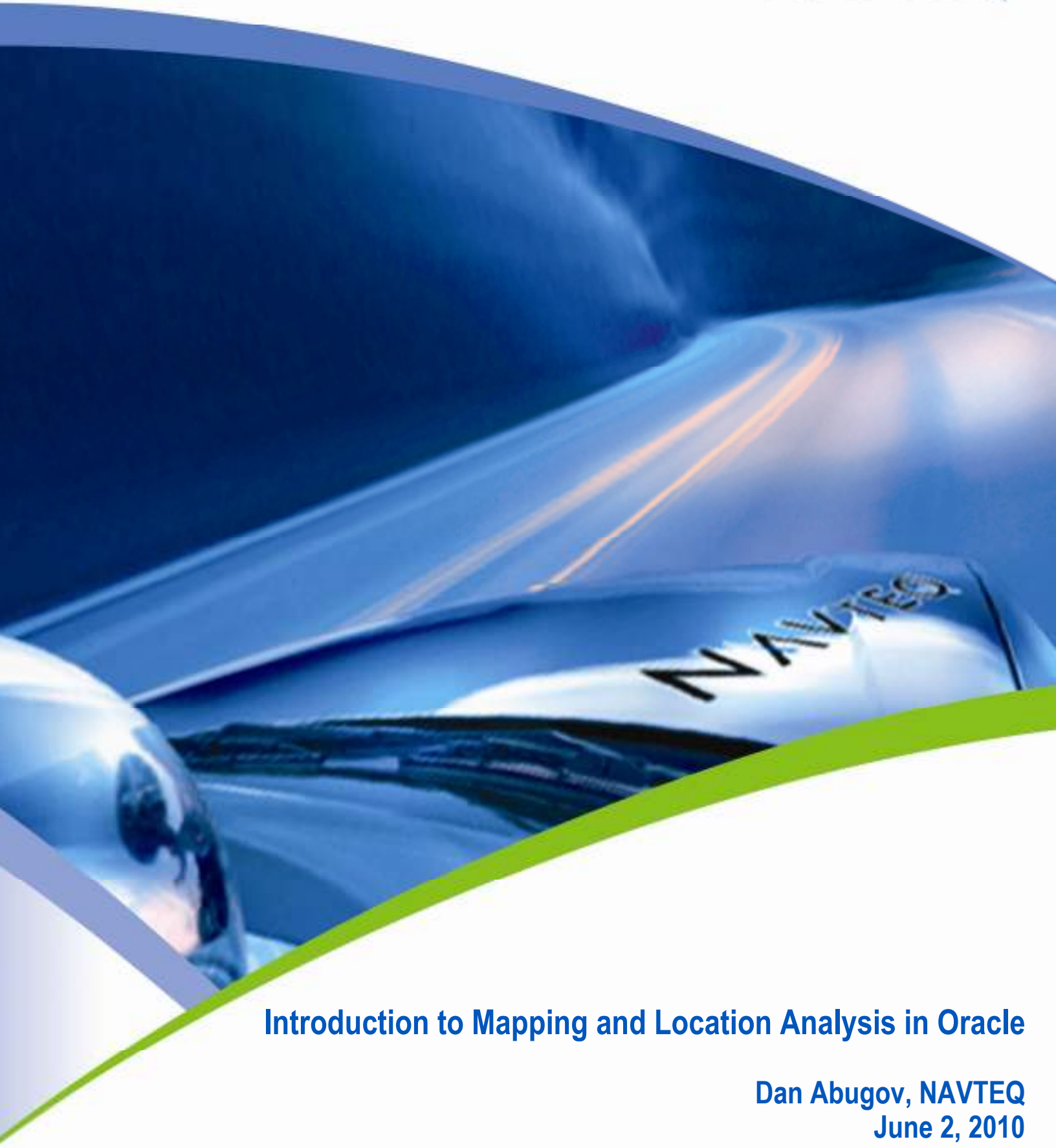
NAVTEQ



Introduction to Mapping and Location Analysis in Oracle

Dan Abugov, NAVTEQ

June 2, 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.

Introduction

Oracle continues to invest making maps and location-enabled applications available to all developers and applications. The use of standard Oracle database components like Oracle Locator and Oracle Spatial continues to expand rapidly, and Oracle MapViewer is included in every Oracle Application Server/Fusion Middleware installation. These technologies are enabling next generation mapping and location analysis capabilities in the Oracle database and application server.

