# Integrate dynamically with Google Maps

2010

*An Oracle "How To" Paper*

*Presented By:*


*Jeremy Blankenship*

*William Indest*

*Geoff Duckwall*


ORACLE | SIEBEL

# Introduction

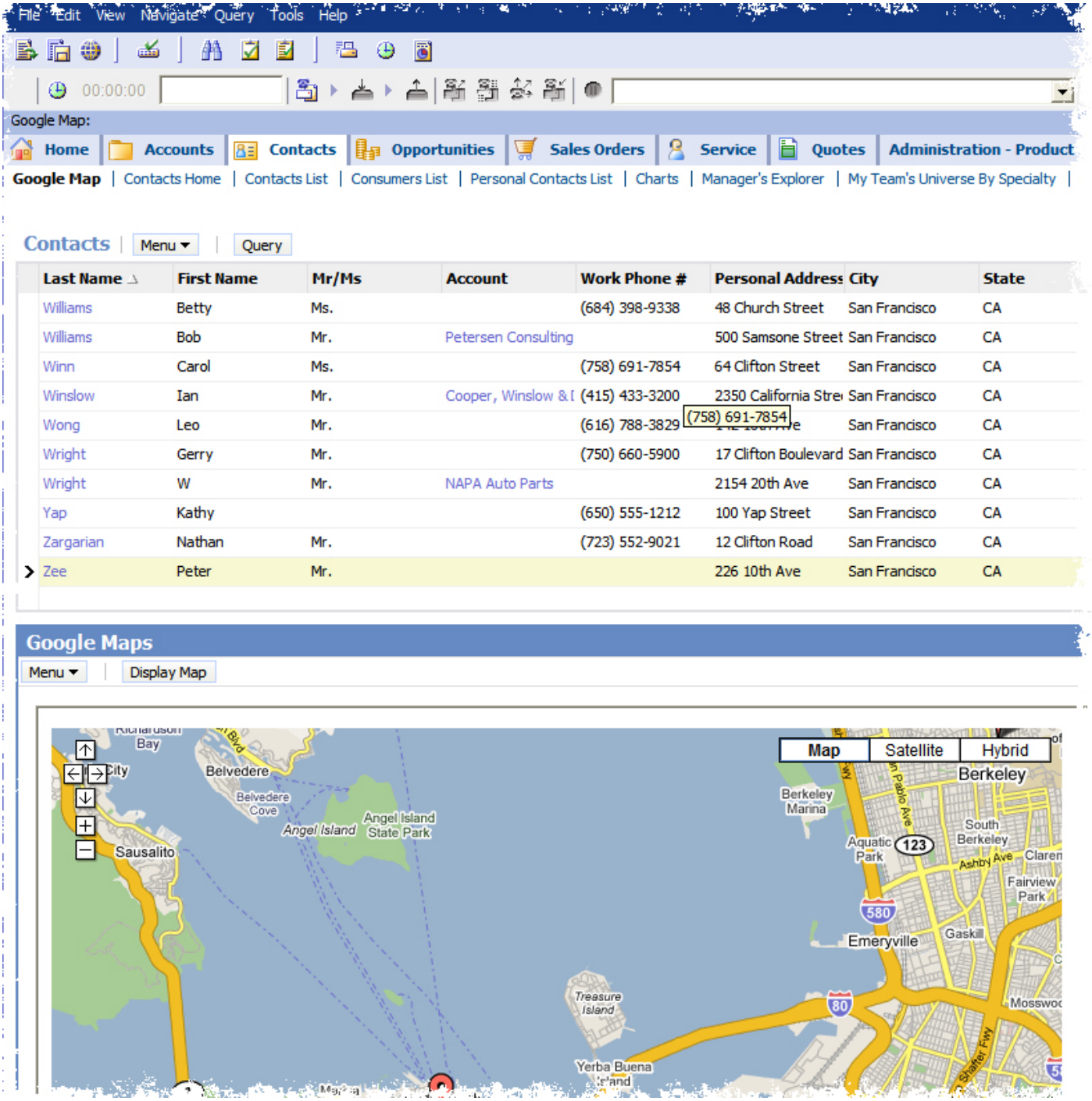
*I get to go to lots of overseas places, like Canada.*

