ORACLE SPATIAL eLOCATION QUICK START GUIDE

Contents

QUICK START GUIDE 3

OVERVIEW 4
OBJECTIVES 4
AUDIENCES 4

PART I : INSTALLING ORACLE SPATIAL COMPOMENTS 5
BEFORE YOU START 5
THE OVERVIEW OF ORACLE SPATIAL COMPONENTS 5
REQUIRED SOFTWARE 5
ORACLE SPATIAL AND MAPVIEWER COMPONENTS 6
DEPLOYING ORACLE MAPVIEWER 6
DEPLOYING ORACLE GEOCODER 9
DEPLOYING ORACLE ROUTESERVER 12
DEPLOYING ORACLE ELOCATION 15

PART II : ORACLE ELOCATION XML API 20
MAPPING XML API 20
GEOCODING XML API 24
ROUTING XML API 26

Quick Start Guide

Overview

Objectives

This document describes the installation of eLocation application and XML API of mapping, geocoding, and routing services.

The first part describes how to install and configure Oracle Spatial Components such as Oracle Mapviewer, Oracle Geocoder, Oracle RouteServer, and Oracle eLocation. It is based on Oracle Weblogic Server.

The second part describes how to use XML API to communicate with Oracle eLocation.

Audiences

This document is intended for spatial enabled application developers. You should understand Oracle database concept and Oracle weblogic server. You should understand Javascript, AJAX concept and DHTML. You should also be familiar with Oracle Spatial or Oracle Locator concepts, or at least have access to Oracle Spatial Developer's guide.

PART I : Installing Oracle Spatial Compoments

This part describes how to install and configure the Oracle Spatial Web Services components separately. There are four main components and one sample web to provide the Oracle Spatial feautures: Oracle Mapviewer, Oracle Geocoder, Oracle RouteServer and Oracle eLocation.

> Note : If you want to install Oracle Spatial Web Services in production environment, we recommend that the heap size of a domain where Mapviewer is deployed to needs to be set over 1G bytes memory and the heap size of a domain where RouteServer is deployed to needs to be set over 6G bytes memory. It is more stable to install Mapviewer and RouteServer to separate domain and server.

If you don't want to import Oracle Spatial Components in you system manually, you can skip this part.

Before you start

Before you install Oracle spatial component, you may need to install database and map data to provide spatial services using Oracle Spatial Components.

You can get the partner data as the following site.

- Oracle Spatial Partners' Data
  http://www.oracle.com/technetwork/database/options/spatialandgraph/spatial/downloads/partners/spatial-partners-data-087203.html

The overview of Oracle Spatial Components

- Oracle Mapviewer

The Oracle Mapviewer service enables users to find features (land, territories, roads and buildings and so on) through images like WMS or web features like WFS. It usually named "Mapping" or "Mapping services".

- The Oracle Geocoder

Oracle Geocoder service enables users to transform address information to spatial location information such as longitude and latitude and vice versa. It is usually named "Geocoding" or "Geocoding services".

- Oracle RouteServer

Oracle RouteServer enables users to route between designated locations and provides driving information such as directions, instruction, distance and time. It is usually named "Routing" or "Routing services".

- Oracle eLocation

Oracle eLocation provides dispatcher services which enables user to invoke mapping, geocoding, routing service via eLocation.

Required Software

The Oracle Spatial components are compatible with J2EE Web Application Server such as Oracle Weblogic Server. Deploying components is based on the Oracle Weblogic in this document. Before installing and deploying these components, you should install Oracle Weblogic Server on your environment.

- The installation guide for Oracle Weblogic is as the following URL.

  http://www.oracle.com/technetwork/middleware/weblogic/documentation/index.html

Oracle Spatial and MapViewer Components

You need four ear files as below to completely install all the components of eLocation.

- mapviewer.ear

  See http://www.oracle.com/technetwork/middleware/mapviewer/overview/index-090261.html for more information on MapViewer.

- geocoder.ear, routeserver.ear, and elocation.ear are shipped with Oracle DB install. These ear files are located under $ORACLE_HOME/md/jlib/ directory.

Deploying Oracle Mapviewer

This section explains how to configure and deploy Mapviewer to Oracle Weblogic server. You can download and copy to specific directory in your system. You should unpack the mapviewer.ear to any directory you want because this directory will be used for working and saving files like generated map images. You can see the following instruction to unpack and name like that. It assumes that you have created the directory named "/u01/webapps". In addtion, mapviewer.jar needs xmlparsev2.jar file because classgen.jar file depends on it. xmlparserv2.jar is not included in the component. You can find it from WEBLOGIC_HOME/util/ccr/lib/xmlparserv2.jar.

The instruction of unpacking mapviewer.ear is as following.

- Copy mapviewer.ear to /u01/webapps
- Go to /u01/webapps
- Rename mapviewer.ear to mapviewer1.ear
- Create subdirectory named mapviewer.ear under /u01/webapps
- Unpack mapviewer1.ear into mapviewer.ear directory
- Go to mapviewer.ear directory
- Rename web.war to web1.war
- Create subdirectory named web.war in the /u01/webapps/mapviewer.ear directory
- Unpack web1.war into web.war directory
- Copy xmlparserv2.jar to WEB-INF/lib (xmlparserv2.jar is not included in the package)

The example of unpacking mapviewer.ear is as following.

```
[oracle@spatialelocation webapps]$ ls
elocation.ear  geocoder.ear  mapviewer.ear  oraclemap.ear  routeserver.ear
xmlparserv2.jar
[oracle@spatialelocation webapps]$ mv mapviewer.ear mapviewer1.ear
[oracle@spatialelocation webapps]$ mkdir mapviewer.ear
[oracle@spatialelocation webapps]$ mv mapviewer1.ear mapviewer.ear
[oracle@spatialelocation webapps]$ cd mapviewer.ear
[oracle@spatialelocation mapviewer.ear]$ unzip mapviewer1.ear
[oracle@spatialelocation mapviewer.ear]$ mv web.war web1.war
[oracle@spatialelocation mapviewer.ear]$ mkdir web.war
[oracle@spatialelocation mapviewer.ear]$ mv web1.war web.war
[oracle@spatialelocation mapviewer.ear]$ cd web.war
[oracle@spatialelocation web.war]$ unzip web1.war
[oracle@spatialelocation web.war]$ cp /u01/webapps/xmlparserv2.jar WEB-INF/lib
```

Next, you will modify the configuration file in the WEB-INF/conf/mapViewerConfig.xml

- Go to WEB-INF/conf
- Backup mapViewerConfig.xml to mapViewerConfig.xml.old
- Modify mapViewerConfig.xml

You can see the detail information how to config the mapViewerConfig.xml as following URL.

- 1.5.2 Configuring MapViewer specified in Oracle® Fusion Middleware User's Guide for Oracle MapViewer 11g Release 1 (11.1.1) Part Number E10145-04

  http://download.oracle.com/docs/cd/E14571_01/web.1111/e10145/toc.htm

The example of modifying mapViewerConfig.xml is as following.

```
[oracle@spatialelocation web.war]$ cd WEB-INF/conf
[oracle@spatialelocation conf]$ vi mapViewerConfig.xml
  <map_data_source name="elocation"
                   jdbc_host="spatialelocation"
                   jdbc_sid="orcl"
                   jdbc_port="1521"
                   jdbc_user="navteq_sf"
                   jdbc_password="!oracle"
                   jdbc_mode="thin"
                   number_of_mappers="3"
                   allow_jdbc_theme_based_foi="true"
   />
  <map_data_source name="mvdemo"
                   jdbc_host="spatialelocation"
                   jdbc_sid="orcl"
                   jdbc_port="1521"
                   jdbc_user="mvdemo "
                   jdbc_password="!oracle"
                   jdbc_mode="thin"
                   number_of_mappers="3"
                   allow_jdbc_theme_based_foi="true"
   />

[oracle@spatialelocation conf]$
```

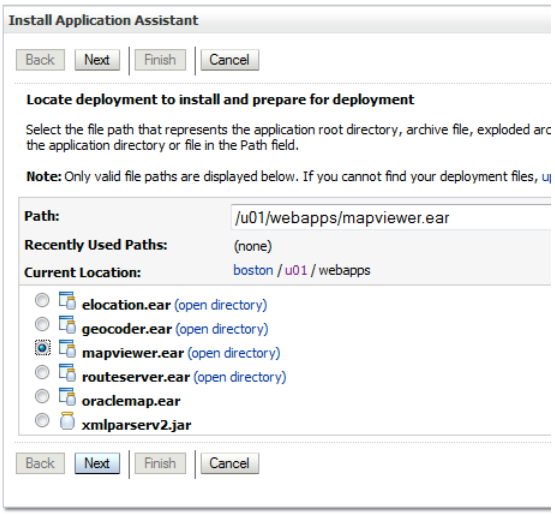
Note: You should precede the jdbc_password value with a '!' (exclamation point). When mapViewer starts next time, it will encrypt and replace the clear text password. If you miss a '!', mapViewer can't not recognize it automatically when the server restart. Or you can use the container data sources.