More and more, businesses are requiring the capability to show assets on a map, and, more importantly, to visualize their key performance indicators geographically in a clear and concise way. This is driving applications developers, who understand the need to easily add maps and into their applications.

Oracle and NAVTEQ have been collaborating to provide turn-key map and other content specifically created for the Oracle platform. This content conforms to the Oracle Locator and Oracle Spatial standards for Mapping and Location Analysis, Geocoding, and Routing. NAVTEQ content includes pre-built tables and indexes and installs in minutes using Oracle Transportable Tablespaces. NAVTEQ data is utilized by many Oracle Applications.

This paper will discuss the underlying technologies that make mapping and location analysis available in Oracle.

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 can 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 convention center?
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.

The Spatial Data Type: SDO_GEOMETRY

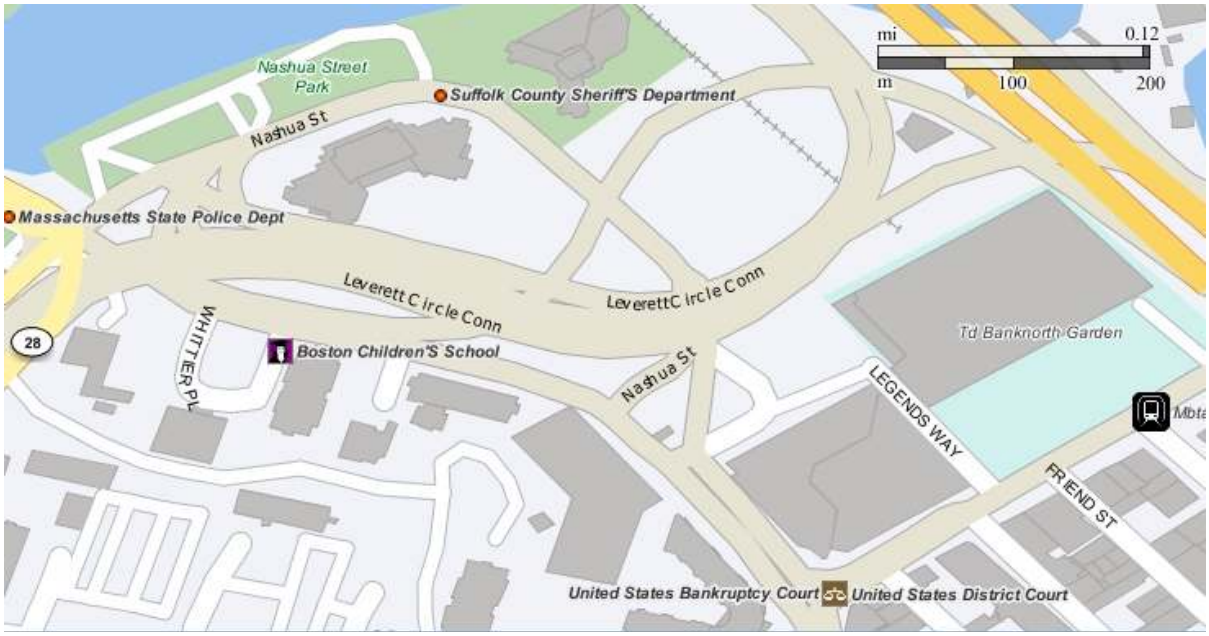
The Oracle database includes the SDO_GEOMETRY data type, which is used to store location data. Any table can include one or more columns of type SDO_GEOMETRY. When creating a table use the SDO_GEOMETRY data type for the column definition, or add a location column to an existing table in Oracle by using the ALTER TABLE command. The syntax for adding a location column is:

```
ALTER TABLE CUSTOMERS
  ADD COLUMN LOCATION(SDO_GEOMETRY);
```

Column naming conventions follow normal Oracle identifier rules. Above, the column name is LOCATION, but any identifier can be used.