*Britney Spears*
*Pop Singer*

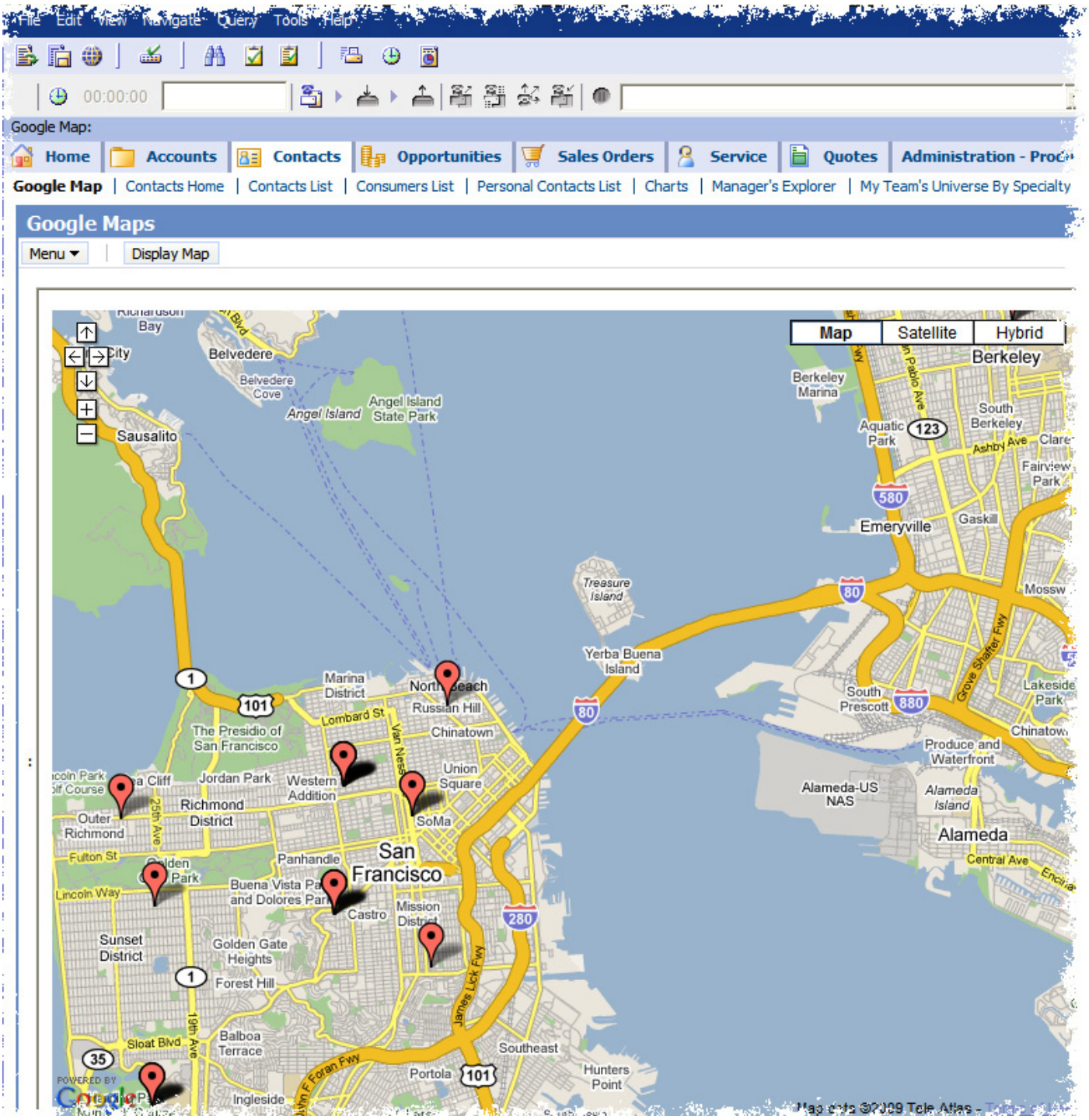
Integration with mapping software can be hard or easy. Here's an easy way. In this paper we show you how to create an applet that integrates with Google Maps using their API to display a map within a form applet. This technique mirrors our approach so elegantly presented in another runaway hit "How To" paper, entitled *Integrate with ESRI*. We focus on the Call Center application's Contact business component and show how to display a map when the linked business component, Personal Address, has latitude, and longitude coordinates. If you haven't slept through the authors' previous "How To" paper, *Create A Tag Cloud*, then you will recognize a similar design pattern.

