



ORACLE

Geocoder – A Spatial Feature of Oracle Database

Technical Brief

May, 2023, Version 1.0
Copyright © 2023, Oracle and/or its affiliates

Public

Purpose statement

This document provides an overview of features and enhancements included in the spatial geocoding feature of Oracle Database. It is intended solely to help you assess the business benefits of using these features and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Introduction	4
How Geocoding is Used	5
Database Geocoding Features	5
Data	5
Types of Geocoding	5
International Address Format Support	6
Levels of Address Matching	6
How To Use	7
Database on-premises or Base Database Cloud Service	7
Oracle Autonomous Database	9
Benefits	12
Conclusion	13

Introduction

Geocoding is the process of converting an address, such as street, city, and state, to its geographic coordinates on the earth’s surface, usually expressed as longitude and latitude. The resulting location can then be displayed on a map and used for data analysis. For example, “1 Oracle Drive, Nashua, NH, 03062” is represented as (longitude=-71.45937, latitude=42.70783).

Geocoding is important because it can be used by every business to prepare their data for location-based analysis. The geocoded locations of customers can be viewed in a table or on a map and used in many different applications.

For example, service industries can make sure their delivery vehicles are routed to the precise customer address. Marketing campaigns can target customers within a specific geographic area or zip code. Sales managers are able to identify all customers within a particular sales territory.

Reverse geocoding is the opposite - it finds a street address from a set of geographic coordinates. For example, a driver has an accident and uses his cell phone to call 911. The driver’s location is identified by his cell phone global positioning system so the emergency vehicle knows where to go.

The geocoder is a spatial feature of Oracle Database that lets you quickly geocode any business data with associated addresses and perform sophisticated spatial analysis with spatial SQL queries and functions.

This technical brief highlights the key features of the Oracle geocoder and how it is used. It serves as an introductory guide for partners or customers who want to geocode their data using this technology.

