Contents

[Overview 3](#_Toc388616887)

[Assumptions 4](#_Toc388616888)

[Environment 4](#_Toc388616889)

[Instructions 5](#_Toc388616890)

[JDeveloper Application 5](#_Toc388616891)

[Database schema 5](#_Toc388616892)

[Datasource configuration 5](#_Toc388616893)

[Oracle Sales Cloud connection Configuration 6](#_Toc388616894)

[Create credential for Oracle Sales Cloud connection in JCS credential store 7](#_Toc388616895)

[Build and deploy the DataSync application 10](#_Toc388616896)

[Build the ear application 10](#_Toc388616897)

[Deploy the DataSync application into JCS 13](#_Toc388616898)

[Build the Phramcy ear application 18](#_Toc388616899)

[Deploy the application into JCS 21](#_Toc388616900)

[Oracle Sales Cloud Custom Objects 26](#_Toc388616901)

[Custom Object: Pres\_c 26](#_Toc388616902)

[Custom Object: Pres\_Med\_c 27](#_Toc388616903)

[Custom Object: Med\_c 28](#_Toc388616904)

[Steps to create a custom object 30](#_Toc388616905)

[Import Oracle Sales Cloud certificate 34](#_Toc388616906)

[Import JCS certificate 55](#_Toc388616907)

[Test the data synchronization 55](#_Toc388616908)

[Attribute Mapping 55](#_Toc388616909)

[Create Customer in Oracle Sales Cloud 56](#_Toc388616910)

[Set Customer Id 58](#_Toc388616911)

[Create a prescription in JCS 71](#_Toc388616912)

[Send a prescription to the pharmacy 75](#_Toc388616913)

[Scheduler 80](#_Toc388616914)

[Customization 81](#_Toc388616915)

[Mapping 81](#_Toc388616916)

[Data Synchronization 82](#_Toc388616917)

**Disclaimer**

All sample code is provided by Oracle for illustrative purposes only.

These sample code examples have not been thoroughly tested under all conditions. Oracle, therefore, cannot guarantee or imply security, reliability, serviceability, or function of the sample code.

All sample code contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

REST services built using REST stacks like Jersey JAX-RS can be secured using standard JEE or OWSM Policies. While there are technical options for securing REST services, each with its pros and cons, Oracle recommends and supports using OWSM policies for interoperability, security and usability in Oracle Cloud.

Oracle recommends that for authentication:

* REST services be secured with the OWSM server policy "oracle/multi\_token\_over\_ssl\_rest\_service\_policy".

      This policy supports any of the following authentication mechanisms:

* HTTP Basic Username/Password over SSL
* SAML 2.0 Signed Bearer token in the HTTP header over SSL
* JWT token in the HTTP header over SSL
* REST clients, based on the use case, can be secured with the corresponding OWSM client policies "oracle/http\_basic\_auth\_over\_ssl\_client\_policy" (supports HTTP Basic Username/Password over SSL).

# Overview

The purpose of this document is to guide you how to use this sample application to develop a connecter between Oracle Sales Cloud and a partner application.

The connector is about data synchronization between a partner application and Oracle Sales Could/JCS . The data can be sync to the partner app from OSC/JCS , another direction or both.

There is an admin mapping UI that can be used to configure object-attribute mapping. The UI lists all source objects. For each object, it lists all attributes needed to be configured. For each source attribute, we can select the target attribute to be mapped. All these object-attribute mapping configurations are stored in DBCS. Later on, when the data sync is executed, mapping data will be retrieved from DBCS in order to set target value based on source value and mapping attributes.

Data synchronization can also be executed by a scheduler. DBCS’s scheduler is used for data synchronization. There is a scheduler admin UI that can be used to configure data sync like the starting time frequency, repeat interval, disable, enable, run immediately etc. Behind the scene, whenever the data sync scheduler task is triggered, the task will call a REST service in JCS, and that REST service will do the data sync.

In this sample data application, there are two use cases. The first use case is to synchronize from OSC/JCS to the partner application immediately. The second use cause is to load data from healthcare data in JCS database tables and push it to OSC’s custom objects through scheduler. Alternatively, the code can be modified to apply for other use cases as well.

## Assumptions

* It is assumed that the reader is competent with Oracle JDeveloper and general Java development, as well as Oracle Sales Cloud extensibility concepts (i.e. Application Composer).
* Reader is aware of creating jar,war,ear deployment packages
* It’s also assumed that reader is aware of WebLogic deployment

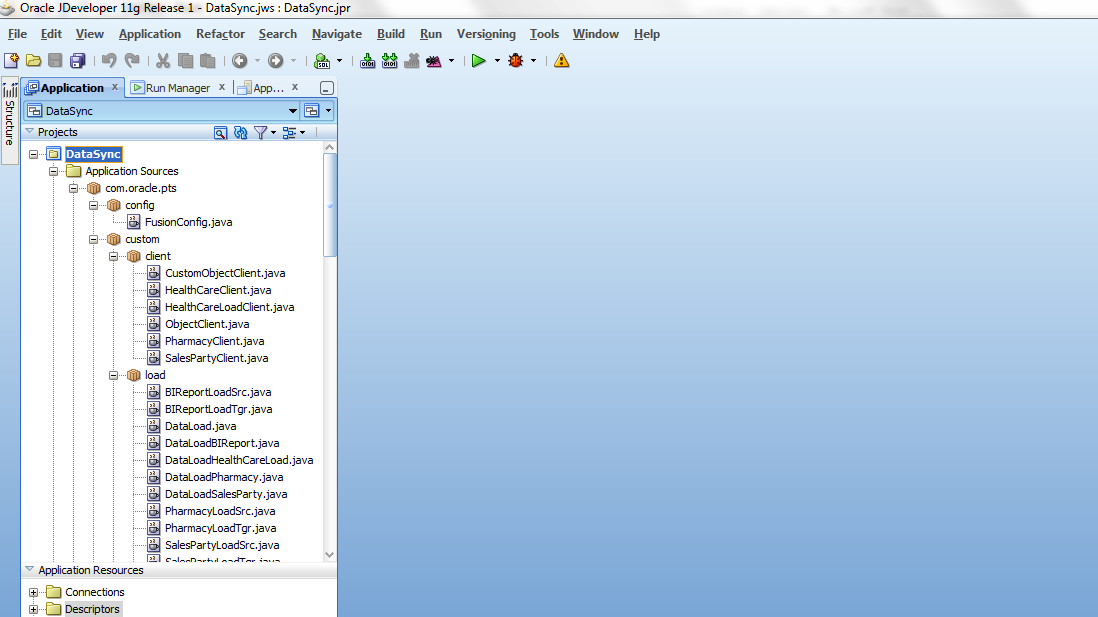
## Environment

* Oracle Sales Cloud Release 8
* Oracle Java Cloud Service 13.2
* Oracle JDeveloper 11.1.1.7.1

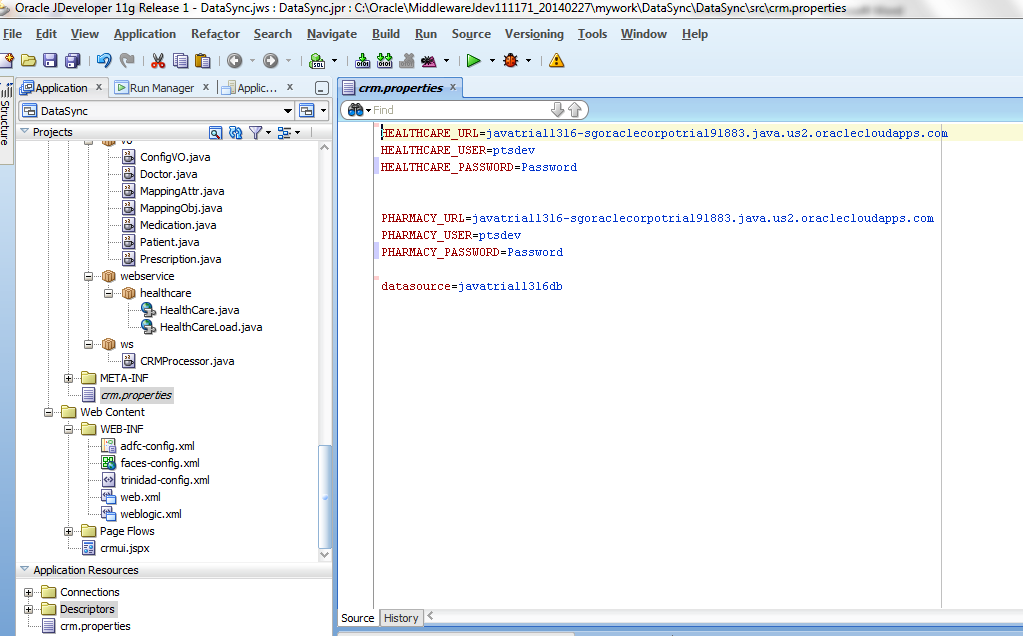
# Instructions

## DataSync Application

* Open DataSync application DataSync project in JDeveloper



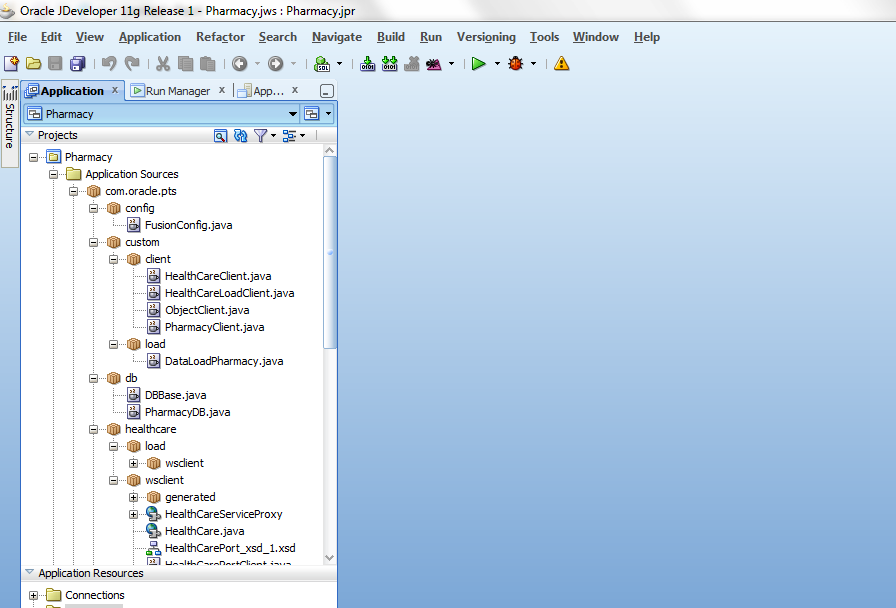
* Open crm.properties in JDeveloper and set database datasource name that is used for the JCS.