Next, you can deploy mapviewer.ear to Oracle Weblogic Server.

- Log in to the Weblogic Console with weblogic server credential, it is typically, weblogic/welcome1.

  Weblogic console URL : http://hostname:port/console

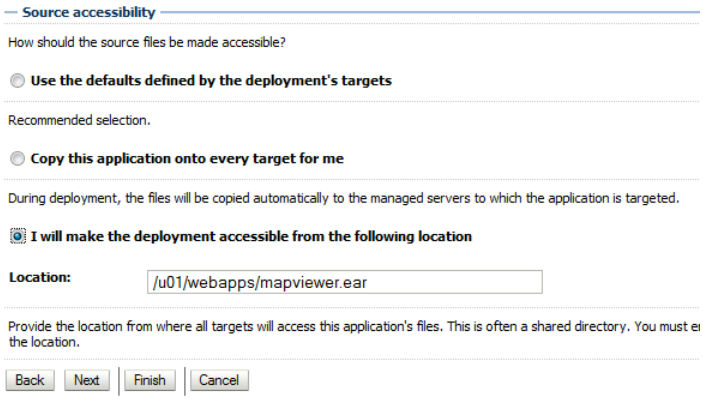
- Click Deployment under the left domain structure menu bar.
- Under Deployment, Click Install
- Select mapviewer.ear under the directory of '/u01/webapps



- Select the targeting style to Install this deployment as an application

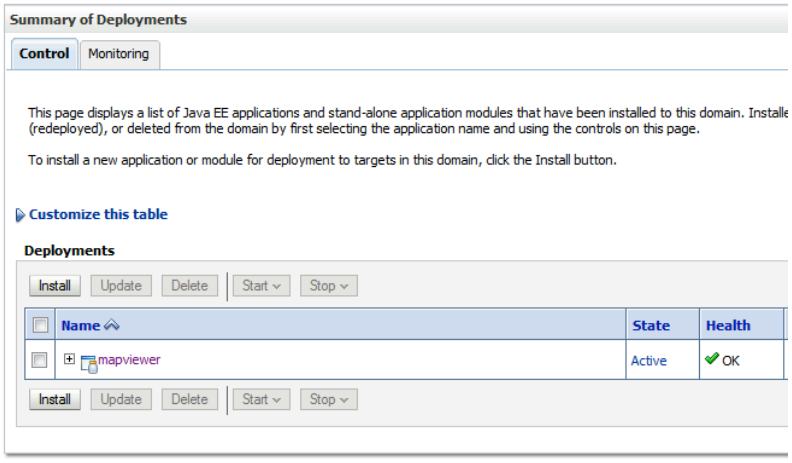
- Select the security to DD only.

- Select the source accessibility to “I will make the deployment accessible from the following location”



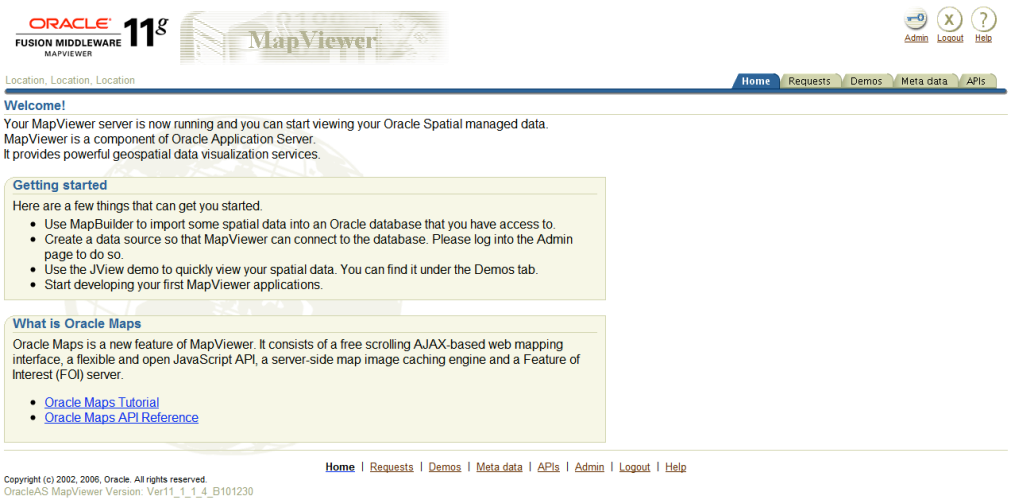
- Select the additional configuration to default, then you can see the deployment’s configuration screen.

- Click Finish



You can see mapviewer page in the web browser to verify the installation is succedded.

- Mapviewer URL : http://hostname:port/mapviwer



Deploying Oracle Geocoder

This section explains how to config and deploy Geocoder to Oracle Weblogic Server. You can download and copy to proper directory in your system. You should unpack the geocoder.ear to any directory you want because this directory will be used for working and saving files. You can see the following instruction to unpack and name like that. It assumes that you have created the directory named /u01/webapps. In addtion, geocoder.jar needs xmlparsev2.jar file. xmlparserv2.jar is not included in the component. You can find it from WEBLOGIC_HOME/util/ccr/lib/xmlparserv2.jar.

The instruction of unpacking geocoder.ear is as following.

- Copy geocoder.ear to /u01/webapps
- Go to /u01/webapps
- Rename geocoder.ear to geocoder1.ear
- Create subdirectory named geocoder.ear under /u01/webapps
- Unpack geocoder1.ear into geocoder.ear directory
- Go to geocoder.ear directory
- Rename web.war to web1.war
- Create subdirectory named web.war in the /u01/webapps/geocoder.ear directory
- Unpack web1.war into web.war directory
- Copy xmlparserv2.jar to WEB-INF/lib (xmlparserv2.jar is not included in the package)

The example of unpacking mapviewer.ear is as following.

```
[oracle@spatialelocation webapps]$ ls
elocation.ear geocoder.ear mapviewer.ear oraclemap.ear routeserver.ear
xmlparserv2.jar
[oracle@spatialelocation webapps]$ mv geocoder.ear geocoder1.ear
[oracle@spatialelocation webapps]$ mkdir geocoder.ear
[oracle@spatialelocation webapps]$ mv geocoder1.ear geocoder.ear
[oracle@spatialelocation webapps]$ cd geocoder.ear
[oracle@spatialelocation geocoder.ear]$ unzip geocoder1.ear
[oracle@spatialelocation geocoder.ear]$ mv web.war web1.war
[oracle@spatialelocation geocoder.ear]$ mkdir web.war
```

```
[oracle@spatialelocation geocoder.ear]$ mv web1.war web.war
[oracle@spatialelocation geocoder.ear]$ cd web.war
[oracle@spatialelocation web.war]$ unzip web1.war
[oracle@spatialelocation web.war]$ cp /u01/webapps/xmlparserv2.jar WEB-INF/lib
```

Next, you will modify the configuration file in the WEB-INF/conf/geocodercfg.xml

- Go to WEB-INF/conf
- Backup geocodercfg.xml to geocodercfg.xml.old
- Modify geocodercfg.xml

You can see the detail information how to config the geocodercfg.xml as following URL.

- 11.7.1.1 Configuring the geocodercfg.xml File specified in Oracle® Spatial Developer's Guide 11g Release 1 (11.1) Part Number B28400-05

  http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_geocode_concepts.htm

The example of modifying geocodercfg.xml is as following.