We created a new view with a list applet on top and a form applet on the bottom to display an HTML page. The new Business Component is called Google MapUrl based on the class, CSSBCVExternalUrl. This BC doesn't do a lot. It has a calculated field for the URL,
"<IFRAME SRC="" + GetProfileAttr("GoogleMapURL") + "" height='700' width='800'></IFRAME>"
that uses a profile attribute, GoogleMapURL, created from the applet's eScript, ShowMap. The GoogleMapURL profile attribute is a string that invokes an ASP page. The ASP page is passed a list of latitudes and longitudes as well as the record Ids. The business component has a LinkField that is used only for the link. The link is between Contact and Google MapUrl; and, we added the Google MapUrl VBC to the Business Object. In summary, we made changes to the Contact Business Object and the Contact Screen, we added a new link, a new business component, a new applet and a new view.

Here's what the map view looks like:


File Edit View Navigate Query Tools Help
00:00:00
Google Map:
Home
Accounts
Contacts
Opportunities
Sales Orders
Service
Quotes
Administration - Product
Google Map | Contacts Home | Contacts List | Consumers List | Personal Contacts List | Charts | Manager's Explorer | My Team's Universe By Specialty |
Contacts | Menu | Query
Last Name | First Name | Mr/Ms | Account | Work Phone # | Personal Address | City | State
Williams | Betty | Ms. | | (684) 398-9338 | 48 Church Street | San Francisco | CA
Williams | Bob | Mr. | Petersen Consulting | | 500 Samsone Street | San Francisco | CA
Winn | Carol | Ms. | | (758) 691-7854 | 64 Clifton Street | San Francisco | CA
Winslow | Ian | Mr. | Cooper, Winslow & | (415) 433-3200 | 2350 California Stre | San Francisco | CA
Wong | Leo | Mr. | | (616) 788-3829 | (758) 691-7854 | San Francisco | CA
Wright | Gerry | Mr. | | (750) 660-5900 | 17 Clifton Boulevard | San Francisco | CA
Wright | W | Mr. | NAPA Auto Parts | | 2154 20th Ave | San Francisco | CA
Yap | Kathy | | | (650) 555-1212 | 100 Yap Street | San Francisco | CA
Zargarian | Nathan | Mr. | | (723) 552-9021 | 12 Clifton Road | San Francisco | CA
Zee | Peter | Mr. | | | 226 10th Ave | San Francisco | CA
Google Maps
Menu | Display Map
Map | Satellite | Hybrid
Belvedere
Belvedere Cove
Angel Island
Angel Island State Park
Sausalito
Berkeley
Berkeley Marina
Aquatic Park
South Berkeley
Fairview Park
Emeryville
Treasure Island
Yerba Buena Island
Shafter Fwy

And here is the lower half of the screen. Note that the query was for the city of San Francisco, CA and that these geocodes came seeded in the Quickstart demonstration environment.