|  |  |
| --- | --- |
| HEALTHCARE\_URL | JCS\_HOSTNAME, JCS hostname |
| HEALTHCARE\_USER | JCS\_USERNAME, JCS username |
| HEALTHCARE\_PASSWORD | JCS\_PASSWORD, JCS password |
| PHARMACY\_URL | JCS\_HOSTNAME, JCS hostname |
| PHARMACY\_USER | JCS\_USERNAME, JCS username |
| PHARMACY\_PASSWORD | JCS\_PASSWORD, JCS password |
| datasource | Datasourcename for DBCS |

## Pharmacy Application

* Open DataSync application DataSync project in JDeveloper



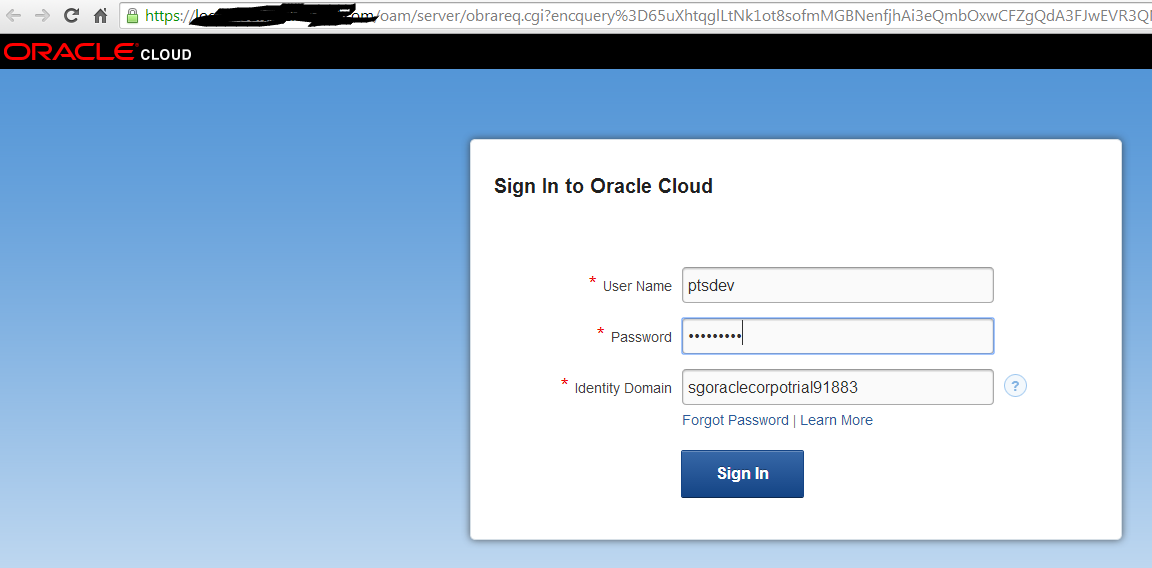
* Open crm.properties in JDeveloper and set database datasource name that is used for the JCS.