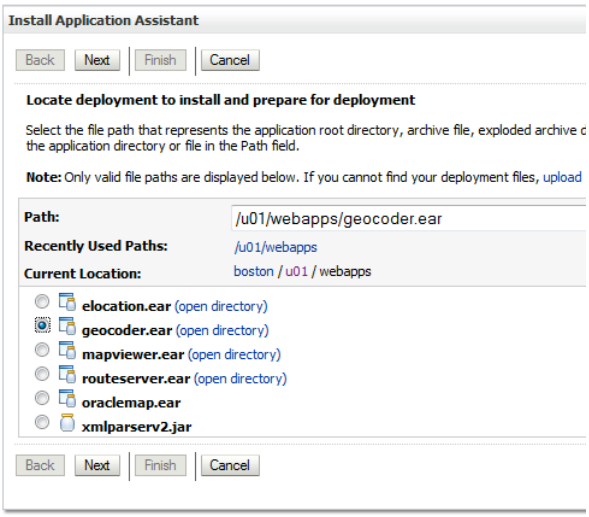
```
[oracle@spatialelocation web.war]$ cd WEB-INF/conf
[oracle@spatialelocation conf]$ vi geocodercfg.xml
        <geocoder>
            <database container_ds="jdbc/geocoder_na"
                    load_db_parser_profiles="true" />
    </geocoder>
 [oracle@spatialelocation conf]$
```

Next, you can deploy geocoder.ear to Oracle Weblogic Server.

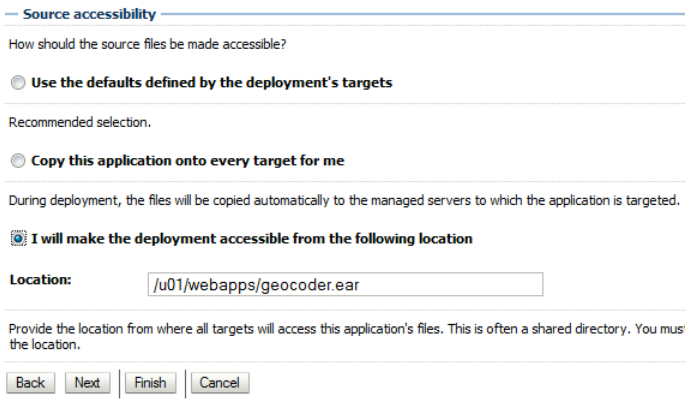
- Log in to the Weblogic Console with weblogic server credential, it is typically, weblogic/welcome1.

  Weblogic console URL : http://hostname:port/console

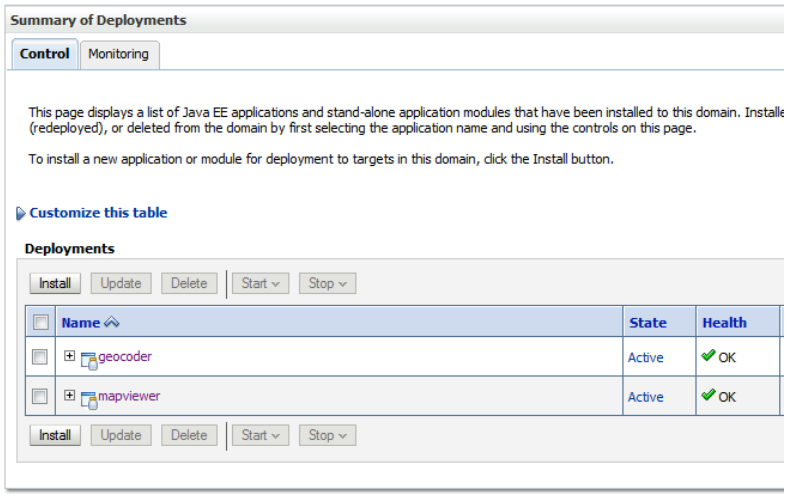
- Click Deployment under the left domain structure menu bar.
- Under Deployment, Click Install
- Select geocoder.ear under the directory of '/u01/webapps



- Select the targeting style to Install this deployment as an application
- Select the security to DD only.
- Select the source accessibility to “I will make the deployment accessible from the following location”.

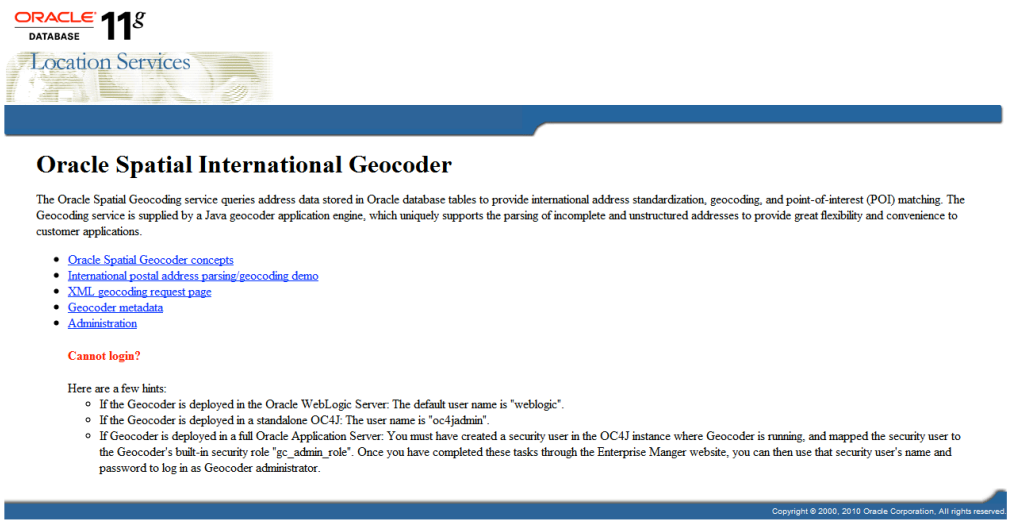


- Select the additional configuration to default, the you can see the deployment’s configuration screen.
- Click Finish



You can see geocoder page in the web browser to verify the installation is succedded.

- Geocoder URL : http://hostname:port/geocoder



Deploying Oracle RouteServer

This section explains how to config and deploy RouteServer to Oracle Weblogic Server. You can download and copy to proper directory in your system. You should unpack the routeserver.ear to any directory you want because this directory will be used for working and saving files. You can see the following instruction to unpack and name like that. It assumes that you have created the directory named /u01/webapps. In addtion, routeserver.jar needs xmlparsev2.jar file. xmlparserv2.jar is not included in the component. You can find it from WEBLOGIC_HOME/util/ccr/lib/xmlparserv2.jar.

The instruction of unpacking routeserver.ear is as following.

- Copy routeserver.ear to /u01/webapps
- Go to /u01/webapps
- Rename routeserver.ear to routeserver1.ear
- Create subdirectory named routeserver.ear under /u01/webapps
- Unpack routeserver1.ear into routeserver.ear directory
- Go to routeserver.ear directory
- Rename web.war to web1.war
- Create subdirectory named web.war in the /u01/webapps/routeserver.ear directory
- Unpack web1.war into web.war directory
- Copy xmlparserv2.jar to WEB-INF/lib (xmlparserv2.jar is not included in the package)

The example of unpacking mapviewer.ear is as following.

```
[oracle@spatialelocation webapps]$ ls
elocation.ear  geocoder.ear  mapviewer.ear  oraclemap.ear  routeserver.ear
xmlparserv2.jar
[oracle@spatialelocation webapps]$ mv routeserver.ear routeserver1.ear
[oracle@spatialelocation webapps]$ mkdir routeserver.ear
[oracle@spatialelocation webapps]$ mv routeserver1.ear routeserver.ear
[oracle@spatialelocation webapps]$ cd routeserver.ear
[oracle@spatialelocation routeserver.ear]$ unzip routeserver1.ear
```

```
[oracle@spatialelocation routeserver.ear]$ mv web.war web1.war
[oracle@spatialelocation routeserver.ear]$ mkdir web.war
[oracle@spatialelocation routeserver.ear]$ mv web1.war web.war
[oracle@spatialelocation routeserver.ear]$ cd web.war
[oracle@spatialelocation web.war]$ unzip web1.war
[oracle@spatialelocation web.war]$ cp /u01/webapps/xmlparserv2.jar WEB-INF/lib
```

Next, you will modify the configuration file in the WEB-INF/conf/web.xml

- Go to WEB-INF
- Backup web.xml to web.xml.old
- Modify web.xml

You can see the detail information how to config the web.xml as following URL.

