:8888/mapviewer/omserver"
             datasource="mvdemo" id="mvHandle" />

    Setting mapviewer parameters...<p>
    <mv:setParam title="Hello World!" bgcolor="#ffffff" width="500" height="375"
                 antialiasing="true" />

    Adding themes from a base map...<p>
    <mv:importBaseMap name="density_map"/>

    Setting initial map center and size...<p>
    <mv:setParam centerX="-122.0" centerY="37.8" size="1.5" />

 <H2> Displaying map: </H2>

    <mv:run/>

 <IMG SRC="<mv:getMapURL />" ALIGN="TOP">

</BODY>
</HTML>
```

The tag library is also in mvclient.jar, its TLD file is MapViewer deployment: mapviewer/web/WEB-INF/lib

Topics

- Overview
- Main features
- Concepts
- APIs
- Fast Start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

MapViewer Fast Start

How to start if you have never used MapViewer before:

- Download latest Quick Start kit from the OTN MapViewer page
- Download the Map Definition Tool (mapdef.jar) utility from OTN under Spatial page.
- Start up Map Definition Tool to “style” your existing spatial data.
- If you do not have any spatial data, download the demo data from OTN, follow instruction to import it into the your database.
- start up OC4J and navigate to http://<host>:<port>/mapviewer page, and have fun!

MapViewer Fast Start

- Deliverables
  - A J2EE .ear file (includes code, sample config files et al.) shipped with Oracle Application Server.
  - A map definition tool utility to manage mapping metadata (download from OTN only)
- Can be deployed in a standalone OC4j (Oracle Container for J2EE) or full Oracle Application Server.
- Configured to run out of the box for new users.

MapViewer Fast Start

Advanced users can configure the MapViewer to:

- save generated maps at a different location, with a customizable life-cycle. Also define how the maps should be referenced using URL.
- output different levels/volumes of logging messages.
- use global map parameters such as: map title, copyright notes, map logo image. Can customize font, position on map.
- customize spatial data cache size or turn it off completely.
- dynamically load custom Image renderers (user classes).
- predefine known data sources.

All by editing the mapViewerConfig.xml file!

Topics

- Overview
- Main features
- Concepts
- APIs
- Fast start
- Map Definition Tool OTN utility: Enterprise-level management of map metadata
- Web resources

Map Definition Tool OTN Utility

Recall that

Mapping Metadata =
definitions of basemaps, predefined themes & styles

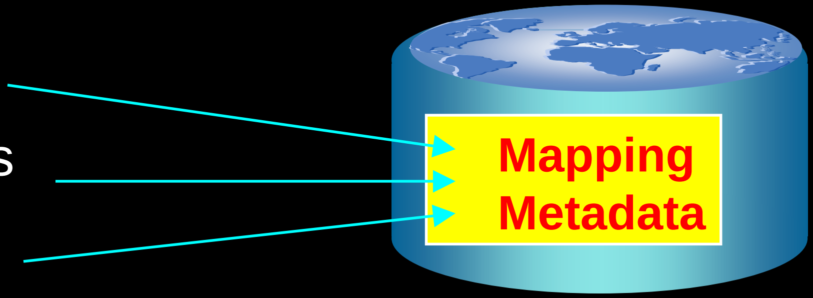
so that

map metadata be shared among many users in an enterprise.

Map Definition Tool OTN utility enables enterprise-wide management of map metadata

Metadata for Mapping

- Three views:
  - user_sdo_maps
  - user_sdo_themes
  - user_sdo_styles



- Users customize their maps by modifying contents in these three metadata views.
- Users modify/manage these views through the Map Definition Tool OTN utility (or directly through SQL).

View USER_SDO_MAPS

user_sdo_maps (name varchar2, description varchar2,
definition CLOB);

- name uniquely references to a basemap, e.g. “us_base”
- definition is an XML document that lists all the themes included in this basemap.

A Sample basemap definition doc:

```
<? XML version=“1.0” >
<MapDefinition  title=“eLocation Customers” >
    <Theme name=“us_state”  min_scale=“0.5” max_scale=“0.002” />
    <Theme name=“us_road_0” min_scale=“0.2” max_scale=“0.002” />
    <Theme name=“us_road_1” />
    <Theme name=“us_parks” min_scale=“0.3” max_scale=“0.05” />
    <Theme name=“us_lakes” min_scale=“0.3” max_scale=“0.05” />
</MapDefinition>
```

View USER_SDO_THEMES

```
user_sdo_themes (name               varchar2,
                 description        varchar2
                 base_table          varchar2,
                 geometry_column    varchar2,
                 styling_rules      CLOB );
```

A Sample styling_rules column:

```
<?xml version=“1.0” ?>
<styling_rules>
        <rule>
                <features style=“m.airport”>  name like ‘% INTL AIRPORT” </features>
                <label column=“substr(name, 1, 3)”  style=“t.airport name”>  1 </lable>
        <rule>
</styling_rules>
```

View USER_SDO_THEMES – contd.

Anatomy of the styling_rules column:

- An XML doc with root element <**styling_rules**> and child nodes <**rule>**s.
- Each <**rule**> node has a mandatory <**feature**> node, and an optional <**label**> node.
- The <**feature**> node specifies features/rows to be included in this theme, using any SQL WHERE clause. It has one attribute “style” specifying the name of the style used in rendering.
- The optional <**label**> node specifies whether the features should be labeled. Any value/SQL expression greater than 0 means labeling is required. The “column” attribute node designates a column in the base table whose values will be used as the label text. The “style” attribute specifies the text style for labeling.

```
<?xml version=“1.0” ?>
<styling_rules>
       <rule>
               <features style=“m.airport”>  name like ‘% INTL AIRPORT” </features>
              <label column=“name”  style=“t.airport name”>  1 </lable>
       <rule>
</styling_rules>
```

View USER_SDO_STYLES

