Postcode Anywhere and Oracle Fusion CRM Integration Summary

Contents

Purpose
The purpose of this document is to provide detailed instructions for manually re-creating customizations within Oracle Fusion Customer Relationship Management (Oracle Fusion CRM) that are required for the functional integration between the partner application and Oracle Fusion CRM. Other information is provided for context only.

Disclaimer
The information in this document was valid at the time of document creation.

Solution Description

One of the International Addressing and Validation services provided by Postcode Anywhere (http://www.postcodeanywhere.co.uk/) is known as Capture+. This document presents the integration between Oracle Fusion CRM and Postcode Anywhere (PCA) Capture+.

The most common use of this integration is when Oracle Fusion CRM users enter customer information. Oracle Fusion CRM produces a search window to search for addresses already validated and known to PCA. As the user enters the name of the customer, a short list of potential targets is presented, with the list becoming shorter as the user continues entering information. The user selects the proper target and imports the customer address into Oracle Fusion CRM.

The use of PCA ensures that addresses are entered into Oracle Fusion CRM with a high degree of quality and consistency. This mitigates any downstream issues involving unintentional duplicate data entry or other *fuzzy matching* complications.

Environment

Oracle Fusion CRM Release 4 (11.1.4.0.0)

Known Prerequisites

This solution requires that some code be deployed to an external-to-Oracle Fusion CRM application server.

UI Integration Summary and Use Case Walkthrough

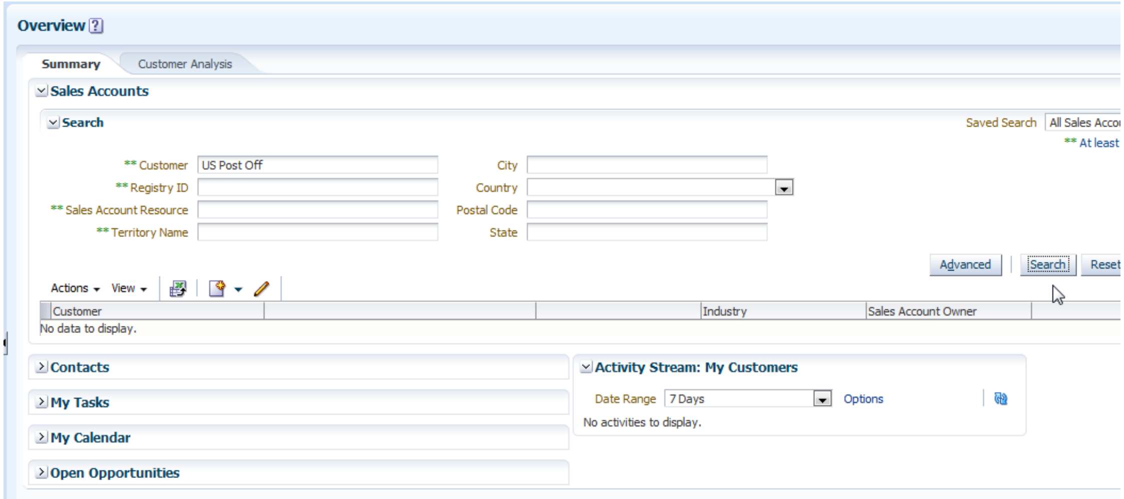
Oracle Fusion CRM Customer Form

A **Postcode Anywhere Search** link is embedded in the Oracle Fusion CRM Create Customer form. The steps for re-creating this integration of PCA and Oracle Fusion CRM technology are documented in the section Creating the Postcode Anywhere Link (Points 1 and 2 of Integration Flow Diagram).

This link triggers a search when the user begins to enter the address in line 1 or a company name. A web service call is made to get suggested addresses. The user is presented with a list of possible matches. The user can either select the result from that list, which causes another web service call to retrieve the result, or continue to enter information that causes another web service call to get additional address options.

Use Case Walkthrough

As a logged-in user, you go to the Customer Entry transaction form. As an example, as a sales representative, you wish to add a customer whose address happens to be a post office in New York. You have searched, but you have not found any existing post offices in the Oracle Fusion CRM system.



** Customer: US Post Off

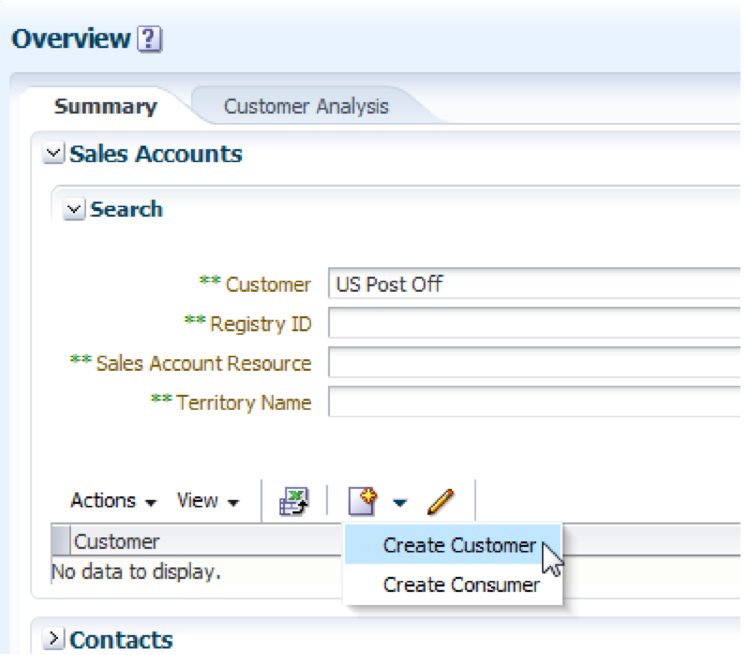
Saved Search: All Sales Acco

Date Range: 7 Days

No data to display.

No activities to display.

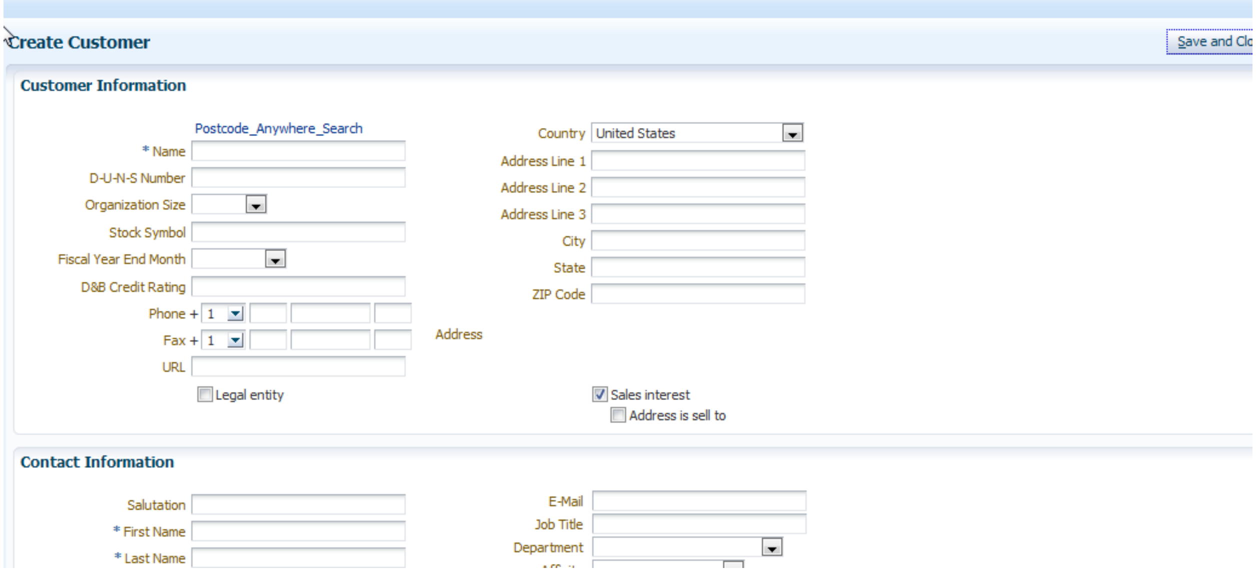
Typically, you will then create a customer.



** Customer: US Post Off

No data to display.

If you are unsure of the exact address to enter, you have the option of clicking the **Postcode Anywhere Search** link that is presented on the Oracle Fusion CRM Create Customer form.



Create Customer

Customer Information

Postcode_Anywhere_Search

* Name
D-U-N-S Number
Organization Size
Stock Symbol
Fiscal Year End Month
D&B Credit Rating
Phone + 1
Fax + 1
URL
□ Legal entity

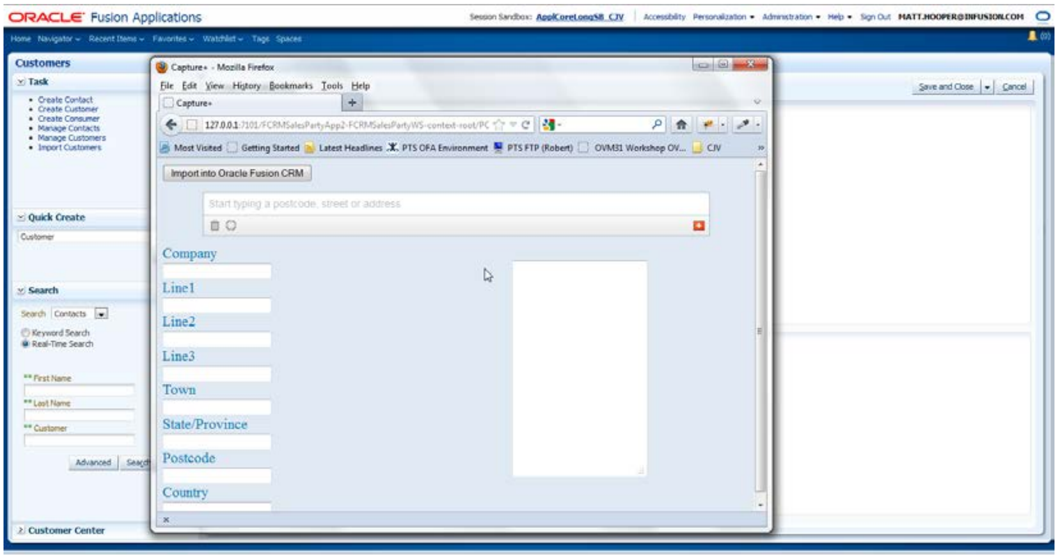
Country United States
Address Line 1
Address Line 2
Address Line 3
City
State
ZIP Code
Address
☑ Sales interest
□ Address is sell to

Contact Information

Salutation
* First Name
* Last Name

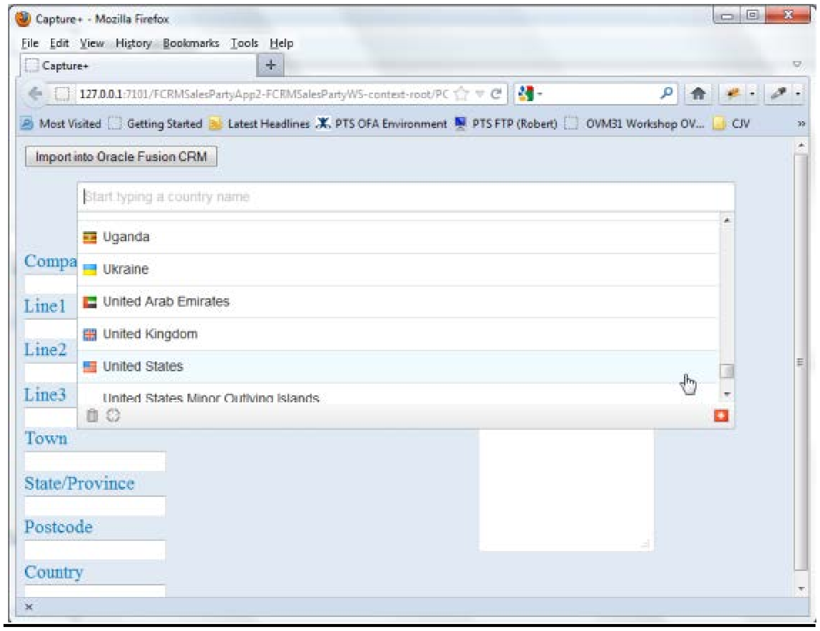
E-Mail
Job Title
Department

A browser window appears that provides a search box. The country in which the search is applied can be a preconfigured default, or it can be based upon the location of the user. For demonstration purposes, the latter is the case here.