- 13.1.1 Configuring the web.xml File specified in Oracle® Spatial Developer's Guide 11g Release 2 (11.2) Part Number B28400-05

  http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_route_server.htm

The example of modifying routeservercfg.xml is as following.

```
[oracle@spatialelocation web.war]$ cd WEB-INF
[oracle@spatialelocation WEB-INF]$ vi web.xml
…
<init-param>
   <param-name>routeserver_schema_jdbc_connect_string</param-name>
   <param-value>
jdbc:oracle:thin:@(description=(address=(host=spatialelocation)(protocol=tcp)(port=1521)
   (connect_data=(sid=ORCL)))
   </param-value>
</init-param>
<init-param>
   <param-name>routeserver_schema_username</param-name>
   <param-value>navteq_sf</param-value>
</init-param>
<init-param>
   <param-name>routeserver_schema_password</param-name>
   <param-value>oracle</param-value>
</init-param>
<init-param>
   <param-name>routeserver_network_name</param-name>
   <param-value>ROUTER_US_NETWORK</param-value>
</init-param>
<init-param>
   <param-name>geocoder_http_url</param-name>
   <param-value>http://spatialelocation:7001/geocoder/gcserver</param-value>
</init-param>
…
[oracle@spatialelocation WEB-INF]$
```

Note: When you specify the routeserver_network_name, you can get name from execuding the query of “select network from user_sdo_network_metadata”. .

Next, you can deploy routeserver.ear to Oracle Weblogic Server.

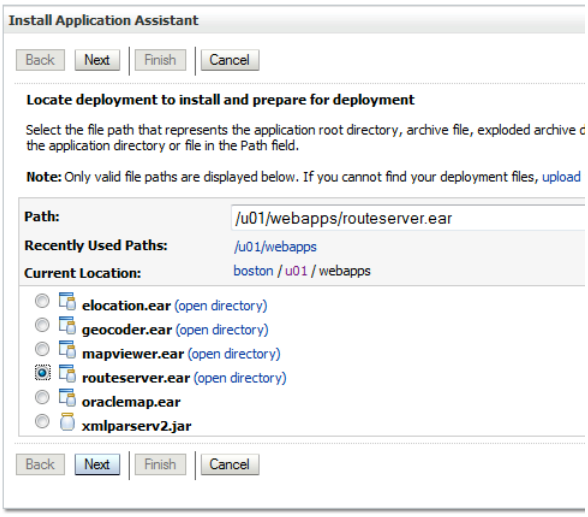
- Log in to the Weblogic Console with weblogic server credential, it is typically, weblogic/welcome1.

  Weblogic console URL : http://hostname:port/console

- Click Deployment under the left domain structure menu bar.

- Under Deployment, Click Install

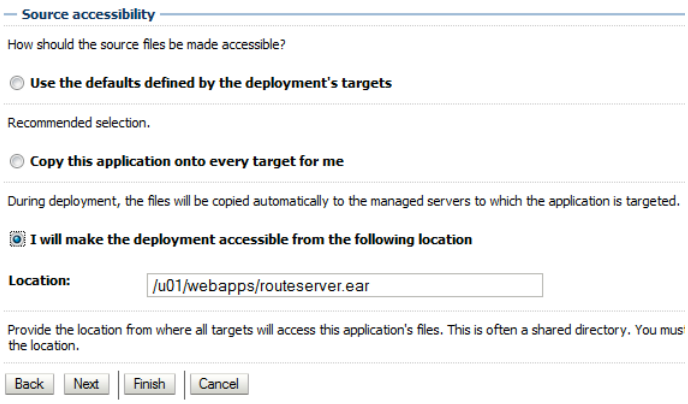
- Select routeserver.ear under the directory of '/u01/webapps



- Select the targeting style to Install this deployment as an application

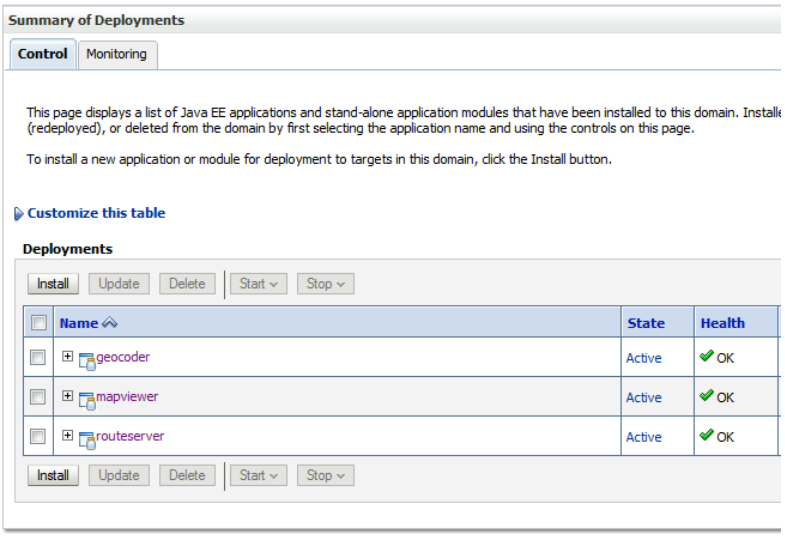
- Select the security to DD only.

- Select the source accessibility to "I will make the deployment accessible from the following location".



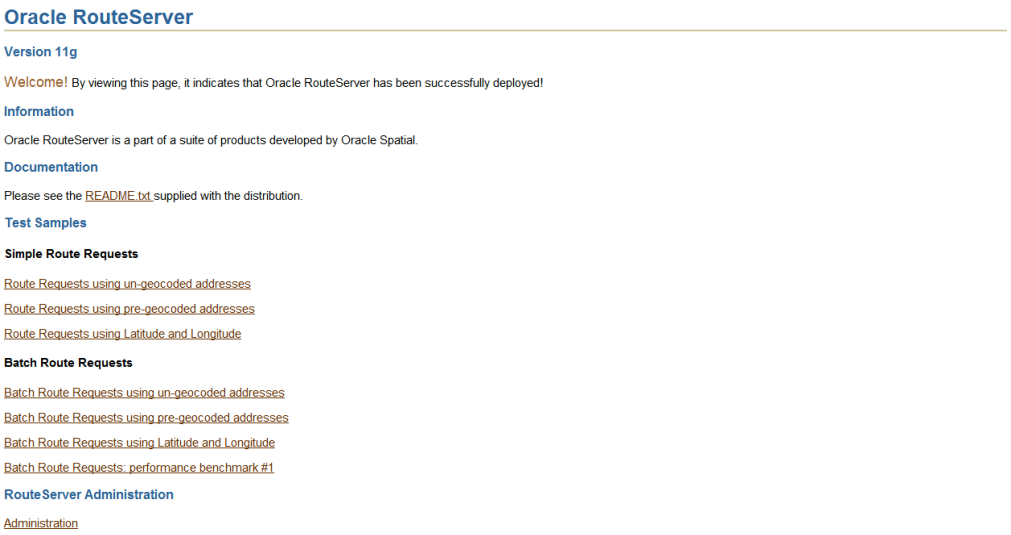
- Select the additional configuration to default, then you can see the deployment's configuration screen.

- Click Finish



You can see routeserver page in the web browser to verify the installation is succedded.

- Routeserver URL : http://hostname:port/routeserver



Deploying Oracle eLocation

This section explains how to configure and deploy eLocation to Oracle Weblogic Server. You can download and copy to proper directory in your system. You need to unpack the elocation.ear to adirectory since this will be used for working and saving files.Follow the instruction below to unpack and name eLocation required files and folders. It is assumed that you have created the directory named /u01/webapps where you will place your eLocation installation folder. Additionally, routeserver.jar needs xmlparsev2.jar file. xmlparserv2.jar is not included in the component. You can find it from WEBLOGIC_HOME/util/ccr/lib/xmlparserv2.jar.

The instructions for unpacking elocation.ear are as follows:

- Copy elocation.ear to /u01/webapps
- Go to /u01/webapps
- Rename elocation.ear to elocation1.ear
- Create subdirectory named elocation.ear under /u01/webapps
- Unpack elocation1.ear into elocation.ear directory

- Go to elocation.ear directory
- Rename web.war to web1.war
- Create subdirectory named web.war in the /u01/webapps/elocation.ear directory
- Unpack web1.war into web.war directory
- Copy xmlparserv2.jar to WEB-INF/lib (xmlparserv2.jar is not included in the package)

A command line example for unpacking elocation.ear is as follows:

```
[oracle@spatialelocation webapps]$ ls
elocation.ear  geocoder.ear  mapviewer.ear  oraclemap.ear  routeserver.ear
xmlparserv2.jar
[oracle@spatialelocation webapps]$ mv elocation.ear elocation1.ear
[oracle@spatialelocation webapps]$ mkdir elocation.ear
[oracle@spatialelocation webapps]$ mv elocation1.ear elocation.ear
[oracle@spatialelocation webapps]$ cd elocation.ear
[oracle@spatialelocation elocation.ear]$ unzip elocation1.ear
[oracle@spatialelocation elocation.ear]$ mv web.war web1.war
[oracle@spatialelocation elocation.ear]$ mkdir web.war
[oracle@spatialelocation elocation.ear]$ mv web1.war web.war
[oracle@spatialelocation elocation.ear]$ cd web.war
[oracle@spatialelocation web.war]$ unzip web1.war
[oracle@spatialelocation web.war]$ cp /u01/webapps/xmlparserv2.jar WEB-INF/lib
```

Next, you will modify the configuration file in the WEB-INF/config/dispatchercfg.xml

- Go to WEB-INF
- Backup dispatchercfg.xml to dispatchercfg.xml.old
- Modify dispatchercfg.xml

The example of modifying dispatchercfg.xml is as following.