|  |
| --- |
| # Sample JCS datasoruce name  datasource=javatrial1316db |

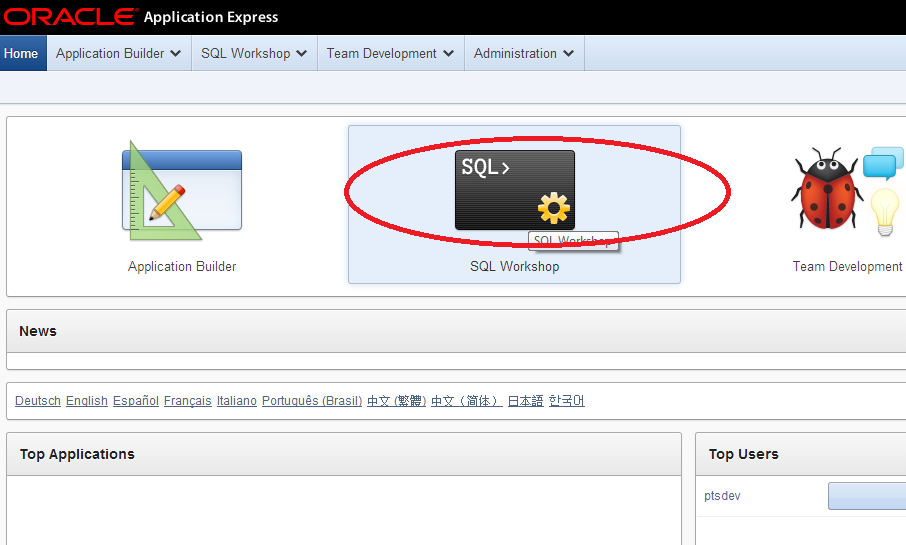
# 

## Load Database sql script

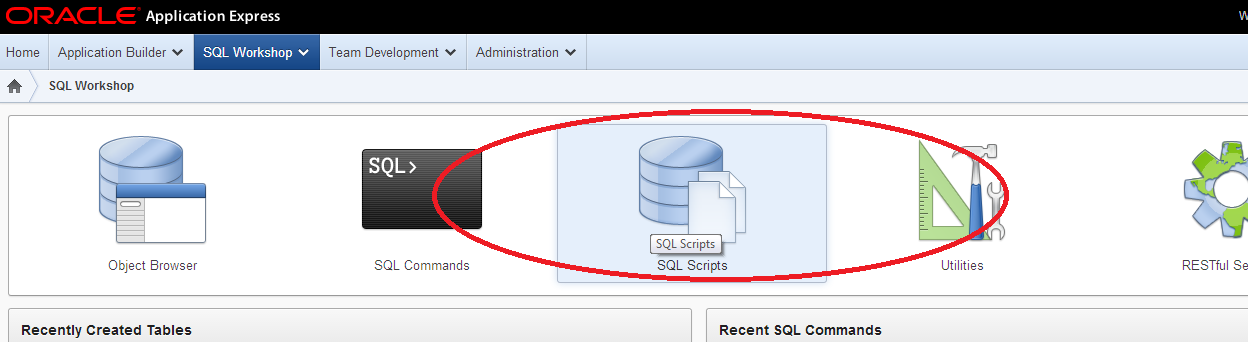
* Login to DBCS admin console



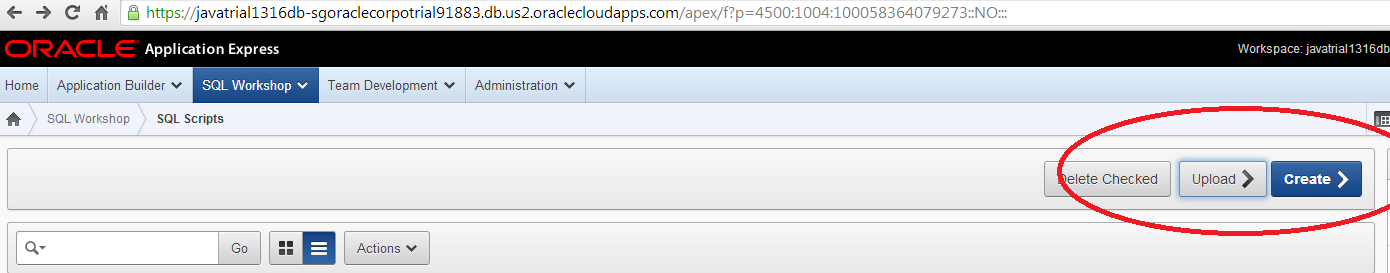
* Click SQL Workshop



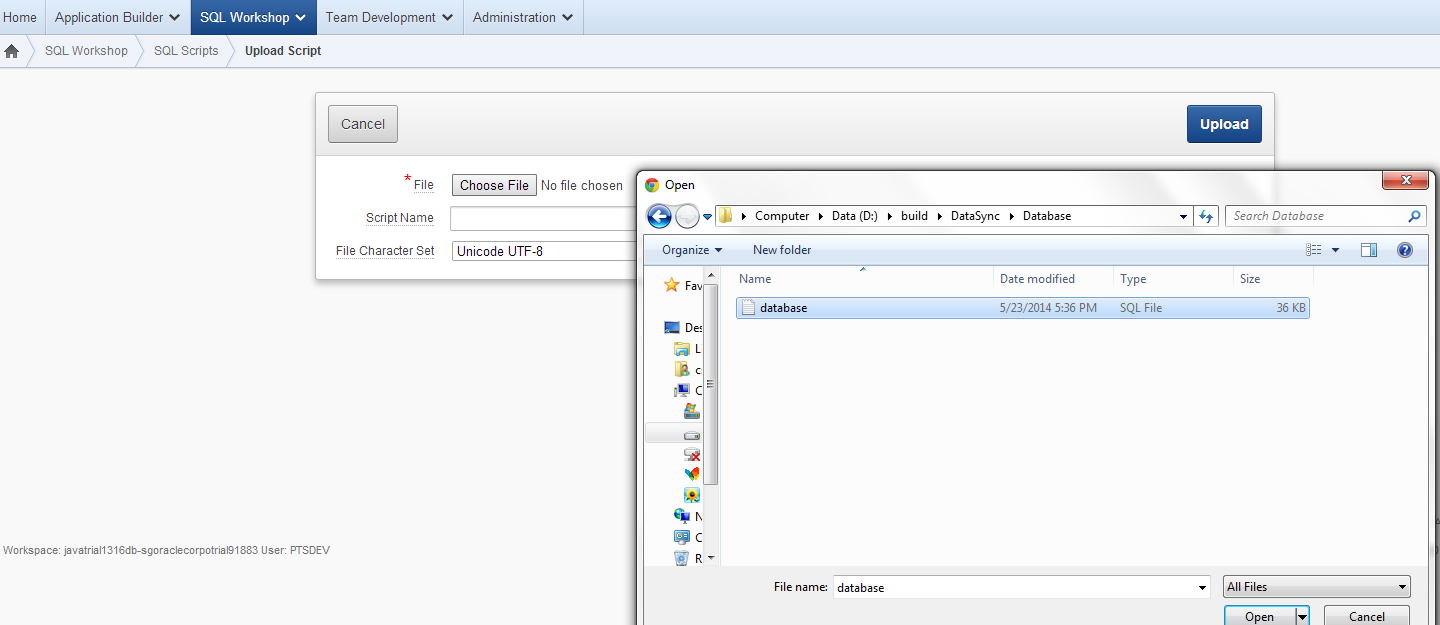
* Click SQL Script



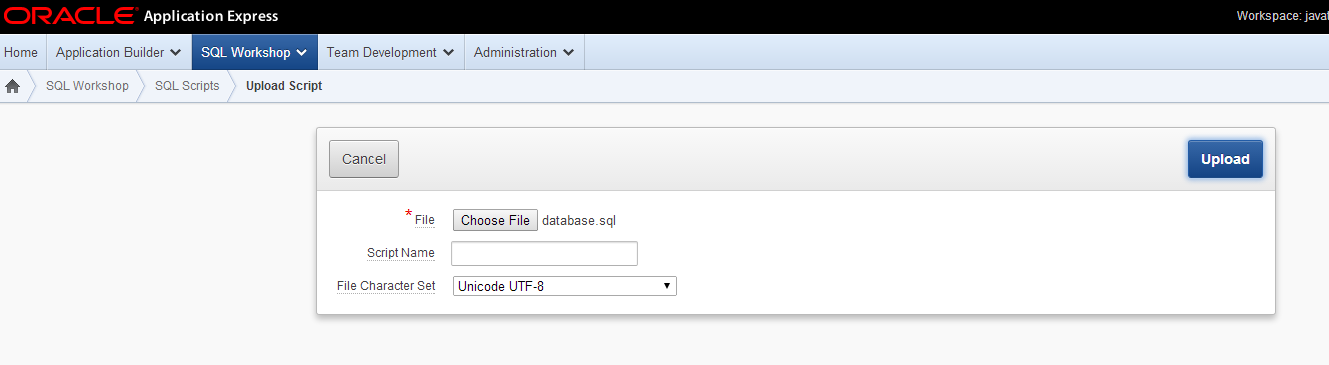
* Click Upload



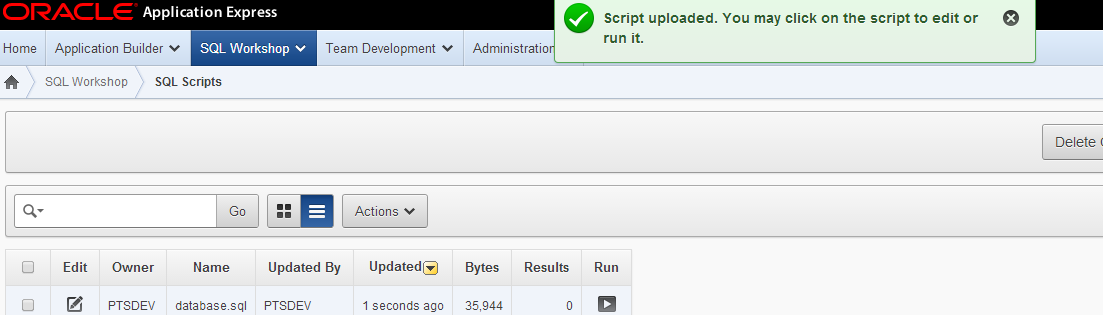
* Choose database.sql from Database folder of the download zip file



* Click Upload



* Click Run



## Oracle Sales Cloud connection Configuration

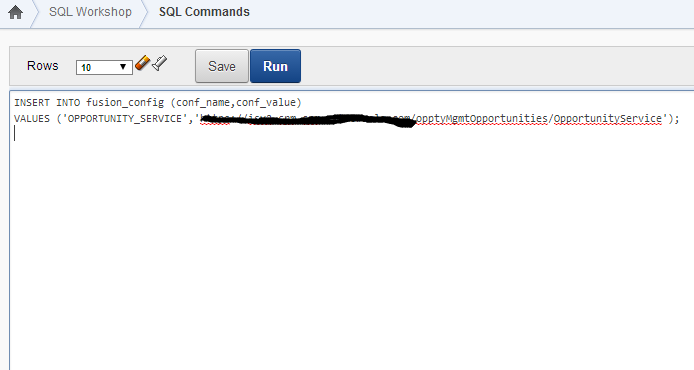
* Set Oracle Sales Cloud connection info in fusion\_config database where OSC\_HOME is Oracle Sales Cloud hostname

|  |  |
| --- | --- |
| OPPORTUNITY\_SERVICE | https://<OSC\_HOME>/opptyMgmtOpportunities/OpportunityService |
| CUSTOMOBJECT\_SERVICE | https://<OSC\_HOME>/crmCommonCustExtn/CustomerCenterCustomObjectService |
| SALESPARTY\_SERVICE | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService |
| FUSION\_USER | Osckey |
| FUSION\_SalesPartyService\_SCHEMA | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService?XSD=/oracle/apps/crmCommon/salesParties/salesPartiesService/SalesParty.xsd |
| FUSION\_OptyService\_SCHEMA | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService?XSD=/oracle/apps/crmCommon/salesParties/salesPartiesService/SalesParty.xsd |

For each row above, run SQL command

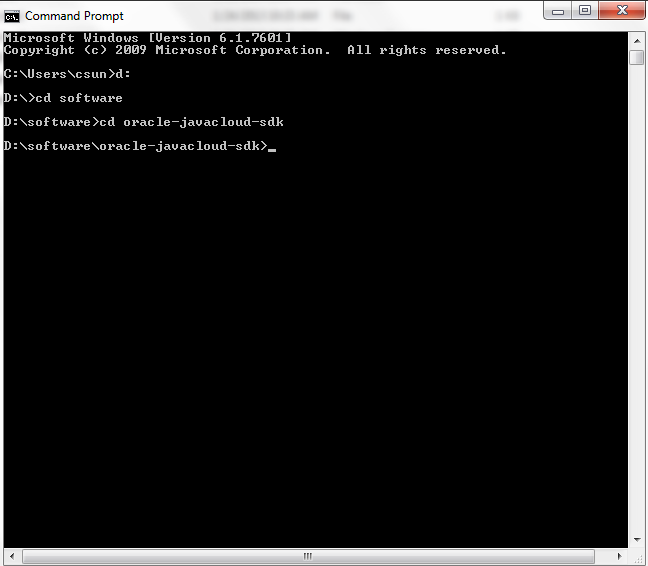
INSERT INTO fusion\_config (conf\_name,conf\_value)

VALUES ('name',' value');



# Create credential for Oracle Sales Cloud connection in JCS credential store

* Download Oracle Java Cloud Service SDK (release 14.1.2.0) from <http://www.oracle.com/technetwork/middleware/weblogic/downloads/java-cloud-sdk-1848874.html>
* Unzip the downloaded SDK
* Open a command prompt and go to the SDK location



* Create a credential

Sample:

javacloud -id sgoraclecorpotrial91883 -si javatrial1316 -u ptsdev -p "jcs\_admin\_pass" -adminurl https://javaservices.us2.cloud.oracle.com -key osckey -keyuser Matt.Hooper -keypassword osc\_pass -set-credential

-id: JCS identity domain

-si: java instance name

-u: JCS admin user

-p: JCS admin password

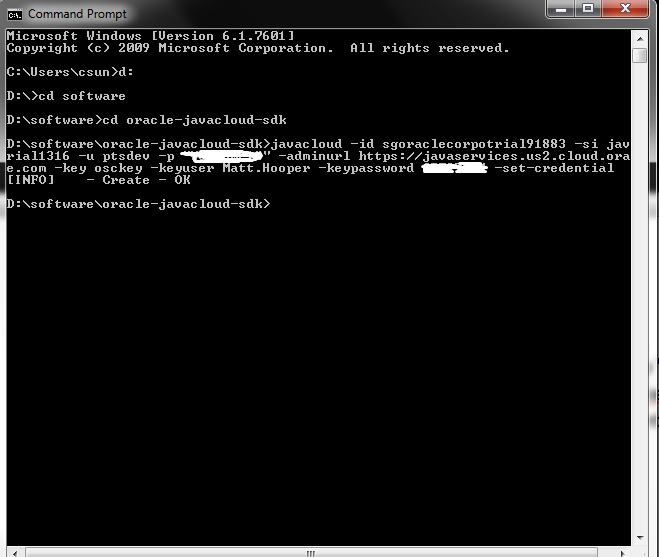
-adminurl: JCS admin url

-key: the name of this credential

-keyuser: the user used to connect to Oracle Sales Cloud

-keypassword: the associated password for the keyuser

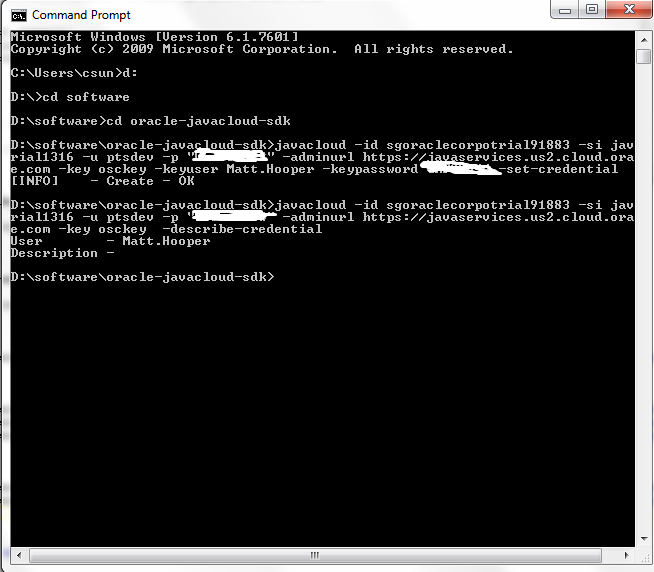
For complete command line reference, go to doc folder in SDK utility



* Check if the key is created

Sample:

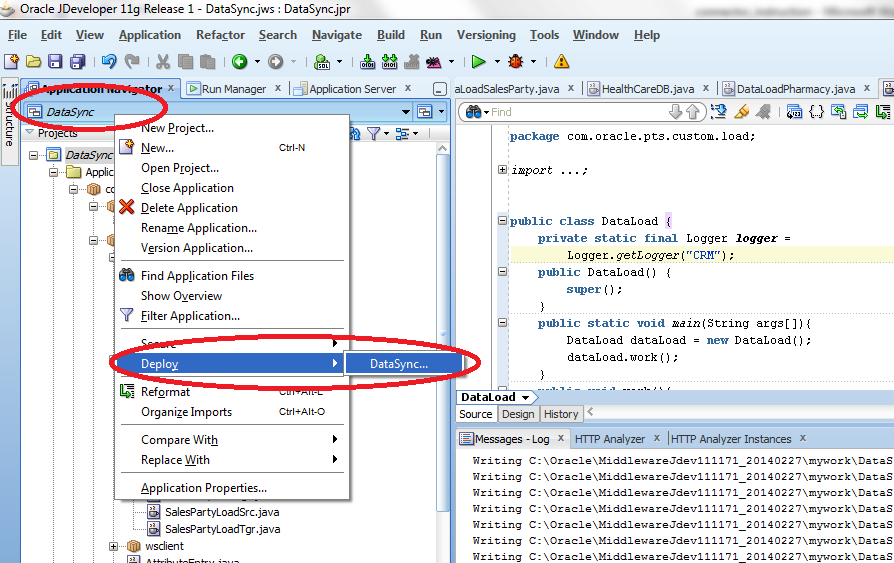
javacloud -id sgoraclecorpotrial91883 -si javatrial1316 -u ptsdev -p " jcs\_admin\_pass " -adminurl https://javaservices.us2.cloud.oracle.com -key osckey -describe-credential