```
user_sdo_styles (name             varchar2,
                 type             varchar2
                description       varchar2,
                definition        CLOB,
                image             BLOB,
                geometry          MDSYS.SDO_GEOMETRY );
```

All six types of styles are stored in this schema. For details please see the MapViewer User’s Guide.

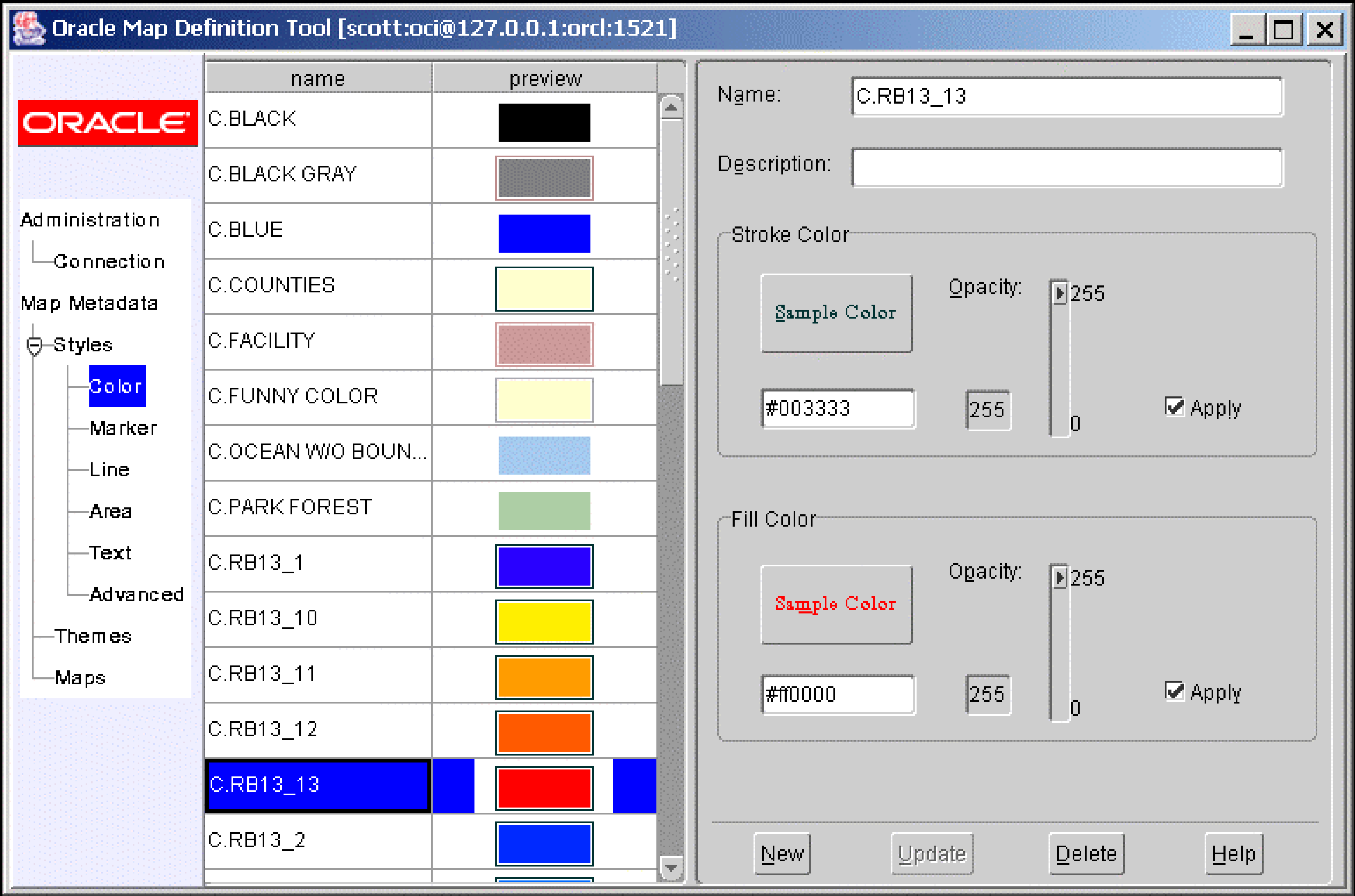
Web Resources

- Oracle Application Server MapViewer site on OTN: http://otn.oracle.com/products/mapviewer
- Oracle Spatial site on OTN: http://otn.oracle.com/products/spatial
- Oracle Discussion Forum on Spatial/MapViewer: http://forums.oracle.com/forums/forum.jsp?forum=76

Appendix

Screen Shots

Screen Shots of Map Definition Tool – Managing Color Styles



name	preview
C.BLACK	⬛
C.BLACK GRAY	⬛
C.BLUE	🟦
C.COUNTIES	🟨
C.FACILITY	🟫
C.FUNNY COLOR	🟨
C.OCEAN W/O BOUN...	🟦
C.PARK FOREST	🟩
C.RB13_1	🟦
C.RB13_10	🟨
C.RB13_11	🟧
C.RB13_12	🟧
C.RB13_13	🟥
C.RB13_2	🟦

Name: C.RB13_13

Description:

Stroke Color

Sample Color

Opacity: 255

#003333 255 0 ☑ Apply

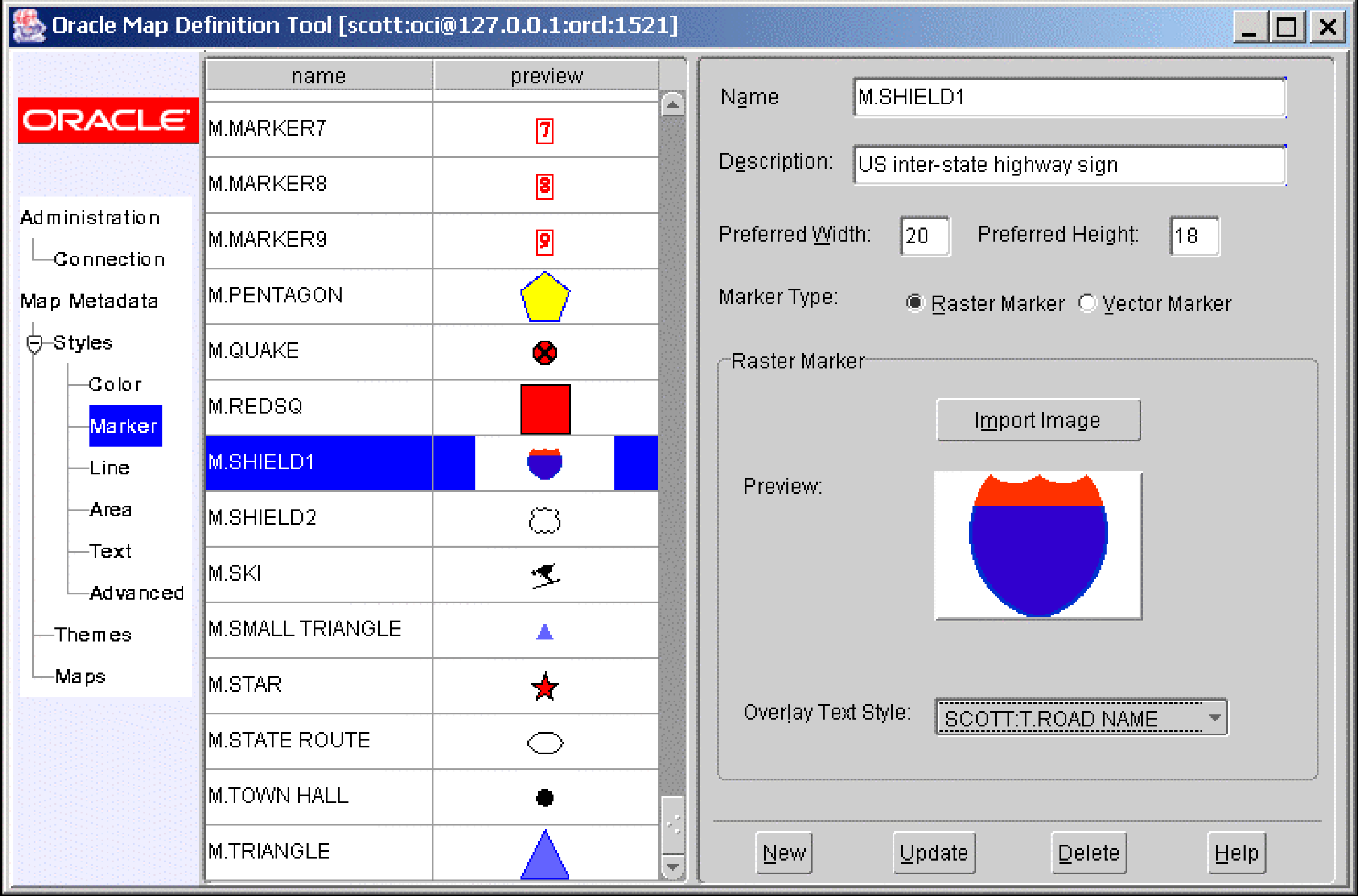
Fill Color

Sample Color

Opacity: 255