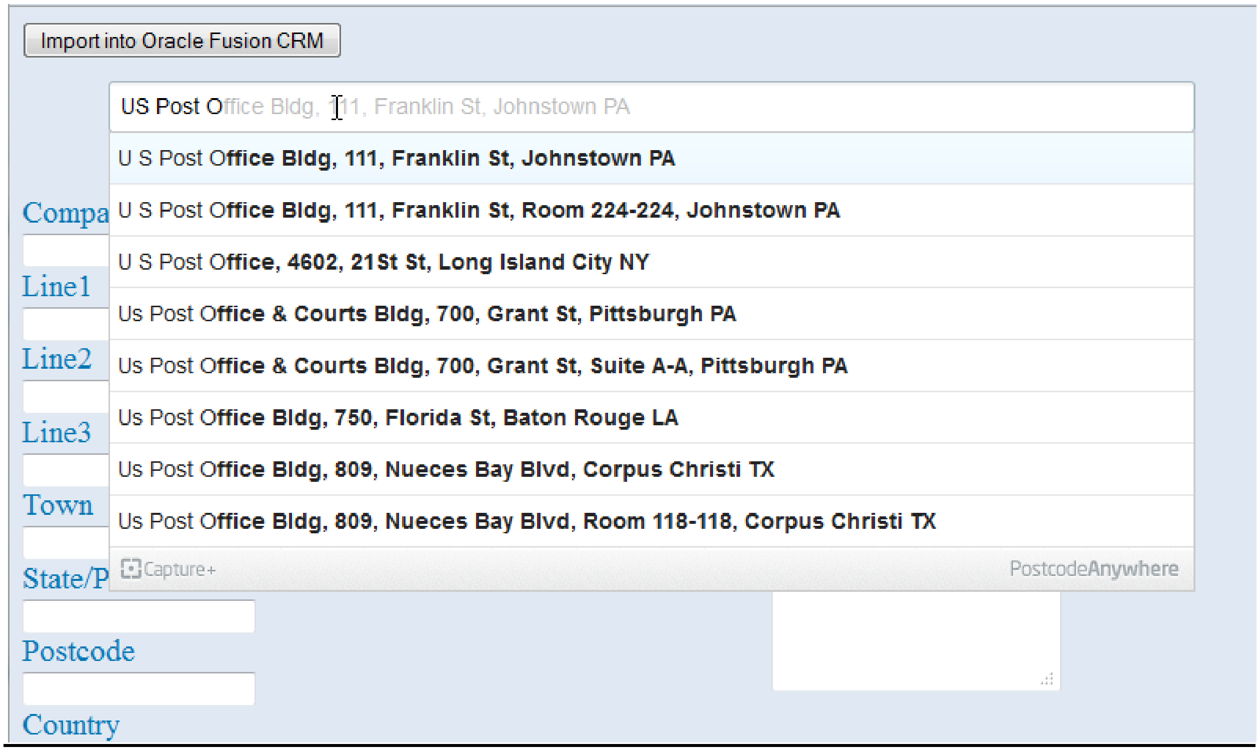

ORACLE Fusion Applications
Customers
Task
Create Contact
Create Customer
Create Consumer
Manage Contacts
Manage Customers
Import Customers
Quick Create
Search
Customer Center
Capture+ - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Capture+
127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-context-root/PC
Import into Oracle Fusion CRM
Start typing a postcode, street or address
Company
Line1
Line2
Line3
Town
State/Province
Postcode
Country
Save and Close
Cancel

You switch the scope of the search to United States.

Note: This can be made a default constant value, but this example highlights that PCA is an international address validation tool, valid for many countries.


Capture+ - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Capture+
127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-context-root/PC
Most Visited
Getting Started
Latest Headlines
PTS OFA Environment
PTS FTP (Robert)
OVM31 Workshop OV...
CJV
Import into Oracle Fusion CRM
Start typing a country name
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States
United States Minor Outlying Islands
Compa
Line1
Line2
Line3
Town
State/Province
Postcode
Country

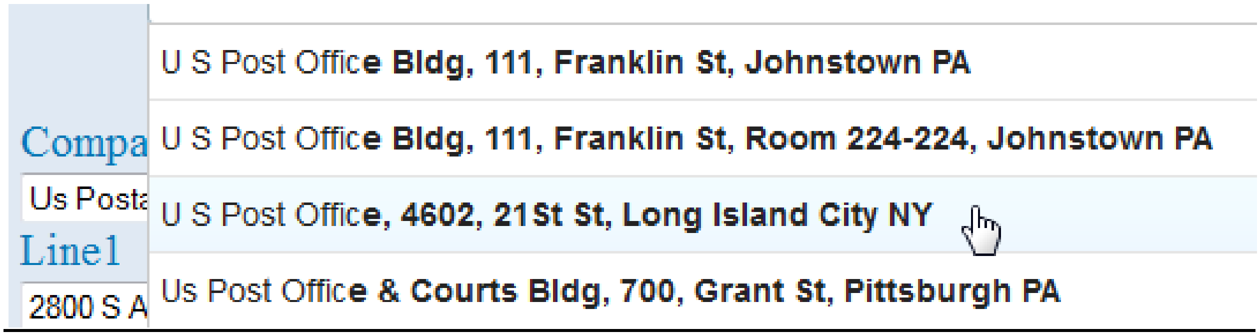
As you enter "U," then "S," then space, then "P," and so on, the browser makes calls to PCA and returns a list of potential targets. Each time you enter a letter, the list of targets is reduced, until you see the address for which you are searching.



US Post Office Bldg, 111, Franklin St, Johnstown PA

U S Post Office Bldg, 111, Franklin St, Johnstown PA
U S Post Office Bldg, 111, Franklin St, Room 224-224, Johnstown PA
U S Post Office, 4602, 21St St, Long Island City NY
Us Post Office & Courts Bldg, 700, Grant St, Pittsburgh PA
Us Post Office & Courts Bldg, 700, Grant St, Suite A-A, Pittsburgh PA
Us Post Office Bldg, 750, Florida St, Baton Rouge LA
Us Post Office Bldg, 809, Nueces Bay Blvd, Corpus Christi TX
Us Post Office Bldg, 809, Nueces Bay Blvd, Room 118-118, Corpus Christi TX

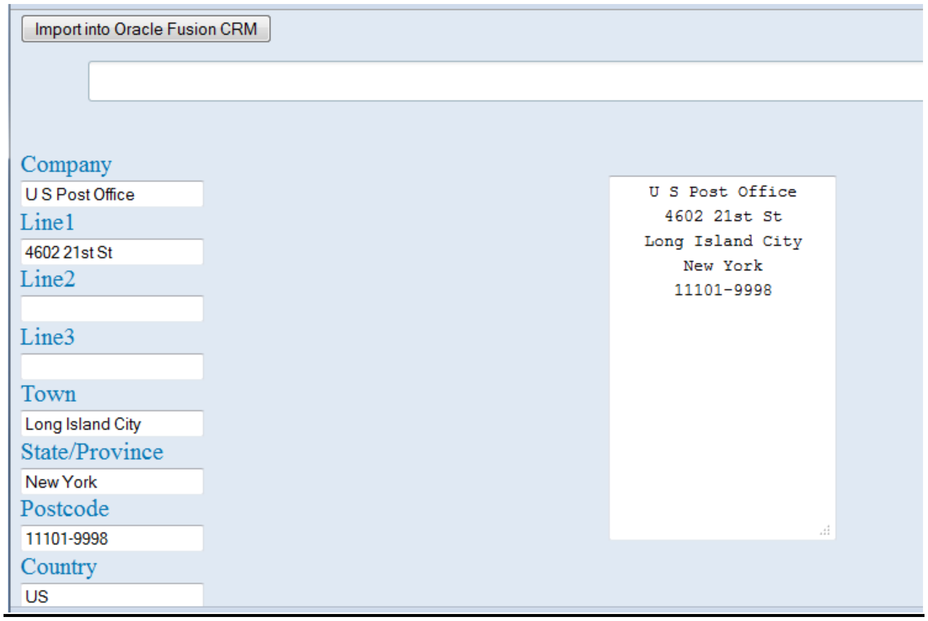
In this case, the post office in Long Island City, New York:



Us Posta
2800 S A

U S Post Office Bldg, 111, Franklin St, Johnstown PA
U S Post Office Bldg, 111, Franklin St, Room 224-224, Johnstown PA
U S Post Office, 4602, 21St St, Long Island City NY
Us Post Office & Courts Bldg, 700, Grant St, Pittsburgh PA

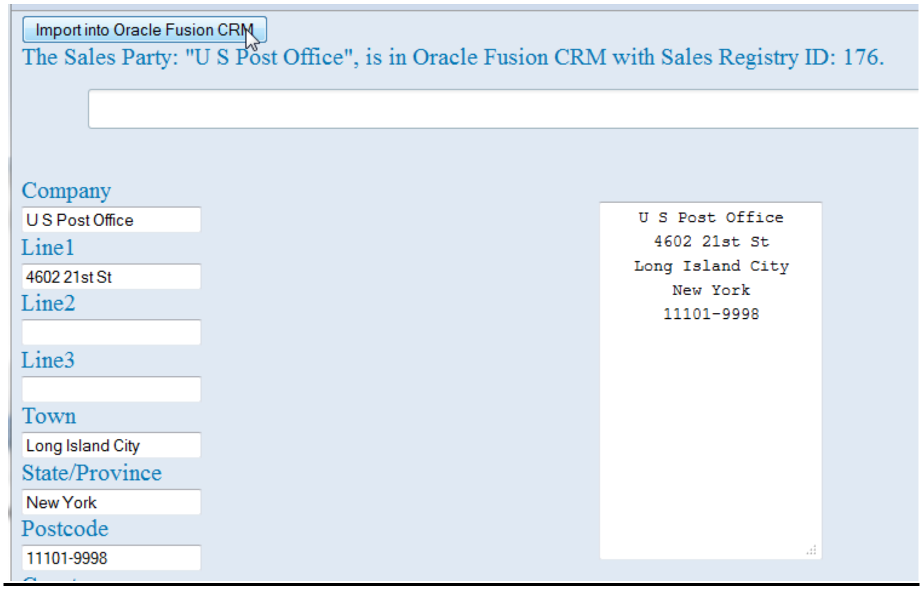
Click that post office and the details are displayed in the browser form:



Company: U S Post Office
Line1: 4602 21st St
Line2:
Line3:
Town: Long Island City
State/Province: New York
Postcode: 11101-9998
Country: US

```
U S Post Office
4602 21st St
Long Island City
New York
11101-9998
```

Click the **Import into Oracle Fusion CRM** button. The browser invokes the Oracle Fusion CRM web services required to import this customer into Oracle Fusion CRM. Note the Registry ID (176) is displayed here. It is the Oracle Fusion CRM customer number (PartyNumber) that was just created in Oracle Fusion CRM.

