

HOW TO DEPLOY ON PREMISE LOCATION SERVICES: A QUICK START GUIDE

ORACLE®

Contents

QUICK START GUIDE	3
OVERVIEW	4
OBJECTIVES	4
AUDIENCES	4
PART I : INSTALLING ORACLE SPATIAL AND GRAPH COMPONENTS	5
BEFORE YOU START	5
THE OVERVIEW OF ORACLE SPATIAL AND GRAPH COMPONENTS	5
REQUIRED SOFTWARE	6
ORACLE SPATIAL AND GRAPH, MAPVIEWER COMPONENTS	6
DEPLOYING ORACLE FUSION MIDDLEWARE MAPVIEWER	6
DEPLOYING THE GEOCODER	8
DEPLOYING THE ROUTESERVER	11
DEPLOYING THE ELOCATION SERVICES	14
PART II : ORACLE ELOCATION XML API	18
MAPPING XML API	18
GEOCODING XML API	21
ROUTING XML API	23

A Quick Start Guide

Overview

Objectives

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

The first part describes how to install and configure Oracle Spatial and Graph Components such as Oracle FMW MapViewer, the Oracle geocoder, the Oracle routeServer, and the eLocation Services. It is based on Oracle Weblogic Server.

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

Audiences

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

PART I: Installing Oracle Spatial and Graph Components

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

If you want to install Oracle Spatial and Graph Web Services in a production environment, we recommend that the memory size of the WebLogic Server where MapViewer is deployed needs to be set over 2 gbytes. The geocoder memory size should also be over 2 gbytes. The memory size of the WL server where RouteServer is deployed to needs to be set over 6G bytes memory. The RouteServer performance is tied directly to the number of partitions that are in memory. Use the following guidelines for WL Server memory sizing:

- North America – 12-20 gbytes
- Europe – 12-30 gbytes
- South America – 4-10 gbytes
- Individual countries depending on size of road network – 2-6 gbytes

It is more stable to install MapViewer, geocoder and RouteServer to separate domains and servers.

If you don't want to import Oracle Spatial and Graph Components in your 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 and Graph Components.

You can get the partner data as the following site.

- Oracle Spatial and Graph Partners' Data

<http://www.oracle.com/technetwork/database-options/spatialandgraph/downloads/spatial-partners-data-087203.html>

The overview of Oracle Spatial and Graph Components

- Oracle FMW 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 Geocoder

The 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".

- The RouteServer

The 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".

- eLocation Services

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

Required Software

The Oracle Spatial and Graph 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 Graph 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 Fusion MiddleWare 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 addition, mapviewer.jar needs xmlparserv2.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@spatiallocation webapps]$ ls
elocation.ear geocoder.ear mapviewer.ear oraclemap.ear routeserver.ear
xmlparserv2.jar
[oracle@spatiallocation webapps]$ mv mapviewer.ear mapviewer1.ear
[oracle@spatiallocation webapps]$ mkdir mapviewer.ear
[oracle@spatiallocation webapps]$ mv mapviewer1.ear mapviewer.ear
[oracle@spatiallocation webapps]$ cd mapviewer.ear
[oracle@spatiallocation mapviewer.ear]$ unzip mapviewer1.ear
[oracle@spatiallocation mapviewer.ear]$ mv web.war web1.war
[oracle@spatiallocation mapviewer.ear]$ mkdir web.war
[oracle@spatiallocation mapviewer.ear]$ mv web1.war web.war
[oracle@spatiallocation mapviewer.ear]$ cd web.war
[oracle@spatiallocation web.war]$ unzip web1.war
```

```
[oracle@spatiallocation 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@spatiallocation web.war]$ cd WEB-INF/conf
[oracle@spatiallocation conf]$ vi mapViewerConfig.xml
<map_data_source name="elocation"
    container_ds="jdbc/maperDS"
    number_of_mappers="3"
    allow_jdbc_theme_based_foi="true"
/>

