

# Oracle Forms 10g- Using Static Google Maps with HTTP geocoding service

Hafed Benteftifa- December 2008  
www.degenio.com

## Introduction

In a number of back-end applications, users as well as entities are registered with their address information. In some cases, it is desired to provide a visual representation of this address information.

A free service that is provided by Google allows us to obtain the map for a given address as well as additional info such as directions from a given point to this address.

This is a bit difficult to implement in Oracle Forms 10g since the Google API is based on Javascript. In those cases where Javascript is blocked or not possible, Google provides another service which is called Google Static maps.

What this service provides is a static image of the map based on the address information. This can be sufficient in a number of Forms applications.

In this note, we provide a javabean that will display the static map in addition to the latitude and longitude corresponding to the address provided by the user (or fetched from a table).

First, we will use the Google Maps HTTP geocoding service to retrieve the latitude and longitude. Then, we use this data to fetch the static map from the Google static map service.

## Geocoding

In most cases, users won't provide locations as a latitude and longitude pair. Instead, the location will be most likely provided as an address entered in a free format text.

In geographical terms, location on earth can be described by a coordinate: the **latitude** and **longitude**.

**Geocoding** is the process of assigning geographical information like latitude and longitude to street addresses. There are a number of services that can provide this geocoding service.

The service that will be used in this javabean is provided by Google.

## Google Maps HTTP geocoding service

The Google Maps API Geocoder is a basic REST query, so all the parameters that should be passed to the web service are included within the query string of the request.

The base request URL is <http://maps.google.com/maps/geo?>. The following parameters are required and have to be included in the string:

- **q:** The address that is to be geocoded. This is a string with all the information related to the location (street, city, state, country).
- **Key:** Your API key. Please note that this key has to be provided in your Forms module. The key is tied to your domain. For more information, visit <http://code.google.com/apis/maps/signup.html> to sign up for a free Google Maps API key.
- **Output:** The format in which the output should be generated. The options are xml, kml, csv, or (default) json.

In our case, we will be using the csv output since we are only interested in the latitude and longitude data. Note that the XML output provides a whole lot more information and can easily be integrated with this javabeen.

Detailed info about this service can be found here

<http://code.google.com/apis/maps/documentation/services.html#Geocoding>

The following link provides more details about the data used in the Google map javabeen.

[http://code.google.com/apis/maps/documentation/services.html#Geocoding\\_Direct](http://code.google.com/apis/maps/documentation/services.html#Geocoding_Direct)

## CSV Result returned by Google geocoding

The data that is used in the javabeen geocoding section is based on the **csv** data provided by the Google geocoding service. The result will consist of four numbers, separated by commas:

1. HTTP status code (integer, enumerated type)
2. Accuracy (integer, enumerated type)
3. Latitude (decimal)
4. Longitude (decimal)

## Usage

Data submitted From Oracle Forms should be in the following format:

Address\_to\_map\_&output=csv&gZoom|gMaptype|gFrame|gSensor|gKey

Where:

gZoom: should be between **0** and **19**. Default is **10**.

gMaptype: could be **roadmap**, **mobile**, **satellite**, **terrain**, **hybrid**

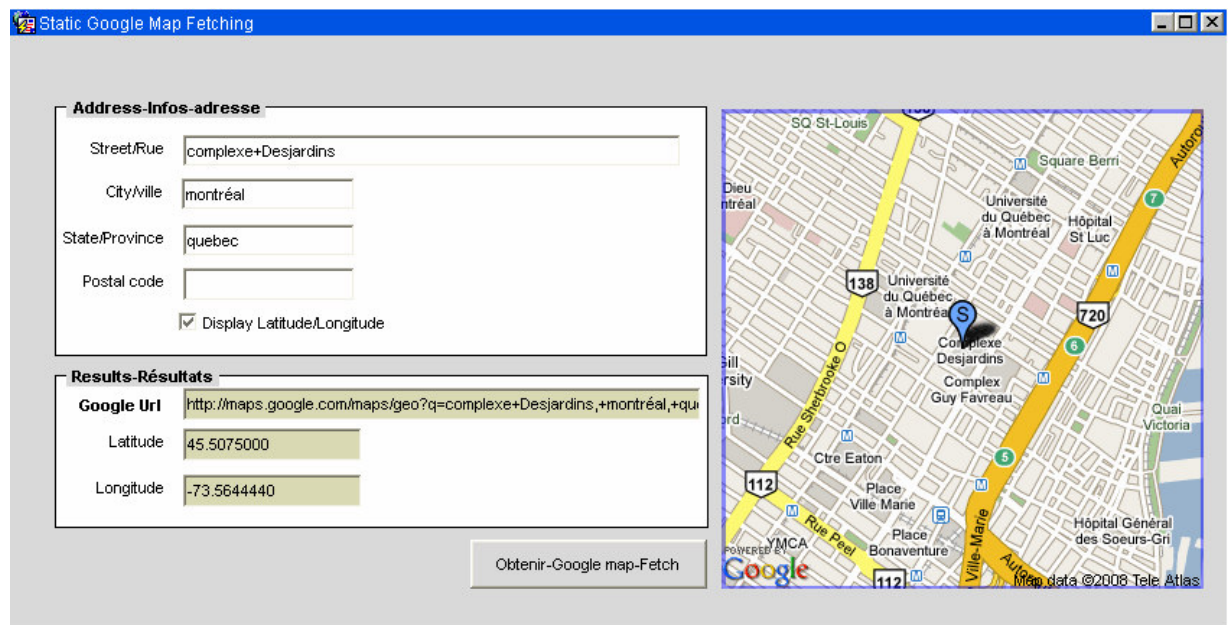
gFrame: **true** or **false**. Used to have a surrounding frame with the image

gSensor: **true** or **false** depending whether you are acquiring the data from a sensor.