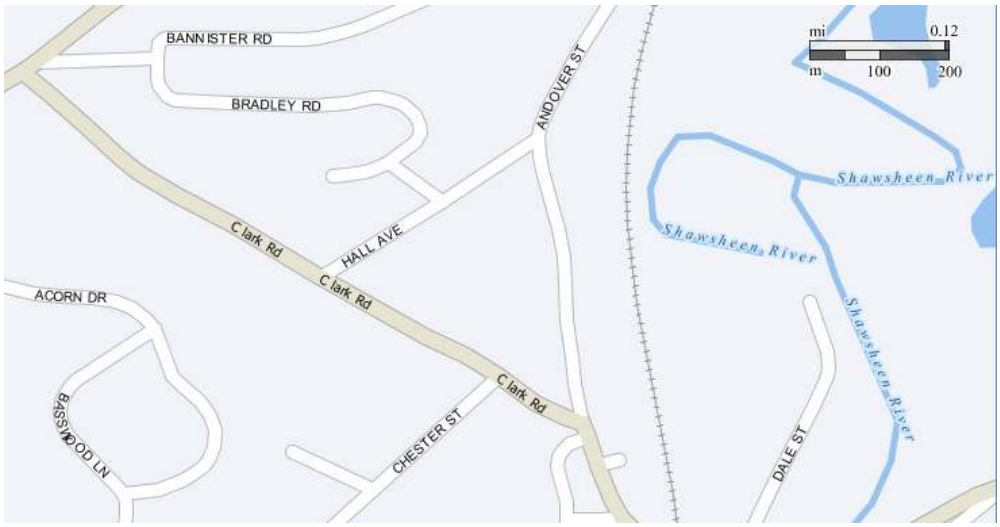
SDO_GEOMETRY is used to store the location on the surface of the Earth of a feature. What kind of feature?

It can define a *point* feature such as an address, an automobile location, a point of interest, a survey marker, etc.



Above, the school, the police and sheriff's department, the courthouse, and the bus stop are all represented by a single point.

SDO_GEOMETRY can also define a *linestring* to represent a linear feature such as a road, a river, or a railway line.



SDO_GEOMETRY can also define a *polygon* to represent an area feature such as a country, state, park, lake, harbor, etc.

When features are included on a map, the map creator may decide to display them differently based on the zoom scale of the map. For instance, San Diego International Airport is appropriately rendered as a point when zoomed out, but when zoomed in it is rendered as a polygon.

Location Queries

Location queries are used to analyze relationships between data. Because location queries are based on simple extensions to SQL any application can enjoy the benefits of location analysis.

Examples of SQL associated with location queries follow. Note the data for most of the queries in this paper are available in downloadable form is available from NAVTEQ Network for Developers (NN4D) site at http://developer.navteq.com.

How many hotels are within two miles of Moscone Center? To answer this question we use a *within distance* query.

```
SELECT COUNT(*)
FROM NTC_MAP_POI_BUSINESS B, NTC_MAP_POI_HOTEL A
WHERE SDO_WITHIN_DISTANCE(A.GEOMETRY,B.GEOMETRY,
  'DISTANCE=2 UNIT=MILE')='TRUE'
   AND B.NAME='MOSCONE CENTER';

COUNT(*)
-------
   319
```

We can extend this query as we can extend any SQL query. For instance, if we wanted to see how many Marriott or Westin hotels are within two miles we could structure our query like this:

```
SELECT A.NAME
FROM NTC_MAP_POI_BUSINESS B, NTC_MAP_POI_HOTEL A
WHERE SDO_WITHIN_DISTANCE(A.GEOMETRY,B.GEOMETRY,
  'DISTANCE=2 UNIT=MILE')='TRUE'
   AND B.NAME='MOSCONE CENTER'
   AND (A.NAME LIKE '%MARRIOTT%'
       OR A.NAME LIKE '%WESTIN%');
NAME
----------------------------------
MARRIOTT-FISHERMAN'S WHARF
JW MARRIOTT-SAN FRANCISCO
WESTIN-ST FRANCIS
MARRIOTT-SAN FRANCISCO-DOWNTOWN
```

WESTIN-SAN FRANCISCO MARKET STREET

Each of the previous queries allow users to do location analysis to solve a typical business problem, and yet none of them required display on a map. One of the interesting and unusual pieces of Oracle's location capabilities is that often the results of location analysis provide information in which the results have no location component.