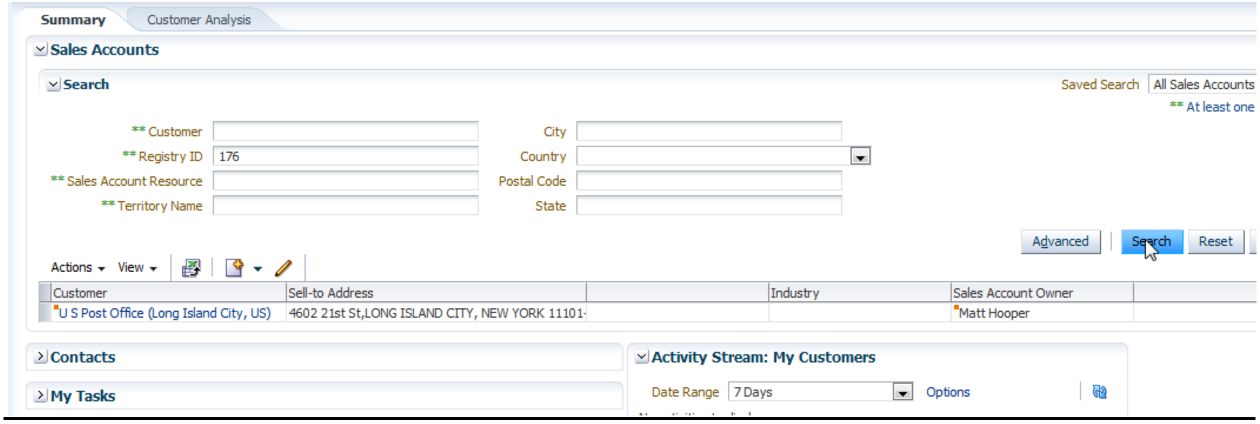


The Sales Party: "U S Post Office", is in Oracle Fusion CRM with Sales Registry ID: 176.

Company: U S Post Office
Line1: 4602 21st St
Line2:
Line3:
Town: Long Island City
State/Province: New York
Postcode: 11101-9998

```
U S Post Office
4602 21st St
Long Island City
New York
11101-9998
```

Close the window and return to Oracle Fusion CRM. Search directly on the Registry ID and find the newly imported customer.



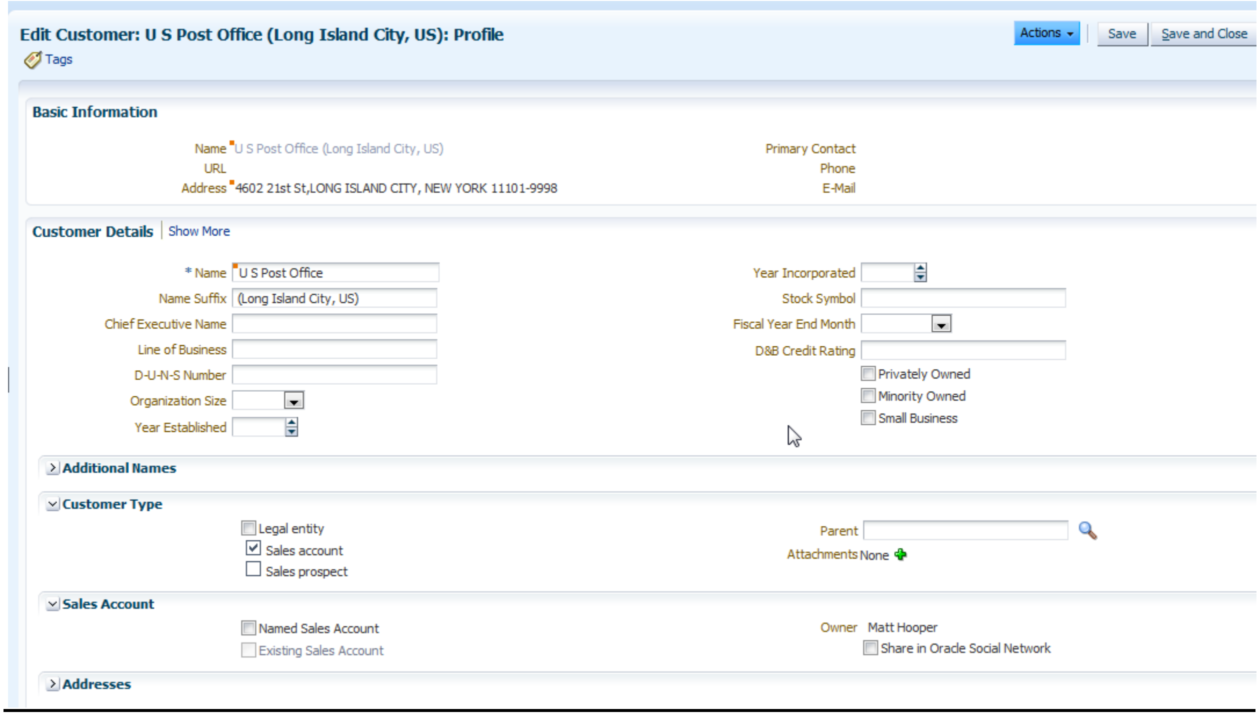
Saved Search: All Sales Accounts
** At least one

** Customer		City	
** Registry ID	176	Country	
** Sales Account Resource		Postal Code	
** Territory Name		State	

Customer	Sell-to Address		Industry	Sales Account Owner
U S Post Office (Long Island City, US)	4602 21st St,LONG ISLAND CITY, NEW YORK 11101-			Matt Hooper

Date Range: 7 Days

At this point, you can update all the information and attributes managed by Oracle Fusion CRM, knowing that the address of this customer has been entered consistently and with a high degree of data quality.



Edit Customer: U S Post Office (Long Island City, US): Profile

Basic Information

Name	U S Post Office (Long Island City, US)	Primary Contact	
URL		Phone	
Address	4602 21st St,LONG ISLAND CITY, NEW YORK 11101-9998	E-Mail	

Customer Details

* Name	U S Post Office	Year Incorporated	
Name Suffix	(Long Island City, US)	Stock Symbol	
Chief Executive Name		Fiscal Year End Month	
Line of Business		D&B Credit Rating	
D-U-N-S Number			□ Privately Owned
Organization Size			□ Minority Owned
Year Established			□ Small Business

Customer Type

□ Legal entity
☑ Sales account
□ Sales prospect

Parent:
Attachments None

Sales Account

□ Named Sales Account
□ Existing Sales Account

Owner: Matt Hooper
□ Share in Oracle Social Network

Authentication and Authorization

To request an account for PostCode Anywhere, visit their website (http://www.postcodeanywhere.co.uk/) and click the **Sign-Up** button.

After you register on the Postcode Anywhere site, you will be provided with an API key. This key is to be included in the URL when calling PCA web services, usually using an HTTP GET request, for example:

https://services.postcodeanywhere.co.uk/CapturePlus/Interactive/AutoComplete/v2.00/json3.ws?key=<PCAClientAPIKey>&Country=US&SearchTerm=Oracle

You have the option to ask PCA to monitor for any unusual usage patterns to ensure that no one else is using your key.

Oracle Fusion CRM Web Services Invoked

The following services are invoked by Oracle-created code that is to be deployed in an external-to-Oracle Fusion CRM software as a service (SaaS) application server:

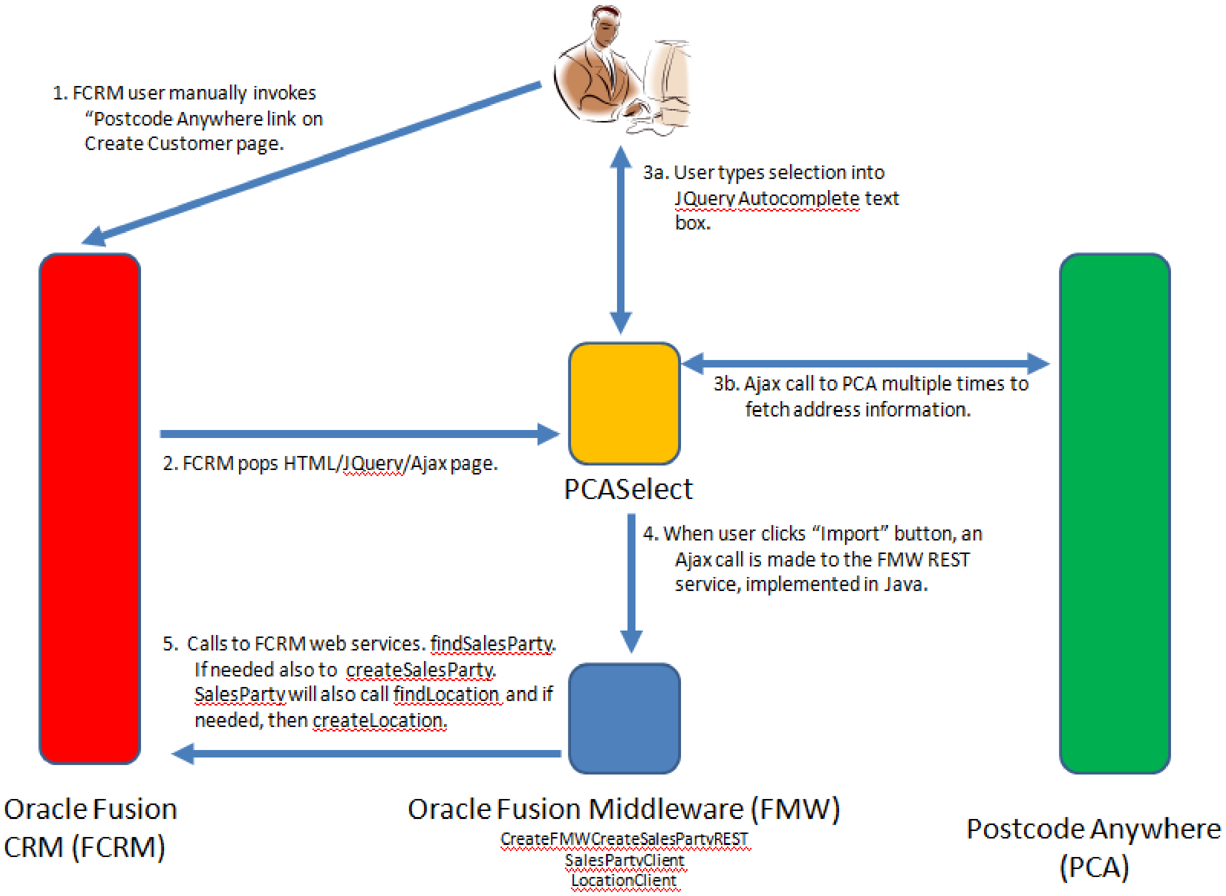
- SalesPartyService
    - findSalesParty
    - createSalesParty
- LocationService
    - findLocation
    - createLocation

SOAP Request Examples

The SOAP endpoints are invoked by Java web service calls. See the extensive example code in the Appendixes.

Creating the Postcode Anywhere and Oracle Fusion CRM Integration

The integration flow is depicted in the following diagram. The enumerated flows will be discussed in more detail to explain how to implement the integration.

