Integrating Oracle Application Express with Google Maps

Introduction ....................................................................................................... 3
Mapping Application ........................................................................................ 4
  Create an Application ................................................................................. 4
  Create an Application Substitution for Google Maps API Key .......... 4
Create Address Form....................................................................................... 4
  Create an HTML Region For Address Form Items ............................... 4
  Create Form Items to Capture Address .................................................. 5
Create Button to Submit Page ..................................................................... 6
Create Response Region ............................................................................... 6
  Create HTML Region ................................................................................ 7
  Create Response Items ............................................................................ 7
Call Google Geocode Service......................................................................... 7
  Create PL/SQL Page Process to Call Geocode Service ....................... 7
  Test the Application ................................................................................ 8
Google Map Page............................................................................................ 9
  Create a Pop-up Page to Embed the Google Map ............................... 9
  Create an HTML Region on the Popup Page for the Embedded Map 10
  Call Google Map API With JavaScript .................................................. 10
Invoke the Map Pop-up Page........................................................................ 11
  Create a Button to Call the Pop-up Map Page ................................. 11
  Test the Application .............................................................................. 12
Conclusion....................................................................................................... 13
INTRODUCTION

Oracle Application Express – a feature of the Oracle Database – is a powerful and easy to use web application development platform. With Oracle Application Express, you can quickly develop and deploy applications in a matter of hours, often without writing a single line of code. Application Express can be easily integrated with the Google Maps API. The Google Maps API allows you to create online mapping applications and incorporate geo-coding into them.

Embedding Google Maps into your Application Express applications is accomplished using JavaScript to invoke the Google Maps API. You must sign up for a Google Maps API key in order to use the API in your applications. Signing up for a Google Maps API key is free and easy at the following URL:

http://code.google.com/apis/maps/signup.html
MAPPING APPLICATION

Once you have signed up for and received your Google Maps API key from the URL provided in the introduction, you are ready to create an Application Express application and create an application substitution for that key.

Create an Application

To create an Application Express application:

1. Login to your Application Express instance.
   Please note that this tutorial cannot be demonstrated on apex.oracle.com because it does not support external network call-outs. You will need to use your own instance to be able to execute the resulting application.

2. Select Application Builder

3. Click Create >

4. Supply information as required by the Create Application wizard

5. Add one blank page on the Pages step of the wizard

6. Once you have created the application, if your environment requires a proxy server to reach pages on the Internet, supply the proxy server in the Proxy Server field of the Application Definition (Shared Components > Definition - under Application)

Create an Application Substitution for Google Maps API Key

Now that you have your application, you are ready to create an application substitution for your Google Maps API key. To create an application substitution:

1. From within your application, navigate to Shared Components

2. Select Definition (under Application)

3. Click Substitutions quick link >

4. Enter API_KEY in the first Substitution String field

5. Enter your Google Maps API key in the first Substitution Value field.
   Note that an API key is only good for a single directory on a Web server. For an Application Express instance, you should supply the URL including the apex DAD reference, for example http://myserver.com:7777/pls/apex.

6. Click Apply Changes

CREATE ADDRESS FORM

First, create a form to capture the address to map.

Create an HTML Region For Address Form Items

To create an HTML region:
1. Navigate to the Page Definition for page one of the application
2. Click the create icon in the Regions section
3. Select HTML for the region type and click Next >
4. Select HTML from the region container list and click Next >
5. Enter Address to Map in the Title field
6. Choose Form Region from the Region Template list and click Create

**Create Form Items to Capture Address**

Next you create the form items to capture the address data. Use the create multiple items using tabular form option to quickly create multiple items.

To create multiple items using a tabular form:

1. Click the create icon in the Items section
2. Click Create multiple items using tabular form at the bottom of the page
3. Enter P1_STREET in the first Name field
4. Enter Street in the first Label field
5. Enter P1_CITY in the second Name field
6. Enter City in the second label field
7. Enter P1_STATE in the third Name field
8. Enter State in the third Label field
9. Enter P1_COUNTRY in the fourth Name field
10. Enter Country in the fourth Label field
11. Leave all selections in the Type lists defaulted to Text
12. Click Create Multiple Items
Create Button to Submit Page

Next create a button that submits the page. Later you will create a process to call out to the Google geocoding service to get the location coordinates.

To create a button to submit the page:

1. Click the create icon in the Buttons section
2. Choose **Address to Map** from the region list and click **Next >**
3. Choose **Create a button in a region position** and click **Next >**
4. Enter **GEOCODE** in the Name field
5. Enter **Get Geocode** in the Label field and click Next >
6. Choose **Button** in the Button Template list and click **Next >**
7. Choose **Region Template Position #NEXT#** in the Position list and click **Next >**
8. Enter **1** in the Branch to Page field
9. Click **Create Button**

CREATE RESPONSE REGION

The mapping application needs a region to display and hold the geocode information that is returned from the Google geocoding service. You create an HTML region and some items in that region.
**Create HTML Region**

To create an HTML region:

1. Navigate to the Page Definition for page one of the application
2. Click the create icon in the Regions section
3. Select HTML for the region type and click Next >
4. Select HTML from the region container list and click Next >
5. Enter Geocode Response in the Title field and click Next >
6. Click Create Region

**Create Response Items**

Create two text items, an item to show the raw response from the geocode service, and an item to hold the latitude and longitude coordinates. Use the create multiple items using tabular form method to quickly create the two items.