#ff0000 255 0 ☑ Apply

Screen Shots of Map Definition Tool - Marker Styles



Oracle Map Definition Tool [scott:oci@127.0.0.1:orcl:1521]

name	preview
M.MARKER7	7
M.MARKER8	8
M.MARKER9	9
M.PENTAGON	⬟
M.QUAKE	⊗
M.REDSQ	🟥
M.SHIELD1	
M.SHIELD2	
M.SKI	
M.SMALL TRIANGLE	▲
M.STAR	★
M.STATE ROUTE	⬭
M.TOWN HALL	●
M.TRIANGLE	▲

Name: M.SHIELD1

Description: US inter-state highway sign

Preferred Width: 20 Preferred Height: 18

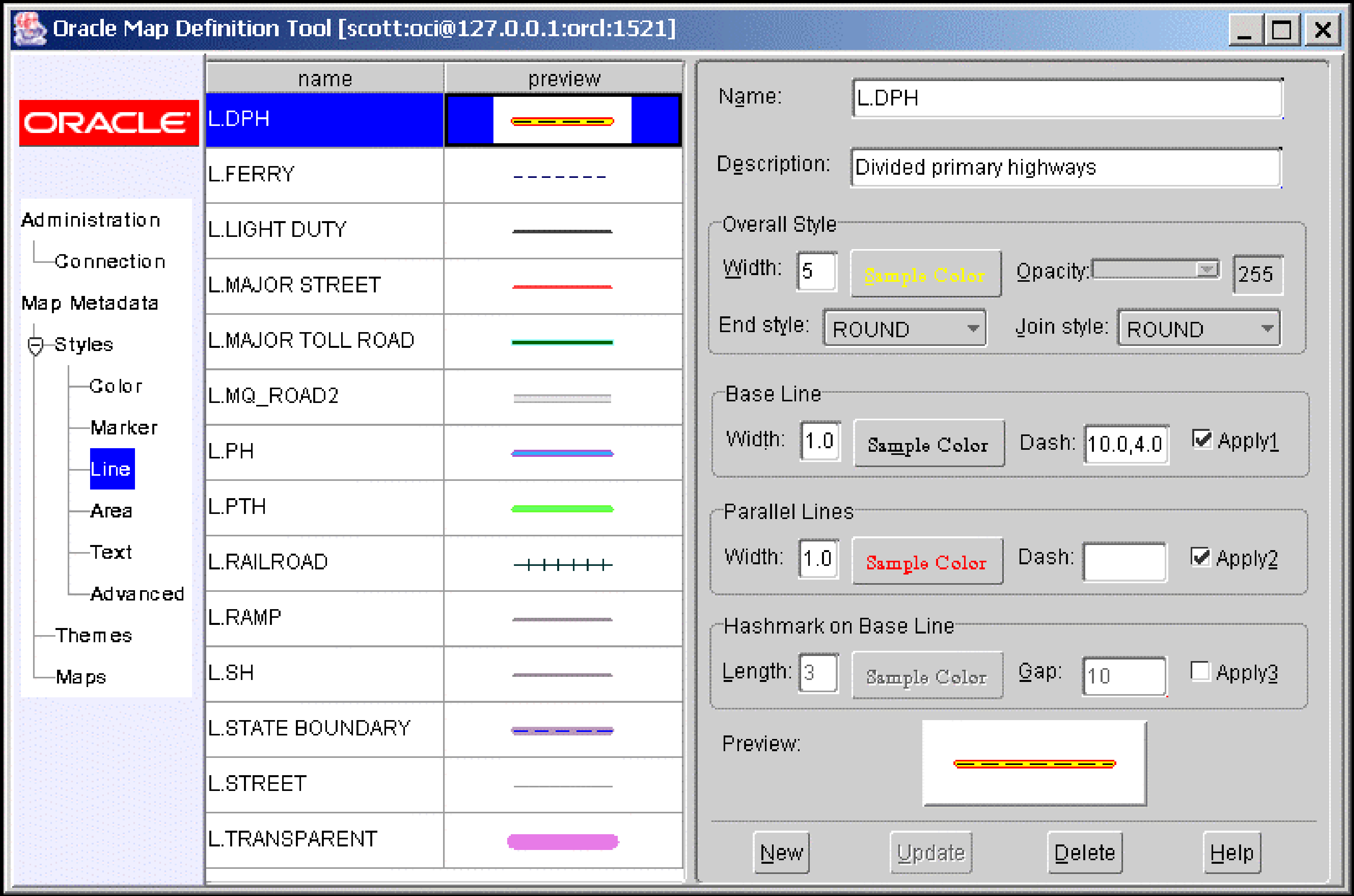
Marker Type: ● Raster Marker ○ Vector Marker

Raster Marker

Preview:

Overlay Text Style: SCOTT:T.ROAD NAME

Screen Shots of Map Definition Tool - Line Styles



Oracle Map Definition Tool [scott:oci@127.0.0.1:orcl:1521]

name	preview
L.DPH	
L.FERRY	
L.LIGHT DUTY	
L.MAJOR STREET	
L.MAJOR TOLL ROAD	
L.MQ_ROAD2	
L.PH	
L.PTH	
L.RAILROAD	
L.RAMP	
L.SH	
L.STATE BOUNDARY	
L.STREET	
L.TRANSPARENT	

Name: L.DPH

Description: Divided primary highways

Overall Style

Width: 5 Sample Color Opacity: 255

End style: ROUND Join style: ROUND

Base Line

Width: 1.0 Sample Color Dash: 10.0,4.0 ☑ Apply1

Parallel Lines

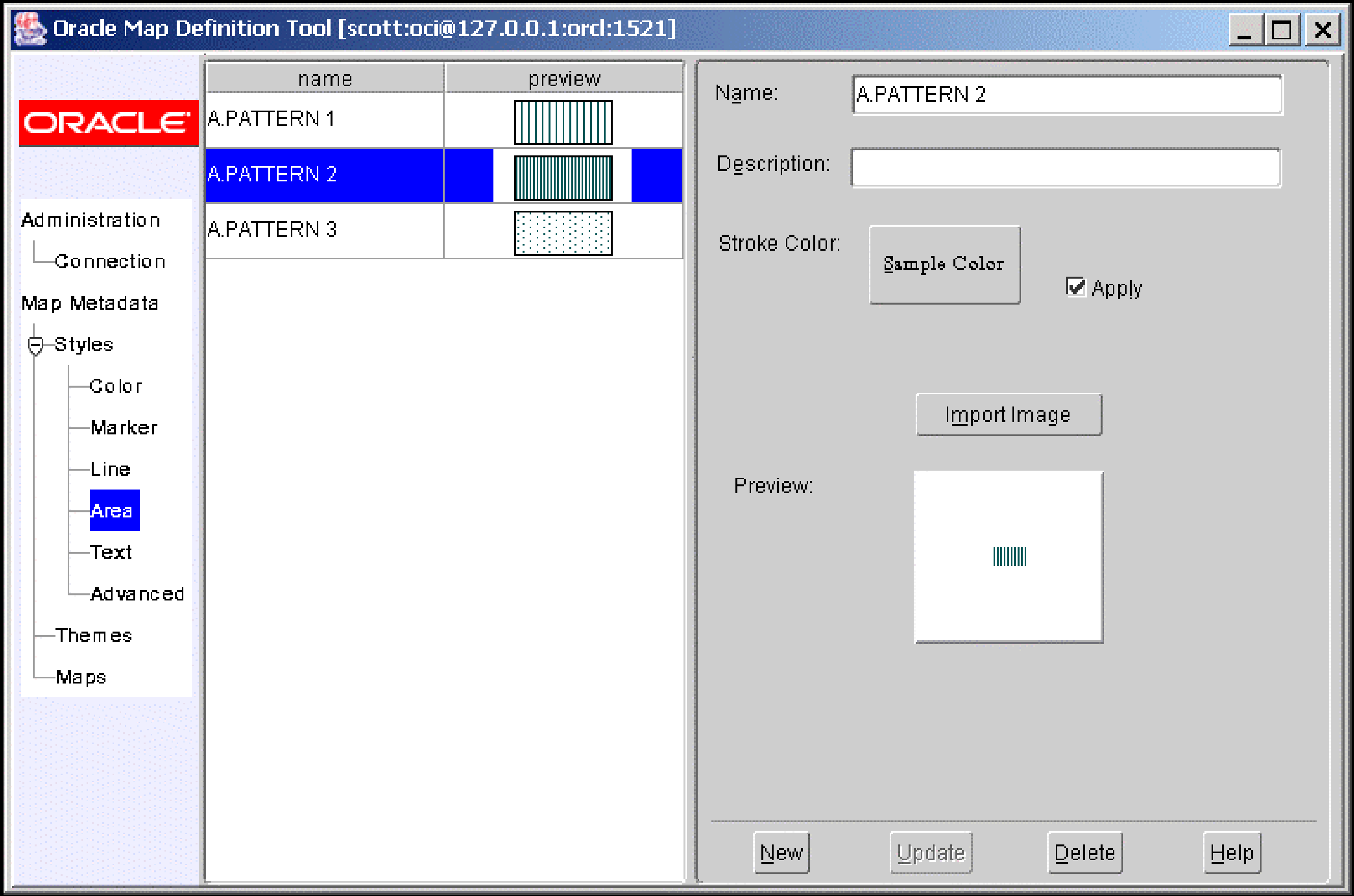
Width: 1.0 Sample Color Dash: ☑ Apply2