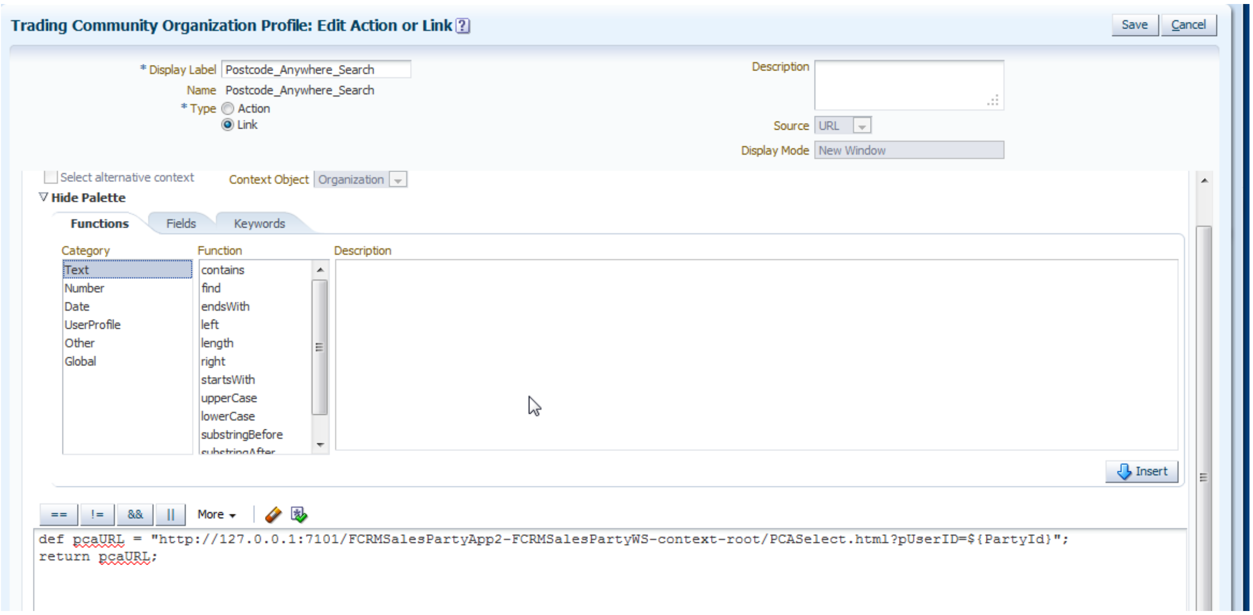

1. FCRM user manually invokes “Postcode Anywhere link on Create Customer page.
3a. User types selection into JQuery Autocomplete text box.
3b. Ajax call to PCA multiple times to fetch address information.
2. FCRM pops HTML/JQuery/Ajax page.
PCASelect
4. When user clicks “Import” button, an Ajax call is made to the FMW REST service, implemented in Java.
5. Calls to FCRM web services. findSalesParty. If needed also to createSalesParty. SalesParty will also call findLocation and if needed, then createLocation.
Oracle Fusion CRM (FCRM)
Oracle Fusion Middleware (FMW)
CreateFMWCreateSalesPartyREST
SalesPartyClient
LocationClient
Postcode Anywhere (PCA)

Creating the Postcode Anywhere Link (Points 1 and 2 of Integration Flow Diagram)

Follow these steps to add the Postcode Anywhere link (backed by a Groovy script detailed as follows) to the Create object. The URL provides the Postcode Anywhere application and passes some identifier parameters:

1. Select **Navigator**, then **Tools**, and then **Application Composer.**
2. From the Oracle Fusion CRM application drop-down list, select the **Common** application.
3. Expand the **Standard Objects** tree node.
4. Expand the **Trading Community Organization Profile** object node.
5. Click the **Actions and Links** link.
6. At the top of the Actions and Links table, click the **Create** button.
7. Set Display Label to **Postcode Anywhere Search.**
8. Set Name to **Postcode Anywhere Search**.
9. Set Type to **Link** (source should automatically set to **URL**).
10. In the script text field, copy and paste this Groovy script:

```
def pcaURL = "http://127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-
context-root/PCASelect.html?pUserID=${PartyId}";
return pcaURL
```



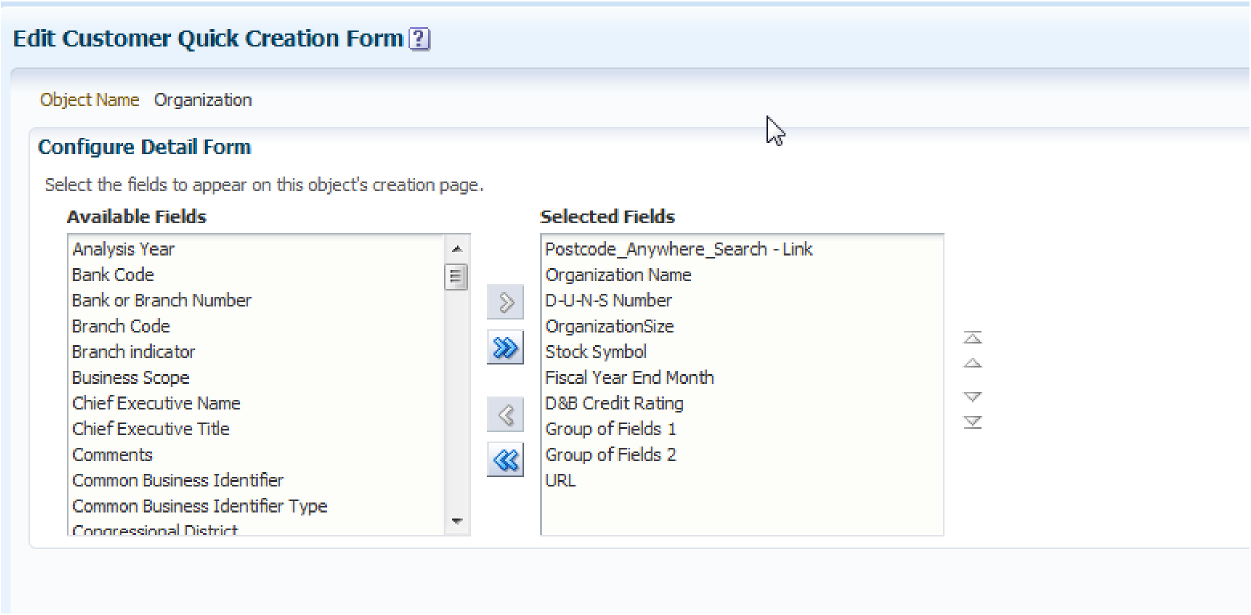
Trading Community Organization Profile: Edit Action or Link

* Display Label Postcode_Anywhere_Search
Name Postcode_Anywhere_Search
* Type ○ Action ● Link
Description
Source URL
Display Mode New Window
□ Select alternative context
Context Object Organization

```
def pcaURL = "http://127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-context-root/PCASelect.html?pUserID=${PartyId}";
return pcaURL;
```

11. After pasting the script, click the Groovy expression validation icon to verify that the script is valid.
12. After validation is successful, save the link definition and exit the edit screen.
13. Click the **Pages** icon.

14. Click the **Edit Customer Quick Creation Form**.
15. In the **Configure Detail Form**, move **Postcode Anywhere Search** from the **Available Fields** list to the **Selected Fields** list and move the link to the desired position.



Edit Customer Quick Creation Form

Object Name Organization

Configure Detail Form

Select the fields to appear on this object's creation page.

Available Fields

Analysis Year
Bank Code
Bank or Branch Number
Branch Code
Branch indicator
Business Scope
Chief Executive Name
Chief Executive Title
Comments
Common Business Identifier
Common Business Identifier Type
Congressional District

Selected Fields

Postcode_Anywhere_Search - Link
Organization Name
D-U-N-S Number
OrganizationSize
Stock Symbol
Fiscal Year End Month
D&B Credit Rating
Group of Fields 1
Group of Fields 2
URL

16. Click **Save and Close**.

PCASelect Page Using JQuery/Ajax (Points 3a and 3b of Integration Flow Diagram)

The value proposition of Postcode Anywhere (PCA) is best exposed when the calls to PCA are embedded in a dynamic client. A browser page appears and includes elements of JQuery/JavaScript to achieve active, client-side effects, especially AutoComplete. Within a JQuery program, you can dynamically call the PCA web services by using Ajax (`http://en.wikipedia.org/wiki/Ajax_(programming))` technology.

There are some well-known limitations on using Ajax to call certain services. The most well-known is the Cross Domain issue. By design, most implementations of Ajax have a default implementation that inhibits browsers from making calls to services in other domains. There are some workarounds to these limitations, such as the Cross-Origin Resource Sharing – CORS (`http://en.wikipedia.org/wiki/Cross-origin_resource_sharing`) approach, which involves setting the HTTP header prior to sending the service request. However, the implementation of this approach is subject to various interpretations by both browsers and servers. Due to its relatively recent arrival, different browsers and servers implement CORS with different behaviors.

For this reason, the Ajax calls using the JavaScript Object Notation - JSON (`http://en.wikipedia.org/wiki/JSON`) approach has been chosen for implementation. Due to its maturity as an approach, it seems to have a much more consistent implementation across heterogeneous environments. Fortunately, Postcode Anywhere exposes its services using JSON (as well as XML/SOAP). In particular, the PCA JSON endpoints used are:

- AutoComplete, for example, `https://services.postcodeanywhere.co.uk/CapturePlus/Interactive/AutoComplete/v2.00/json3.ws?key=<PCAClientKey>&Country=US&SearchTerm=Oracle`
- RetrieveById, for example, `https://services.postcodeanywhere.co.uk/CapturePlus/Interactive/RetrieveById/v2.00/json3.ws?key=<PCAClientKey>&Id=<AnIdFromAutoComplete>`

The approach is first to call AutoComplete and pass in SearchTerm. A JSON message will be returned similar to the following:

```
{"Items":[{"Id":"USA|CT|321|","Match":"Oracle","Suggestion":"
AZ","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|ST|1721|1240796|","Match":"Oracle","Suggestion":" Blvd, Colorado Springs
CO","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|PR|41932829|0","Match":"Oracle","Suggestion":" Corp, 500, Oracle Pkwy,
Redwood City CA","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|PR|28026247|0","Match":"Oracle","Suggestion":" Corporation,
233, S Wacker Dr, Suite 4500, Chicago
IL","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|ST|2455|592122|","Match":"Oracle","Suggestion":" Ct, Ellijay
GA","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|ST|29650|592122|","Match":"Oracle","Suggestion":" Ct, Johnson City
TN","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|ST|3686|592122|","Match":"Oracle","Suggestion":" Ct, Newbury Park
CA","IsRetrievable":false,"AutoSelect":false},{"Id":"USA|ST|3686|5921230|","Match":"Oracle","Suggestion":" Long
Island","IsRetrievable":false,"AutoSelect":false}]}
```