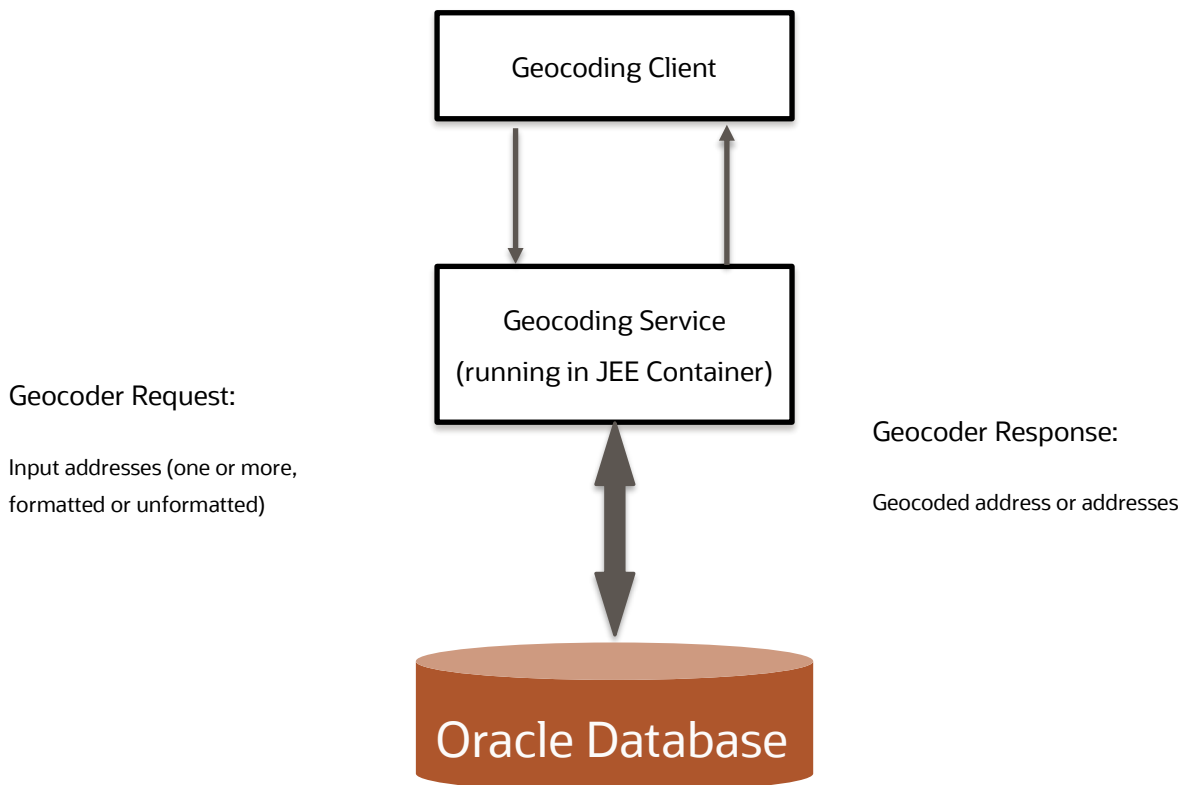


Figure 1. Spatial Geocoding Service Workflow

How Geocoding is Used

Geocoding is used in a wide range of applications through a variety of interfaces. Web portals and mapping sites use geocoding to convert addresses into coordinates that are used to display locations on maps or locate nearby points of interest.

The Oracle geocoder may also be used for *table geocoding* to convert entire tables of addresses in bulk and add a longitude/latitude column to the data set. This is commonly used by applications that depend on the addresses of clients, customers and suppliers, for example, Human Resources, Purchasing, Sales and Marketing, and Customer Relationship Management.

A unique benefit of the Oracle geocoder is that it can also be invoked dynamically through database triggers and included in database stored procedures. This allows for address data to be automatically and transparently converted into and stored as longitude/latitude when entered via a database trigger. This is particularly beneficial to Call Center and other real-time and near real-time service-based applications.

Database Geocoding Features

Oracle Spatial offers complete geocoding. It provides interpolation-based (or standard) geocoding, point-based geocoding (or point address geocoding), reverse geocoding, batch geocoding, international address standardization, and other geocoding functions. Its unique unparsed address support adds great flexibility and convenience to customer applications. SQL and XML APIs for geocoding are provided. The geocoder comes with the database and it can also be deployed in a supported JEE container. The XML API shown in Figure 1 provided in the middle-tier can be accessed via a web service.

Data

The Oracle geocoder is dependent on external reference data. The world's leading data providers and many regional vendors offer this data for Oracle's geocoder. HERE Map content is delivered in Oracle Delivery Format (ODF_GEOMETRY) and TomTom's data is provided in an Oracle format. Both data formats are out-of-the box ready for geocoding. The reference data is necessary for Oracle's geocoder to determine the location for a specified address.

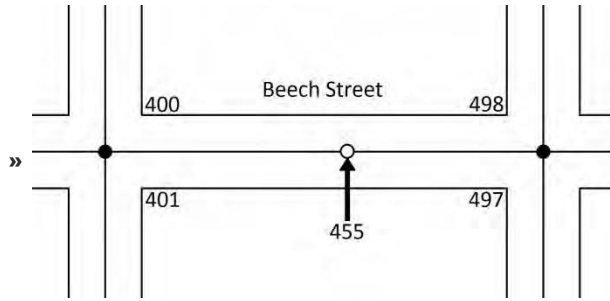
Oracle also publishes a geocoding schema so any partner or customer can take other map data (i.e. OpenStreetMap or local map data) and populate it. Once the data is stored in the database, the geocoder is used to parse addresses and search reference data so that coordinates and other information are returned to the user.

External reference data is also available from other data providers. For more information on data providers, please visit <http://www.oracle.com/technetwork/database-options/spatialandgraph/downloads/spatial-partners-data-087203.html>.

Types of Geocoding

Oracle supports three types of geocoding: interpolation-based geocoding, point-based geocoding, and reverse geocoding.

- For interpolation-based geocoding, streets have address ranges. The location for a given address is approximated based on the start and end values of the range.
- For example, the input address is 455 Beech Street. A block of Beech Street runs from 400 to 499. Odd-numbered addresses fall on the east side of Beech Street and even-numbered addresses on the west side of the street. 455 Beech Street would probably be located about halfway up the block on the east side of the street. A point would be mapped at that location along the street.



- Point-based geocoding is exact as data sets include the exact location (longitude and latitude) of addresses, intersections, and points of interest. Point-based geocoding returns more accurate results and can be used in situations where interpolation is not possible. It is more suitable for countries like India and Japan where address ranges are not used for street addresses.
- Reverse geocoding converts a point location (longitude and latitude) to the corresponding street address and is useful for many applications. If a GPS device is used to track delivery trucks, for example, reverse geocoding lets the user know the approximate street addresses of the trucks.