[oracle@spatiallocation conf]$
```

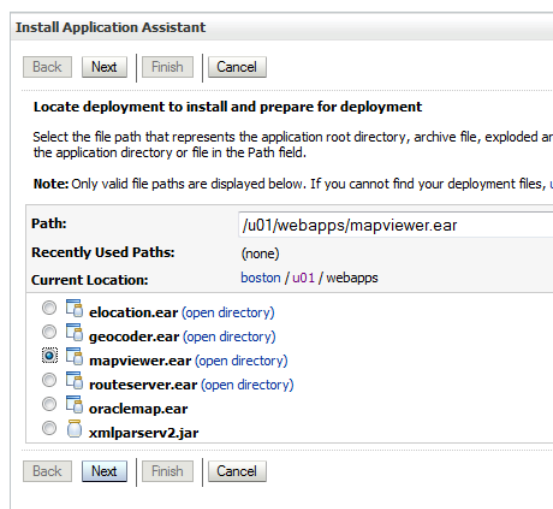
Note: You should use the container data source defined in WLS instead of a direct JDBC connection.

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"

Source accessibility

How should the source files be made accessible?

☒ **Use the defaults defined by the deployment's targets**

Recommended selection.

☐ **Copy this application onto every target for me**

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

☒ **I will make the deployment accessible from the following location**

Location:

Provide the location from where all targets will access this application's files. This is often a shared directory. You must enter the location.

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

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Install (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Name	State	Health
mapviewer	Active	OK

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

- MapViewer URL : <http://hostname:port/mapviewer>

ORACLE 11g FUSION MIDDLEWARE MAPVIEWER

MapViewer

Admin Logout Help

Location, Location, Location **Home** Requests Demos Meta data APIs

Welcome!

Your MapViewer server is now running and you can start viewing your Oracle Spatial managed data. MapViewer is a component of Oracle Application Server. It provides powerful geospatial data visualization services.

Getting started

Here are a few things that can get you started.

- Use MapBuilder to import some spatial data into an Oracle database that you have access to.
- Create a data source so that MapViewer can connect to the database. Please log into the Admin page to do so.
- Use the JView demo to quickly view your spatial data. You can find it under the Demos tab.
- Start developing your first MapViewer applications.

What is Oracle Maps

Oracle Maps is a new feature of MapViewer. It consists of a free scrolling AJAX-based web mapping interface, a flexible and open JavaScript API, a server-side map image caching engine and a Feature of Interest (FOI) server.

- [Oracle Maps Tutorial](#)
- [Oracle Maps API Reference](#)

Home | Requests | Demos | Meta data | APIs | Admin | Logout | Help

Copyright (c) 2002, 2006, Oracle. All rights reserved.
OracleAS MapViewer Version: Ver11_1_T_4_B101230

Deploying the 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 addition, geocoder.jar needs xmlparserv2.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@spatiallocation webapps]$ ls
elocation.ear geocoder.ear mapviewer.ear oraclemap.ear routeserver.ear
xmlparserv2.jar
[oracle@spatiallocation webapps]$ mv geocoder.ear geocoder1.ear
[oracle@spatiallocation webapps]$ mkdir geocoder.ear
[oracle@spatiallocation webapps]$ mv geocoder1.ear geocoder.ear
[oracle@spatiallocation webapps]$ cd geocoder.ear
[oracle@spatiallocation geocoder.ear]$ unzip geocoder1.ear
[oracle@spatiallocation geocoder.ear]$ mv web.war web1.war
[oracle@spatiallocation geocoder.ear]$ mkdir web.war
[oracle@spatiallocation geocoder.ear]$ mv web1.war web.war
[oracle@spatiallocation geocoder.ear]$ cd web.war
[oracle@spatiallocation web.war]$ unzip web1.war
[oracle@spatiallocation 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 and Graph Developer's Guide 12c Release 1 (12.1)** E17896-10
http://docs.oracle.com/cd/E16655_01/appdev.121/e17896/sdo_geocode_concepts.htm#CIHECECC

The example of modifying geocodercfg.xml is as following.

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

Install Application Assistant

Back Next Finish Cancel

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive or the application directory or file in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, [upload](#)