Hashmark on Base Line

Length: 3 Sample Color Gap: 10 ☐ Apply3

Preview:

Screen Shots of Map Definition Tool - Area Styles



Oracle Map Definition Tool [scott:oci@127.0.0.1:orcl:1521]

name	preview
A.PATTERN 1	▥
A.PATTERN 2	▥
A.PATTERN 3	░

Name: A.PATTERN 2

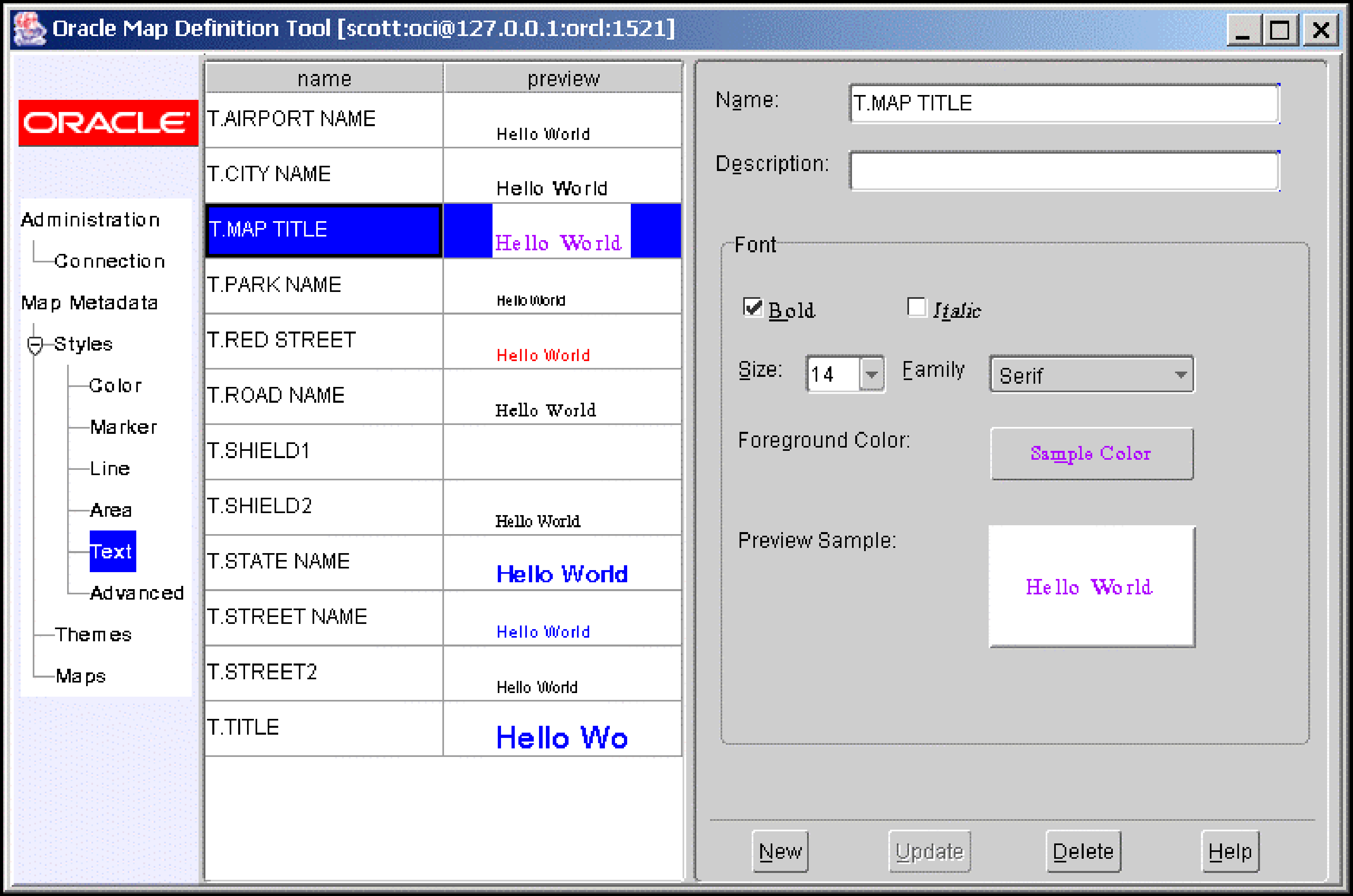
Description:

Stroke Color: Sample Color

☑ Apply

Preview:

Screen Shots of Map Definition Tool - Text Styles



Oracle Map Definition Tool [scott:oci@127.0.0.1:orcl:1521]

name	preview
T.AIRPORT NAME	Hello World
T.CITY NAME	Hello World
T.MAP TITLE	Hello World
T.PARK NAME	Hello World
T.RED STREET	Hello World
T.ROAD NAME	Hello World
T.SHIELD1	
T.SHIELD2	Hello World
T.STATE NAME	Hello World
T.STREET NAME	Hello World
T.STREET2	Hello World
T.TITLE	Hello Wo

Name: T.MAP TITLE

Description:

Font

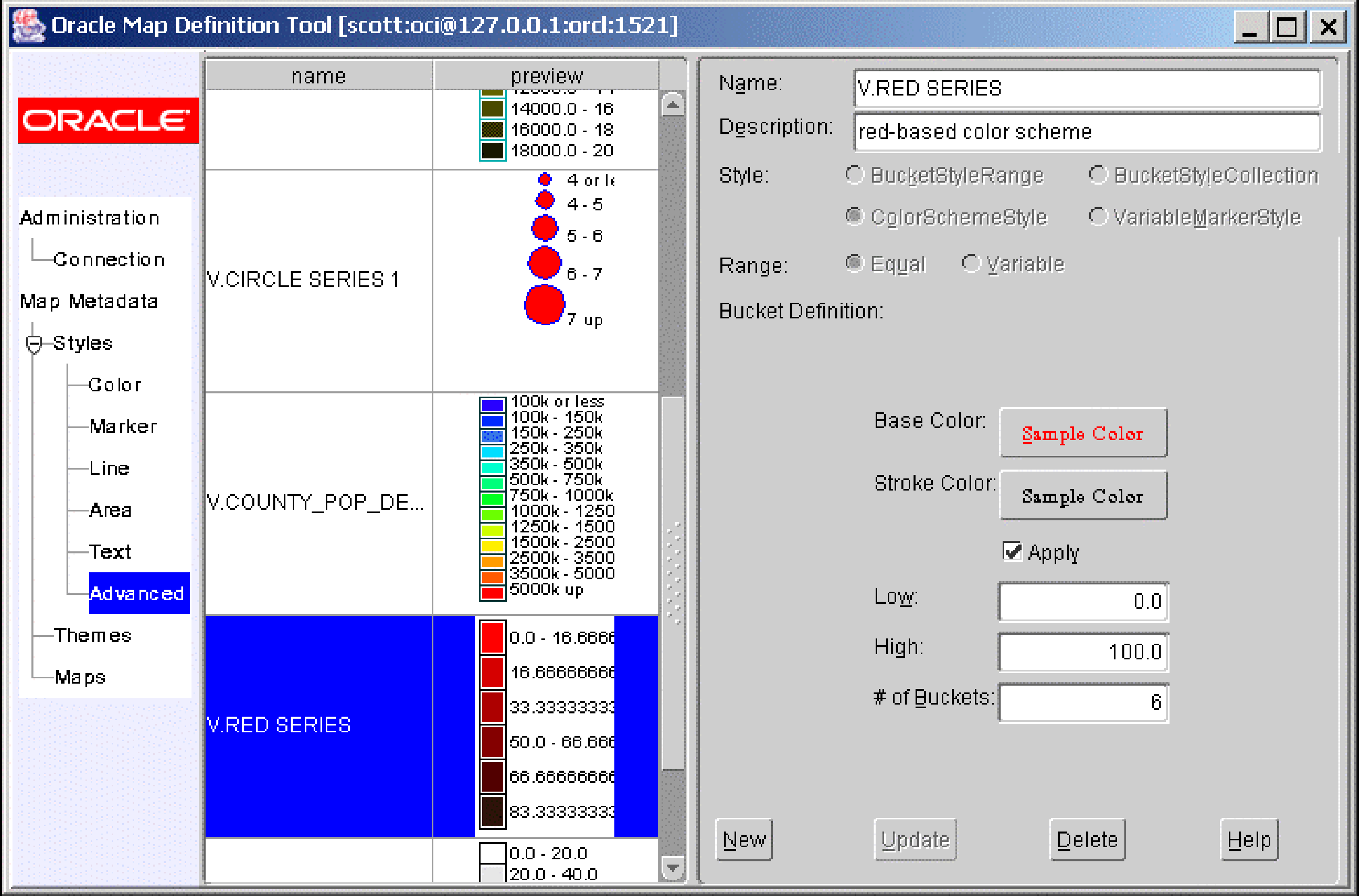
☑ Bold ☐ Italic

Size: 14 Family Serif

Foreground Color: Sample Color

Preview Sample: Hello World

Screen Shots of Map Definition Tool - Advanced Styles



Oracle Map Definition Tool [scott:oci@127.0.0.1:orcl:1521]

- Administration
  - Connection
- Map Metadata
  - Styles
    - Color
    - Marker
    - Line