This is an array containing possible matches. Note the ID field. For each row in the array, call RetrieveById, passing in the ID for that row. For each call, a JSON array is received. As an illustration, if the ID “USA|ST|2455|592122|” (the ID for Oracle in Johnson City) is passed, a message similar to this is received:

```
{"Items":[{"FieldGroup":"Common","FieldName":"Id","FormattedValue":"USA|ST|2455|592122|","FieldType":"Unknown","FieldSequence":0,
"Language":"ENG"},{"FieldGroup":"Common","FieldName":"Company","FormattedValue":"","FieldType":"Unknown","FieldSequence":1,"Langu
age":"ENG"},{"FieldGroup":"Common","FieldName":"Line1","FormattedValue":"Oracle
Ct","FieldType":"Unknown","FieldSequence":2,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"Line2","FormattedValue":"","Fie
ldType":"Unknown","FieldSequence":3,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"City","FormattedValue":"Ellijay","Field
Type":"Unknown","FieldSequence":4,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"ProvinceCode","FormattedValue":"GA","Fiel
dType":"Unknown","FieldSequence":5,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"ProvinceName","FormattedValue":"Georgia"
,"FieldType":"Unknown","FieldSequence":6,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"CountryCode","FormattedValue":"USA
","FieldType":"Unknown","FieldSequence":7,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"CountryName","FormattedValue":"Un
ited
States","FieldType":"Unknown","FieldSequence":8,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"PostalCode","FormattedValue
":"30540","FieldType":"Unknown","FieldSequence":9,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"Type","FormattedValue":"R
esidential","FieldType":"Unknown","FieldSequence":10,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"District","FormattedVa
lue":"","FieldType":"Unknown","FieldSequence":11,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"SubBuilding","FormattedVal
ue":"","FieldType":"Unknown","FieldSequence":12,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"BuildingNumber","FormattedV
alue":"","FieldType":"Unknown","FieldSequence":13,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"BuildingName","FormattedV
alue":"","FieldType":"Unknown","FieldSequence":14,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"Street","FormattedValue":
"Oracle
Ct","FieldType":"Unknown","FieldSequence":15,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"Label","FormattedValue":"Oracl
e Ct\nEllijay GA 30540\nUnited
States","FieldType":"Unknown","FieldSequence":16,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"AdminAreaCode","FormattedV
alue":"123","FieldType":"Unknown","FieldSequence":17,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"AdminAreaName","Format
tedValue":"Gilmer","FieldType":"Unknown","FieldSequence":18,"Language":"ENG"},{"FieldGroup":"Common","FieldName":"CountryCode2","
FormattedValue":"US","FieldType":"Unknown","FieldSequence":19,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"Urbanization
","FormattedValue":"","FieldType":"Unknown","FieldSequence":0,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"ZIP","Format
tedValue":"30540","FieldType":"Unknown","FieldSequence":1,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"AddOn","Formatte
dValue":"","FieldType":"Unknown","FieldSequence":2,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"POBox","FormattedValue"
:"","FieldType":"Unknown","FieldSequence":3,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"StreetPrefixDirection","Format
tedValue":"","FieldType":"Unknown","FieldSequence":4,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"StreetName","Formatte
dValue":"Oracle","FieldType":"Unknown","FieldSequence":5,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"StreetType","Form
attedValue":"Ct","FieldType":"Unknown","FieldSequence":6,"Language":"ENG"},{"FieldGroup":"Country","FieldName":"StreetSuffixDirec
tion","FormattedValue":"","FieldType":"Unknown","FieldSequence":7,"Language":"ENG"}]}
```

Each array item represents an attribute of the address. Certain address attributes that are of interest are extracted and presented to the user. The user will manually decide if one of these results fulfills the search.

For flexibility, two versions of the PCASelect.html page are created:

- In the first version, full control of the calls to the PCA APIs is managed by the PCASelect.html page itself. A call is made to AutoComplete and then there are multiple calls to RetrieveById for each returned array item.
- In the second version, a JavaScript function that is directly supported by Postcode Anywhere is leveraged. The APIs are not directly invoked. Rather, the Postcode Anywhere JavaScript function is invoked and returns a document with fields that can be accessed. The PCA JavaScript also has a prebuilt AutoComplete select field that allows the user to specify the country prior to invoking the APIs. This is the version used in the Use Case Walkthrough presented earlier.

For a complete listing of both approaches to the the PCASelect code, see the Appendixes.

Importing PCA Data (Point 4 of Integration Flow Diagram)

Regardless of which approach is used to call the PCA APIs and present the response, at some point the sales representative clicks the **Import to Oracle Fusion CRM** button and the Oracle Fusion CRM web services are called to import the customer and address into Oracle Fusion CRM.

As discussed earlier in this document, there are some challenges to be faced when calling from a client browser to SOAP web services. Despite using CORS to avoid cross-origin issues, the various implementations cannot yet be relied upon to produce consistent behavior. As a consequence, a JSON approach is used to mitigate those issues when calling PCA APIs. However, Oracle Fusion CRM does not expose JSON services. Instead, it exposes SOAP web services.

Therefore, to ensure consistent behavior of the application, a Representational State Transfer (REST) service in Java, resident in the same application server that is supporting the PCASelect page, has been chosen for implementation. This avoids any cross-origin issues. This service must be deployed to an application server that is external to both Oracle Fusion CRM (SaaS) and PCA.

The REST service calls Java code that is resident in the same server. See the section Java Client Proxies for Web Services (Point 5 of the Integration Flow Diagram) for the details of that code.

To expose Java methods as REST services, the open source Jersey JAX-RS (`http://jersey.java.net/`) project is leveraged. Attached to this document is a zip file that contains the Java code, organized in an Oracle JDeveloper application. The Jersey .jar files are resident in the public_html/lib directory of the FCRMSalesPartyWS project. To extract the attached file named **PCAFCRMSalesPartyApp.zippy**:

1. Right click on the **Attachment** icon here.
2. Select **Save Embedded File to Disk**.
3. Locate the saved file. Change the file extension to .zip. Right click on the file and **Rename** the file from **PCAFCRMSalesPartyApp.zippy** to **PCAFCRMSalesPartyApp.zip**.
4. Extract the contents of the zipped file.

For the full listing of the REST Java service, see Appendix C: CreateFCRMSalesPartyREST.java.

Java Client Proxies for Web Services (Point 5 of the Integration Flow Diagram)

Model

Oracle Fusion CRM exposes SOAP web services for many business objects in Oracle Fusion CRM. The main business object of concern here is SalesParty. SalesParty contains all the information related to a customer. SalesParty has a flexible hierarchy of information that can be represented loosely as:

- SalesParty
    - OrganizationParty
        - OrganizationProfile
        - PartySite
            - ✓ PartySiteUse (contains a foreign-key reference to Location)
        - PartyUsageAssignment
    - SalesAccount
        - SalesAccountResource
        - OrganizationDetails

To create SalesParty, you must build a SOAP message that reflects this hierarchy.

If you create SalesParty without SalesAccount, it is classified as a Sales Prospect. To view Sales Prospect in the Oracle Fusion CRM user interface, go to a Real-Time search, which is sometimes not an intuitive step for casual users of Oracle Fusion CRM. To avoid that requirement, create any SalesParty with SalesAccount.

SalesAccount is required to have one SalesResource assigned. Therefore, create SalesResource, which is the PartyId of the sales representative who started the whole process. A Groovy script is used in the Oracle Fusion CRM **Postcode Anywhere Search** link (detailed in the Appendixes) to discover the PartyId of the sales representative in Oracle Fusion CRM and then pass that information through to the creation of SalesParty and SalesResource.

Also, note that the PartySiteUse node contains a foreign-key reference to Location. Therefore, you must pass the address information to LocationClient. LocationClient will call the Oracle Fusion CRM Location web service methods and return the LocationId of an existing Location with the same information, or the LocationId of a newly created Location. This new Location will be included in the SalesParty -- Organization Party -- PartySite node.

Java Web Service Proxies

To call the SalesParty and Location web services previously mentioned from Java, you must create Java web service proxies. The Oracle JDeveloper (JDEV) wizard is used to generate these proxies. The proxies help to invoke the various methods to prepare the JAXB elements that will be submitted as the SOAP message.

The Oracle Fusion CRM web services are very robust and provide a wide range of methods with which to manipulate the business objects. Because there are so many methods contained in the services, the JDEV wizard generates a large amount of code and files. Rather than include all of that auto-generated code in this document, you are encouraged to access the JDEV application that is included in this document and examine the content in the JDeveloper application.

However, the web services proxy client code is included in this document. This is code that has been written to create the SOAP messages and invoke the web services. The JDEV wizard creates an empty shell for the client. All the needed method calls are added to that shell. The JDEV naming convention for these shells is <ServiceName>SoapHttpPortClient.java. In this case, there are two clients called SalesPartyServiceSoapHttpPortClient.java and LocationServiceSoapHttpPortClient.java.

See the Appendixes for the complete listings for both of these clients.