The second question we asked is how many of my customers have average household income of greater than $67,000? To answer this question we use a *relationship* query, which tells us if there is a spatial interaction between an area of interest and a search column. In the case below, the area of interest is each of the census demographic areas that have a median income > $67,000, and the search column is the stored location of each of the customers.

```
SELECT COUNT(*)
FROM BLOCK_GROUP B, CUSTOMERS A
WHERE B.MED_HOUSEHOLD_INCOME > 67000
  AND SDO_ANYINTERACT (A.LOCATION,GEOMETRY)='TRUE';
```

As with the previous query, this analysis can be extended to see what these customers tend to buy so marketing outreach can be properly targeted. Also demographic analysis can be extended to look at number of children, population density, race, or any one of a number of factors the census keeps track of. In addition, other spatial analysis can be incorporated (do people who live near golf courses buy golf equipment, or do people who live near bodies of water buy insect repellent and/or boat equipment and/or fishing gear).

The third question we asked above was: What is the closest ATM to my current location? To answer this question we use a *nearest neighbor* query. Current location can be determined easily through interfaces creates for GPS and mobile devices, and this kind of query is similar to those executed through programs running on those devices:

```
SELECT NAME
FROM NTC_MAP_POI_BUSINESS A
WHERE SDO_NN(A.GEOMETRY,
          SDO_GEOMETRY(2001,8307,SDO_POINT_TYPE(
            -122.40953, 37.79388, NULL),NULL,NULL),
            'SDO_BATCH_SIZE=8')='TRUE'
  AND A.CAT_ID = 3578
  AND ROWNUM < 4;

NAME
------------------------------
BANK OF THE WEST
BNP PARIBAS
FIRST REPUBLIC BANK
```

To return the closest ATM of a specific type (in this case Bank of America) the previous query can be modified:

```
SELECT NAME||', '||POI_HOUSE_NUMBER||' '||POI_STREET_NAME AS ADDR
FROM NTC_MAP_POI_BUSINESS A
WHERE SDO_NN(A.GEOMETRY,
          SDO_GEOMETRY(2001,8307,SDO_POINT_TYPE(
            -122.40953, 37.79388, NULL), NULL, NULL),
          'SDO_BATCH_SIZE=8')='TRUE'
  AND A.CAT_ID = 3578
  AND NAME LIKE '%BANK OF AMERICA%'
  AND ROWNUM < 4;

ADDR
---------------------------------
BANK OF AMERICA, 163 BRANNAN ST
BANK OF AMERICA, 345 3RD ST
BANK OF AMERICA, 55 HAWTHORNE ST
```

Oracle Locator and Oracle Spatial

Oracle Locator is component of every Oracle Database. All of the functionality referenced above, including the ability to store, index, and query location information is included in Oracle Locator. Oracle Locator also includes other functionality which is beyond the scope of this paper.

Oracle Spatial takes all of the functionality of Oracle Locator and expands it in many ways, some of which are useful to the many applications that can utilize location analysis, and others which are quite specialized in their application and use of the technology.

Some of the additional capabilities of Oracle Spatial include:

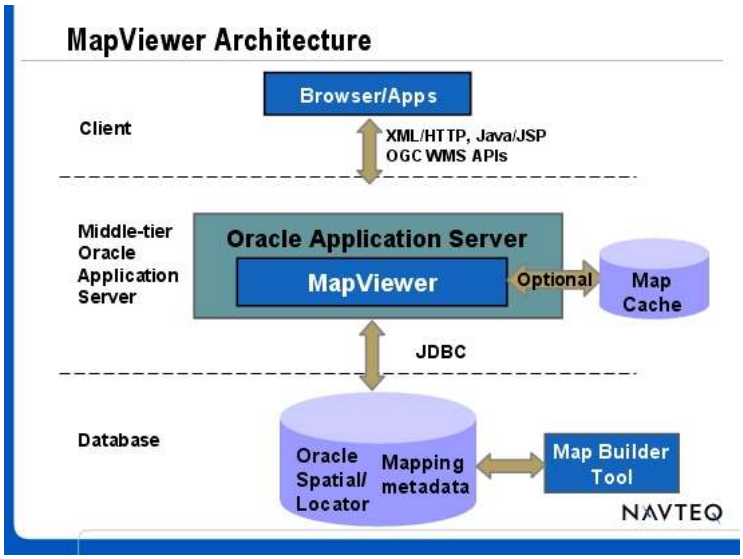
- Geocoding
  - Geocoding is the process of assigning a longitude and latitude (the location on the Earth) to an address. It is a major requirement of business and mapping applications. Once locations on the Earth are assigned to addresses, spatial analysis and mapping is possible. NAVTEQ content enables geocoding in Oracle.
- Raster Imagery Storage and Retrieval
- Linear Referencing
- Spatial Boolean Functions
- Topology Data Model
- Network Data Model
- LIDAR Data Storage and Retrieval (Point Clouds)
- Spatial Analysis and Mining
- Tin Storage and Retrieval
- Spatial Aggregation
- Spatial Web Services including:
  - Geocoding
  - Routing
    - Routing used in Oracle applications to help people easily navigate between locations. In addition, routing is used for transportation analysis (how much should I charge a customer if I have to pick something up at location A and deliver it to location B) and for marketing analysis (if I put a store at a particular location, how many customers can drive to it within a ten minute drive time). NAVTEQ provides the routing data that is used by Oracle applications.
  - Yellow Pages
  - Web Catalog Service