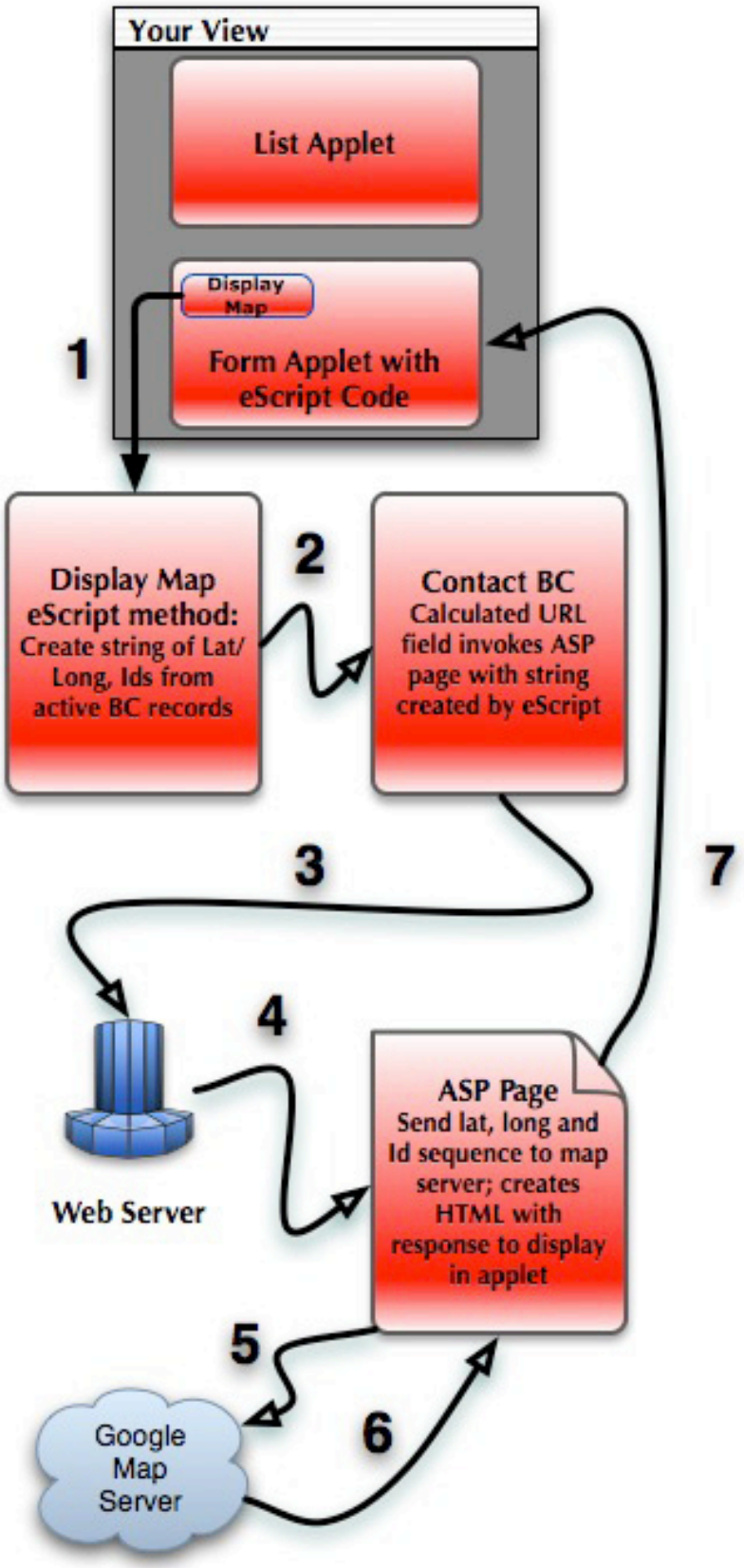


*The internet is a great way to get on the net.*

*Bob Dole*
*former Republican presidential candidate*

# How It Flows

The html source that generated the map may be found in the Appendix. Let's look at how it flows...



## Step By Step

Here is an overview of the process:

**Step 1**
Click the "Display Map" button, invoke the ShowMap eScript.

**Step 2**
Iterate over all active records creating a string consisting of record ids, latitude and longitude.

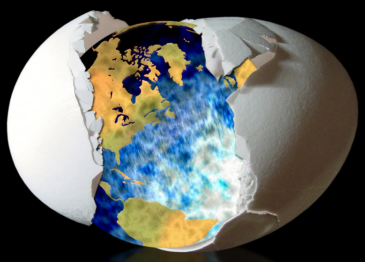
**Step 3**
The eScript on the applet sets the Profile Attribute and queries the Business Component which makes a request via the URL in the calculated field. The ASP page, ShowMap.asp, is invoked with a string of record ids and latitude/longitude pairs.