    - Area
    - Text
    - Advanced
  - Themes
  - Maps

name	preview
	14000.0 - 16 16000.0 - 18 18000.0 - 20
V.CIRCLE SERIES 1	● 4 or le ● 4 - 5 ● 5 - 6 ● 6 - 7 ● 7 up
V.COUNTY_POP_DE...	100k or less 100k - 150k 150k - 250k 250k - 350k 350k - 500k 500k - 750k 750k - 1000k 1000k - 1250 1250k - 1500 1500k - 2500 2500k - 3500 3500k - 5000 5000k up
V.RED SERIES	0.0 - 16.6666 16.66666666 33.33333333 50.0 - 66.666 66.66666666 83.33333333
	0.0 - 20.0 20.0 - 40.0

Name: V.RED SERIES

Description: red-based color scheme

Style: ○ BucketStyleRange ○ BucketStyleCollection ● ColorSchemeStyle ○ VariableMarkerStyle

Range: ● Equal ○ Variable

Bucket Definition:

Base Color: Sample Color

Stroke Color: Sample Color

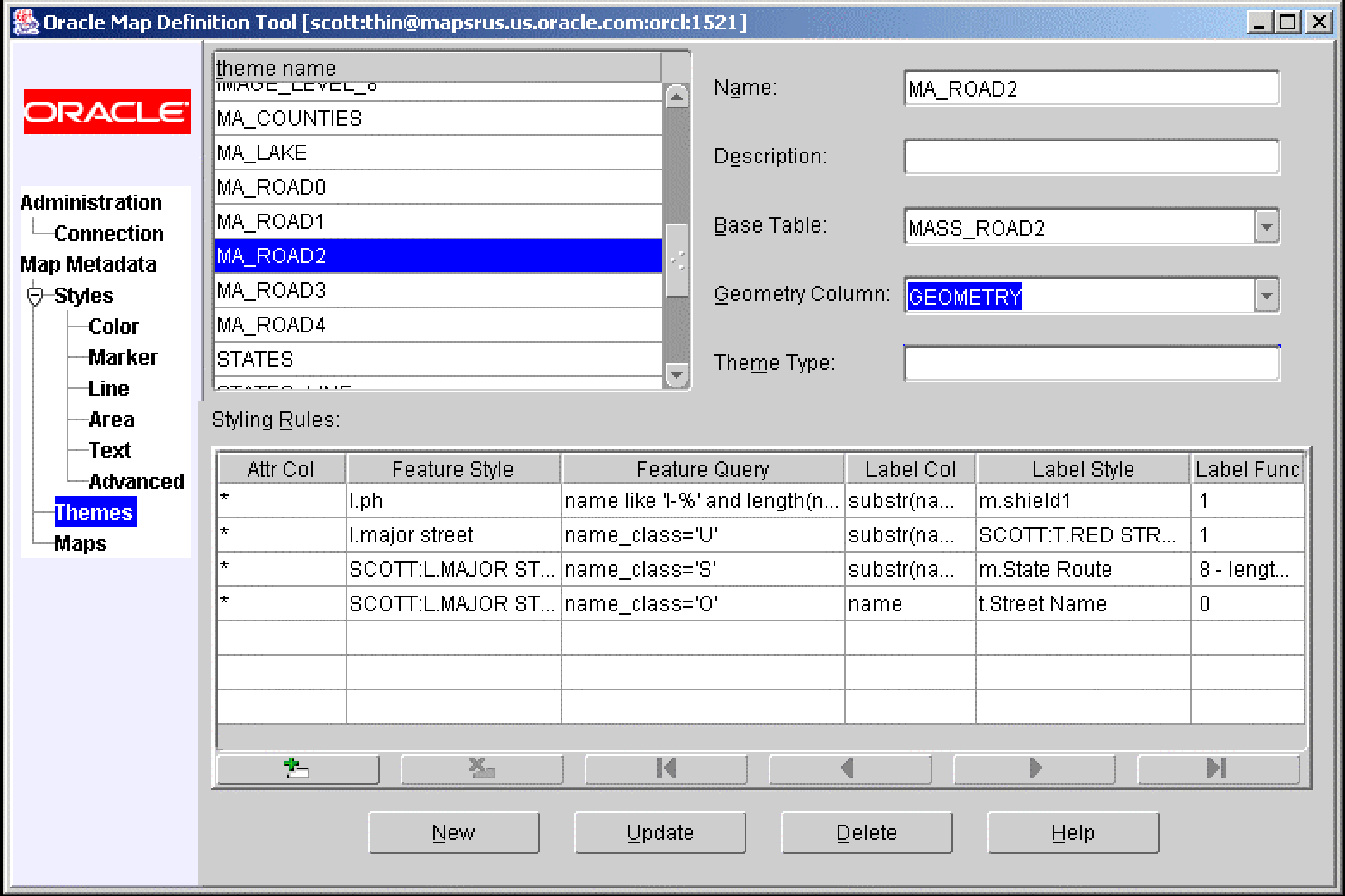
☑ Apply

Low: 0.0

High: 100.0

# of Buckets: 6

Screen Shots of Map Definition Tool - Themes



Oracle Map Definition Tool [scott:thin@mapsrus.us.oracle.com:orcl:1521]

theme name
IMAGE_LEVEL_6
MA_COUNTIES
MA_LAKE
MA_ROAD0
MA_ROAD1
MA_ROAD2
MA_ROAD3
MA_ROAD4
STATES
STATES_LINE

Name: MA_ROAD2

Description:

Base Table: MASS_ROAD2

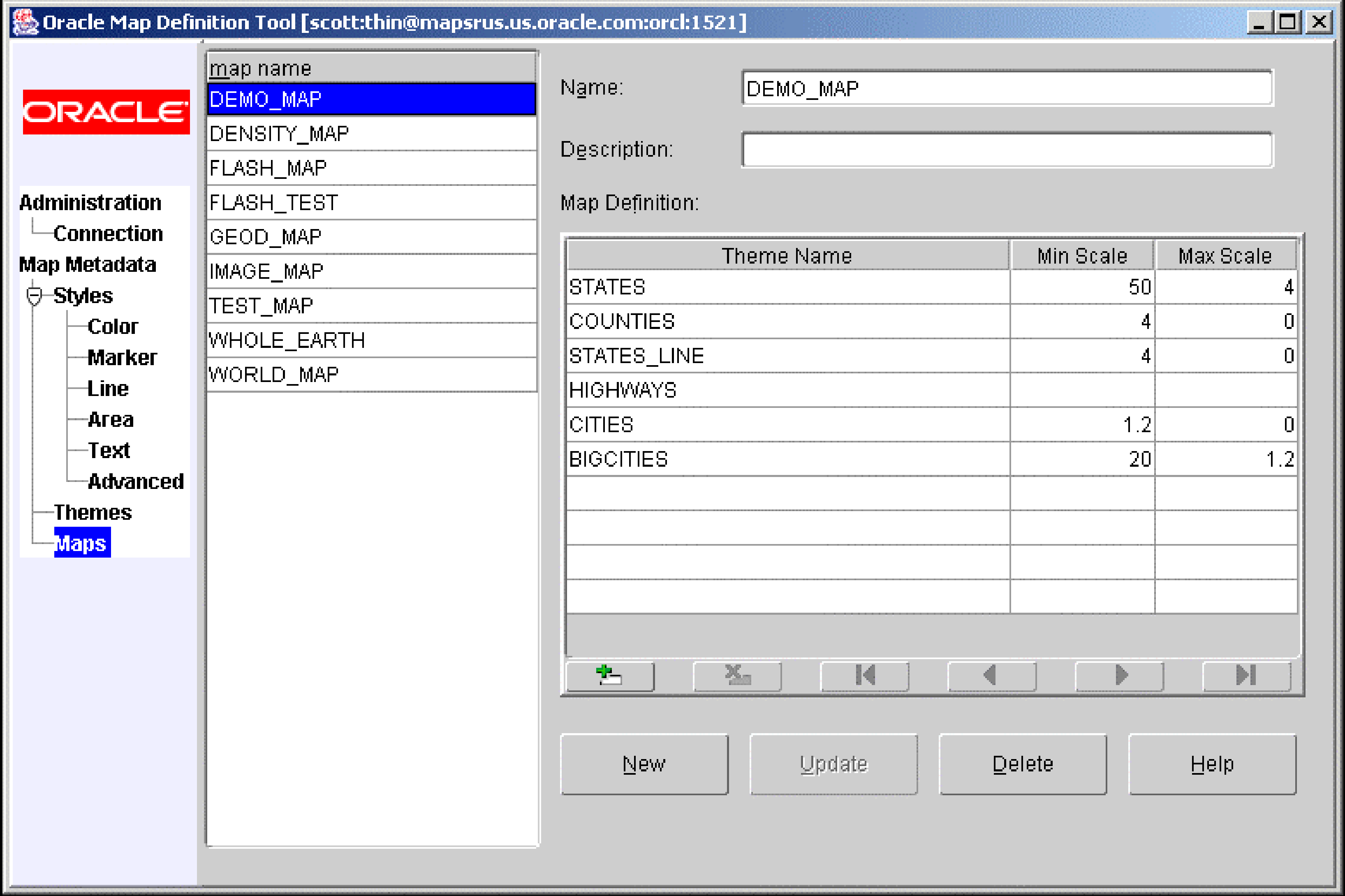
Geometry Column: GEOMETRY