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



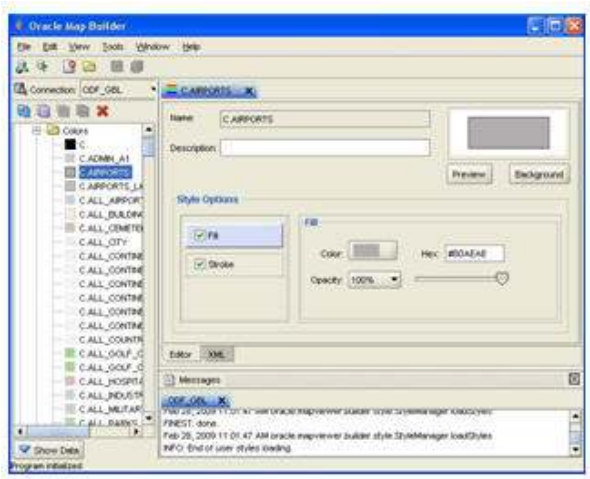
Below is an example of a map rendered by Oracle Fusion Middleware MapViewer.

Oracle Map Builder

Oracle Fusion Middleware MapViewer uses metadata to describe the visual representation of the features stored in the database. There are three different types of metadata that describe how maps are rendered:

- Styles
  - Can control fill color, border color, line thickness, line style, text font and size, and so on
  - Stored as XML in view USER_SDO_STYLES
- Themes
  - A geometry column associated with a style
  - A text label column can also be associated with a theme. The label column has its own style (size, font, color, and other attributes)
  - Stored as XML in view USER_SDO_THEMES
- Maps
  - One or more themes
  - Stored as XML in view USER_SDO_MAPS