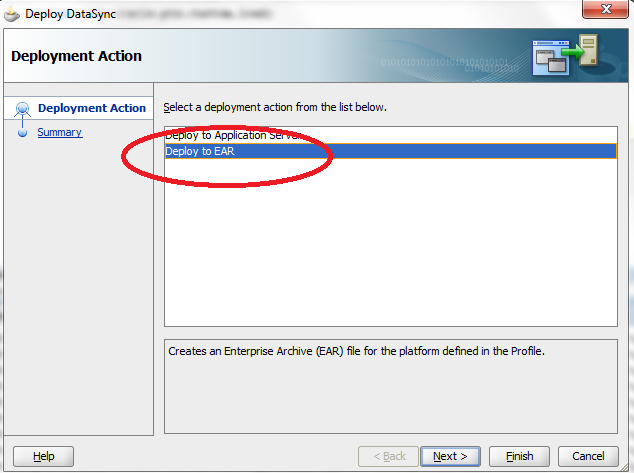
# Build and deploy the DataSync application

## Build the ear application

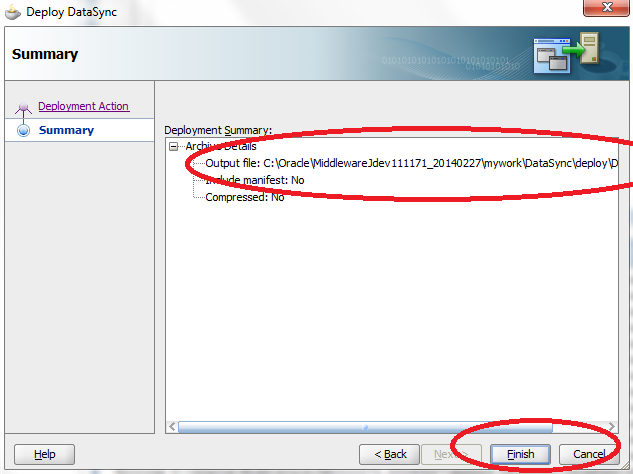
* Right click DataSync application and click Deployment -> DataSync



* Select Deploy ear and click next

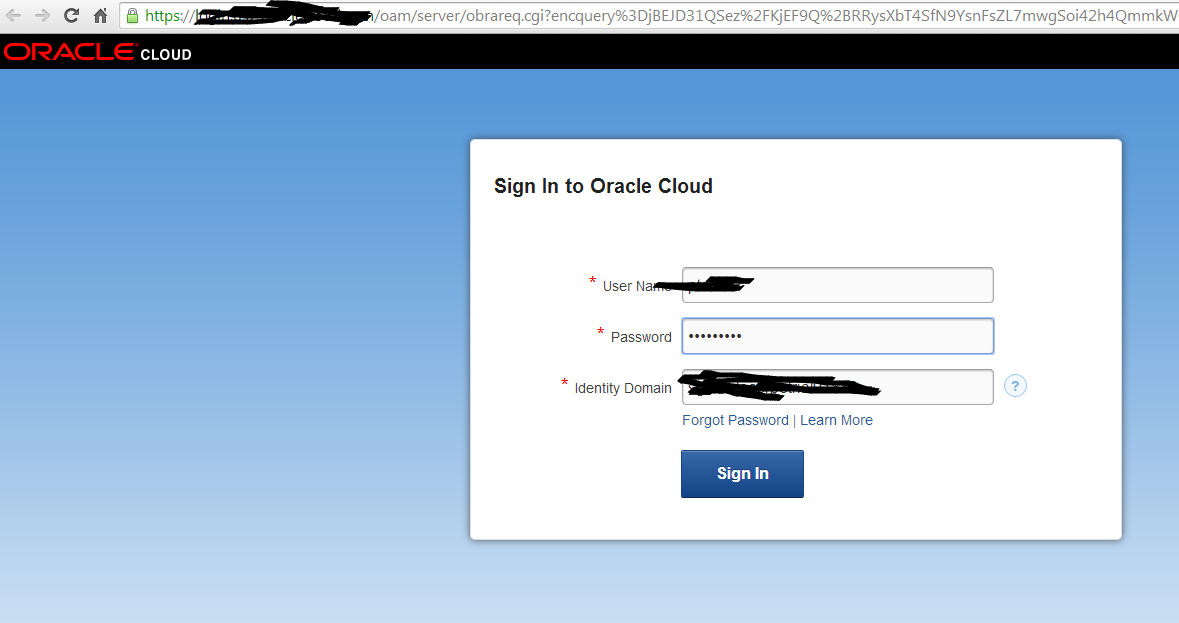


* Click Finish and remember output ear file location.

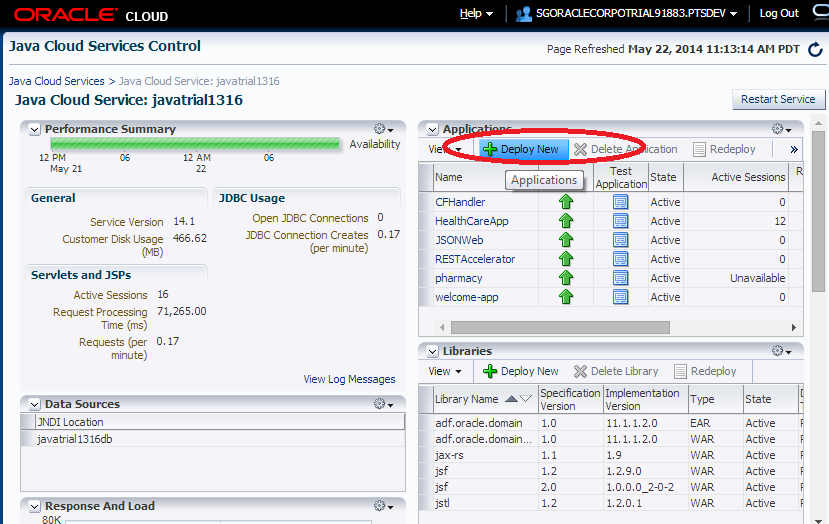


## Deploy the DataSync application into JCS

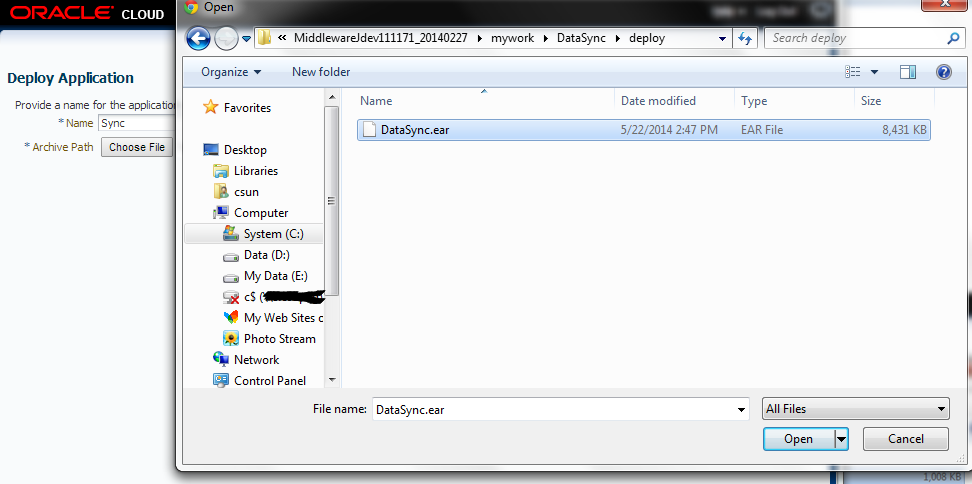
* Login to JCS admin console



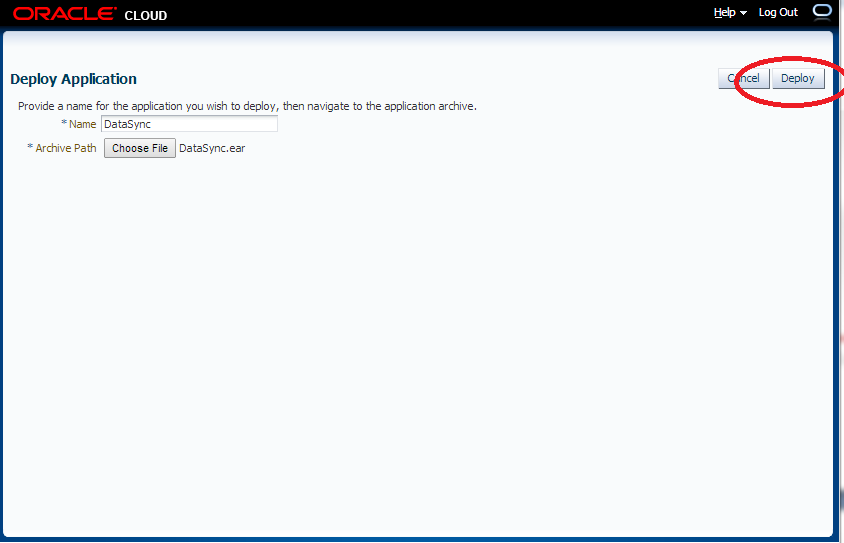
* Click Deploy New



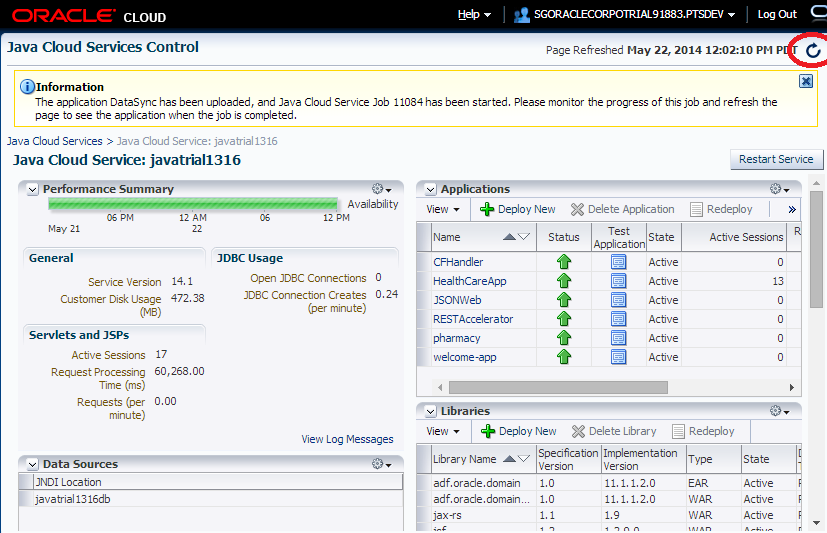
* Set Name as DataSync and select ear application from the previous steps