**Step 4**
The ASP page chews on the string, creating the latitude and longitude pairs and the id sequence.

**Step 5**
The ASP page formulates a request and then sends the request to the Google map server .

**Step 6**
Google Maps returns an HTML page based upon input latitude and longitudes.

**Step 7**
The HTML page is displayed in the applet.



*How to Cook It*

*The earth is what we all have in common.*

*Wendell Barry*
*Nature Lover*

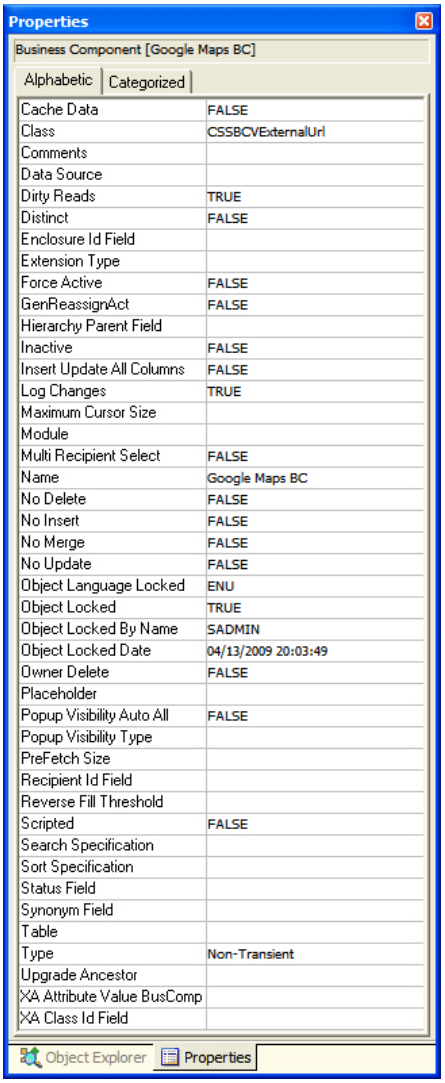
**Configuration Notes**

Although this configuration is straightforward we provide you with a sif file to help you to have this wonderful configuration up and running as soon as possible. However if you don't have the sif files here are the properties for this configuration.

All of the properties are shown below the skeletal instructions. First, create a BC, add it to the appropriate Business Object. Create a new applet then create a new view with an existing list applet and your new applet. (Remember to add the new view to the appropriate responsibility.)

The `blank.htm` and the `ShowMap.asp` files should be placed in the directory: `X:\Inetpub\wwwroot\google` (or whatever you choose to name it). The new Google MapURL Applet should excite you; this is where all the action happens. The web page with the tag field, URL, on the Business Component is displayed here and the button triggers the generation of the Profile Attribute that the BC uses to display the URL.

Here are the properties for the various components, the BC is first.



Properties

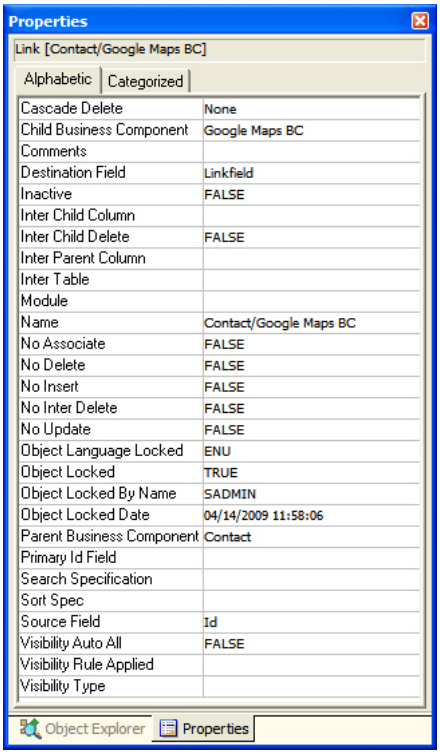
**Google Map VBC Properties:**
There are two fields for this VBC, a Linkfield to link the VBC to Contact and the MapURL field. Here are the properties of each:

Linkfield: type = DTYPE_TEXT

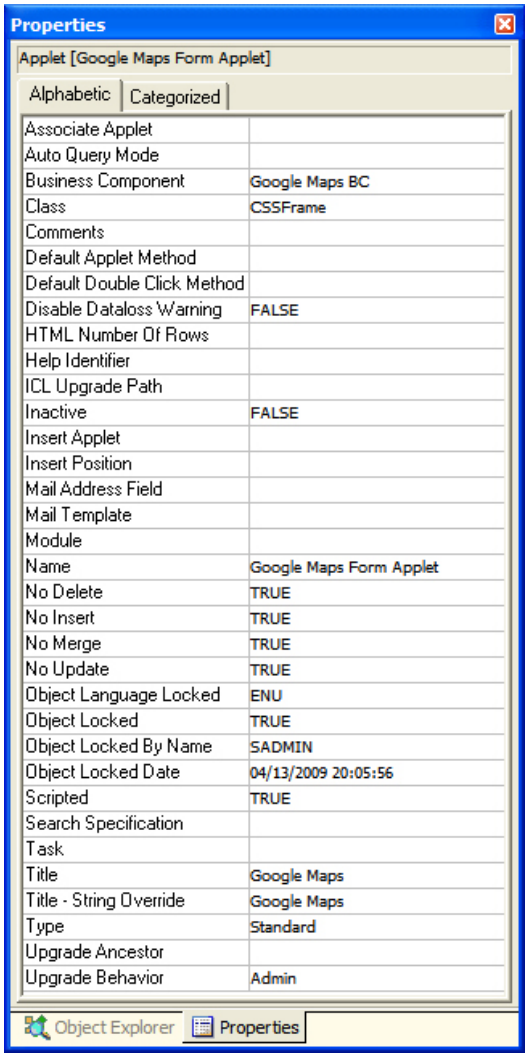
MapURL field:

| | |
|---|---|
| Calculated | TRUE |
| Calculated Value | "<IFRAME SRC="" + GetProfileAttr("GoogleMapURL") + "" height='700' width='800'></IFRAME> |

Here are the properties for the link between the Contact BC and the Google Maps VBC:




And now the applet which is where the magic happens:




Applet Web Template: Applet Form 4 Column Basic

There are two controls on the applet:
Button control: ShowMap with the Caption: Display Map;
and, the field control, MapURL, with the following properties:

| | |
|---|---|
| Field | MapURL |
| Field Type | BC Field |
| HTML Display Mode | DontEncodeData |
| HTML Only | FALSE |
| HTML Row Sensitive | TRUE |
| HTML Type | Text |
| Height | 120 |
| Inactive | FALSE |
| Name | MapURL |
| Parent Name | Google Maps Form Applet |
| Prompt | FALSE |
| Read Only | FALSE |
| Runtime | FALSE |
| Show Popup | FALSE |

| | |
|---|---|
| Sort | TRUE |
| Text Alignment | Left |
| Text Alignment-Label | Right |
| Visible | TRUE |

eScript

There are three scripts associated with the Google Maps Form applet, 1) WebApplet_Load which sets the Profile Attribute; 2) the WebApplet_PreInvokeMethod which calls the showmap script; and, 3) the WebApplet_PreCanInvokeMethod which allows the ShowMap script to be invoked.

*Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning.*

*Rich Cook*
*too many google hits*

```
function WebApplet_Load ()
{
//    TheApplication().SetProfileAttr("GoogleMapURL","http://"+  Clib.
getenv("COMPUTERNAME") + ".oracleads.com/google/blank.htm");

//  The above code is more machine agnostic if you are running this
from a demonstration instance. You should change the setting

//  of the ProfileAttr, "GoogleMapURL" to the proper server which
has a Google Maps license.

//  Note that the Google maps license is tied to a specific server

       TheApplication().SetProfileAttr("GoogleMapURL","http://wp7101.
oracleads.com/google/blank.htm");

}
```