Oracle Map Builder is Oracle's map authoring tool created to manage map metadata. Oracle Map Builder creates and manages XML metadata associated with styles, themes, and maps (plus a lot of other things) via a simple GUI.




```
<?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 JDeveloper

Oracle JDeveloper is an integrated development environment (IDE) with end-to-end support for modeling, developing, debugging, optimizing, and deploying Java applications and Web services. Oracle JDeveloper supports the full application development life cycle.

Oracle JDeveloper simplifies the development of rich internet applications by providing an Application Development Framework (ADF) that minimizes the requirements to write infrastructure code. It is based on a Model-View-Controller design pattern:

- Model – Handles data interactions and business logic
- View – Handles user interface components
- Controller – Handles application flow/interactions between Model and View components

Rich internet applications require full-featured user interfaces, and JDeveloper meets this need with ADF Faces. ADF Faces includes a large set of predefined GUI components such as pie charts, bar charts, graphs, dials, etc. ADF Faces also supports partial page rendering/AJAX.

Importantly, for this application ADF Faces also includes a geoMap component which simplifies the process of adding a map and showing your business data in your internet application.

NAVTEQ Data

NAVTEQ provides turnkey content for mapping, location analytics, geocoding and routing that meets the specification for inclusion in Oracle Application via Oracle Locator and Spatial. A data sample 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 same data you can download for sample use 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 is:

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

Setup to use NAVTEQ Data in MapViewer

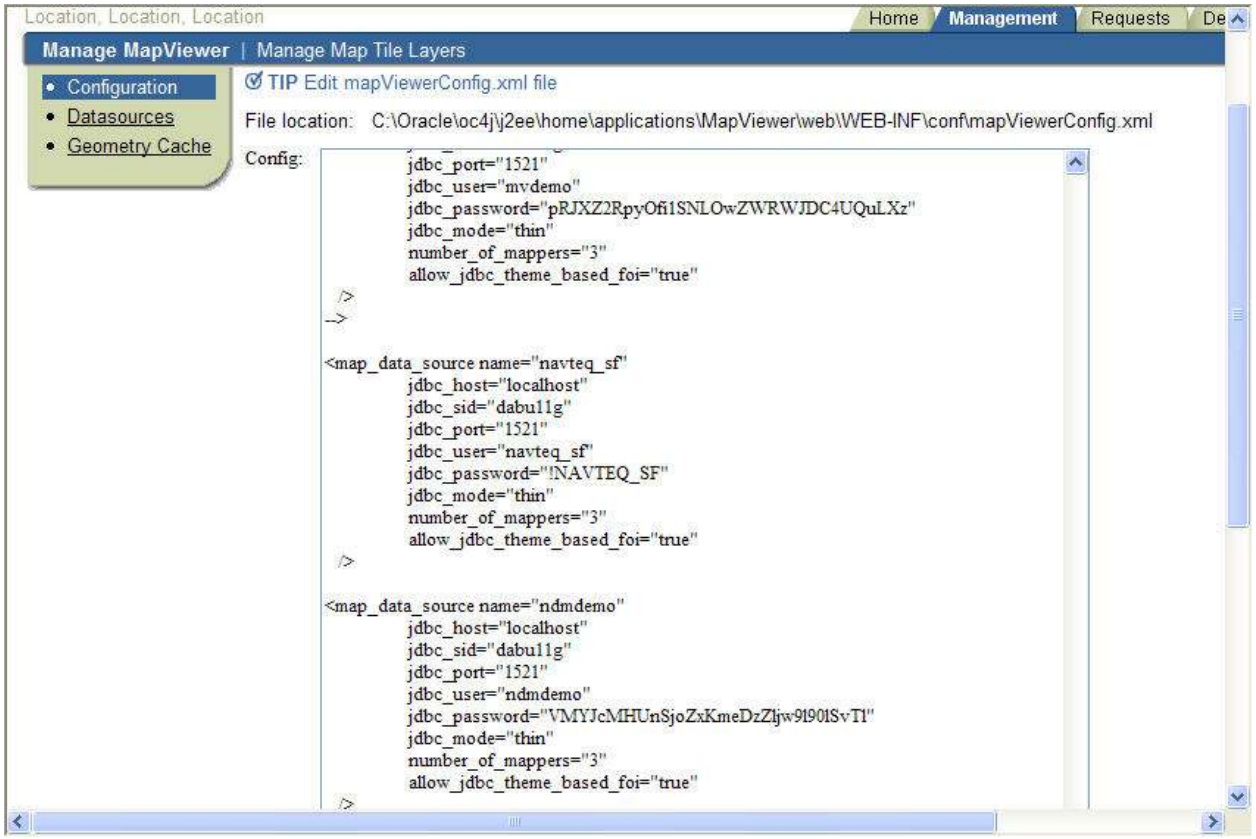
To use the sample content in a mapping application, follow the steps below.

After deploying Oracle Fusion Middleware MapViewer into the application server, make sure to create a jdbc connection to the database that is started whenever the application server is started. To do this, go to the MapViewer deployment (for example: http://localhost:8888/mapviewer).

Next, click on the Admin button in the upper right hand side of the MapViewer page:

Next, click on Configuration in the upper right:

On the configuration page, slide way down until you come to where the connection strings are defined. Add a connection string for the NAVTEQ_SF user or the user you imported the sample data into. Make sure the connection string is not inside an XML comment (not between <!-- and --> :



Note the jdbc_password is preceded by an exclamation point (!). When the application server is restarted the password is encrypted.

At the bottom of the page, click on Save & Restart:

Now you are ready to use NAVTEQ content in a mapping application.