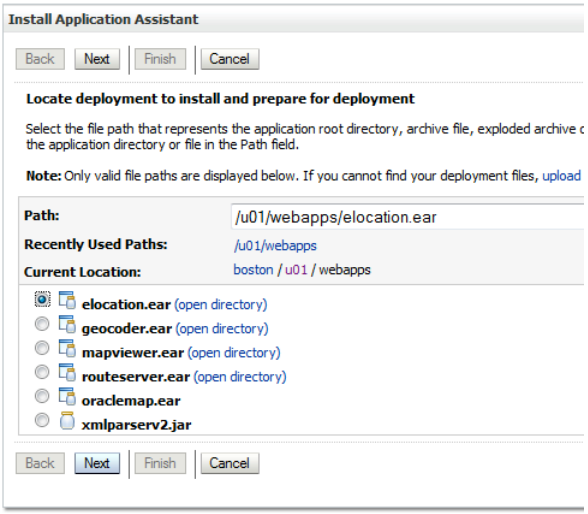
```
[oracle@spatialelocation web.war]$ cd WEB-INF/config
[oracle@spatialelocation config]$ vi dispatchercfg.xml
…
<mapper_cluster>
    <mapper url="http://spatialelocation:7001/mapviewer/omserver" />
</mapper_cluster>
<router_cluster>
    <router
url="http://spatialelocation:7001/routeserver/servlet/RouteServerServlet" />
</router_cluster>
<geocoder_cluster>
    <geocoder url="http://spatialelocation:7001/geocoder/gcserver" />
</geocoder_cluster>
  …
 [oracle@spatialelocation config]$
```

Next, you can deploy elocation.ear to Oracle Weblogic Server.

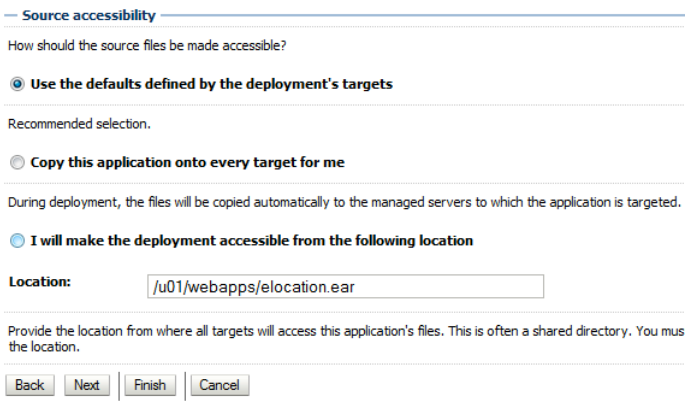
- Log in to the Weblogic Console with your Weblogic server administrator credentials (it is typically weblogic/welcome1):

  Weblogic console URL: http://hostname:port/console

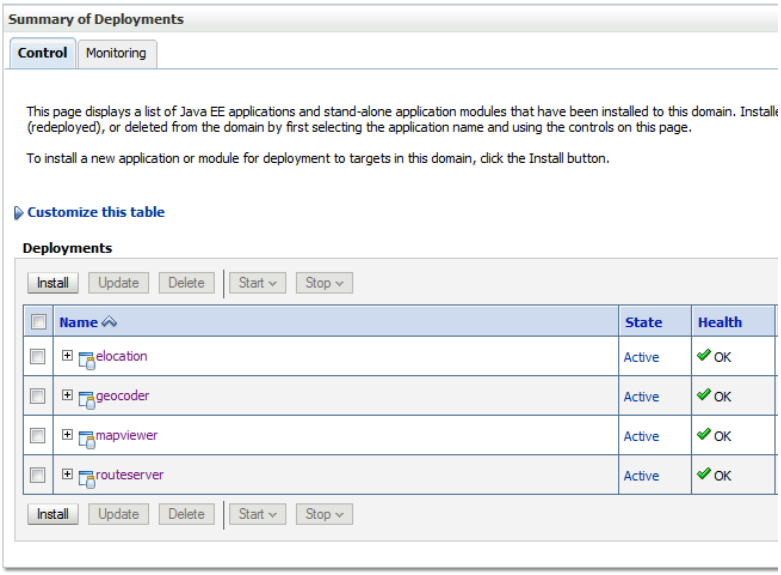
- Click Deployment under the left domain structure menu bar.
- Under Deployment, Click Install
- Select elocation.ear under the directory of '/u01/webapps



- Select the targeting style to Install this deployment as an application
- Select the security to DD only.
- Select the source accessibility to "I will make the deployment accessible from the following location".

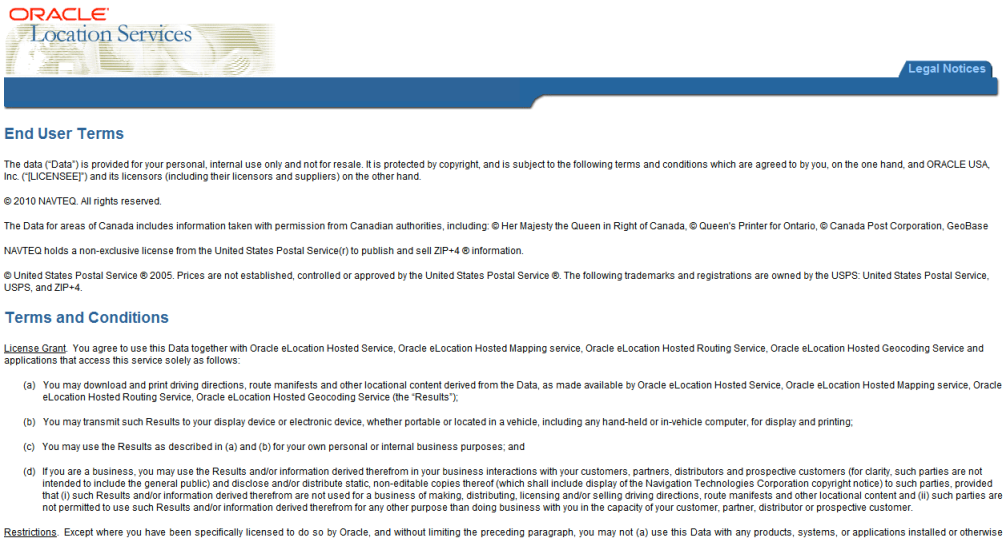


- Select the additional configuration to default, then you can see the deployment's configuration screen.
- Click Finish



You can see eLocation page in the web browser to verify the installation was successful.

- eLocation URL : http://hostname:port/elocation



- eLocation URL : http://hostname:port/elocation/ajax/index.html

  For this default application to work, it needs a data surce called “elocation” and a basemap with a name “world_map” in the MapViewer instance.

ORACLE

Location Services

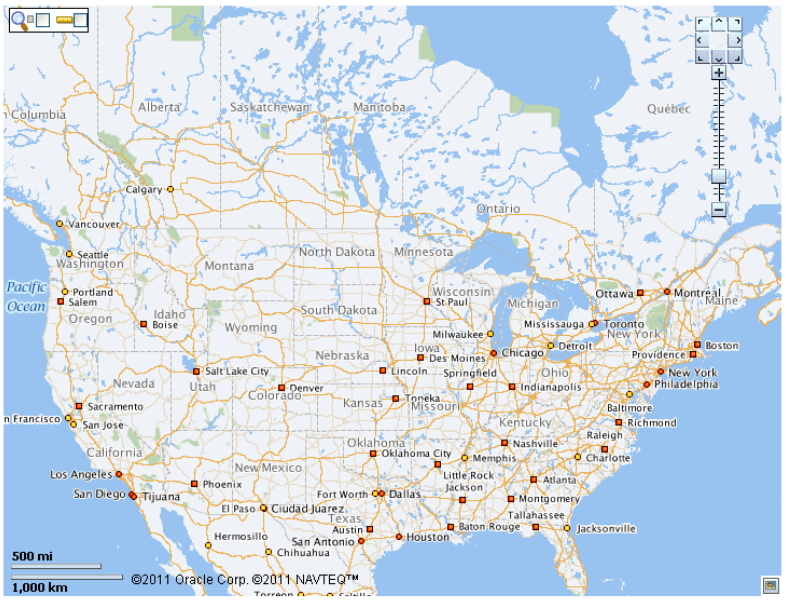
Mapping | Geocoding | Routing | Proximity Search | Oracle Spatial Technology Network

Search GO

Find Map | Directions

A Address, City, State or Postal Code...