All of these geocoding methods can be performed at single record level or in batch (either serially or in parallel). Parallel batch geocoding and parallel reverse geocoding are much faster and more efficient than multiple single requests since they have better utilization of database resources.

International Address Format Support

Oracle geocoder supports international postal address parsing and geocoding. By parsing the postal address, the geocoder is able to extract address information, such as street name, house number, name of city and postal code from postal addresses in different international address formats. The geocoder uses the parsed street address information to search the database and resolve the input address into a longitude/latitude location.

Levels of Address Matching

Geocoded data may have various levels of precision – from complete address matching to matching part of the address. Some applications, such as property insurance, may require an exact match between the address and the data used for geocoding. Other applications, such as mapping voters in a particular voting district, may not have to be as precise so that geocoding can be performed despite errors or missing information in the input addresses.

The geocoder has fuzzy address matching support to cleanse incorrect addresses for more correct output. It may correct commonly misspelled street names or use “road” versus “street” or “drive”. The result is a clean address with correct formatting and spelling, and any missing elements, such as the zip code.

The MATCH_MODE parameter lets the user decide how closely the input address must match the stored address. There are 9 MATCH-MODE parameters to choose from, including the default. Some of these include:

- EXACT – best used with addresses that are known to be accurate
- RELAX_STREET_TYPE – if the input address of “Washington Street” doesn’t exist, but “Washington Road” does, then “Washington Road” will be returned
- RELAX_HOUSE_NUMBER – if the input address is “13 Washington Street”, then “13 Washington Road” or “13 Washington Place” would also match
- RELAX_BUILTUP_AREA – searches for the address outside the city specified, but within the same county

The returned results give a status – or a MATCHCODE attribute – that indicates the general quality of the match. The MATCHVECTOR attribute further details the quality and precision of the match by telling the user how each individual address element matched. Used together, these attributes let your application decide whether to accept the results of a geocode operation or reject the results and flag the record for later resolution.

How To Use

The geocoder is a spatial feature of Oracle Database and can be accessed via a PL/SQL interface or it can be deployed in Oracle WebLogic Server and accessed as a web service via XML requests. Before either of these methods can be used, the reference data must be loaded.

Oracle Autonomous Database users, however, do not need to obtain and load the reference data. They can use additional functions included in the SDO_GCDR package only in Autonomous Database instances. The details of these functions and their usage are described later in this document.

Database on-premises or Base Database Cloud Service

When using the geocoder via a PL/SQL interface in the database, the following steps occur:

- Addresses are entered into the subprograms using specific geocoding datatypes.
- Calls are sent to the subprograms of the SDO_GCDR PL/SQL package.
- Geocoded results are returned in the same datatypes.

Example: Geocoding PL/SQL Request

```
SQL> SELECT SDO_GCDR.GEOCODE(USER, SDO_KEYWORDARRAY('10 Clay Street', 'San Francisco, CA'),
'US' , 'DEFAULT') FROM DUAL;
```

Geocoding Results

```
SDO_GEO_ADDR(0, SDO_KEYWORDARRAY(), NULL, 'CLAY ST', NULL, NULL, 'SAN FRANCISCO', 'SAN
FRANCISCO', ' CA', 'US', '94111', NULL, '94111', NULL, '10', 'CLAY', 'ST', 'F', 'F', NULL,
NULL, 'L', .826086957, 796013931, '????#ENUT?B281CP?', 1, 'DEFAULT', -122.39587,
37.7955978, '???10101010??004?', 8307)
```

Web Service

A web service can be set up to perform geocoding calls. Geocoding requests, expressed in XML, are sent to the web service, and the results are also in XML.

An XML geocoding request containing an input address is sent to the geocoder via HTTP. The geocoder parses the input request and invokes the geocoder to look up the input address in the database. The geocoder then sends the geocoded result in XML or JSON formats to the client via HTTP. The JSON format is compliant to OpenAPI Specification Version 3.