Path: /u01/webapps/geocoder.ear

Recently Used Paths: /u01/webapps

Current Location: boston / u01 / webapps

- ☐ elocation.ear (open directory)
- ☒ geocoder.ear (open directory)
- ☐ mapviewer.ear (open directory)
- ☐ routeserver.ear (open directory)
- ☐ oraclemap.ear
- ☐ xmlparserv2.jar

Back Next Finish Cancel

- 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".

Source accessibility

How should the source files be made accessible?

☐ Use the defaults defined by the deployment's targets

Recommended selection.

☐ Copy this application onto every target for me

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

☒ I will make the deployment accessible from the following location

Location: /u01/webapps/geocoder.ear

Provide the location from where all targets will access this application's files. This is often a shared directory. You must provide the location.

Back Next Finish Cancel

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

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Install (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install Update Delete Start Stop

Name	State	Health
geocoder	Active	OK
mapviewer	Active	OK

Install Update Delete Start Stop

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

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

Oracle Spatial International Geocoder

The Oracle Spatial Geocoding service queries address data stored in Oracle database tables to provide international address standardization, geocoding, and point-of-interest (POI) matching. The Geocoding service is supplied by a Java geocoder application engine, which uniquely supports the parsing of incomplete and unstructured addresses to provide great flexibility and convenience to customer applications.

- [Oracle Spatial Geocoder concepts](#)
- [XML geocoding request page](#)
- [Geocoder metadata](#)
- [Administration](#)

Cannot login?

Here are a few hints:

- If the Geocoder is deployed in the Oracle WebLogic Server. The default user name is "weblogic".
- If the Geocoder is deployed in a standalone OC4J. The user name is "oc4jadmin".
- If Geocoder is deployed in a full Oracle Application Server. You must have created a security user in the OC4J instance where Geocoder is running, and mapped the security user to the Geocoder's built-in security role "gc_admin_role". Once you have completed these tasks through the Enterprise Manager website, you can then use that security user's name and password to login as Geocoder administrator.

Copyright © 2000, 2010 Oracle Corporation. All rights reserved.

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 addition, routeserver.jar needs xmlparserv2.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@spatiallocation webapps]$ ls
elocation.ear geocoder.ear mapviewer.ear oraclemap.ear routeserver.ear xmlparserv2.jar
[oracle@spatiallocation webapps]$ mv routeserver.ear routeserver1.ear
[oracle@spatiallocation webapps]$ mkdir routeserver.ear
[oracle@spatiallocation webapps]$ mv routeserver1.ear routeserver.ear
[oracle@spatiallocation webapps]$ cd routeserver.ear
[oracle@spatiallocation routeserver.ear]$ unzip routeserver1.ear
[oracle@spatiallocation routeserver.ear]$ mv web.war web1.war
[oracle@spatiallocation routeserver.ear]$ mkdir web.war
[oracle@spatiallocation routeserver.ear]$ mv web1.war web.war
[oracle@spatiallocation routeserver.ear]$ cd web.war
[oracle@spatiallocation web.war]$ unzip web1.war
[oracle@spatiallocation 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 and Graph Developer's Guide 12c Release 1 (12.1)** E17896-10

http://docs.oracle.com/cd/E16655_01/appdev.121/e17896/sdo_route_server.htm#CCHGDJJD

The example of modifying routeservercfg.xml is as following.