Search



PART II : Oracle eLocation XML API

This part describes how to submit a request in XML format to Oracle Spatial eLocation Web Services, such as mapping, geocoding, and routing. It explains XML document type definitions for each service and shows you the examples of requests and responses.Oracle Spatial web services have XML APIs to provide data and metadata of their services through HTTP protocol.

This part does not explain all APIs but just show you some example how to use and DTDs or schema file for each service. You can download the XML request and response sample Java code from the following link. Or you can find it in the middle-tier virtual box. It gives you the sample project from Eclipse which has various examples from mapping, geocoding, and routing.

- The sample Java project file for Oracle Spatial Web Services XML API

  You can find sample source code in the virual box kit for Oracle Spatial

The sample project file has the following examples java file which can execute.

- Mapping

  MapRequestSimple.java

  MapRequestWithBox.java

  MapRueqestWithCenter.java

  MapRequestWithFeatures.java

  MapRequestWithLegend.java

  MapRequestWithScalebar.java

  MapRequestWithStatistics.java

- Geocoding

  GeocodingRequestWithGenForm.java

  GeocodingRequestWithMulti.java

  GeocodingRequestWithUnformmatted.java

  GeocodingRequestWithUSForm1.java

  GeocodingRequestWithUSForm2.java

- Routing

  BatchRoutingRequest.java

  RoutingRequestWithAddress.java

  RoutingRequestWithMultiAddress.java

Mapping XML API

This section provides examples of mapping request and response. You can see the DTD of map request is as following.

- 3.2 Map Request DTD specified in Oracle® Fusion Middleware User's Guide for Oracle MapViewer 11g Release 1 (11.1.1) E10145-04

  http://download.oracle.com/docs/cd/E14571_01/web.1111/e10145/vis_xmlapi.htm#i1017276

The example of request mapping with center is as following.

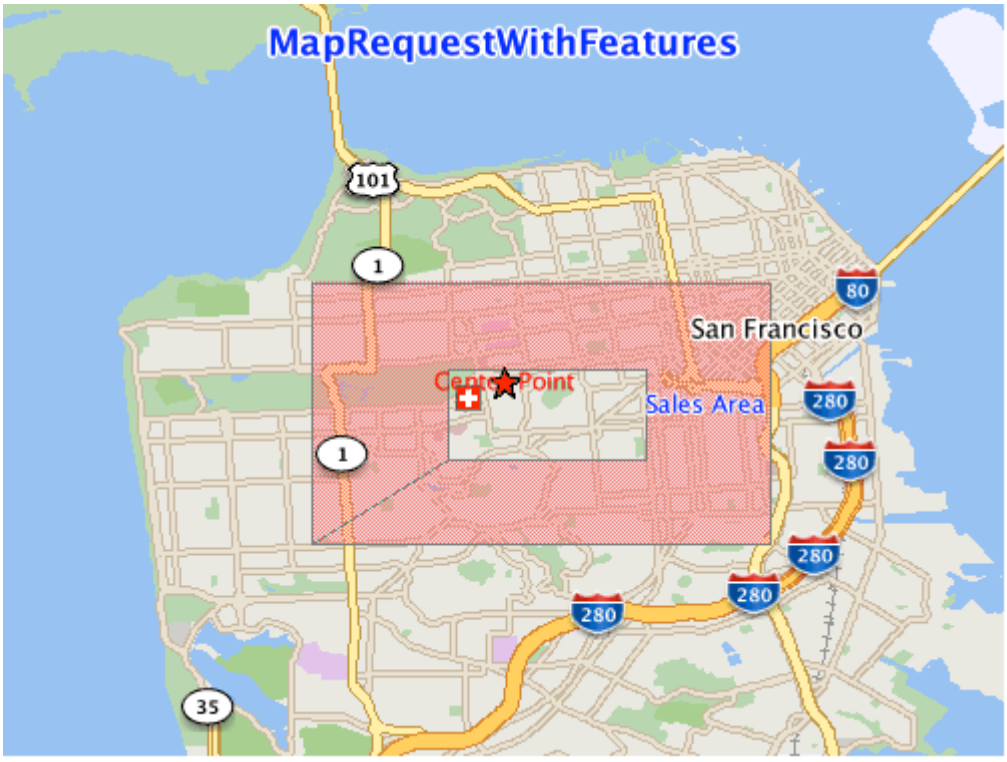
```
<?xml version="1.0" encoding="UTF-8"?>
<map_request datasource="elocation" title="MapRequestWithFeatures"
basemap="world_map" width="500" height="375" bgcolor="#a6cae0" format="GIF_URL">>
<center size="0.125">
<geoFeature label="Center Point" text_style="M.STAR" render_style="M.STAR" >
<geometricProperty typeName="center">
<Point>
<coordinates>-122.45, 37.7706</coordinates>
</Point>
</geometricProperty>
</geoFeature>
</center>
<srs>SDO:8307</srs>
<themes>
<theme name="M_MAJOR_ROADS_S08"/>
</themes>
<geoFeature typeName="nil" label="Sales Area" text_style="9988"
render_style="A.CCTR_SUPPORT_F97E80">
<geometricProperty>
<MultiPolygon srsName="SDO:8307">
<polygonMember>
<Polygon>
<innerBoundaryIs>
<LinearRing>
<coordinates>-122.4820,37.7434,-122.4057,37.7434,-122.4057,37.7869,-
122.4820,37.7869,-122.4820,37.7434,-122.4593,37.7573,-122.4593,37.7724,-
122.4262,37.7724,-122.4262,37.7573,-122.4593,37.7573</coordinates>
</LinearRing>
</innerBoundaryIs>
</Polygon>
</polygonMember>
</MultiPolygon>
</geometricProperty>
</geoFeature>
<geoFeature render_style="M.HOSPITAL_1" text_style="9987">
<geometricProperty>
<Point>
<coordinates>-122.4559, 37.7678</coordinates>
</Point>
</geometricProperty>
</geoFeature>
</map_request>
```

The example of response mapping with center is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<map_response>
 <map_image>
```