In order to start using geocoder as a web service, follow these steps:

- Unzip the geocoder.ear.zip file found in your \$ORACLE_HOME/md/jlib directory into a suitable directory. Your resulting directory structure should be: \$geocoder.ear/web.war/... .
- Log on to the WebLogic Server console (for example, <http://<hostname>:7001/console>); and from Deployments install the geocoder.ear file, accept the Name geocoder for the deployment and choose the Location option. Make the deployment accessible from the location in the next step.
- Launch the geocoder welcome page in a web browser using the URL <http://<hostname>:<port>/geocoder>. On the welcome page, select the Administration link and enter the administrator (weblogic) user name and password.
- Modify the geocoder configuration file (geocodercfg.xml). Uncomment at least one <geocoder> element, and change the <database> element attributes of that <geocoder> element to reflect the configuration of your database.
- Save the changes to the file and restart the geocoder.

- Test the database connection by going to the welcome page at URL `http://<hostname>:<port>/geocoder` and running the XML geocoding request page. (This demo requires geocoder data for the United States.) Examples are available to demonstrate various capabilities of the geocoding service. Reviewing the examples at URL `http://<hostname>:<port>/geocoder/gcxmlreq_exp_af.html` is a good way to learn the XML API, which is described in Geocoding Request XML Schema Definition and Example Section of Oracle Spatial Developer's Guide.

Example: Geocoding XML Request

```
<?xml version="1.0" standalone="yes"?>
<geocode_request vendor="elocation">
<address_list>
<input_location id="27010">
<input_address match_mode="relax_street_type">
<us_form2 street="one oracle drive" city="nashua" state="nh"/>
</input_address>
</input_location>
</address_list>
</geocode_request>
```

Geocoding Results (in XML format)

```
<geocode_response>
<geocode id="27010" match_count="1">
<match sequence="0" longitude="-71.45937" latitude="42.70783" match_code="1"
error_message="???#ENUT?B281CP?" match_vector="???10101010??004?" srid="8307">
<output_address name="" house_number="1" street="Oracle Dr" settlement="Nashua"
builtup_area="Nashua" municipality="" order1_area="NH" order8_area="" country="US"
postal_code="03062" postal_addon_code="" side="L" percent="0.0" edge_id="22325991"/>
</match>
</geocode>
</geocode_response>
```

Example: Geocoding XML Request for JSON format

Note that format parameter is used and set to "JSON" for this purpose.


```
http://hostname:port/geocoder/gcserver?format=JSON&xml_request=<geocode_request
vendor="elocation">
```

```
<address_list>
  <input_location id="27010"
    <input_address match_mode="DEFAULT">
      <us_form2 street="500 oracle pky" city="redwood city" state="ca" />
    </input_address>
  </input_location>
</address_list>
</geocode_request>
```

Reverse Geocoding Results

```
<geocode_response>
<geocode id="27010" match_count="1">
<match sequence="0" longitude="-71.45937" latitude="42.70783" match_code="1"
error_message="" match_vector="???14141414??404?" srid="8307">
<output_address name="" house_number="1" street="Oracle Dr" settlement="Nashua"
builtup_area="Nashua" municipality="Hillsborough" order1_area="New Hampshire"
order8_area="" country="US" postal_code="03062" postal_addon_code="" side="L" percent="0.0"
edge_id="22325991"/>
</match>
</geocode>
</geocode_response>
```

Oracle Autonomous Database

The Oracle Autonomous Database includes a special version of the SDO_GCDR package that eliminates the need to obtain and load reference data required for address geocoding. These functions are called SDO_GCDR.ELOC_GEOCODE and SDO_GCDR.ELOC_GEOCODE_AS_GEOM and can be used for geocoding address data and for reverse geocoding longitude/latitude data to a street address.

The first step, however, is granting access to these functions to the user. This must be done as the ADMIN user of the Autonomous Database.

Replace <db_user_name> in the function call below. Then execute it as ADMIN to **grant access**.

```
exec sdo_gcdr.eloc_grant_access('<db_user_name>');
```

There is a corresponding function to revoke access.

```
exec sdo_gcdr.eloc_revoke_access('<db_user_name>');
```

Once the user has access they can use the SDO_GCDR.ELOC_GEOCODE function to geocode a formatted or unformatted street or reverse geocode Latitude/Longitude coordinates for a location. The function returns a JSON object.

Let's look at some examples.