Appendixes

Appendix A: PCASelect (Full Control of PCA Service Calls)

```
<html>
<head>
    <style>
    body {
      color:#0476B3;
      background-color:#E0E9F3;
      font-family:"Times New Roman", Times, serif;
      font-size:24px;
      }

    .ui-autocomplete {
                  position: absolute;
                  top: 100%;
                  left: 0;
                  z-index: 1000;
                  float: left;
                  display: none;
                   min-width: 160px;
                   padding: 20px 0;
                   margin: 2px 0 0 0;
                   list-style: none;
                   color: #0476B3;
                   font-size: 18px;
                   background-color: ##E0E9F3;
                   border-color: #ccc;
                   border-color: rgba(0, 0, 0, 0.2);
                   border-style: solid;
                   border-width: 1px;
                   -webkit-border-radius: 5px;
                   -moz-border-radius: 5px;
                   border-radius: 5px;
                   -webkit-box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
                   -moz-box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
                   box-shadow: 0 5px 10px rgba(0, 0, 0, 0.2);
                   -webkit-background-clip: padding-box;
                   -moz-background-clip: padding;
                   background-clip: padding-box;
                  *border-right-width: 2px;
                  *border-bottom-width: 2px;

                  .ui-menu-item > a.ui-corner-all {
                         display: block;
                         padding: 3px 15px;
                         clear: both;
                         font-weight: normal;
                         line-height: 14px;
                         color: ##0476B3;
                         font-color: ##0476B3;
                         white-space: nowrap;

                         &.ui-state-hover, &.ui-state-active {
                              color: ##0476B3;
                              text-decoration: none;
                              background-color: #0088cc;
                              border-radius: 0px;
                              -webkit-border-radius: 0px;
                              -moz-border-radius: 0px;
                              background-image: none;
                             }
                   }
            }
     </style>
    <script language="javascript" src="jquery-1.8.2.js"></script>
    <script src="jquery-ui.js"></script>
    <script type="text/javascript">

    $(function(){
    var pcaResultsArray = [{label: "",
                            value: ""}];

    function pcaRetrieveById(key,Id,j,pcaSearchTerm) {

        var url = 'https://services.postcodeanywhere.co.uk/CapturePlus/Interactive/RetrieveById/v2.00/json3.ws?Key=' + key +
'&Id=' + Id;
        // var url = "pcajsonRetrieveById_" + j + ".ws"; // this calls pre-fetched results to avoid using PCA credits during
testing.

        var jqxhr = $.ajax({url:url,
                        type: "get",
                        dataType: 'json'
                       })
         .done  (function(response){})
         .fail  (function (XHR, textStatus, errorThrown) {
                                    $('.pcaResult').append(XHR);
                                    $('.pcaResult').append(textStatus);
                                    $('.pcaResult').append(errorThrown);
                                   })
         .always(function(response){
                               // initialize variables in case they have been set before
                               var pcaId              ='';
                               var pcaCompany         ='';
                               var pcaLine1           ='';
                               var pcaLine2           ='';
                               var pcaCity            ='';
                               var pcaProvinceCode    ='';
                               var pcaProvinceName    ='';
                               var pcaCountryCode     ='';
                               var pcaPostalCode      ='';
                               var pcaPOBox           ='';

                               for (var i = 0; i < response.Items.length; i++) {
                                    var Item = response.Items[i];

                                    switch (Item.FieldName)
                                        {
                                        case "Id":
                                                            pcaId = Item.FormattedValue;
                                                break;
                                        case "Company":
                                                            pcaCompany = Item.FormattedValue;
                                                break;
                                        case "Line1":
                                                            pcaLine1 = Item.FormattedValue;
                                                break;
                                        case "Line2":
                                                            pcaLine2 = Item.FormattedValue;
                                                break;
                                        case "City":
                                                            pcaCity = Item.FormattedValue;
                                                break;
                                        case "ProvinceCode":
                                                            pcaProvinceCode = Item.FormattedValue;
                                                break;
                                        case "ProvinceName":
                                                            pcaProvinceName = Item.FormattedValue;
                                                break;
                                        case "CountryCode":
                                                            pcaCountryCode = Item.FormattedValue;
                                                break;
                                        case "PostalCode":
                                                            pcaPostalCode = Item.FormattedValue;
                                                break;
                                        case "POBox":
                                                            pcaPOBox = Item.FormattedValue;
                                                break;
                                        default:
                                                break;
                                       } // end of switch

                                 }  // end of for loop

                                 // $('.pcaResult').append(outputline);
                                 var outputline  = "";
                                     if (pcaCompany == null || pcaCompany == "" ){
                                       outputline += pcaSearchTerm;
                                     } else {
                                       outputline += pcaCompany;
                                     }
                                     outputline += ":";
                                     outputline += pcaLine1;
                                     outputline += ":";
                                     outputline += pcaCity;
                                     outputline += ":";
                                     outputline += pcaProvinceName;
                                     outputline += ":";
                                     outputline += pcaPostalCode;
                                     outputline += ":";
                                     outputline += pcaCountryCode;

                                   pcaResultsArray[j] = { label: outputline, value: outputline };

                                 });  // end of always and jqxhr

     }  // end of pcaRetrievById

     function pcaSearch (key,SearchTerm) {

        var url = "https://services.postcodeanywhere.co.uk/CapturePlus/Interactive/AutoComplete/v2.00/json3.ws";
        url += "?Key=" + key;
        url += "&Country=US";
        url += "&SearchTerm=" + SearchTerm;
```

```
        // url = 'pcajson.ws';  // this calls a pre-fetched result during testing to avoid using PCA credits. Overrides pca url
above when uncommented.

        var jqxhr = $.ajax({url: url,
                            type: "get",
                        dataType: 'json'
                       })
         .done  (function(response){})
         .fail  (function (XHR, textStatus, errorThrown) {
                                    $('.pcaResult').append(XHR);
                                    $('.pcaResult').append(textStatus);
                                    $('.pcaResult').append(errorThrown);
                                   })
         .always(function(response){
                              // pcaResultsArray = [{label:"Search Item Not Found"
                              //                    , value:"Search Item Not Found"
                              //                    }];
                               var itemsLength = response.Items.length;
                               itemsLength = itemsLength - 1;
                               for (var i = 1; i < itemsLength; i++) { //starting from 1 because 0 has error in test file switch
to 0 in prod
                                      var Item = response.Items[i];
                                      pcaRetrieveById(key,Item.Id,i,SearchTerm);
                                   }  // end of for loop
                               }); // end of always and var jqxhr
     }  // end of pcaSearch

        $("#pcaResults").autocomplete({
                       //minLength : 4,
                           // delay : 3,
                           scroll: true,
                         position: { collision: "flip" },
                          source : function(request, response) {
                            pcaSearch('KY95-DJ54-TT62-XN69', request.term);
                            for (var k = 0; k < pcaResultsArray.length; k++) { //starting from 1 because 0 has error in test file
switch to 0 in prod
                               if (pcaResultsArray == "") {
                                   pcaResultsArray[k] = [null];
                                } // end of if
                            }  // end of for loop
                            response(pcaResultsArray);
                          }  // end of source function
                       });      //end of autocomplete
     });                          // end of outer function

       function pcaImportIntoFcrm(form,pFcrmUserID) {

         var pca = form.pcaResults.value;
         var pcastrs = pca.split(':');

         var pcaCompany      = pcastrs[0];
         var pcaLine1        = pcastrs[1];
         var pcaLine2        = pcastrs[2];
         var pcaLine3        = pcastrs[3];
         var pcaCity         = pcastrs[4];
         var pcaProvinceName = pcastrs[5];
         var pcaPostalCode   = pcastrs[6];
         var pcaCountryCode  = pcastrs[7];

         var url = "";
         url += 'http://127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-context-root/jersey/createFCRMSalesParty';
         url +=  "?pPartyName=";
         url +=   pcaCompany;
         url += "&pAddress1=" ;
         url +=   pcaLine1;
         url += "&pAddress2=" ;
         url +=   pcaLine2;
         url += "&pAddress3=" ;
         url +=   pcaLine3;
         url += "&pCity=";
         url +=   pcaCity;
         url += "&pState=";
         url +=   pcaProvinceName;
         url += "&pPostalCode=";
         url +=   pcaPostalCode;
         url += "&pCountry=";
         url += "US";  // hardcoding to US for now because pcaCountryCode returns USA.
         url += "&pFcrmUserID=";
         url +=   pFcrmUserID;

         var jqXHR = $.ajax({
                        type: 'GET',
                         url: url,
                    dataType: 'html',
                  beforeSend:function (XHR){
                              XHR.setRequestHeader("Access-Control-Allow-Origin", "*");
                              }
                   })
         .done  (function(response){})
         .fail  (function (XHR, textStatus, errorThrown) {
                                    $('.pcaResult').append(XHR);
                                    $('.pcaResult').append(textStatus);
                                    $('.pcaResult').append(errorThrown);
                                   })
         .always(function(response){
                                  $('.pcaResult').append(response);
                                   }); // end of always and var jqxhr
       }

    </script>
</head>

<body onload="resizeTo(1200,400);moveTo(200,200);">

<script type="text/javascript">
           var searchTermURL           = document.URL;
           var searchTermURLsplit      = searchTermURL.split("?");
           var searchTermURLsplitAgain = searchTermURLsplit[1].split("=");
           var pFcrmUserID             = searchTermURLsplitAgain[1];
</script>
<div id="container">
   <FORM NAME="pcaSearchTermForm" ACTION="" METHOD="GET">
    <INPUT TYPE="button" NAME="button" Value="Import into Oracle FCRM" onClick="pcaImportIntoFcrm(this.form,pFcrmUserID)">
    <label for="pcaResults">Enter Postcode Anywhere Search Term: </label>
    <p>
    <input id="pcaResults" size=75/>
   </FORM>
</div>

<div id="pcaResult" class="pcaResult">
</div>

</body>
</html>
```

Appendix B: PCASelect (Surrendering Control of PCA Calls to PCA JavaScript Function)

```
<!DOCTYPE html>
<html>
    <head>
        <meta content=""/>
        <title>Capture+</title>
        <meta http-equiv="X-UA-Compatible" content="IE=9" />

        <script language="javascript" src="jquery-1.8.2.js"   type="text/javascript"></script>
        <script src="jquery-ui.js"></script>
        <script type="text/javascript" src="https://www.postcodeanywhere.co.uk/scripts/main-1-0.js"></script>

        <style type="text/css">

            body {
                           color:#0476B3;
                background-color:#E0E9F3;
                     font-family:"Times New Roman", Times, serif;
                       font-size:20px;
                 }

            .ui-helper-clearfix:after
            {
                content: inherit;
            }
            .ui-datepicker
            {
                padding: 0px;
            }
            .pcaAutoComplete .inputArea .search
            {
                top: 1px;
            }
        </style>
    </head>
<body onload="resizeTo(850,650);moveTo(200,200);">

<script type="text/javascript">
    function CapturePlusCallback(uid, response) {

        var addressLabel = "";
        if (document.getElementById('CompanyParts').value != "") {
            addressLabel += document.getElementById('CompanyParts').value + "\n";
        }
        if (document.getElementById('Line1Parts').value != "") {
            addressLabel += document.getElementById('Line1Parts').value + "\n";
        }
        if (document.getElementById('Line2Parts').value != "") {
            addressLabel += document.getElementById('Line2Parts').value + "\n";
        }
        if (document.getElementById('Line3Parts').value != "") {
            addressLabel += document.getElementById('Line3Parts').value + "\n";
        }
        if (document.getElementById('TownParts').value != "TownParts") {
            addressLabel += document.getElementById('TownParts').value + "\n";
        }
        if (document.getElementById('CountyParts').value != "CountyParts") {
            addressLabel += document.getElementById('CountyParts').value + "\n";
        }
        if (document.getElementById('PostcodeParts').value != "") {
            addressLabel += document.getElementById('PostcodeParts').value + "\n";
        }
        if (document.getElementById('CountryParts').value != "CountryParts") {
            addressLabel += document.getElementById('CountryParts').value + "\n";
        }
        document.getElementById('AddressLabel').value = addressLabel;
    }

       function pcaImportIntoFcrm(form,pFcrmUserID) {
         var url = "";
         url += 'http://127.0.0.1:7101/FCRMSalesPartyApp2-FCRMSalesPartyWS-context-root/jersey/createFCRMSalesParty';
         url +=  "?pPartyName=";
         url +=   document.getElementById('CompanyParts').value;
         url += "&pAddress1=" ;
         url +=   document.getElementById('Line1Parts').value;
         url += "&pAddress2=" ;
         url +=   document.getElementById('Line2Parts').value;
         url += "&pAddress3=" ;
         url +=   document.getElementById('Line3Parts').value;
         url += "&pCity=";
         url +=   document.getElementById('TownParts').value;
         url += "&pState=";
         url +=   document.getElementById('CountyParts').value;
         url += "&pPostalCode=";
         url +=   document.getElementById('PostcodeParts').value;
         url += "&pCountry=";
         //url +=  document.getElementById('CountryParts').value;
         url += "US"  // This is hardcoded until we get the country passed.
         url += "&pFcrmUserID=";
         url +=   pFcrmUserID;

         var jqXHR = $.ajax({
                        type: 'GET',
                         url: url,
                    dataType: 'html',
                  beforeSend:function (XHR){
                              XHR.setRequestHeader("Access-Control-Allow-Origin", "*");
                              }
                   })
         .done  (function(response){})
         .fail  (function (XHR, textStatus, errorThrown) {
                                    $('.pcaResult').append(XHR);
                                    $('.pcaResult').append(textStatus);
                                    $('.pcaResult').append(errorThrown);
                                   })
         .always(function(response){
                                  $('.pcaResult').append(response);
                                   }); // end of always and var jqxhr
       }

</script>

<style type="text/css">

.formRow .control
{
    width:250px;
}
.pcaAutoComplete .inputArea .search {
    height: 30px;
}
</style>
<div class="clear h20"></div>
<div class="box">
    <div id="container">
       <script type="text/javascript">
          var searchTermURL            = document.URL;
          var searchTermURLsplit       = searchTermURL.split("?");
          var searchTermURLsplitAgain = searchTermURLsplit[1].split("=");
          var pFcrmUserID              = searchTermURLsplitAgain[1];
       </script>
     <form NAME="pcaSearchTermForm" ACTION="" METHOD="GET">
       <INPUT TYPE="button" NAME="button" Value="Import into Oracle Fusion CRM"
onClick="pcaImportIntoFcrm(this.form,pFcrmUserID)">
    </form>
    </div>
    <div id="pcaResult" class="pcaResult"></div>

    <div class="formRow Top">
    <link rel="stylesheet" type="text/css" href="http://services.postcodeanywhere.co.uk/css/captureplus-1.34.min.css?key=UA93-
NT23-JC37-GD2" />
        <script type="text/javascript" src="https://services.postcodeanywhere.co.uk/js/captureplus-1.31.js?key=UA93-NT23-JC37-
GD23&app=7649"></script>
        <div style="width:690px; margin-left:auto; margin-right:auto; margin-top:14px; margin-bottom:40px;">
        <div id="ua93nt23jc37gd237649"></div>
        </div>
    </div>

    <div style="width:450px; float:left;">
    <div class="formRow Top">
                <label for="CompanyParts">
                    Company</label>
                <div class="control">
                    <input id="CompanyParts" name="CompanyParts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                <label for="Line1Parts">
                    Line1</label>
                <div class="control">
                    <input id="Line1Parts" name="Line1Parts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                <label for="Line2Parts">
                    Line2</label>
                <div class="control">
                    <input id="Line2Parts" name="Line2Parts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                <label for="Line3Parts">
                    Line3</label>
                <div class="control">
                    <input id="Line3Parts" name="Line3Parts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                <label for="TownParts">
                    Town</label>
                <div class="control">
                    <input id="TownParts" name="TownParts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                <label for="CountyParts">
                    State/Province</label>
                <div class="control">
                    <input id="CountyParts" name="CountyParts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
            <div class="formRow">
                 <label for="PostcodeParts">
                        Postcode</label>
                    <div class="control">
                        <input id="PostcodeParts" name="PostcodeParts" type="text" value="" />
                    </div>
                <div class="clear h0">
                </div>
            </div>

            <div class="clear h0">
            </div>

            <div class="formRow">
                <label for="CountryParts">
                    Country</label>
                <div class="control">
                    <input id="CountryParts" name="CountryParts" type="text" value="" />
                </div>

                <div class="clear h0">
                </div>
            </div>
</div>
            <div style="float:left; width:300px; height:369px; border-bottom: 1px solid #d7d7d7;">
                <textarea id="AddressLabel" style="margin-left: 27px; margin-top: 20px; height: 290px; text-align:center; font-
size:14px; line-height:20px;"></textarea>
                <div class="clear h0"></div>
            </div>

</div>

</body>
</html>
```