```
  <map_content url="http://spatialelocation:7001/mapviewer/images/omsmap5_25.gif?refresh=-
2892865288326176637" />
  <box srsName="sdo:8307">
    <coordinates>
-122.53333333333333,37.7081  -122.36666666666666,37.8331     </coordinates>
  </box>
  <themes>
    <theme name="M_BACKDROP_LAND_S11" />
    <theme name="M_BUILTUP_S11" />
    <theme name="M_OCEAN_DEEP_S11" />
    <theme name="M_HARBOR_BAY_S11" />
    <theme name="M_ISLANDS_S11" />
    <theme name="M_FACILITIES_S11" />
    <theme name="M_PARKS_S11" />
    <theme name="M_WATER_FRESH_S11" />
    <theme name="M_RAILWAYS_S11" />
    <theme name="M_SECONDARY_ROADS_S11" />
    <theme name="M_MAJOR_ROADS_S11" />
    <theme name="M_MINOR_HIGHWAYS_S11" />
    <theme name="M_MAJOR_HIGHWAYS_S11" />
    <theme name="M_CITIES_POINTS_S11" />
    <theme name="M_MAJOR_ROADS_S08" />
  </themes>
  <xfm matrix="3.333333333333428E-4 0.0 0.0 -3.333333333333428E-4 -
122.53333333333333 37.83310000000001" />
  <WMTException version="1.0.0" error_code="SUCCESS">
  </WMTException>
 </map_image>
</map_response>
```

The screen captured image of response mapping with center is as following.



The example of request mapping with center is as following.

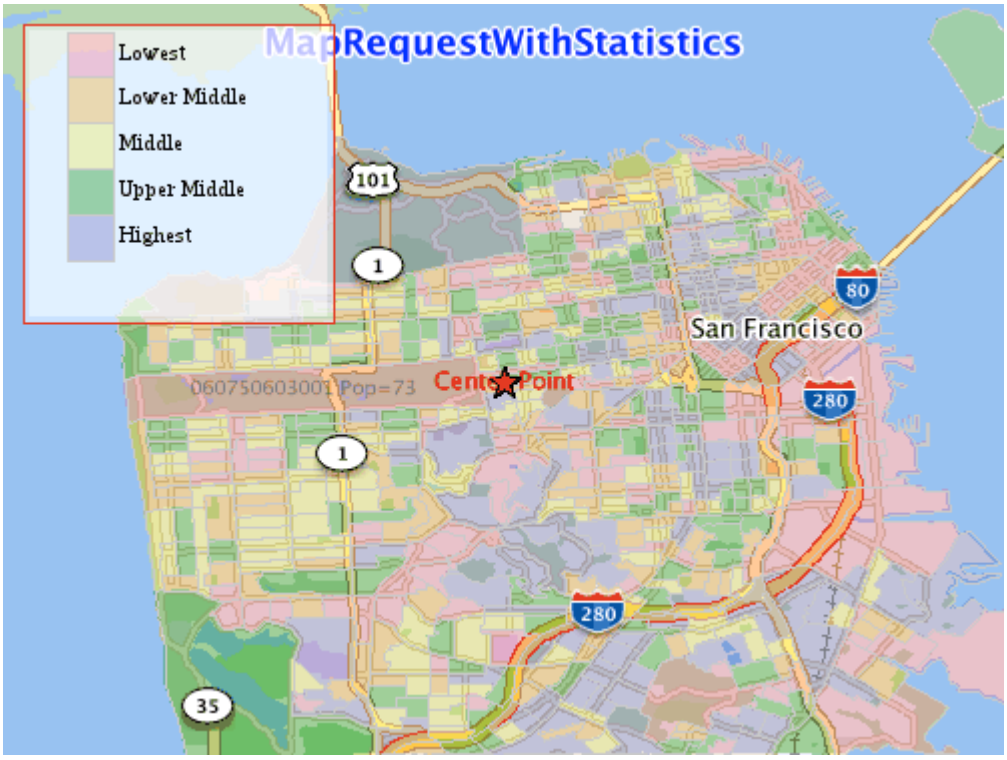
```
<?xml version="1.0" encoding="UTF-8"?>
<map_request datasource="elocation" title="MapRequestWithStatistics"
basemap="world_map" width="500" height="375" bgcolor="#a6cae0" format="GIF_URL">>
<center size="0.125">
<geoFeature label="Center Point" text_style="M.STAR" render_style="M.STAR" >
<geometricProperty typeName="center">
<Point>
<coordinates>-122.45, 37.7706</coordinates>
</Point>
</geometricProperty>
</geoFeature>
</center>
<srs>SDO:8307</srs>
<themes>
<theme name="BLK_GRP_POPULATION"/>
</themes>
<legend bgstyle="fill:#ffffff;fill-opacity:128;stroke:#ff0000" position="NORTH_WEST">
<column>
<entry style="V.BLOCK_GROUPS_POPULATION" tab="1"/>
</column>
</legend>
</map_request>
```

The example of response mapping with center is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<map_response>
 <map_image>
  <map_content
url="http://spatialelocation:7001/mapviewer/images/omsmap9_25.gif?refresh=4021167672648852594"
/>
  <box srsName="sdo:8307">
    <coordinates>
-122.53333333333333,37.7081  -122.36666666666666,37.8331     </coordinates>
  </box>
  <themes>
    <theme name="M_BACKDROP_LAND_S11" />
    <theme name="M_BUILTUP_S11" />
    <theme name="M_OCEAN_DEEP_S11" />
    <theme name="M_HARBOR_BAY_S11" />
    <theme name="M_ISLANDS_S11" />
    <theme name="M_FACILITIES_S11" />
    <theme name="M_PARKS_S11" />
    <theme name="M_WATER_FRESH_S11" />
    <theme name="M_RAILWAYS_S11" />
    <theme name="M_SECONDARY_ROADS_S11" />
    <theme name="M_MAJOR_ROADS_S11" />
    <theme name="M_MINOR_HIGHWAYS_S11" />
    <theme name="M_MAJOR_HIGHWAYS_S11" />
    <theme name="M_CITIES_POINTS_S11" />
    <theme name="BLK_GRP_POPULATION" />
  </themes>
  <xfm matrix="3.333333333333428E-4 0.0 0.0 -3.333333333333428E-4 -122.53333333333333
37.83310000000001" />
  <WMTException version="1.0.0" error_code="SUCCESS">
```

```
 </WMTException>
 </map_image>
</map_response>
```

The screen captured image of response mapping with center is as following.



Geocoding XML API

This section provides examples of geocoding request and response. You can see the schema of geocoding request is as following.

- 11.7 Using the Geocoding Service (XML API) specified in Oracle® Spatial Developer's Guide 11g Release 2 (11.2) E11830-07

  http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11830/sdo_geocode_concepts.htm#CIHFABDJ

The example of request geocoding with USForm2 is as following.

```
<?xml version="1.0" standalone="yes"?>
<geocode_request vendor="elocation">
<address_list>
<input_location id="1">
<input_address match_mode="RELAX_ALL">
<us_form2 name="San Francisco State University" state="ca"/>
</input_address>
</input_location>
</address_list>
</geocode_request>
```

The example of response geocoding with USForm2 is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<geocode_response>
<geocode id="1" match_count="1">
<match sequence="0" longitude="-122.47711" latitude="37.72081" match_code="1"
```

```
error_message="???O#ENUT?B281CP?" match_vector="???04141114??004?">
<output_address name="SAN FRANCISCO STATE UNIVERSITY" house_number="1600"
street="HOLLOWAY AVE" builtup_area="SAN FRANCISCO" order1_area="CA"
order8_area="" country="US" postal_code="94132" postal_addon_code="" side="R"
percent="0.0" edge_id="199227588" />
</match>
</geocode>
</geocode_response>
```

The example of request geocoding with multi is as following.

```
<?xml version="1.0" standalone="yes"?>
<geocode_request vendor="elocation">
<address_list>
<input_location id="1">
<input_address match_mode="RELAX_ALL">
<us_form2 street="500 oracle pky" city="redwood city" state="ca"/>
</input_address>
</input_location>
<input_location id="2">
<input_address match_mode="RELAX_ALL">
<us_form2 street="500 oracle pky" city="redwood city" state="ca"/>
</input_address>
</input_location>
<input_location id="3">
<input_address match_mode="RELAX_ALL">
<gen_form street="oracle" city="redwood city" region="CA" postal_code=""  country="US"/>
</input_address>
</input_location>
<input_location id="4">
<input_address match_mode="RELAX_ALL">
<unformatted country="UNITED STATES">
<address_line value="oracle"/>
<address_line value="redwood city"/>
<address_line value="CA"/>
</unformatted></input_address>
</input_location>
</address_list>
</geocode_request>
```

The example of response geocoding with USForm2 is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<geocode_response>
<geocode id="1" match_count="2">
<match sequence="0" longitude="-122.2271" latitude="37.48417" match_code="4"
error_message="??????????B281CP?" match_vector="???13131310??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94063" postal_addon_code=""
side="L" percent="0.0" edge_id="199138467" />
</match>
<match sequence="1" longitude="-122.17498" latitude="37.32519" match_code="4"
error_message="??????????B281CP?" match_vector="???13131310??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94022" postal_addon_code=""
side="L" percent="0.0" edge_id="198576594" />
```

```
</match>
</geocode>
<geocode id="2" match_count="2">
<match sequence="0" longitude="-122.2271" latitude="37.48417" match_code="4"
error_message="??????????B281CP?" match_vector="???13131310??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94063" postal_addon_code=""
side="L" percent="0.0" edge_id="199138467" />
</match>
<match sequence="1" longitude="-122.17498" latitude="37.32519" match_code="4"
error_message="??????????B281CP?" match_vector="???13131310??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94022" postal_addon_code=""
side="L" percent="0.0" edge_id="198576594" />
</match>
</geocode>
<geocode id="3" match_count="2">
<match sequence="0" longitude="-122.2271" latitude="37.48417" match_code="4"
error_message="??????????B281CP?" match_vector="???11131110??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94063" postal_addon_code=""
side="L" percent="0.0" edge_id="199138467" />
</match>
<match sequence="1" longitude="-122.17498" latitude="37.32519" match_code="4"
error_message="??????????B281CP?" match_vector="???11131110??004?">
<output_address name="" house_number="" street="" builtup_area="REDWOOD CITY"
order1_area="CA" order8_area="" country="US" postal_code="94022" postal_addon_code=""
side="L" percent="0.0" edge_id="198576594" />
</match>
</geocode>
<geocode id="4" match_count="1">
<match sequence="0" longitude="-122.26195" latitude="37.53202" match_code="2"
error_message="???O#ENU??B281CP?" match_vector="???04101110??004?">
<output_address name="ORACLE" house_number="500" street="ORACLE PKY"
builtup_area="REDWOOD CITY" order1_area="CA" order8_area="" country="US"
postal_code="94065" postal_addon_code="" side="L" percent="0.0" edge_id="199238503" />
</match>
</geocode>
</geocode_response>
```

Routing XML API

This section provides examples of routing request and response. You can see the DTD of map request is as following link.