Theme Type:

Styling Rules:

Attr Col	Feature Style	Feature Query	Label Col	Label Style	Label Func
*	l.ph	name like 'I-%' and length(n...	substr(na...	m.shield1	1
*	l.major street	name_class='U'	substr(na...	SCOTT:T.RED STR...	1
*	SCOTT:L.MAJOR ST...	name_class='S'	substr(na...	m.State Route	8 - lengt...
*	SCOTT:L.MAJOR ST...	name_class='O'	name	t.Street Name	0

Screen Shots of Map Definition Tool - basemaps



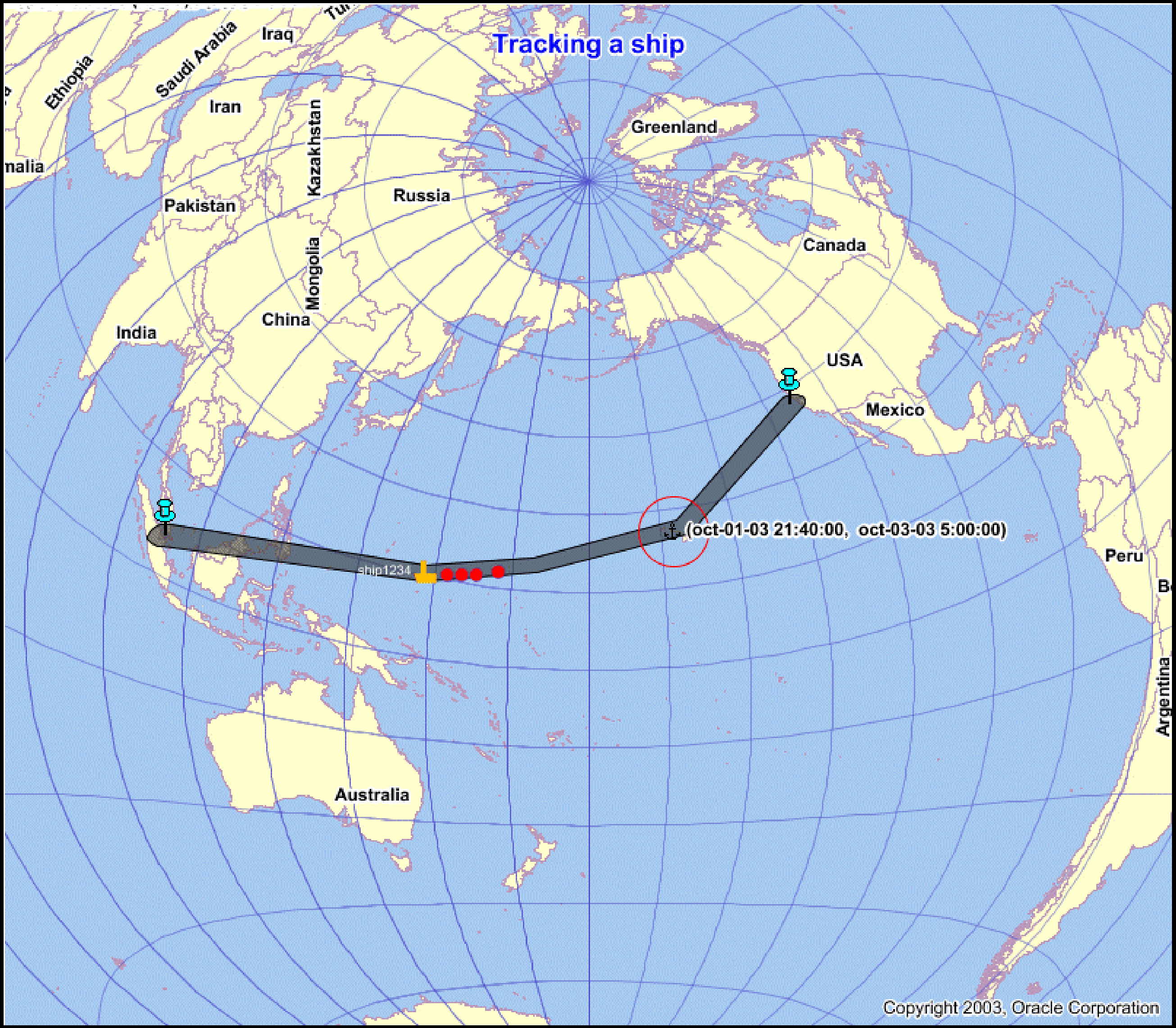
map name
DEMO_MAP
DENSITY_MAP
FLASH_MAP
FLASH_TEST
GEOD_MAP
IMAGE_MAP
TEST_MAP
WHOLE_EARTH
WORLD_MAP

Name: DEMO_MAP

Description:

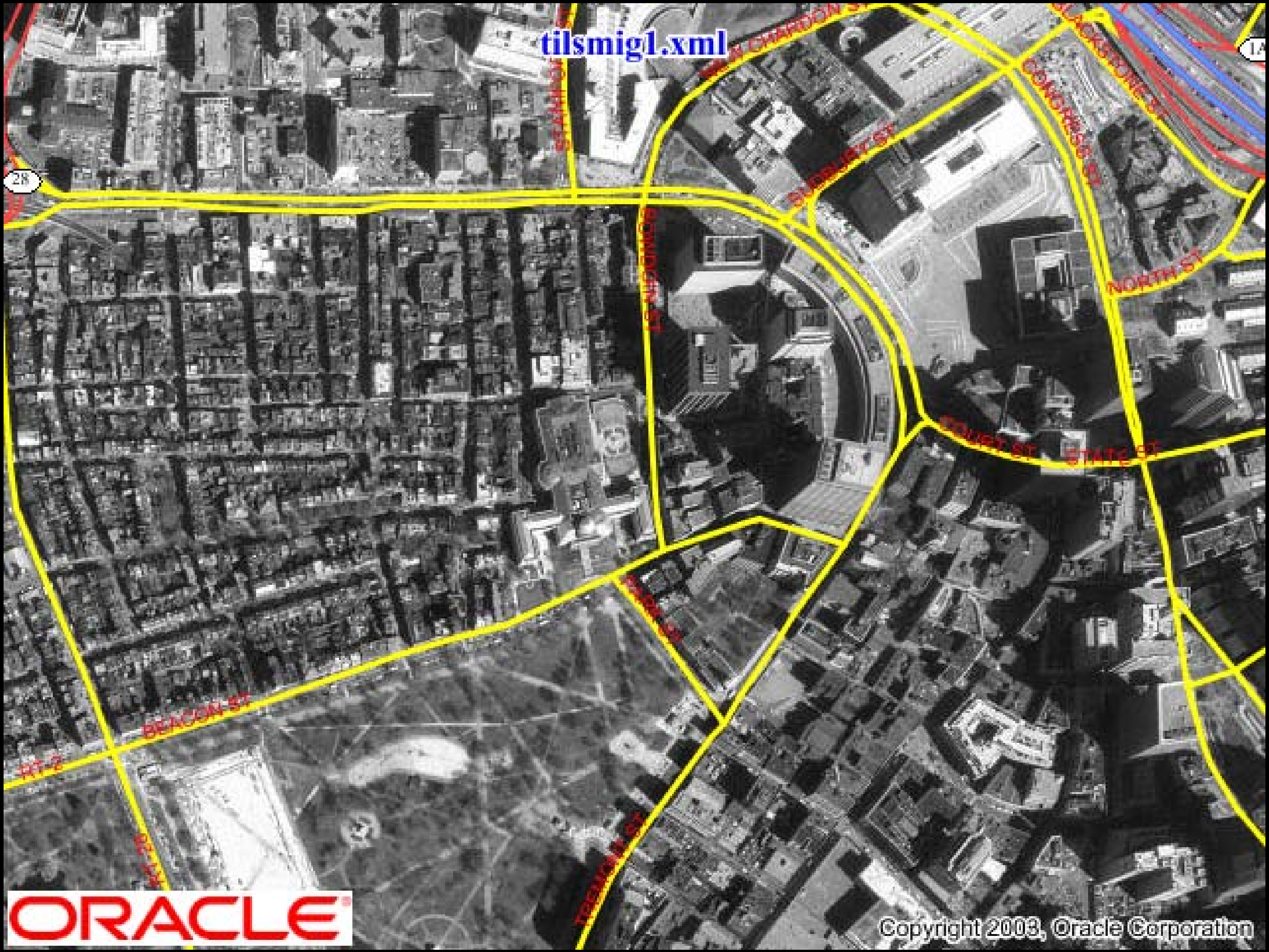
Map Definition:

Theme Name	Min Scale	Max Scale
STATES	50	4
COUNTIES	4	0
STATES_LINE	4	0
HIGHWAYS		
CITIES	1.2	0
BIGCITIES	20	1.2



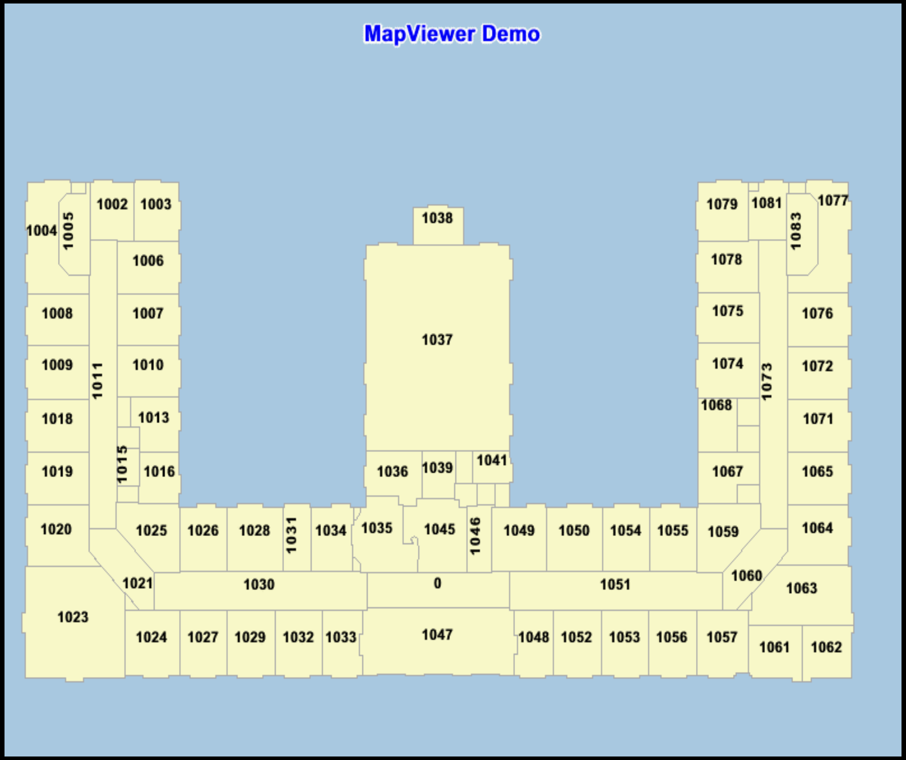
A simple application tracking a ship’s locations as it travels from San Francisco to Singapore.

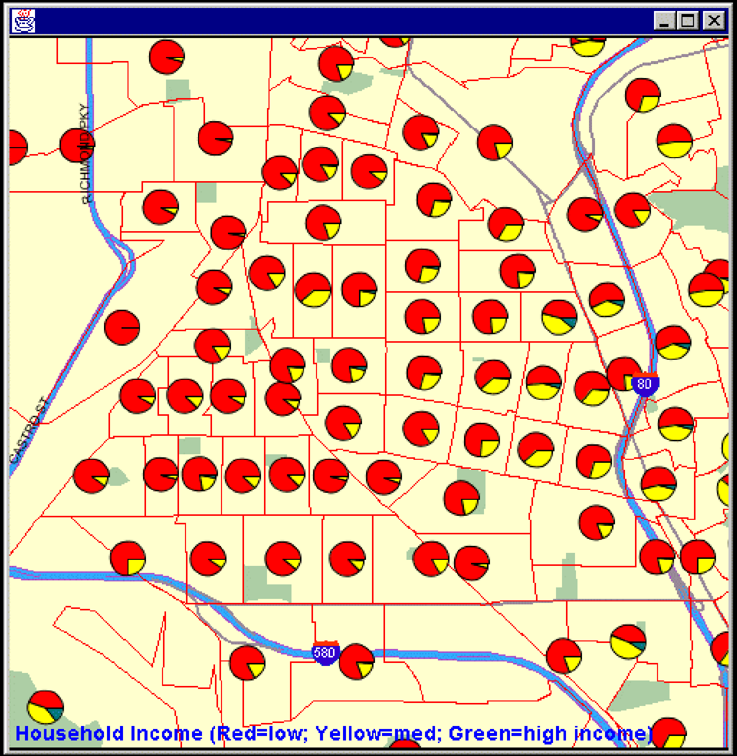
Overlaying vector data (roads) on top of an image theme (aerial photos of downtown Boston)



Enhanced Mapping and Visualization Capabilities

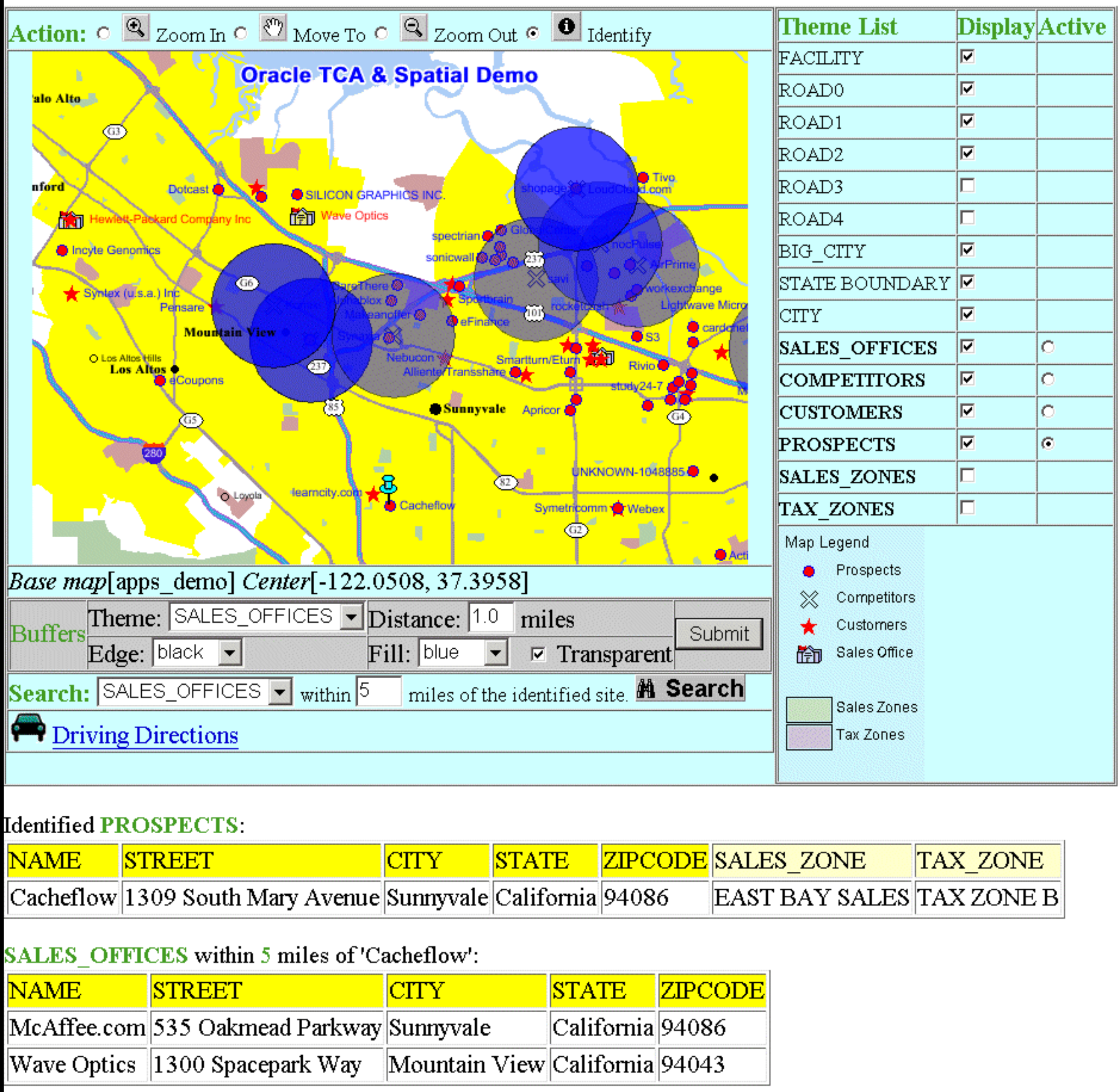
Sample Screenshots