Appendix C: CreateFCRMSalesPartyREST.java

```
package createFCRMSalesParty;

import FCRMSalesParty.SalesPartyServiceSoapHttpPortClient;
import FCRMSalesParty.ServiceException;
import java.util.*;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;

@Path("/createFCRMSalesParty")
public class CreateFCRMSalesPartyREST {
    @GET
    @Produces("text/plain")
    //Read
    public  String createFCRMSalesParty(@QueryParam("pPartyName")   String pPartyName
                                       ,@QueryParam("pAddress1")      String pAddress1
                                       ,@QueryParam("pAddress2")      String pAddress2
                                       ,@QueryParam("pAddress3")      String pAddress3
                                       ,@QueryParam("pCity")              String pCity
                                       ,@QueryParam("pState")             String pState
                                       ,@QueryParam("pPostalCode")   String pPostalCode
                                       ,@QueryParam("pCountry")        String pCountry
                                       ,@QueryParam("pFcrmUserID") String pFcrmUserID
                                       )
    {
        String salesPartyClientResponse = new String();
        try {salesPartyClientResponse =
             SalesPartyServiceSoapHttpPortClient.SalesPartyClient(pPartyName, pAddress1,               pAddress2, pAddress3,
pCity, pState, pPostalCode, pCountry, pFcrmUserID);
        } catch (ServiceException e) {
             e.printStackTrace();
        }
 return ("The Sales Party: \"" + pPartyName + "\", is in Oracle Fusion CRM with Sales Registry ID: " +  salesPartyClientResponse
+ ".");
    }
}
```

Appendix D: LocationServiceSoapHttpPortClient.java

```
package proxy;

import com.sun.xml.ws.api.addressing.AddressingVersion;
import com.sun.xml.ws.api.addressing.WSEndpointReference;
import com.sun.xml.ws.developer.WSBindingProvider;
import com.sun.xml.ws.message.StringHeader;

import java.net.Authenticator;

import java.util.List;
import java.util.Map;
import java.util.UUID;

import javax.xml.bind.JAXBElement;
import javax.xml.ws.WebServiceRef;

import proxy.types.Conjunction;
import proxy.types.FindControl;
import proxy.types.FindCriteria;
import proxy.types.Location;
import proxy.types.LocationInformation;
import proxy.types.LocationResult;
import proxy.types.ObjectFactory;
import proxy.types.ViewCriteria;

import proxy.types.ViewCriteriaItem;
import proxy.types.ViewCriteriaRow;

import weblogic.wsee.jws.jaxws.owsm.SecurityPoliciesFeature;
// !THE CHANGES MADE TO THIS FILE WILL BE DESTROYED IF REGENERATED!
// This source file is generated by Oracle tools
// Contents may be subject to change
// For reporting problems, use the following
// Version = Oracle WebServices (11.1.1.0.0, build 111209.0821.28162)

public class LocationServiceSoapHttpPortClient
{
  @WebServiceRef
  private static LocationService_Service locationService_Service;

  static String pAddress1;
  static String pAddress2;
  static String pAddress3;
  static String pCity;
  static String pState;
  static String pPostalCode;
  static String pCountry;

    public static String LocationClient(
                                     String pAddress1
                                    ,String pAddress2
                                    ,String pAddress3
                                    ,String pCity
                                    ,String pState
                                    ,String pPostalCode
                                    ,String pCountry) throws ServiceException {

        System.out.println(pAddress1 + " " + pCity + " " + pState + " " + pPostalCode + " " + pCountry);

    locationService_Service = new LocationService_Service();
      // Configure security feature
      SecurityPoliciesFeature securityFeatures =
          new SecurityPoliciesFeature(new String[] { "oracle/wss_username_token_over_ssl_client_policy" });
    LocationService locationService = locationService_Service.getLocationServiceSoapHttpPort(securityFeatures);

    // Add your code to call the desired methods.
    MyAuthenticator myAuth = new MyAuthenticator("matt.hooper@infusion.com","XeJcome10");
    Authenticator.setDefault(myAuth);

    // Get the request context to set the outgoing addressing properties
    WSBindingProvider wsbp = (WSBindingProvider)locationService;

    Map<String, Object> requestContext = wsbp.getRequestContext();

    //Map to appropriate Fusion user ID, no need to provide password with SAML authentication
    requestContext.put(WSBindingProvider.USERNAME_PROPERTY, "matt.hooper@infusion.com");
    requestContext.put(WSBindingProvider.PASSWORD_PROPERTY, "XeJcome10");

    try {

        FindCriteria findCriteria = new FindCriteria();

        findCriteria.setFetchStart(0);
        findCriteria.setFetchSize(-1);

        // Here is the filter to retrieve Locations that match the input
        ViewCriteria vc;
        ViewCriteriaRow vcr;
        ViewCriteriaItem vci;

        // a Filter is a Group including ViewCriteria
        // <ns2:filter>
        //    <ns2:conjunction>And</ns2:conjunction>
        vc = new ViewCriteria();
        vc.setConjunction(Conjunction.fromValue("And"));
        // a ViewCriteria is made of ViewCriteriaRow(s)
        //    <ns2:group>
        //      <ns2:conjunction>And</ns2:conjunction>
        //      <ns2:upperCaseCompare>false</ns2:upperCaseCompare>
        vcr = new ViewCriteriaRow();
        vcr.setConjunction(Conjunction.fromValue("And"));
        vcr.setUpperCaseCompare(true);
        // a ViewCriteriaRow is made of ViewCriteriaItem(s)
        //
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>Address1</typ1:attribute>
        //   <typ1:operator>=</typ1:operator>
        //   <typ1:value>Infusion St.</typ1:value>
        //</typ1:item>
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>City</typ1:attribute>
        //   <typ1:operator>=</typ1:operator>
        //   <typ1:value>Liberty Corner</typ1:value>
        //</typ1:item>
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>State</typ1:attribute>
        // <typ1:operator>=</typ1:operator>
        //   <typ1:value>NJ</typ1:value>
        //</typ1:item>
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>PostalCode</typ1:attribute>
        //   <typ1:operator>=</typ1:operator>
        //   <typ1:value>07938</typ1:value>
        //</typ1:item>
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>Country</typ1:attribute>
        //   <typ1:operator>=</typ1:operator>
        //   <typ1:value>US</typ1:value>
        //</typ1:item>

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("Address1");
        vci.setOperator("=");
        vci.getValue().add(pAddress1);
        vcr.getItem().add(vci);
/*
        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("Address2");
        vci.setOperator("=");
        vci.getValue().add(pAddress2);
        vcr.getItem().add(vci);

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("Address3");
        vci.setOperator("=");
        vci.getValue().add(pAddress3);
        vcr.getItem().add(vci);
 */
        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("City");
        vci.setOperator("=");
        vci.getValue().add(pCity);
        vcr.getItem().add(vci);

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("State");
        vci.setOperator("=");
        vci.getValue().add(pState);
        vcr.getItem().add(vci);

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("PostalCode");
        vci.setOperator("=");
        vci.getValue().add(pPostalCode);
        vcr.getItem().add(vci);

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("Country");
        vci.setOperator("=");
        vci.getValue().add(pCountry);
        vcr.getItem().add(vci);

        vc.getGroup().add(vcr);
        findCriteria.setFilter(vc);

        findCriteria.setExcludeAttribute(false);

        FindControl findControl = new FindControl();
        findControl.setRetrieveAllTranslations(false);

        // first we will try to see if a Location with these attributes already exists.
        LocationResult findLocation = locationService.findLocation(findCriteria, findControl);
        List<Location> locationList = findLocation.getValue();

        if (locationList.size() == 0) {  // no locations with these criteria found, so create a new Location
            System.out.println("No Locations exist already. Proceeding to create new Location");

            Location locationToBeCreated = new Location();
            ObjectFactory factory = new ObjectFactory();

            locationToBeCreated.setAddress1(pAddress1);
            locationToBeCreated.setAddress2(factory.createLocationAddrElementAttribute2(pAddress2));
            locationToBeCreated.setAddress3(factory.createLocationAddrElementAttribute2(pAddress3));
            locationToBeCreated.setCity(factory.createLocationCity(pCity));
            locationToBeCreated.setState(factory.createLocationState(pState));
            locationToBeCreated.setPostalCode(factory.createLocationPostalCode(pPostalCode));
            locationToBeCreated.setCountry(pCountry);
            locationToBeCreated.setCreatedByModule(factory.createLocationCreatedByModule("HZ_IMPORT"));

            // call method createLocation
            LocationResult returnedLocationResult = locationService.createLocation(locationToBeCreated);

            List<Location> returnedLocationList = returnedLocationResult.getValue();
            Location returnedLocation = returnedLocationList.get(0);  // List should only have one element

            System.out.println("Location: LocationId is: " + returnedLocation.getLocationId().toString());
            return returnedLocation.getLocationId().toString();

        }  // end of if no locations with these criteria found
        {  // else at least one Location was found. We will use the first Location in the returned List
            System.out.println("Locations exist already. We will use the first Location found.");
            Location location = locationList.get(0);
            System.out.println("Location: LocationId is: " + location.getLocationId().toString());
            return location.getLocationId().toString();
        } // end of else
    } catch (ServiceException e) {
        System.out.println(e.getMessage());
    }

    System.out.println ("Error with Location. We should not get here"); // designates error
    return "00000000";
    }

    public static void main(String[] args) throws Exception {
        Object locationClient;
        System.out.println("Getting ready to call LocationClient");
        String LocationClientOut = LocationClient("Infusion St.","Infusion Plaza","Infusion Business
District","Newark","NJ","07938","DE");
        System.out.println(LocationClientOut);
        }
  }
```