- 13.3 Routing Engine XML API specified in Oracle® Spatial Developer's Guide 11g Release 2 (11.2) E11830-07

  http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11830/sdo_route_server.htm#insertedID3

The example of request routing with multi address is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<route_request id="1" route_preference="fastest" road_preference="LOCAL"
return_driving_directions="true" distance_unit="mile" time_unit="minute">
```

```
<start_location>
<input_location id="1">
<input_address>
<us_form2 name="San Francisco State University" state="ca"/>
</input_address>
</input_location>
</start_location>
<location>
<input_location id="1" longitude="-122.4174" latitude="37.7669" />
</location>
<end_location>
<input_location id="2" longitude="-122.4259" latitude="37.7485" />
</end_location>
</route_request>
```

The example of response request routing with multi address is as following.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Oracle Routeserver version 11.2.0.2.11 (data version 10.2.0.5.0) -->
<route_response>
  <route id="1" step_count="22" distance="30.52065351103741" distance_unit="mile"
time="35.07154261271159" time_unit="minute" start_location="1" end_location="2">
    <subroute id="1" step_count="13" distance="28.77214909044305" distance_unit="mile"
time="30.80810546875" time_unit="minute" start_location="1" end_location="1">
      <segment sequence="1" instruction="Start out on FIVE CANYONS PKY (Going
Northwest)" distance="1.721966088703345" time="2.8866250276565553"/>
      <segment sequence="2" instruction="Turn LEFT onto E CASTRO VALLEY BLVD (Going
West)" distance="0.3260175236444954" time="0.5465208341677984"/>
      <segment sequence="3" instruction="Turn LEFT onto RAMP (Going West)"
distance="0.22076057230701043" time="0.37007290522257485"/>
      <segment sequence="4" instruction="Turn SLIGHT RIGHT onto I-580 W (Going West)"
distance="17.653576130820486" time="15.499845046301683"/>
      <segment sequence="5" instruction="Stay STRAIGHT to go onto RAMP (Going West)"
distance="0.8104703866812483" time="0.8360833406448365"/>
      <segment sequence="6" instruction="Stay STRAIGHT to go onto I-80 W (Going West)"
distance="0.03431926896304998" time="0.04184090693791707 4"/>
      <segment sequence="7" instruction="Stay STRAIGHT to go onto RAMP (Going West)"
distance="0.9787796230051862" time="1.0097115675608317"/>
      <segment sequence="8" instruction="Stay STRAIGHT to go onto I-80 W (Going West)"
distance="1.7096625865968755" time="2.0843636274337767"/>
      <segment sequence="9" instruction="Stay STRAIGHT to go onto BAY BRG (Going
West)" distance="2.990182095419025" time="3.645530273516973"/>
      <segment sequence="10" instruction="Stay STRAIGHT to go onto I-80 W (Going
Southwest)" distance="1.3486360509511306" time="1.644212124745051"/>
      <segment sequence="11" instruction="Turn SLIGHT RIGHT onto RAMP (Going
Southwest)" distance="0.18485677106721357" time="0.30988541841506956"/>
      <segment sequence="12" instruction="Turn LEFT onto HARRISON ST (Going
Southwest)" distance="0.5674330448782823" time="1.3835909048716226"/>
      <segment sequence="13" instruction="Turn RIGHT onto 15TH ST (Going West)"
distance="0.22549067365518005" time="0.5498214165369669"/>
    </subroute>
    <subroute id="2" step_count="9" distance="1.7485044205943578" distance_unit="mile"
time="4.263437143961588" time_unit="minute" start_location="1" end_location="2">
      <segment sequence="1" instruction="Start out on 15TH ST (Going West)"
distance="0.013451721716688618" time="0.032799782355626424"/>
```

```
        <segment sequence="2" instruction="Turn LEFT onto S VAN NESS AVE (Going South)"
distance="0.5571242095239937" time="1.3584545334180196"/>
        <segment sequence="3" instruction="Turn RIGHT onto 20TH ST (Going West)"
distance="0.057639969015242945" time="0.14054546356201172"/>
        <segment sequence="4" instruction="Turn LEFT onto CAPP ST (Going South)"
distance="0.21832473177451475" time="0.5323484738667806"/>
        <segment sequence="5" instruction="Turn RIGHT onto 22ND ST (Going West)"
distance="0.20635679881805172" time="0.503166651725769"/>
        <segment sequence="6" instruction="Turn LEFT onto SAN JOSE AVE (Going South)"
distance="0.2207046591279254" time="0.5381515423456827"/>
        <segment sequence="7" instruction="Turn RIGHT onto 24TH ST (Going West)"
distance="0.06957062019789514" time="0.1696363608042399"/>
        <segment sequence="8" instruction="Turn LEFT onto GUERRERO ST (Going South)"
distance="0.22246317933687687" time="0.5424393971761068"/>
        <segment sequence="9" instruction="Turn RIGHT onto 26TH ST (Going West)"
distance="0.18286868575227747" time="0.44589481751124066"/>
      </subroute>
    </route>
</route_response>
```

The example of request batch routing is as following.

```
<?xml version="1.0" encoding="UTF-8"?>
<batch_route_request id="1" route_preference="fastest" road_preference="LOCAL"
sort_by_distance = "true" cutoff_distance="35" distance_unit="mile" time_unit="minute">
<start_location>
<input_location id="1">
<input_address>
<us_form2 name="San Francisco State University" state="ca"/>
</input_address>
</input_location>
</start_location>
<end_location>
<input_location id="2" longitude="-122.4174" latitude="37.7669" />
</end_location>
<end_location>
<input_location id="3" longitude="-122.4259" latitude="37.7485" />
</end_location>
</batch_route_request>
```

The example of response request routing with multi address is as following.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- Oracle Routeserver version 11.2.0.2.11 (data version 10.2.0.5.0) -->
<batch_route_response id="1">
<route id="2" step_count="0" distance="28.77214909044305" distance_unit="mile"
time="30.80810546875" time_unit="minute" />
<route id="3" step_count="0" distance="30.792201966693593" distance_unit="mile"
time="34.73306477864583" time_unit="minute" />
</batch_route_response>
```

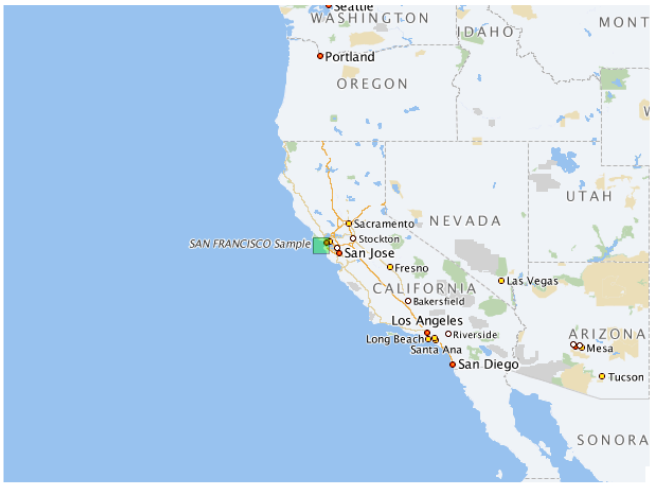
ORACLE

eLocation Services

Mapviewer | Geocoder | Route | Elocation | Tutorial | Mapping API | elocation API | Quickstartguide

Search | Directions

search



Copyright © 2011, Oracle Corp. All Rights Reserved

Legal Notices | Terms and Conditions | NAVTEQ | Feedback