```
[oracle@spatiallocation web.war]$ cd WEB-INF
[oracle@spatiallocation WEB-INF]$ vi web.xml
...
<init-param>
  <param-name>container_ds</param-name>
  <param-value>jdbc/router_na</param-value></init-param>
</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://spatiallocation:7001/geocoder/gcserver</param-value>
</init-param>
...
[oracle@spatiallocation WEB-INF]$
```

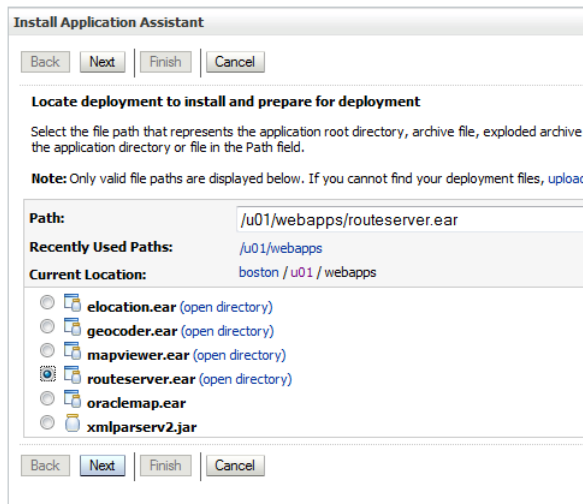
Note: When you specify the routeserver_network_name, you can get name from executing the query of "select network from user_sdo_network_metadata". .

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”.

Source accessibility

How should the source files be made accessible?

☐ Use the defaults defined by the deployment's targets

Recommended selection.

☐ Copy this application onto every target for me

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

☒ I will make the deployment accessible from the following location

Location:

Provide the location from where all targets will access this application's files. This is often a shared directory. You must provide the location.

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

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Install (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

<input type="checkbox"/>	Name	State	Health
<input type="checkbox"/>	geocoder	Active	OK
<input type="checkbox"/>	mapviewer	Active	OK
<input type="checkbox"/>	routeserver	Active	OK

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

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

ORACLE 12c
DATABASE

Location Services

Oracle Spatial and Graph RouteServer

[RouteServer Demo](#)

Simple Route Requests

[Route Requests using un-geocoded addresses](#)

[Route Requests using pre-geocoded addresses](#)

[Route Requests using Latitude and Longitude](#)

Batch Route Requests

[Batch Route Requests using un-geocoded addresses](#)

[Batch Route Requests using pre-geocoded addresses](#)

[Batch Route Requests using Latitude and Longitude](#)

Administration

[Administration](#)

[Metadata](#)

Copyright 2006-2013 Oracle Corporation. All rights reserved.

Deploying Oracle eLocation

This section explains how to configure and deploy eLocation Services to Oracle Weblogic Server. You can download and copy to proper directory in your system. You need to unpack the elocation.ear to a directory 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, eLocation needs xmlparserv2.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@spatiallocation webapps]$ ls
elocation.ear geocoder.ear mapviewer.ear oraclemap.ear routeserver.ear xmlparserv2.jar
[oracle@spatiallocation webapps]$ mv elocation.ear elocation1.ear
[oracle@spatiallocation webapps]$ mkdir elocation.ear
[oracle@spatiallocation webapps]$ mv elocation1.ear elocation.ear
[oracle@spatiallocation webapps]$ cd elocation.ear
[oracle@spatiallocation elocation.ear]$ unzip elocation1.ear
[oracle@spatiallocation elocation.ear]$ mv web.war web1.war
[oracle@spatiallocation elocation.ear]$ mkdir web.war
[oracle@spatiallocation elocation.ear]$ mv web1.war web.war
[oracle@spatiallocation elocation.ear]$ cd web.war
[oracle@spatiallocation web.war]$ unzip web1.war
[oracle@spatiallocation web.war]$ cp /u01/webapps/xmlparserv2.jar WEB-INF/lib
```