Here is the ShowMap eScript associated with the applet:

```
function WebApplet_PreInvokeMethod (MethodName)

{

  if (MethodName == "ShowMap")

  {

    var sURL;

    var counter = 1;

    var bc;

     //sURL = "http://"+ Clib.getenv("COMPUTERNAME") + ".oracleads.
com/google/showmap.asp?";

    sURL = "http://wp7101.oracleads.com/google/showmap.asp?";

    var ActiveBO = TheApplication().ActiveBusObject();

         if (ActiveBO.Name() == "Contact") {

             var ActiveBC = ActiveBO.GetBusComp("Contact")

               with (ActiveBC){

               var IsRecord = FirstRecord();

               while (IsRecord) {

                   bc = GetMVGBusComp ("Personal Street Address")
```

```
                with (bc){
            var IsRecord2 = FirstRecord();
            while(IsRecord2) {
              bc.ActivateField("Latitude");
              bc.ActivateField("Longitude");
              bc.ActivateField("Id");
              bc.ActivateField("Address Name");
              var sLongitude = GetFieldValue("Longitude");
              var sLatitude = GetFieldValue("Latitude");
              var sId = GetFieldValue("Id");
              var sDesc = GetFieldValue("Address Name");
              if (sDesc == "")
              {
                sDesc = "Contact";
              }
              if (sLatitude != "" && sLongitude != "")
              {
                sURL = sURL + "lat" + counter + "=" + sLatitude ;
                sURL = sURL + "&lon" + counter + "=" + sLongitude;
                sURL = sURL + "&id" + counter + "=" + sId ;
                sURL = sURL + "&d" + counter + "=" + sDesc + "&";
                counter += 1;
              }
              IsRecord2 = NextRecord();
            }
          }
          IsRecord = NextRecord();
               }
           }
        } // if ActiveBC Name is Contact


    if (counter == 1)
    {
```

```
    sURL = “http://”+ Clib.getenv(“COMPUTERNAME”) + “/google/blank.
htm”;

  }

  TheApplication().SetProfileAttr(“GoogleMapURL”,sURL);

  this.BusComp().ExecuteQuery(ForwardBackward);

  return (CancelOperation);

  }


  return (ContinueOperation);

}

function WebApplet_PreCanInvokeMethod (MethodName, &CanInvoke)

{

  if (MethodName == “ShowMap”)

  {

    CanInvoke = “TRUE”;

    return (CancelOperation);

  }

  return (ContinueOperation);


}
```

## Arcane Trickery

One trick we used is setting and getting the user specified Profile Attribute via:

```
TheApplication().SetProfileAttr("GoogleMapURL",sURL);
```

While not a trick but a thoughtful design consideration is the creation of a generic ASP page for you to use independent of the business component for which you choose to create display a map. Even though the we use Contacts here if you pass a string to the ASP for any object with the string, Lat/long and id then you can reuse the ASP page for creating your very own maps.

Google dispenses ids based upon a machine name; we have tried to keep our 7101 standing up for our demonstrations so that we don't have to keep refreshing our Google Maps API id. To get your very own id go to: http://code/google.com/apis/maps/

To test try something like:

http://wp7101.oracleads.com/google/ShowMap.asp?lat0=33,lon0=83.

*No matter where you go – there you are.*

*Confucius popularized by Buckaroo Bonzai in the 8th dimension*

In this appendix we present the source code for two html files used in this configuration.
Here is the extremely complicated html code for blank.htm:

```
<html>

</html>
```

Here is the asp code for showmap.asp:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<%
  Dim latArray(101)
  Dim lonArray(101)
  Dim idArray(101)
  Dim descArray(101)
  Dim serveraddress

  Dim showHighlight

  Dim centerLat
  Dim centerLon
  Dim maxLat
  Dim maxLon
  Dim minLat
  Dim minLon
  Dim height
  Dim width
  Dim debug
  Dim startlat
  Dim startlon

  height = 600
  width = 500
  debug = "false"
  showHighlight = "false"

  serveraddress = Request.ServerVariables("LOCAL_ADDR")

  startlat = Request.QueryString("slat")
  startlon = Request.QueryString("slon")


  for i=1 to 100
     latArray(i) = Request.QueryString("lat" & i)
     lonArray(i) = Request.QueryString("lon" & i)
     idArray(i) = Request.QueryString("id" & i)
     descArray(i) = Request.QueryString("d" & i)
  next

  centerLat = 0
  centerLon = 0
  maxLat = 0
  maxLon = 0
  minLat = 0
  minLon = 0

  maxLat = latArray(1)
  minLat = latArray(1)
```

```
  maxLon = lonArray(1)
  minLon = lonArray(1)

  centerLat = latArray(1)
  centerLon = lonArray(1)