Appendix E: SalesPartyServiceSoapHttpPortClient.java

```
package FCRMSalesParty;

import com.oracle.xmlns.adf.svc.types.Conjunction;

import com.sun.xml.ws.api.addressing.AddressingVersion;
import com.sun.xml.ws.api.addressing.WSEndpointReference;
import com.sun.xml.ws.developer.WSBindingProvider;
import com.sun.xml.ws.message.StringHeader;

import com.oracle.xmlns.adf.svc.types.FindControl;
import com.oracle.xmlns.adf.svc.types.FindCriteria;
import com.oracle.xmlns.adf.svc.types.ViewCriteria;
import com.oracle.xmlns.adf.svc.types.ViewCriteriaItem;
import com.oracle.xmlns.adf.svc.types.ViewCriteriaRow;
import com.oracle.xmlns.apps.cdm.foundation.parties.organizationservice.OrganizationParty;
import com.oracle.xmlns.apps.cdm.foundation.parties.organizationservice.OrganizationPartyResult;
import com.oracle.xmlns.apps.cdm.foundation.parties.organizationservice.OrganizationProfile;
import com.oracle.xmlns.apps.cdm.foundation.parties.partyservice.PartySite;
import com.oracle.xmlns.apps.cdm.foundation.parties.partyservice.PartySiteUse;
import com.oracle.xmlns.apps.cdm.foundation.parties.partyservice.PartyUsageAssignment;
import com.oracle.xmlns.apps.crmcommon.salesparties.salespartiesservice.ObjectFactory;
import com.oracle.xmlns.apps.crmcommon.salesparties.salespartiesservice.SalesAccount;
import com.oracle.xmlns.apps.crmcommon.salesparties.salespartiesservice.SalesAccountResource;
import com.oracle.xmlns.apps.crmcommon.salesparties.salespartiesservice.SalesParty;

import com.sun.xml.ws.developer.WSBindingProvider;

import proxy.*;

import java.io.FileOutputStream;

import java.net.Authenticator;

import java.util.List;
import java.util.Map;
import java.util.UUID;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBElement;
import javax.xml.bind.Marshaller;
import javax.xml.ws.WebServiceRef;


import weblogic.wsee.jws.jaxws.owsm.SecurityPoliciesFeature;
// This source file is generated by Oracle tools.
// Contents may be subject to change.
// For reporting problems, use the following:
// Generated by Oracle JDeveloper 11g Release 2 11.1.2.3.0.6276

public class SalesPartyServiceSoapHttpPortClient {
    public static String SalesPartyClient(
     String pPartyName
    ,String pAddress1
    ,String pAddress2
    ,String pAddress3
    ,String pCity
    ,String pState
    ,String pPostalCode
    ,String pCountry
    ,String pFcrmUserID) throws ServiceException {
        SalesPartyService_Service salesPartyService_Service = new SalesPartyService_Service();

        // Configure security feature
        SecurityPoliciesFeature securityFeatures =
            new SecurityPoliciesFeature(new String[] { "oracle/wss_username_token_over_ssl_client_policy" });
        SalesPartyService salesPartyService =
            salesPartyService_Service.getSalesPartyServiceSoapHttpPort(securityFeatures);

        FCRMSalesParty.MyAuthenticator myAuth = new FCRMSalesParty.MyAuthenticator("matt.hooper@infusion.com","XeJcome10");
        Authenticator.setDefault(myAuth);

        // Get the request context to set the outgoing addressing properties
        WSBindingProvider wsbp = (WSBindingProvider)salesPartyService;

        Map<String, Object> requestContext = wsbp.getRequestContext();

        //Map to appropriate Fusion user ID, no need to provide password with SAML authentication
        requestContext.put(WSBindingProvider.USERNAME_PROPERTY, "matt.hooper@infusion.com");
        requestContext.put(WSBindingProvider.PASSWORD_PROPERTY, "XeJcome10");

        // Add your code to call the desired methods.

        FindCriteria findCriteria = new FindCriteria();

        findCriteria.setFetchStart(0);
        findCriteria.setFetchSize(-1);
        findCriteria.setExcludeAttribute(false);
        // Here is the filter to retrieve records that match the input

        ViewCriteria vc;
        ViewCriteriaRow vcr;
        ViewCriteriaItem vci;

        // a Filter is a Group including ViewCriteria
        // <ns2:filter>
        //    <ns2:conjunction>And</ns2:conjunction>
        vc = new ViewCriteria();
        vc.setConjunction(Conjunction.fromValue("And"));
        // a ViewCriteria is made of ViewCriteriaRow(s)
        //    <ns2:group>
        //      <ns2:conjunction>And</ns2:conjunction>
        //      <ns2:upperCaseCompare>false</ns2:upperCaseCompare>
        vcr = new ViewCriteriaRow();
        vcr.setConjunction(Conjunction.fromValue("And"));
        vcr.setUpperCaseCompare(true);
        // a ViewCriteriaRow is made of ViewCriteriaItem(s)
        //
        //<typ1:item>
        //   <typ1:conjunction>And</typ1:conjunction>
        //   <typ1:upperCaseCompare>true</typ1:upperCaseCompare>
        //   <typ1:attribute>PartyName</typ1:attribute>
        //   <typ1:operator>=</typ1:operator>
        //   <typ1:value>CJV International</typ1:value>

        vci = new ViewCriteriaItem();
        vci.setConjunction(Conjunction.fromValue("And"));
        vci.setUpperCaseCompare(true);
        vci.setAttribute("PartyName");
        vci.setOperator("=");
        vci.getValue().add(pPartyName);
        vcr.getItem().add(vci);

        vc.getGroup().add(vcr);
        findCriteria.setFilter(vc);

        findCriteria.setExcludeAttribute(false);

        FindControl findControl = new FindControl();
        findControl.setRetrieveAllTranslations(false);

        try {
                List<SalesParty> salesList = salesPartyService.findSalesParty(findCriteria,findControl);

                if (salesList.size() == 0) {  // SalesParty does not already exist, so create a new one.

                System.out.println("SalesParty not found so we will now proceed to create SalesParty");

                  // Create Object Factories
                  // Create an Object Factory for organizations
                  com.oracle.xmlns.apps.cdm.foundation.parties.organizationservice.ObjectFactory
                  organizationFactory = new com.oracle.xmlns.apps.cdm.foundation.parties.organizationservice.ObjectFactory ();

                  // Create an Object Factory for Parties - will be important for Party Sites
                  com.oracle.xmlns.apps.cdm.foundation.parties.partyservice.ObjectFactory
                  partyFactory = new com.oracle.xmlns.apps.cdm.foundation.parties.partyservice.ObjectFactory();

                  // Create an Object Factory for SalesParties.
                  // It is currently found in com.oracle.xmlns.apps.crmcommon.salesparties.salespartiesservice
                  ObjectFactory salesPartyFactory = new ObjectFactory();


                  // create an Organization Profile
                  OrganizationProfile organizationProfileToBeCreated = new OrganizationProfile();
                  organizationProfileToBeCreated.setOrganizationName(pPartyName);

organizationProfileToBeCreated.setCreatedByModule(organizationFactory.createOrganizationProfileCreatedByModule("HZ_IMPORT"));

                  // create a PartySite
                  PartySite partySiteToBeCreated = new PartySite();

                  // Party Sites need a location created by CreateLocation
                  String locationId = LocationServiceSoapHttpPortClient.LocationClient(pAddress1, pAddress2, pAddress3, pCity,
pState, pPostalCode, pCountry);
                  if (locationId == "000000") {
                      System.out.println("From SalesPartyClient: Call to LocationClient had errors, stopping ...");
                      return "00000000";
                  }
                  System.out.println("SalesParty: LocationId is: " + locationId);

                  Long locationIdLong = new Long(locationId);
                  partySiteToBeCreated.setLocationId(locationIdLong);
                  partySiteToBeCreated.setCreatedByModule(partyFactory.createPartySiteCreatedByModule("HZ_IMPORT"));
                  partySiteToBeCreated.setPartySiteType(partyFactory.createPartySiteUseSiteUseType("SELL_TO"));

                  // create a PartySiteUse
                  PartySiteUse partySiteUseToBeCreated = new PartySiteUse();
                  partySiteUseToBeCreated.setCreatedByModule(partyFactory.createPartySiteCreatedByModule("HZ_IMPORT"));
                  partySiteUseToBeCreated.setSiteUseType(partyFactory.createPartySiteUseSiteUseType("SELL_TO"));

                  // add the PartySiteUse to the PartySite
                  partySiteToBeCreated.getPartySiteUse().add(partySiteUseToBeCreated);

                  // create a Party Usage
                  PartyUsageAssignment partyUsageAssignmentToBeCreated = new PartyUsageAssignment ();
                  partyUsageAssignmentToBeCreated.setPartyUsageCode("SALES_ACCOUNT");
                  partyUsageAssignmentToBeCreated.setStatusFlag(true);
                  partyUsageAssignmentToBeCreated.setCreatedByModule("HZ_IMPORT");

                  // we need to create a Sales Account
                  SalesAccount salesAccountToBeCreated = new SalesAccount();

                  // we need to create a Sales Account Resource
                  SalesAccountResource salesAccountResourceToBeCreated = new SalesAccountResource();

                  Long resourceIdToBeCreated = new Long(pFcrmUserID);

salesAccountResourceToBeCreated.setResourceId(salesPartyFactory.createSalesAccountResourceResourceId(resourceIdToBeCreated));
                  salesAccountResourceToBeCreated.setLockAssignmentFlag(true);

salesAccountResourceToBeCreated.setAssignmentTypeCode(salesPartyFactory.createSalesAccountResourceAssignmentTypeCode("MANUAL"));

                  // add the Sales Account Resource to the Sales Account
                  salesAccountToBeCreated.getSalesAccountResource().add(salesAccountResourceToBeCreated);

                  // we need to create an Organization Party

                  OrganizationParty organizationPartyToBeCreated = new OrganizationParty();

organizationPartyToBeCreated.setCreatedByModule(organizationFactory.createOrganizationPartyCreatedByModule("HZ_IMPORT"));
                  organizationPartyToBeCreated.getOrganizationProfile().add(organizationProfileToBeCreated);
                  organizationPartyToBeCreated.getPartySite().add(partySiteToBeCreated);
                  organizationPartyToBeCreated.getPartyUsageAssignment().add(partyUsageAssignmentToBeCreated);

                  SalesParty salesPartyToBeCreated = new SalesParty();
                  salesPartyToBeCreated.setPartyName(pPartyName);  // not sure if this can be set here.tbd.
                  salesPartyToBeCreated.setStatus("A");
                  salesPartyToBeCreated.setPartyType("ORGANIZATION");

                  salesPartyToBeCreated.getOrganizationParty().add(organizationPartyToBeCreated);
                  salesPartyToBeCreated.getSalesAccount().add(salesAccountToBeCreated);

                  // call method createSalesParty
                  System.out.println(salesPartyToBeCreated);

                  SalesParty returnedSalesParty = salesPartyService.createSalesParty(salesPartyToBeCreated);

                  String returnedPartyNumber = returnedSalesParty.getOrganizationParty().get(0).getPartyNumber();
                  // return the SalesParty Id just created
                  System.out.println(returnedPartyNumber);
                  return returnedPartyNumber;
                  }
                  {     // else SalesParty already exists
                 System.out.println("SalesParty found so we will not create SalesParty. For now we will report Party Number of
first Sales party found");
                 // Iterating through SalesParty List
                 for (int i = 0; i < 1; i++) {  // just get the first SalesParty in list
                    SalesParty salesParty = salesList.get(i);

                    System.out.println(salesParty.getPartyNumber());
                    return salesParty.getPartyNumber();
                 }  // end of for
            }  // end of else

         } catch (Exception ex) {
       ex.printStackTrace();
     }
        System.out.println("Sales Party Error, we shouldn't get to here.");
        return "00000000";  // 00000000 denotes error
    }

    public static void main(String[] args) throws Exception {
        System.out.println("Getting ready to call SalesPartyClient");
        String SalesPartyClientOut = SalesPartyClient("CJV International Arkansas HQ","Infusion St.","Infusion Plaza","Infusion
Business District","Liberty Corner","AR","07938","US","300000001398160");
        System.out.println(SalesPartyClientOut);
        }
}
```