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

- Go to WEB-INF
- Backup dispatcher.cfg.xml to dispatcher.cfg.xml.old
- Modify dispatcher.cfg.xml

The example of modifying dispatcher.cfg.xml is as following.

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

```
[oracle@spatiallocation 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

Install Application Assistant

Back Next Finish Cancel

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive or the application directory or file in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, [upload](#)

Path: /u01/webapps/elocation.ear

Recently Used Paths: /u01/webapps

Current Location: boston / u01 / webapps

- ☒ elocation.ear (open directory)
- ☐ geocoder.ear (open directory)
- ☐ mapviewer.ear (open directory)
- ☐ routeserver.ear (open directory)
- ☐ oraclemap.ear
- ☐ xmlparserv2.jar

Back Next Finish Cancel

- 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".

Source accessibility

How should the source files be made accessible?

☒ Use the defaults defined by the deployment's targets

Recommended selection.

☐ Copy this application onto every target for me

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

☐ I will make the deployment accessible from the following location

Location: /u01/webapps/elocation.ear

Provide the location from where all targets will access this application's files. This is often a shared directory. You must provide the location.

Back Next Finish Cancel

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

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Install (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install | Update | Delete | Start ▼ | Stop ▼

Name	State	Health
eLocation	Active	✓ OK
geocoder	Active	✓ OK
mapviewer	Active	✓ OK
routeserver	Active	✓ OK

Install | Update | Delete | Start ▼ | Stop ▼

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

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

ORACLE

Location Services

eLocation Demo | XML interface | JS API | Admin

Welcome to Oracle Spatial and Graph eLocation

Oracle Spatial eLocation is a set of web based location aware services (Location-based Services). It enables applications to use location as a new dimension in managing their business processes thereby enriching end user experience.

Oracle Spatial eLocation consists of a set of J2EE applications that take advantage of the location capability provided by Oracle 12c Locator and Spatial. All data is stored inside an Oracle Spatial database.

Oracle Spatial eLocation provides three basic types of services:

1. **Geocoding service** takes user input address (or Point Of Interest name -POI name-) and converts it to a latitude/longitude location.
2. **Mapping service** renders map image with user specified map parameters.
3. **Routing (Driving Directions) service** takes two street addresses (or multiple addresses in the case of batch routing) and return the driving directions based on street network analysis.

Cannot log into the administration page?

Here is a hint:
You must have created a security user in the instance where eLocation is running, and mapped the security user to the eLocation's built-in security role "lbs_admin_role". Once you have completed these tasks through the Enterprise Manager website, you can then use that security user's name and password to log in as eLocation administrator.

Copyright © 2013, Oracle Corp. All Rights Reserved

[Legal Notices](#) | [Terms and Conditions](#) | [Feedback](#)

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

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

Mapping

Geocoding

Routing

Proximity Search

Oracle Spatial Technology Network

Search GO

Find Map

Directions

A

Address, City, State or Postal Code...

Search

500 mi

1,000 km

©2011 Oracle Corp. ©2011 NAVTEQ™

PART II: The eLocation XML API

This part describes how to submit a request in XML format to the 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 and Graph web services have XML APIs to provide data and metadata of their services through HTTP protocol.

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>
```

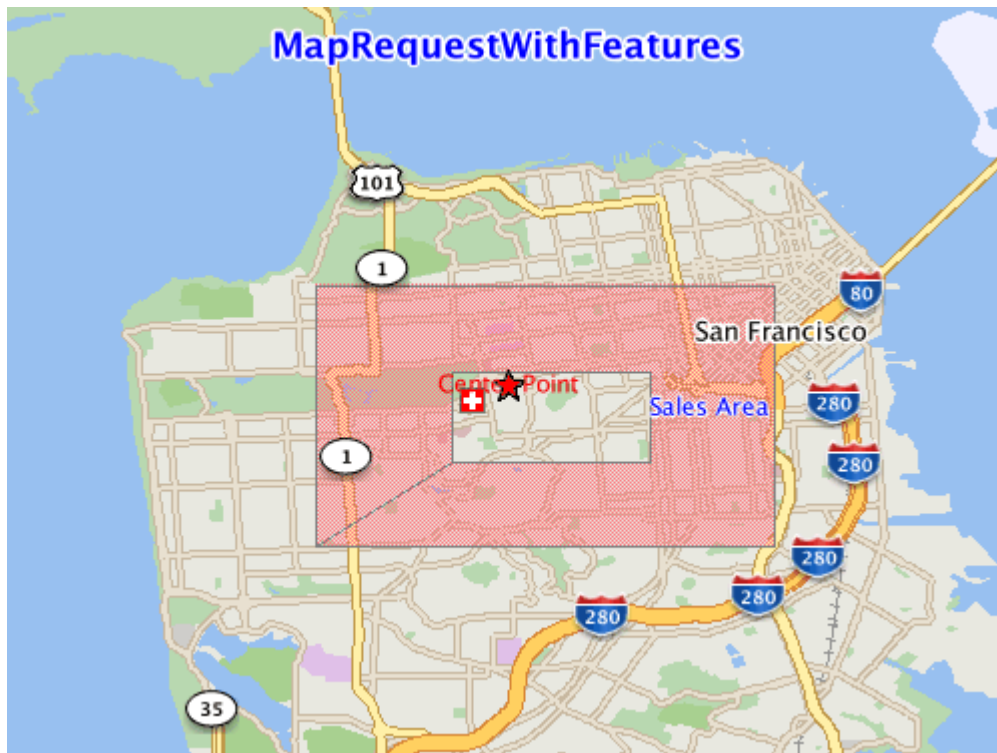
The example of response mapping with center is as following.

```

<?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="3.3333333333333428E-4 0.0 0.0 -3.3333333333333428E-4 -122.53333333333333
37.833100000000001" />
    <WMTEException version="1.0.0" error_code="SUCCESS">
    </WMTEException>
  </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>