To create multiple items using a tabular form:

1. Click the create icon in the Items section
2. Click Create multiple items using tabular form at the bottom of the page
3. Choose Geocode Response from the Create Item(s) in Region list
4. Enter P1_RESPONSE in the first Name field
5. Enter Response in the first Label field
6. Enter P1_LOCATION in the second Name field
7. Enter Location in the second label field
8. Leave all selections in the Type lists defaulted to Text
9. Click Create Multiple Items

**CALL GOOGLE GEOCODE SERVICE**

The application is now ready to call the Google geocode service. This is accomplished by creating a PL/SQL process on the page when the Get Geocode button is pressed. The process uses the supplied PL/SQL package UTL_HTTP to call out to the Google geocoding service. When the process receives the response, it populates the items in the Geocode Response region.

**Create PL/SQL Page Process to Call Geocode Service**

To create a PL/SQL page process:

1. Click the create icon in the Processes section
2. Choose PL/SQL from the process category list and click Next >
3. Enter Call Google Geocode Service in the Name field and click Next >
4. Enter the following code in the PL/SQL Page Process text field:

```
declare
  l_address varchar2(4000);
  l_url varchar2(32000);
  l_response varchar2(32000);
begin
  l_address := :P1_STREET||','||:P1_CITY;
  if :P1_STATE is not null then
    l_address := l_address||','||:P1_STATE;
  end if;
  if :P1_COUNTRY is not null then
    l_address := l_address||','||:P1_COUNTRY;
  end if;
  l_address := replace(l_address,' ','+');
  l_url :=
    'http://maps.google.com/maps/geo?q='||l_address||'&'||'output=csv'||'&'||'key='||:API_KEY;
  l_response := utl_http.request(l_url,
      APEXAPPLICATION.G_PROXY_SERVER);
  :P1_RESPONSE := l_response;
  :P1_LOCATION := substr(l_response,instr(l_response,',',1,2)+1);
end;
```

5. Click Next >

6. Enter **Called Geocode Service** in the Success Message text area

7. Enter **Error Calling Geocode Service** in the Failure Message text area

8. Click Next >

9. Select **GEOCODE** from the When Button Pressed list

10. Click Create Process

**Test the Application**

Run the page to test the process that you created. Enter an address and click the Get Geocode button. Ensure that a response is received.
Now that the application is properly geocoding addresses, the next step is to create a page that will embed the Google Map. You create a new page using the popup template. The popup page will use the Google Map JavaScript API to display the map on the page.

Create a Pop-up Page to Embed the Google Map

To create a new page that uses the pop-up page template:

1. Click Create > from the page definition of page one
2. Choose New page from the list and click Next >
3. Choose Blank Page from the page type list and click Next >
4. Enter 2 in the Page Number field and click Next >
5. Enter Map in the Name field and click Next >
6. Click Next >
7. Click Finish
8. Click Edit Page
9. Click the edit icon in Page section
10. Choose Popup from the Page Template list
11. Click Apply Changes
Create an HTML Region on the Popup Page for the Embedded Map

To create an HTML region on the page:

1. Click the create icon in the Regions section
2. Select HTML for the region type and click Next>
3. Select HTML from the region container list and click Next>
4. Enter Map in the Title field
5. Choose No Template from the Region Template list
6. Click Next>
7. Enter the following in the HTML Text Region Source text area:
   `<div id="map" style="width: 600px; height: 400px"></div>`
8. Click Create Region

Call Google Map API With JavaScript

You now call the Google Map API with JavaScript embedded in the pop-up page. The JavaScript is added to the HTML Header of the page and then the map is initialized in the onload event of the page.

To embed JavaScript in the pop-up page:

1. Click the edit icon in the Page section
2. Add the following JavaScript code in the HTML Header text area:
   ```javascript
   <script src="http://maps.google.com/maps?file=api&v=2&key=API_KEY." type="text/javascript"></script>
   <script type="text/javascript">
     //<![CDATA[
     //globals
     var bounds = new GLatLngBounds();
     function initMap() {
       if (GBrowserIsCompatible()) {
         var map = new GMap2(document.getElementById("map"));
         var title = "&P1_STREET.";
         map.addControl(new GSmallMapControl());
         map.addControl(new GMapTypeControl());
         var point = new GLatLng(&P1_LOCATION.);
         bounds.extend(point);
         map.setCenter(point);
         map.setZoom(map.getBoundsZoomLevel(bounds)-4);
         var marker = new GMarker(point);
         map.addOverlay(marker);
         marker.openInfoWindowHtml('<div class="tiny">' +
         title.replace(/~/g,"<br />") + '</div>');</div>''
     }
     //]]>
   </script>
   
3. Add the following code to the Page HTML Body Attribute text area:
   `onload="initMap()" onunload="GUnload()"`
INVOKE THE MAP POP-UP PAGE

Finally, you need to invoke the pop-up page to display the Google Map for the given address. You create a button on page one of the application in the Geocode Response region.

Create a Button to Call the Pop-up Map Page

To create a button to call the pop-up map page:

1. Navigate to the page definition of page one
2. Click the create icon in the Buttons section
3. Choose Geocode Response from the region list and click Next >
4. Choose Create a button in a region position and click Next >
5. Enter SHOW_MAP in the Button Name field
6. Enter Show Map in the Label field
7. Choose Redirect to URL without submitting page and click Next >
8. Choose Button from the Button Template list and click Next >
9. Choose Region Template Position #NEXT# from the Position list and click Next >

4. Click Apply Changes
10. Choose **URL** from Target list

11. Enter the following in the URL Target text area:
   
   `javascript:popUp2('f?p=&APP_ID.:2:&SESSION.','','650','450');`

12. Click **Create Button**

**Test the Application**

You can now test the mapping application. Enter an address and click the Get Geocode button. Click the Show Map button. Ensure that the location is proper mapped.
CONCLUSION
Oracle Application Express allows developers to easily building on-line mapping applications that use the Google Maps API. First, address information is geocoded using the Google geocoding service. Then the location is mapped by embedding maps with JavaScript using the Google Maps API.