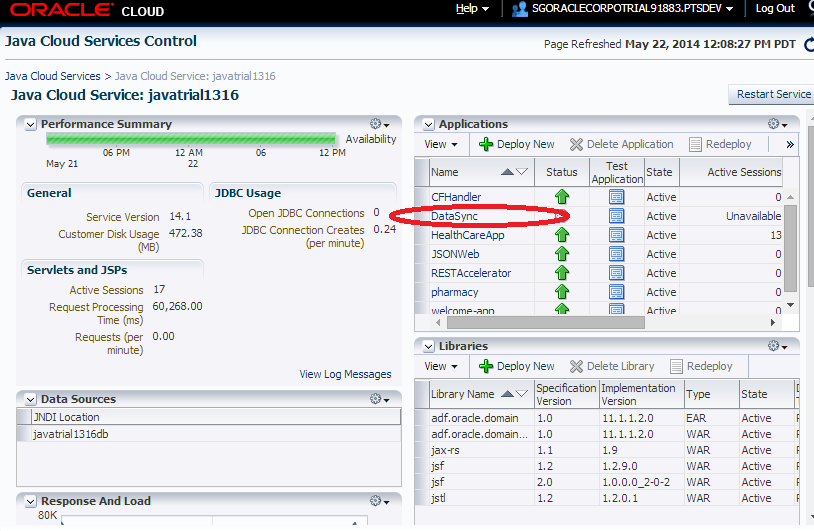
* Click Deploy



* Wait for a minute and click refresh

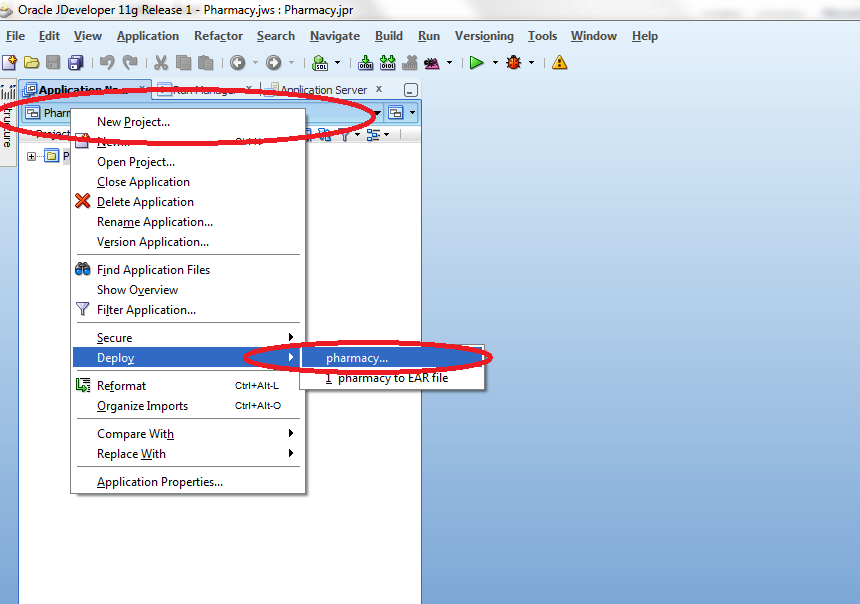


* DataSync application is deployed

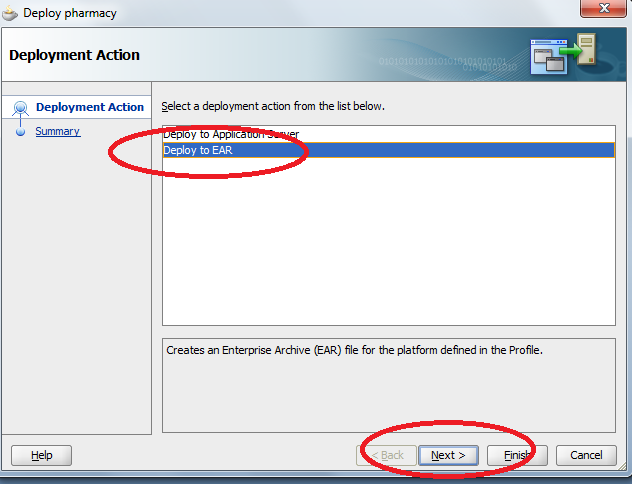


## Build the Phramcy ear application

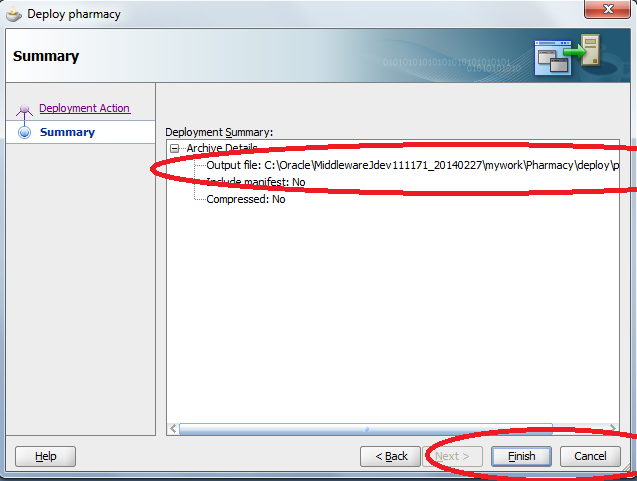
* Right click Pharmacy application and click Deployment -> pharmacy



* Select Deploy ear and click next

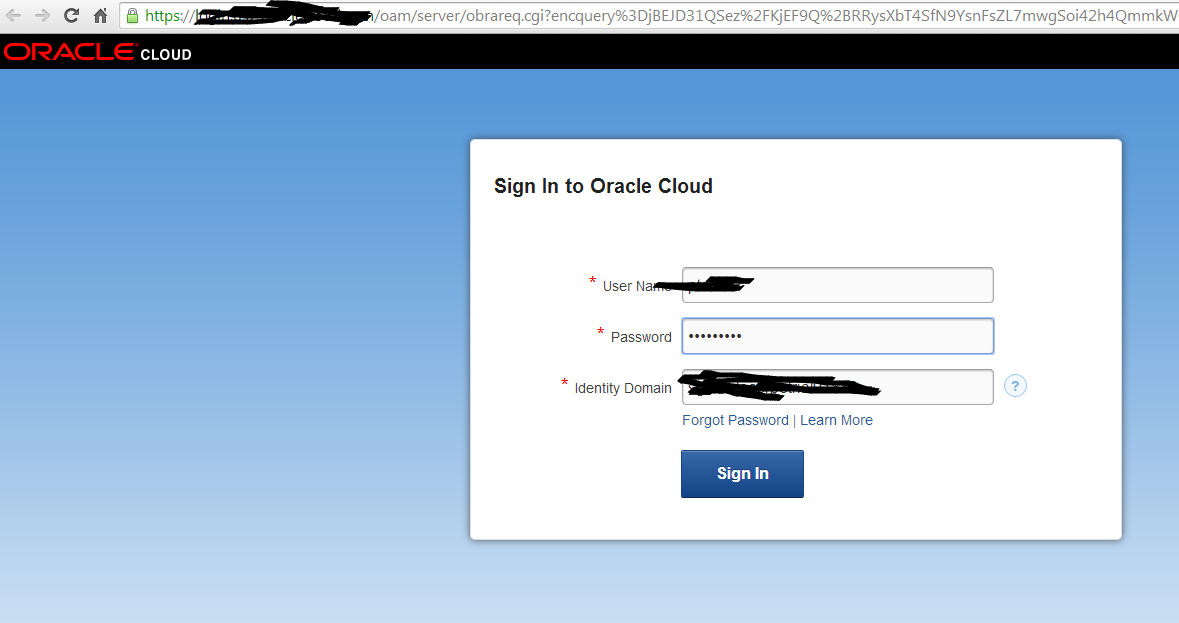


* Click Finish and remember output ear file location.