' Figure out the maximum and minimum lat and lon values
  for i = 2 to 100
     if latArray(i) <> "" then
       if latArray(i) > maxLat then
          maxLat = latArray(i)
       end if

       if latArray(i) < minLat then
         minLat = latArray(i)
       end if

       if lonArray(i) > maxLon then
         maxLon = lonArray(i)
       end if

       if lonArray(i) < minLon then
        minLon = lonArray(i)
       end if
    end if
  next

if startlat <> "" then
    centerlat = startlat
    centerlong = startlon
elseif latArray(2) <> "" then
    centerLat = (cDbl(minLat) + CDbl(maxLat)) / 2
    centerLon = (cDbl(minLon) + CDbl(maxLon)) / 2
end if


%>

 <html xmlns="http://www.w3.org/1999/xhtml"
xmlns:v="urn:schemas-microsoft-com:vml">
  <head>
    <meta http-equiv="content-type" content="text/html;
charset=utf-8"/>
    <title>Google Maps JavaScript API Example: Custom Icon</
title>
    <script src="http://maps.google.com/maps?file=api&amp;v=2&am
p;key=ABQIAAAA-cdXX9Zhk2SupnZPK455vhSCaSD26KZyoBR2lM_EYur-FLlw-
CxRodUbeiVRQ2Iwt8o34-R4ppohbYQ"
            type="text/javascript"></script>
    <script type="text/javascript">
    function initialize() {
      if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_can-
vas"));
        map.setCenter(new GLatLng(<%=centerLat %> ,<%=centerLon
%>), 13);
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());


        // Create a base icon for all of our markers that speci-
fies the
        // shadow, icon dimensions, etc.
        var baseIcon = new GIcon(G_DEFAULT_ICON);
```

```
        baseIcon.shadow = “http://www.google.com/mapfiles/shad-
ow50.png”;
        baseIcon.iconSize = new GSize(20, 34);
        baseIcon.shadowSize = new GSize(37, 34);
        baseIcon.iconAnchor = new GPoint(9, 34);
        baseIcon.infoWindowAnchor = new GPoint(9, 2);


        // Creates a marker based on the type of item to map
        function createMarker(point, description,rowid) {
               //Remove any spaces from description to pull up
correct filename
               var filename = description.split(‘ ‘).join(‘’);
               var descriptionIcon = new GIcon(baseIcon);
         // descriptionIcon.image = “http://<%=serveraddress%>/
google/icons/” + filename + “.png”;

          // Set up our GMarkerOptions object
          markerOptions = { icon:descriptionIcon };
          var marker = new GMarker(point, markerOptions);

          GEvent.addListener(marker, “click”, function() {
            marker.openInfoWindowHtml(description + “ Id:” + ro-
wid );

          });
          return marker;
        }

             var bounds = new GLatLngBounds();
        <%for i = 1 to 100
         if latArray(i) <> “” then %>
          var latlng<%=i%> = new GLatLng(<%=latArray(i) %>,
<%=lonArray(i) %>);
          map.addOverlay(createMarker(latlng<%=i%>, “<%=
descArray(i) %>”, “<%= idArray(i) %>”));
               bounds.extend(latlng<%=i%>);
           <% end if
         next %>
             }

             // If a Starting position has been passed in, mark
the location with a Star icon
             <% if startlat <> “” then %>
             var latlngstart = new GLatLng(<%=startlat%>,<%=star
tlon%>);
             map.addOverlay(createMarker(latlngstart, “Star”,
“None”));
             <% end if%>

             map.setZoom(map.getBoundsZoomLevel(bounds));
             map.setCenter(new GLatLng(<%=centerLat %> ,<%=cen-
terLon %>));
    }
    </script>
  </head>
  <body onload=”initialize()” onunload=”GUnload()”>
    <div id=”map_canvas” style=”width: 760px; height: 600px”></
div>

  </body>
</html>
```

**Title: Integrate dynamically with Google Maps**

**Authors: Jeremy Blankenship, William Indest and Geoff Duckwall**

*Our Bosses*

*Managed By:*

*David Terry*

*Jim Perry*

*Todd Lesser*

*Legal Stuff*

**Oracle Corporation**
**World Headquarters**
**500 Oracle Parkway**
**Redwood Shores, CA 94065**
**U.S.A.**
**Worldwide Inquiries:**
**Phone: +1.650.506.7000**
**Fax: +1.650.506.7200**
**oracle.com**