Geocoding a formatted address

Use the following version of the function when the address elements such as street, city, or postal code are known.

```
ELOC_GEOCODE( street VARCHAR2, city VARCHAR2, region VARCHAR2, postal_code VARCHAR2, cc2
VARCHAR2, match_mode IN VARCHAR2 DEFAULT 'RELAX_POSTAL_CODE')
```

For example,

ORACLE

```
select sdo_gcdr.eloc_geocode('100 N Renfrew St', 'Vancouver', 'BC', 'V5K 4W3', 'CA',  
'RELAX_POSTAL_CODE') from dual;
```

which gives the result:

```
[{  
  "id": "0",  
  "matchCount": "1",  
  "matches": [  
    {  
      "sequence": "0",  
      "x": -123.04406,  
      "y": 49.28577,  
      "houseNumber": "100",  
      "street": "N Renfrew St",  
      "settlement": "Vancouver",  
      "municipality": "Vancouver",  
      "region": "BC",  
      "postalCode": "V5K",  
      "country": "CA",  
      "language": "ENG",  
      "name": "",  
      "edgeId": 811555206,  
      "percent": 0.13,  
      "side": "R",  
      "matchCode": 1,  
      "matchVector": "??010001010??000?"  
    }  
  ]  
}]
```

Geocoding an unformatted address

An unformatted address has all the address elements in one line with commas separating them. E.g.

```
select sdo_gcdr.eloc_geocode('100 N Renfrew St, Vancouver BC V5K 4W3, CA') from dual;
```

will give the same result as above.

Geocoding a place name

Use the unformatted address form to geocode a named location such as well known building.

```
select sdo_gcdr.eloc_geocode('CN Tower, Toronto CA') from dual;
```

which gives the result:

```
[{
  "id": "0",
  "matchCount": "1",
  "matches":
    [{
      "sequence": "0",
      "x": -79.38651,
      "y": 43.6418,
      "houseNumber": "301",
      "street": "BREMNER BLVD",
      "settlement": "TORONTO",
      "municipality": "TORONTO",
      "region": "ON",
      "postalCode": "M5V",
      "country": "CA",
      "language": "ENG",
      "name": "CN TOWER",
      "edgeId": 1242626341,
      "percent": 0.0,
      "side": "L",
      "matchCode": 1,
      "matchVector": "??04101010??000?"
    }
  ]
}]
```

Reverse geocoding a location

Pass in the Longitude and Latitude in that order. E.g. executing

```
select sdo_gcdr.eloc_geocode(-79.38651, 43.6418) from dual;
```

will give the result:

```
[{
  "id": "0",
  "matchCount": "1",
  "matches":
    [{
      "sequence": "0",
      "x": -79.38650275729277,
      "y": 43.641786173013465,
      "houseNumber": "294",
      "street": "BREMNER BLVD",
      "settlement": "TORONTO",
      "municipality": "TORONTO",
      "region": "ON",
      "postalCode": "M5V",
      "country": "CA",
      "language": "ENG",
      "name": "",
      "edgeId": 1242626341,
      "percent": 0.510676156507514,
      "side": "L",
      "matchCode": 1,
      "matchVector": "14141414??404?"
    }
  ]
}]
```

Further details and examples can be found in the reference section for the SDO_GCDR package in the Oracle Spatial Developer's Guide.

Benefits

The Oracle geocoder provides many benefits to users.

- Choice of implementation - in database or middle tier (integration with Oracle Fusion Middleware)

ORACLE

- Highly scalable and fast performance – in-database parallel batch geocoding and reverse geocoding on massive data sets
- For an Oracle database cluster with 24 cores, batch geocodes were 1365 per second and batch reverse geocodes were 3388 per second
- For a 16 core stand-alone Linux box, batch geocodes were 1164 per second
- Scalability and fast performance can be achieved in a middle tier deployment using multiple geocoder instances and Oracle WebLogic Server's own load balancing capabilities
- For a single processor with 4 cores, geocodes were 834 geocodes per second
- Open schema
- Choice of purchasing pre-populated reference data from data providers or populating schema yourself
- Customizable parser
- Can easily add misspelled names and other modifications
- Easy to configure - standard SQL interface with consistent result format
- Fuzzy logic for address input (prefix matching for street names and SOUNDEX-based matching)
- International support – large number of countries

Conclusion

Oracle provides robust geocoding capabilities with Oracle Database. Business data can be quickly geocoded with associated addresses so that sophisticated spatial analysis can be performed on customer data.

The geocoder can be invoked in the database or middle tier and supports interpolation-based geocoding, point-based geocoding, reverse geocoding, and fuzzy address matching.



Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 0000, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.