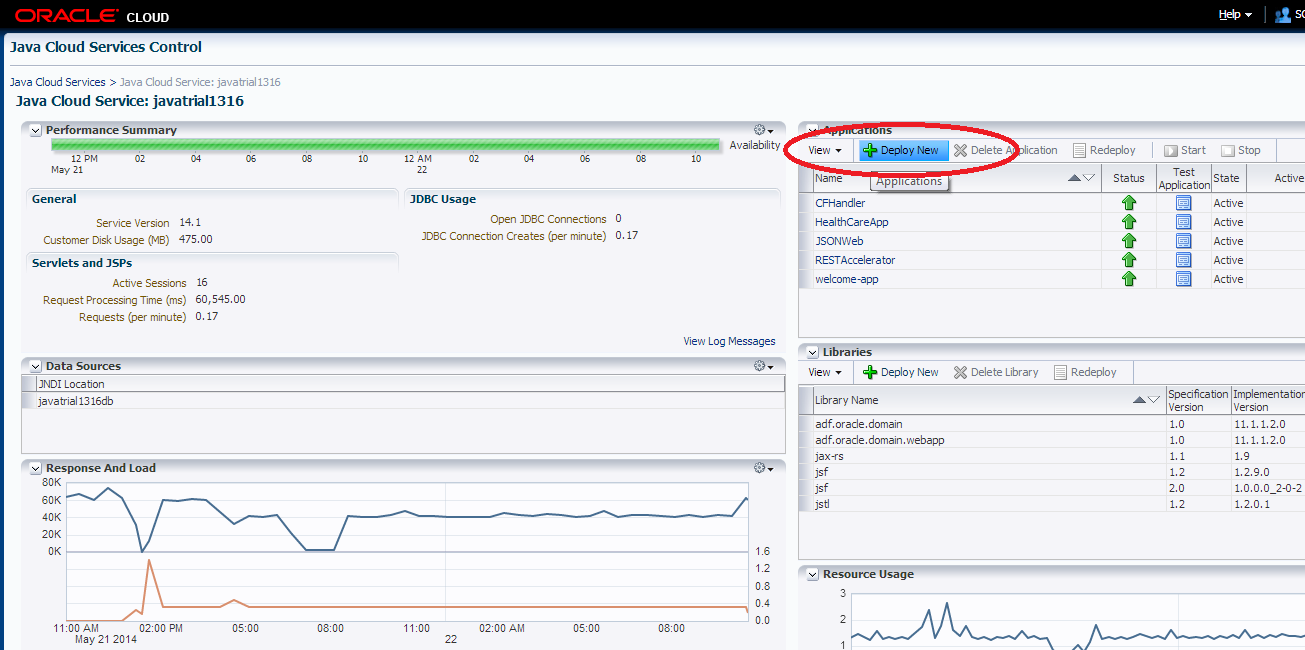


## Deploy the application into JCS

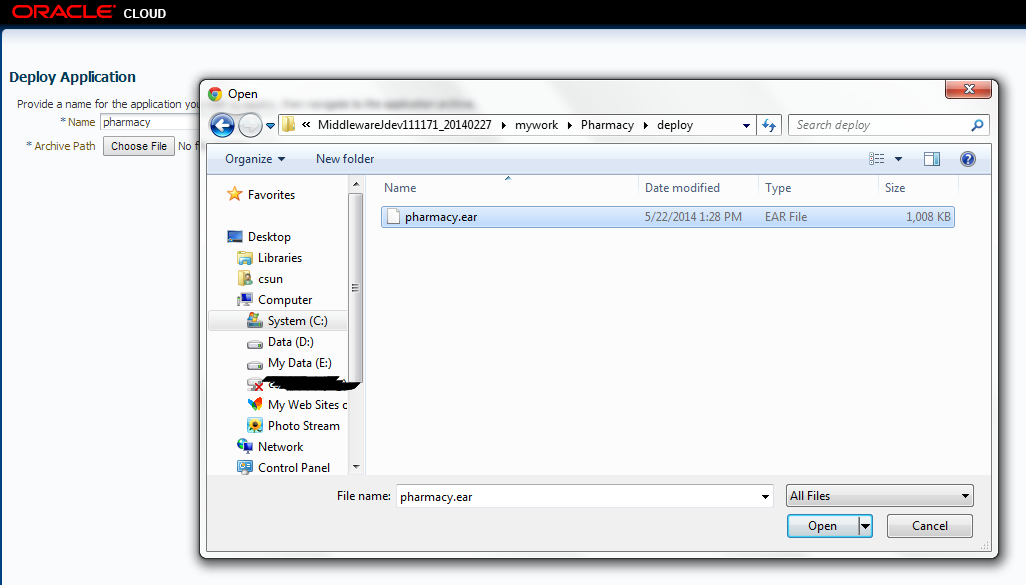
* Login to JCS admin console



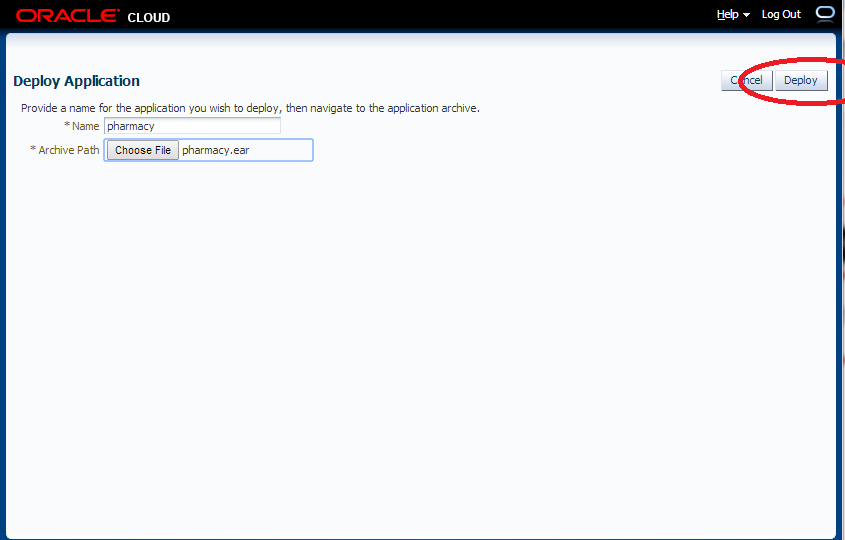
* Click Deploy New



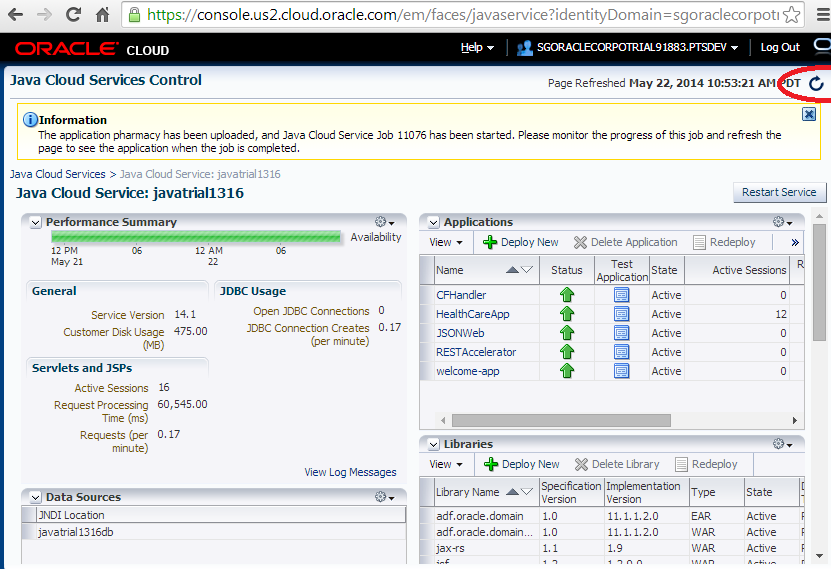
* Set Name as pharmacy and select ear application from the previous steps



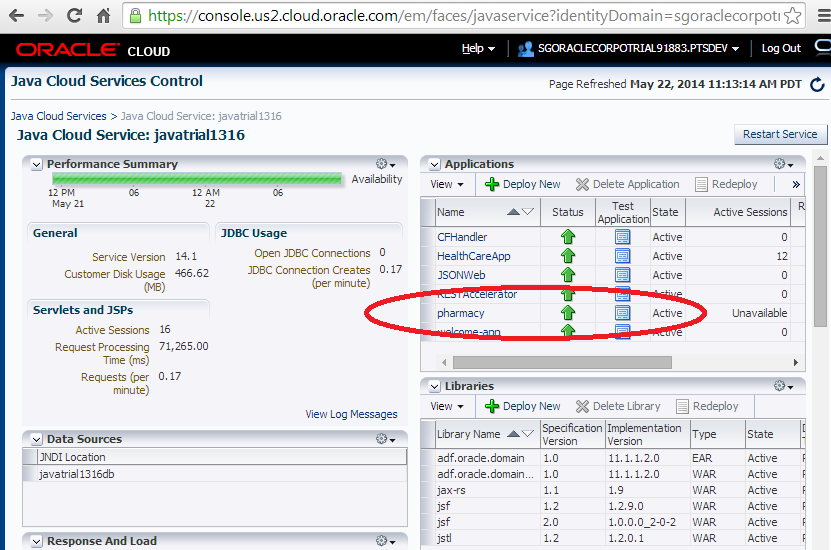
* Click Deploy



* Wait for a minute and click refresh



* pharmacy application is deployed

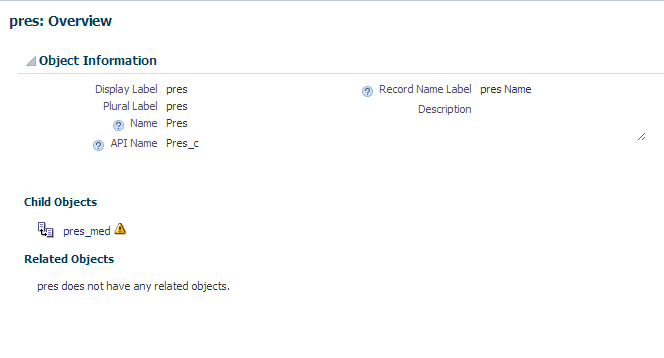


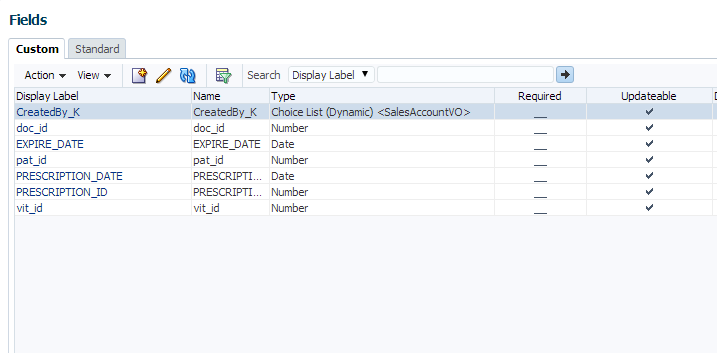
# Oracle Sales Cloud Custom Objects

**There are some Custom Objects which need to be created under Customer Center Application by using Application Composer. The data synchronization by scheduler will push data from JCS to Oracle Sales Cloud. Those objects are used by BI to generate some reports.**

## Custom Object: Pres\_c

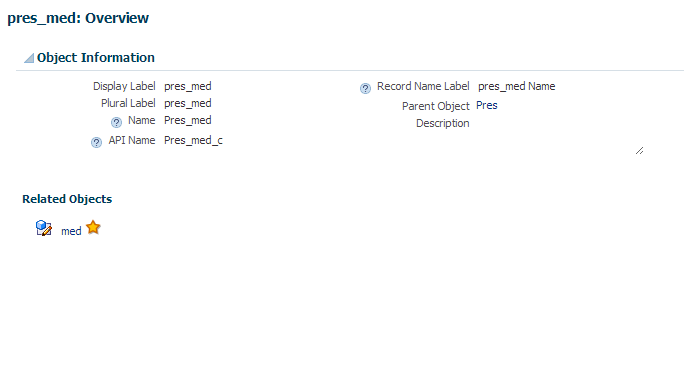
**This object represents a Prescription.**