```

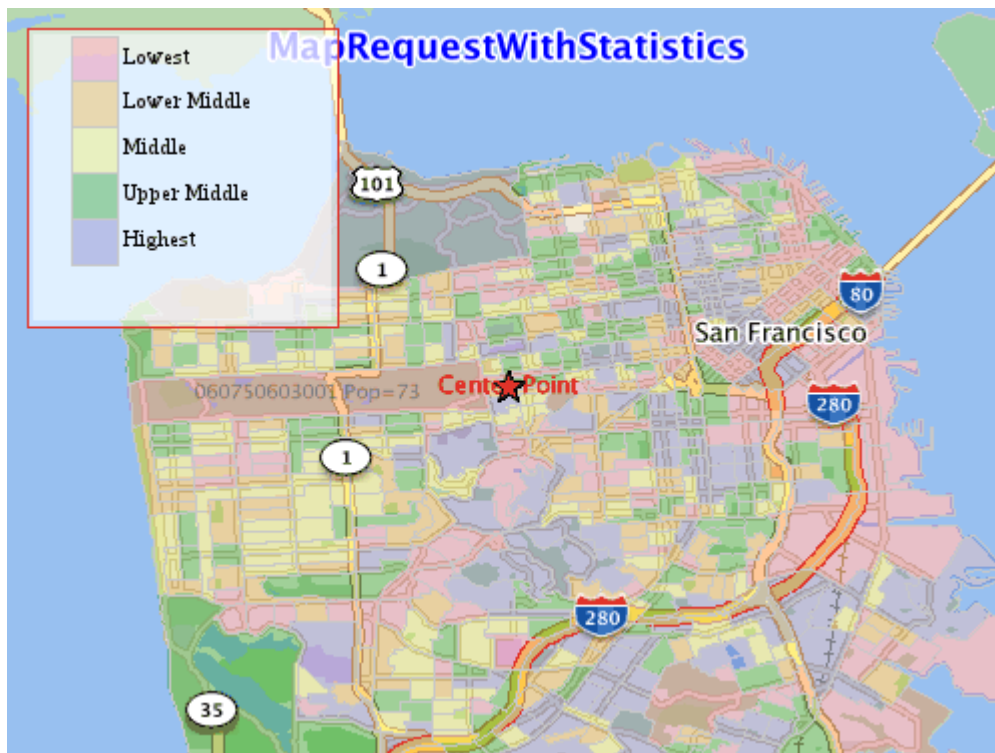
The example of response mapping with center is as following.

```

<?xml version="1.0" encoding="UTF-8"?>
<map_response>
<map_image>
<map_content
url="http://spatialelocation:7001/mapviewer/images/omsmap9_25.gif?refresh=402116767264
8852594" />
<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="3.33333333333333428E-4 0.0 0.0 -3.33333333333333428E-4 -122.53333333333333
37.83310000000001" />
<WMTEException version="1.0.0" error_code="SUCCESS">
</WMTEException>
</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 and Graph Developer's Guide 12c Release 1 (12.1)** E17896-09

http://docs.oracle.com/cd/E16655_01/appdev.121/e17896/toc.htm

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="???0414114??004?">
      <output_address name="SAN FRANCISCO STATE UNIVERSITY" house_number="1600"
      street="HOLLOWAY AVE" buildup_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 and Graph Developer's Guide 12c Release 1 (12.1)** E17896-10
- http://docs.oracle.com/cd/E16655_01/appdev.121/e17896/toc.htm
- 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="2.886625027656553"/>
<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="0.041840906937917074"/>
  <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**

Copyright © 2011, Oracle Corp. All Rights Reserved Legal Notices | Terms and Conditions | NAVTEQ | Feedback