A simple floor plan map with automatic label placement.

Each pie chart represents household income distribution in each block.



Action: ○ Zoom In ○ Move To ○ Zoom Out ● Identify

Theme List	Display	Active
FACILITY	☑	
ROAD0	☑	
ROAD1	☑	
ROAD2	☑	
ROAD3	☐	
ROAD4	☐	
BIG_CITY	☑	
STATE BOUNDARY	☑	
CITY	☑	
SALES_OFFICES	☑	○
COMPETITORS	☑	○
CUSTOMERS	☑	○
PROSPECTS	☑	●
SALES_ZONES	☐	
TAX_ZONES	☐	

Map Legend

●	Prospects
✕	Competitors
★	Customers
⌂	Sales Office
🟩	Sales Zones
🟪	Tax Zones

Base map[apps_demo] *Center*[-122.0508, 37.3958]

Buffers	Theme: SALES_OFFICES ▾	Distance: 1.0 miles
	Edge: black ▾	Fill: blue ▾ ☑ Transparent

Submit

Search: SALES_OFFICES ▾ within 5 miles of the identified site. Search

Driving Directions

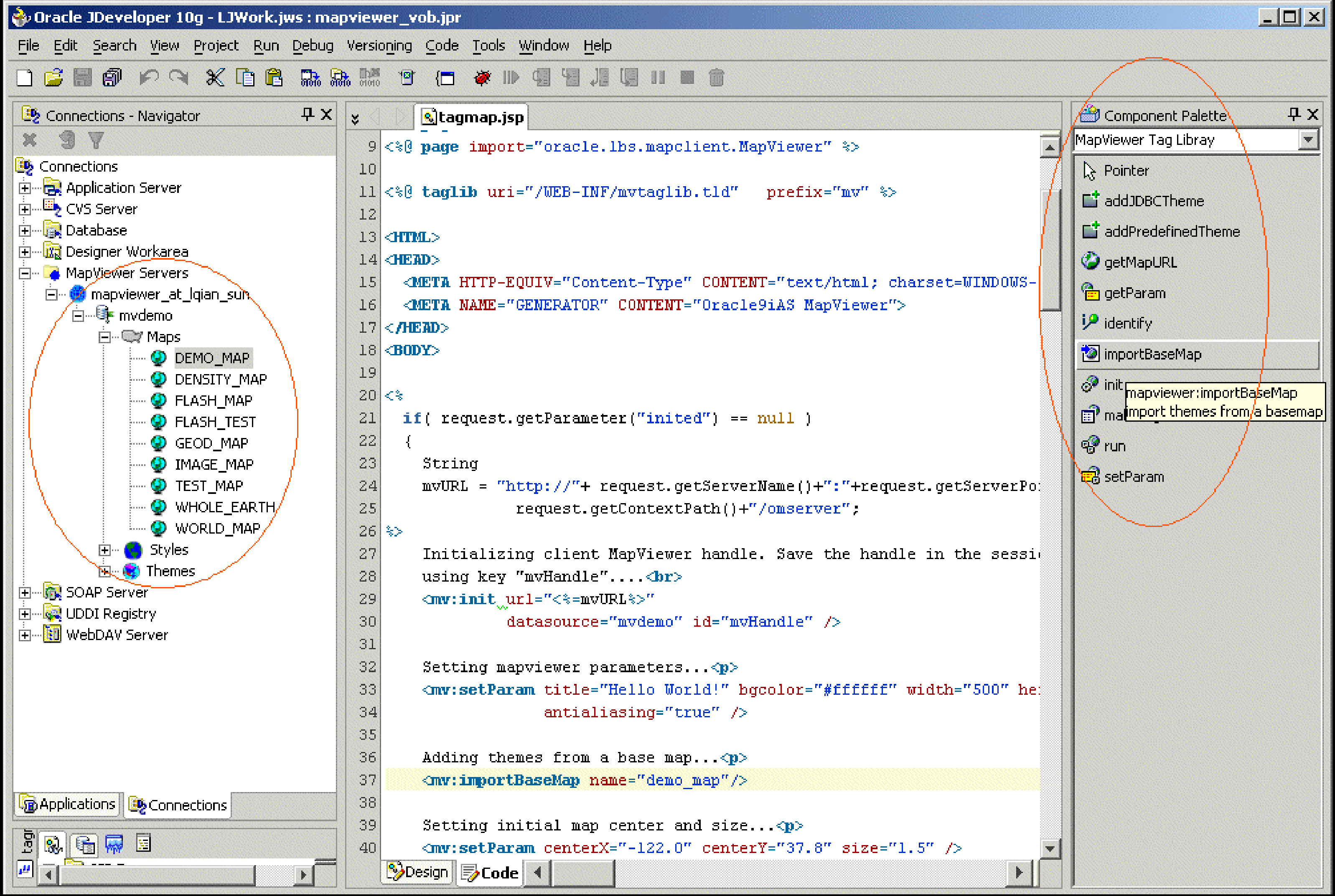
Identified PROSPECTS:

NAME	STREET	CITY	STATE	ZIPCODE	SALES_ZONE	TAX_ZONE
Cacheflow	1309 South Mary Avenue	Sunnyvale	California	94086	EAST BAY SALES	TAX ZONE B

SALES_OFFICES within 5 miles of 'Cacheflow':

NAME	STREET	CITY	STATE	ZIPCODE
McAffee.com	535 Oakmead Parkway	Sunnyvale	California	94086
Wave Optics	1300 Spacepark Way	Mountain View	California	94043