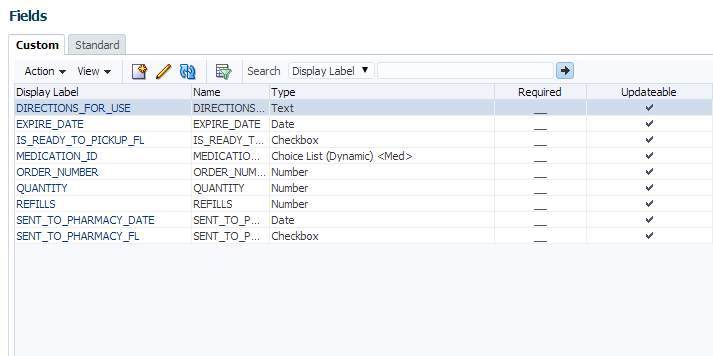




## Custom Object: Pres\_Med\_c

**This object represents a relationship between Prescription and Medication.**

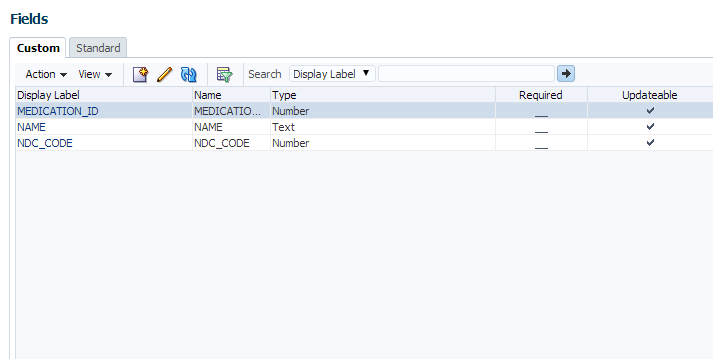




## Custom Object: Med\_c

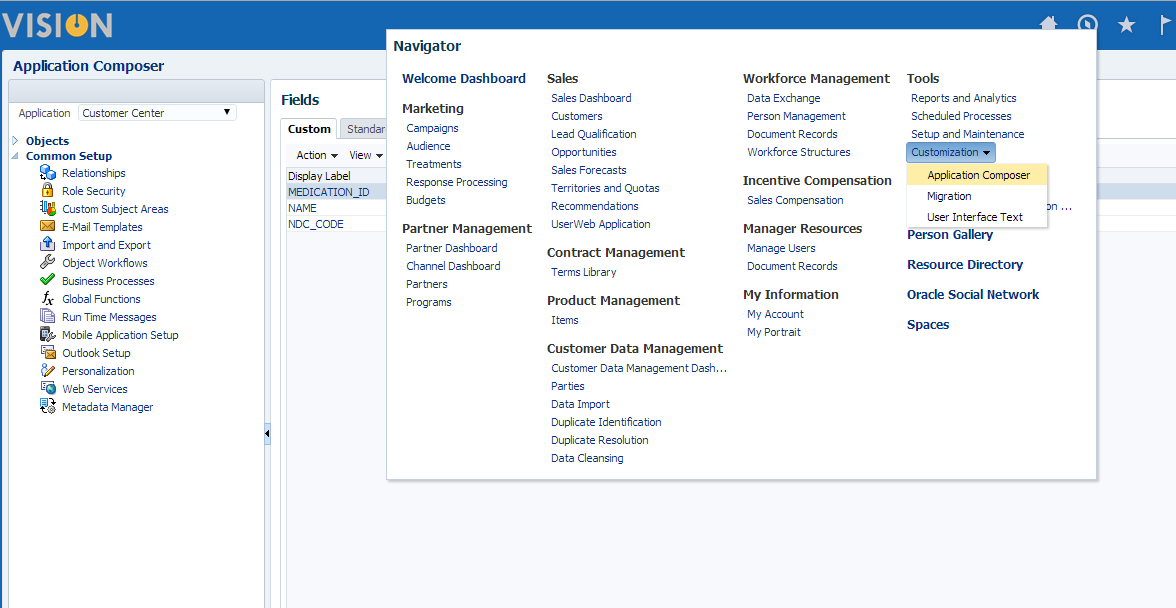
**This object represents a Medication.**



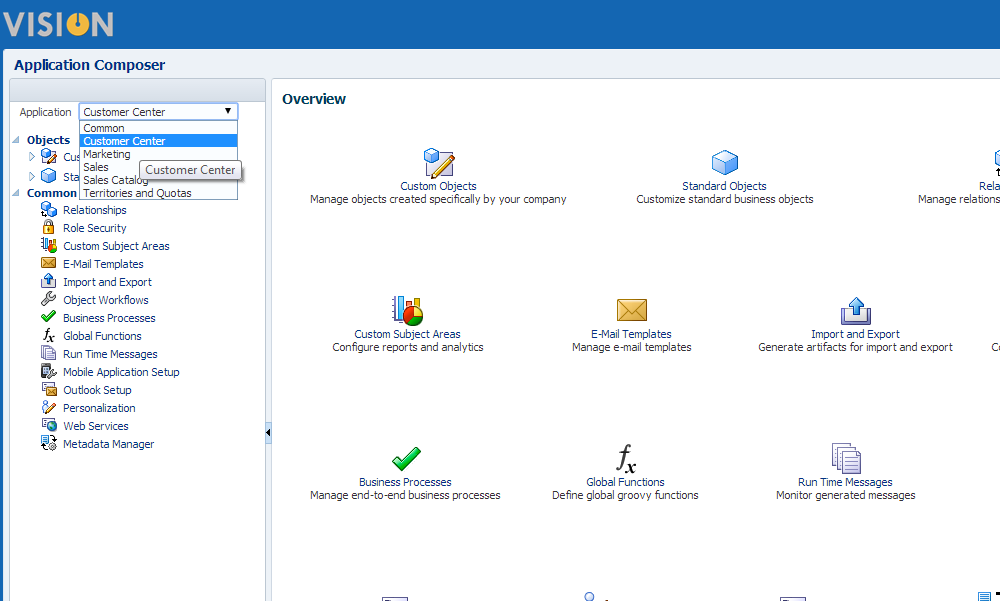


## Steps to create a custom object

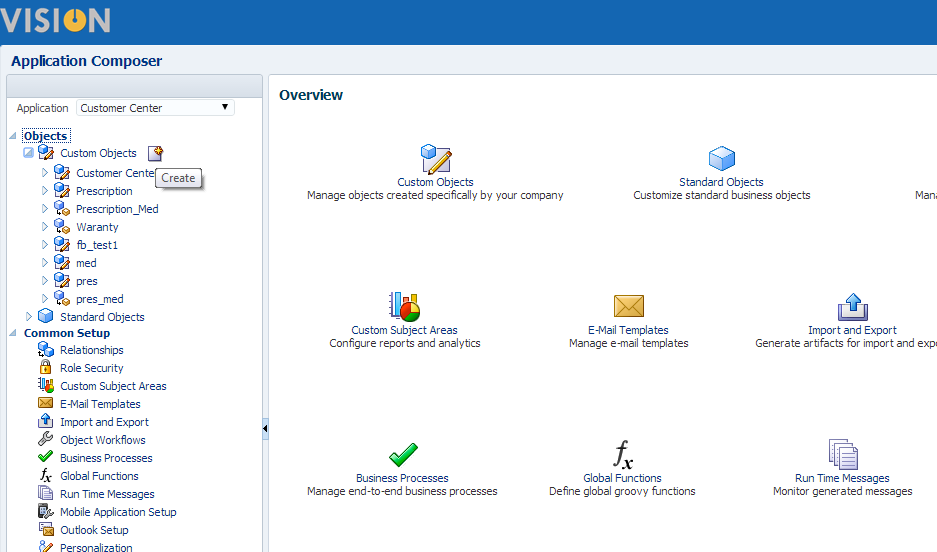
1. **Click on Navigator -> Customization -> Application Composer**



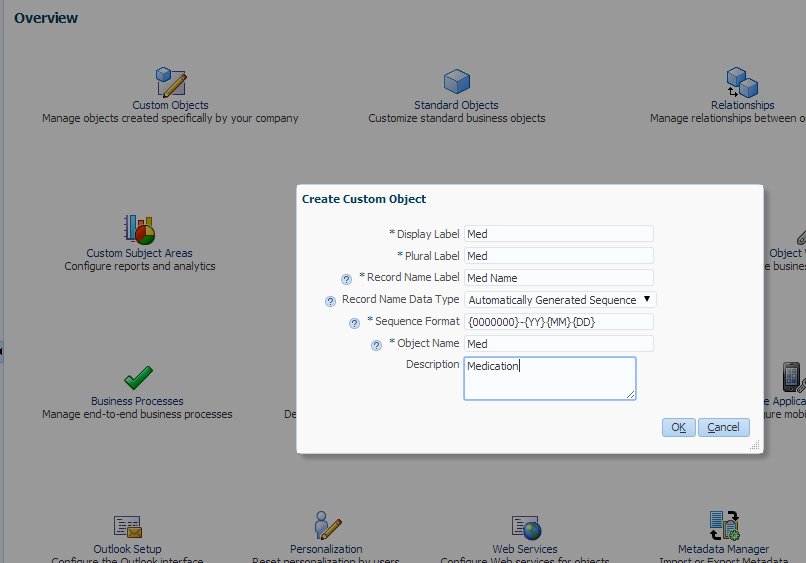
1. **Choose Customer Center**



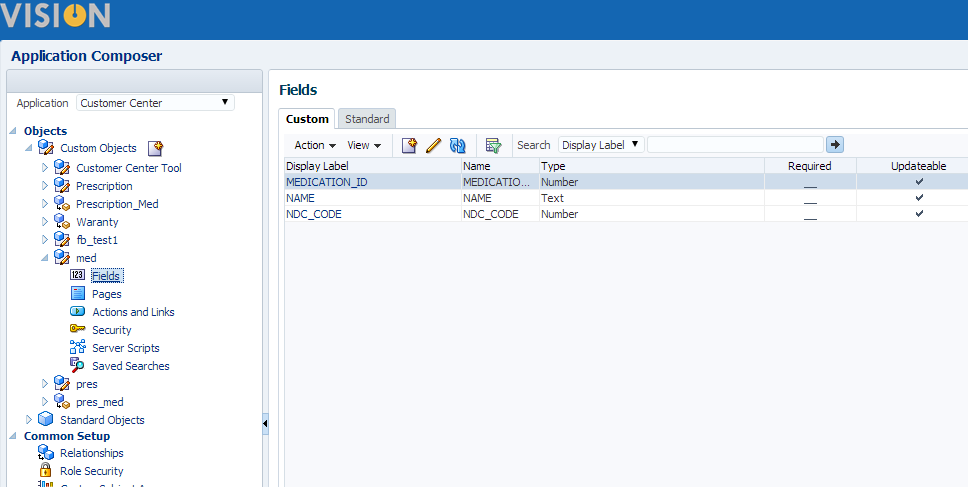
1. **Click a ‘+’ icon to create a custom object**



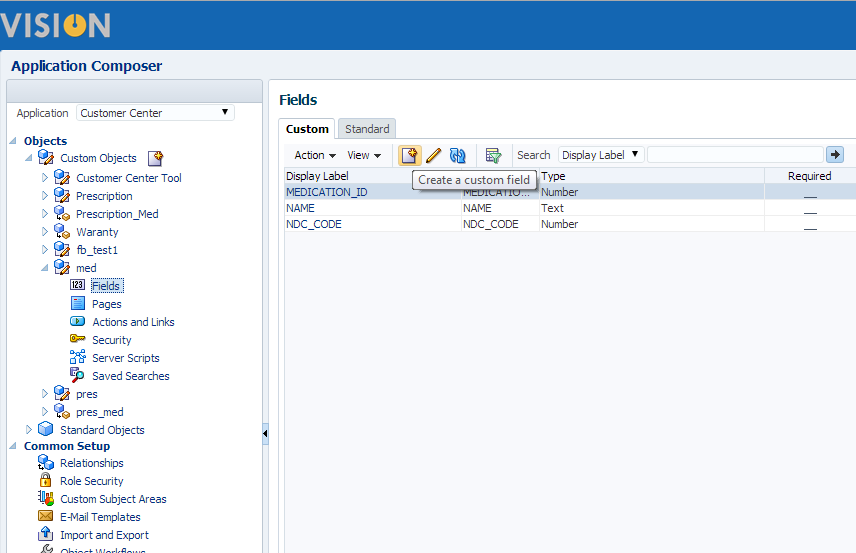
1. **Fill up the required fields and click OK**