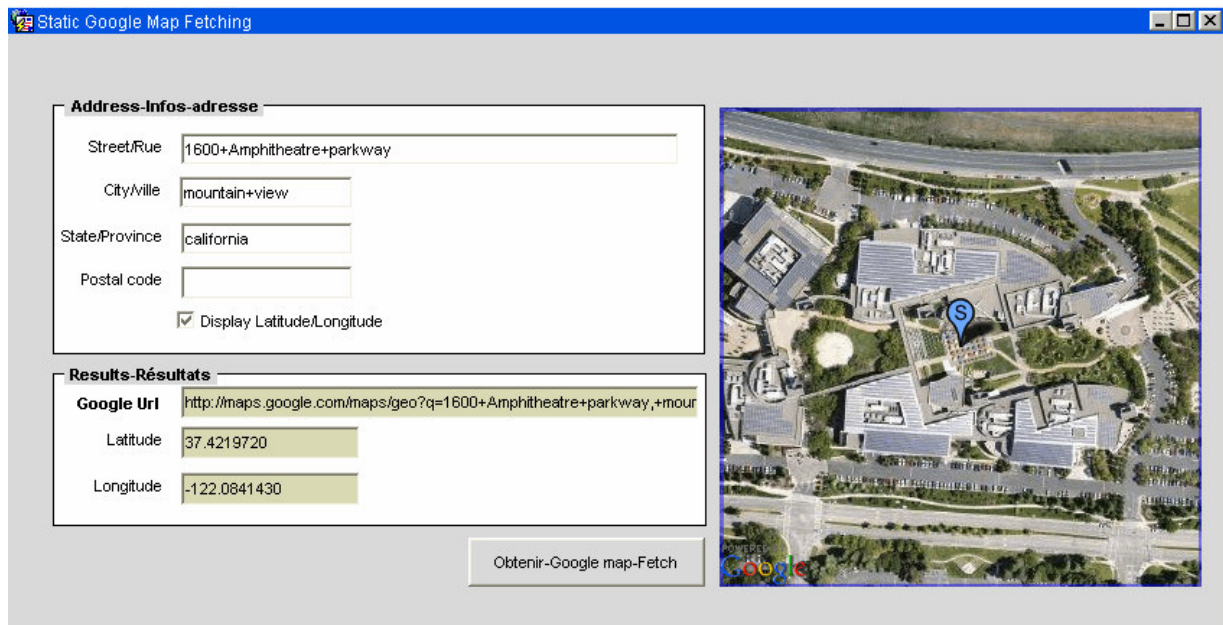
Note: The field separator used in the Javabeen is the character: |

Example:

Address\_to\_map&output=csv&14|roadmap|true|false| ABQIAAAAE3WfGRybLGo3



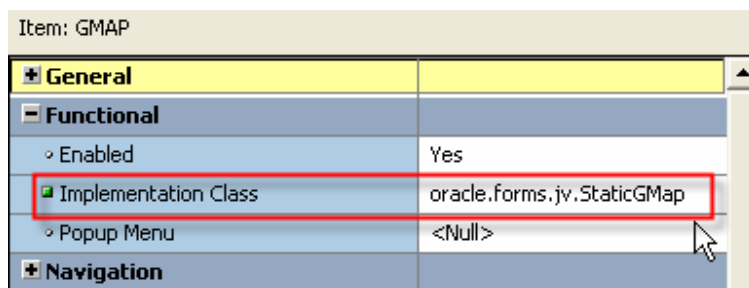
Address\_to\_map&output=csv&17|satellite|true|false| ABQIAAAAE3WfGRybLGo3



## Using the Google Map Bean in your module

The implementation class to use for the Bean Item should be:

oracle.forms.jv. StaticGMap



- The property to set in order to get the Google Map: **SETURL**

set\_custom\_property( 'BL1.GMAP', 1, 'SETURL', :BL1.TX1 );

where :BL1.TX1 will be in the format:

Address\_to\_map\_&output=csv&gZoom|gMaptype|gFrame|gSensor|gKey;

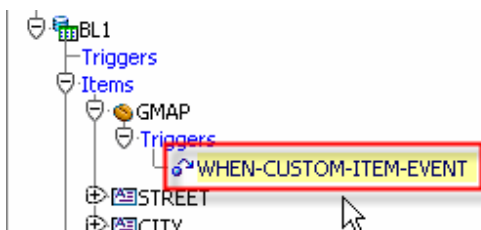
- Getting the Latitude : use property **LATITUDE**

```
get_custom_property('BL1.GMAP',1,'LATITUDE')
```

- Getting the Longitude : use property **LONGITUDE**

```
get_custom_property('BL1.GMAP',1,'LONGITUDE');
```

- Events raised by this bean: **FINISHED**, **INSTANCE** and **ABORTED**



You can use the trigger WHEN-CUSTOM-ITEM-EVENT on the bean to add additional support.

In the accompanying FMB module, I used the events to show how you can alert the users for the various state returned by the bean.

## Installing the JAR

- Download the gmap.zip file
- Unzip the files
- Copy the gmap.jar file in the <ORACLE\_HOME>/forms/java directory
- Edit your /forms/server/formsweb.cfg file to add gmap.jar
- Open the gmap.fmb module (Oracle Forms 9i or 10g)
- Compile all and run the module

The jar file gmap.jar must be signed. The one included in the zip file was already signed. However, I suggest recompiling the StaticGMap.java file and resigning with your own certificates.

## Comments and suggestions

please note that this Javabean is provided without any warranty.

There are a number of improvements (XML format for example) and checks that can be added. Please direct any comments or suggestions to Hafed Benteftifa at [info@degenio.com](mailto:info@degenio.com). Thanks.