A map showing feature identification and location-based queries/searches.



```
<%@ page import="oracle.lbs.mapclient.MapViewer" %>

<%@ taglib uri="/WEB-INF/mvtaglib.tld"   prefix="mv" %>

<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=WINDOWS-
  <META NAME="GENERATOR" CONTENT="Oracle9iAS MapViewer">
</HEAD>
<BODY>

<%
  if( request.getParameter("inited") == null )
  {
    String
    mvURL = "http://"+ request.getServerName()+":"+request.getServerPo
             request.getContextPath()+"/omserver";
%>
    Initializing client MapViewer handle. Save the handle in the sessi
    using key "mvHandle"....<br>
    <mv:init url="<%=mvURL%>"
             datasource="mvdemo" id="mvHandle" />

    Setting mapviewer parameters...<p>
    <mv:setParam title="Hello World!" bgcolor="#ffffff" width="500" he
             antialiasing="true" />

    Adding themes from a base map...<p>
    <mv:importBaseMap name="demo_map"/>

    Setting initial map center and size...<p>
    <mv:setParam centerX="-122.0" centerY="37.8" size="1.5" />
```

JDeveloper with custom MapViewer JSP tags in its component palette, and a map metadata browser in its Connections panel.

ORACLE FUSION MIDDLEWARE