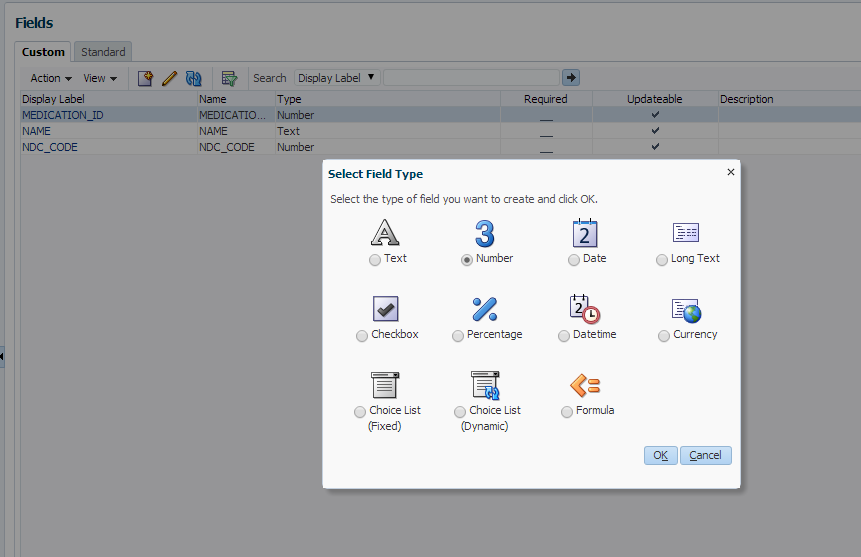
1. **Click on Fields on the left side**



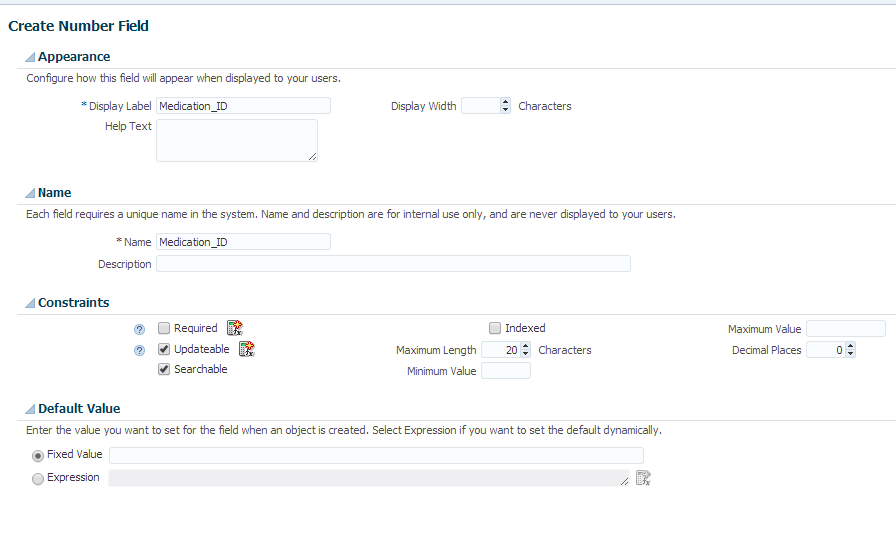
1. **Click ‘+’ icon to create a custom field**



1. **Choose number**



1. **Fill up the required field and click save**

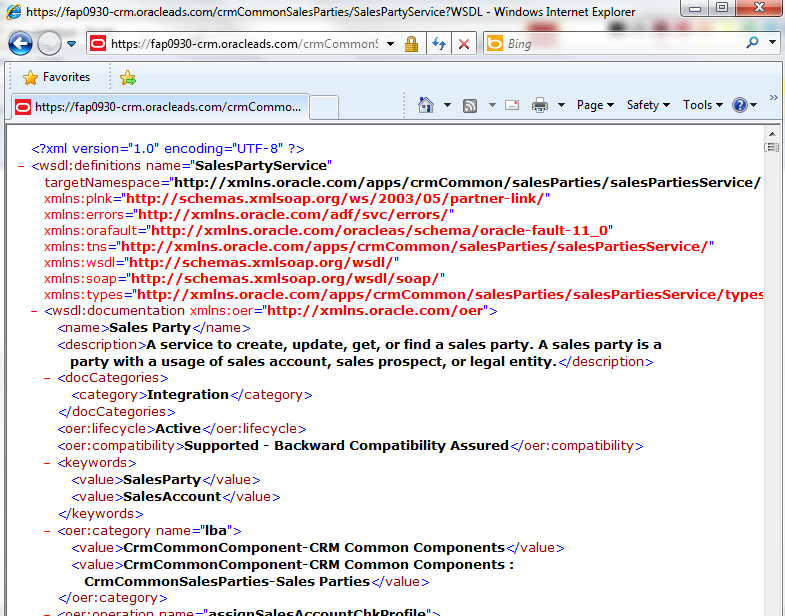


# Import Oracle Sales Cloud certificate

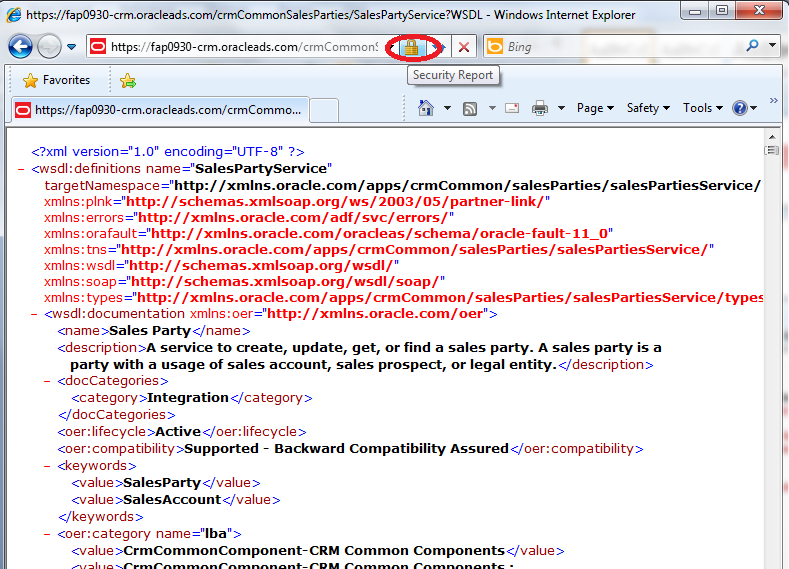
In order to authenticate to Fusion, the server side public certificate must be acquired and added as a trusted cert entry to a keystore used by the client. This keystore stores a reference to the Fusion public certificate and uses the alias "orakey". The Fusion public certificate is obtained from any Fusion Application object WSDL.

* Open IE and type OSC SalesParty web service wsdl URL
* WSDL Document URL : https://<Your OSC host>/crmCommonSalesParties/SalesPartyService?WSDL

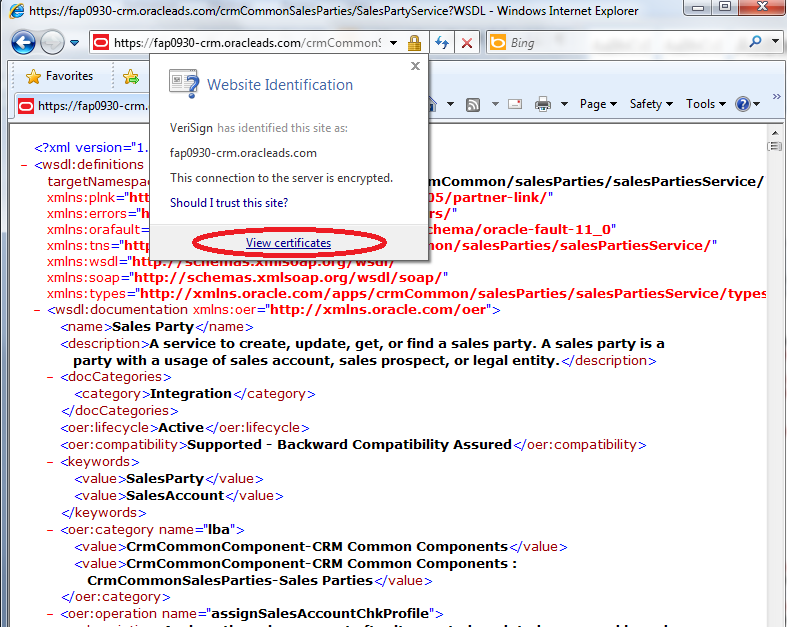
Sample: https://fap0930-crm.oracleads.com/crmCommonSalesParties/SalesPartyService?WSDL



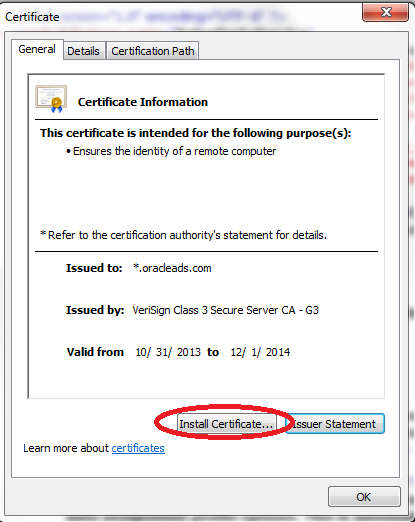
* Click key icon



* Click View certificates



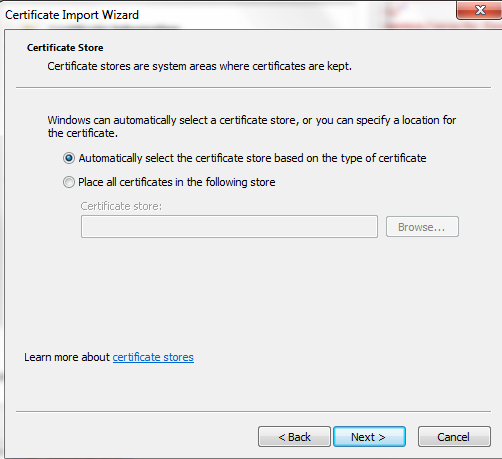
* Click View certificates



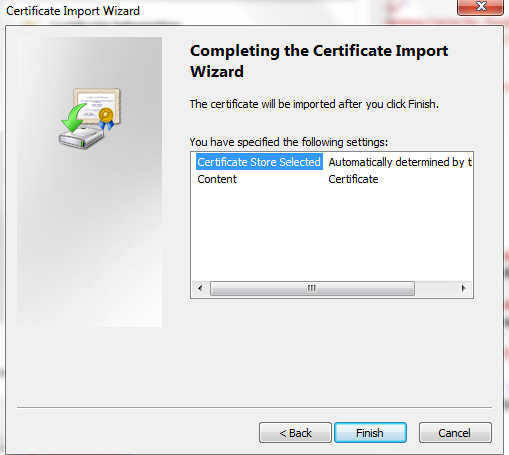
* Click Next



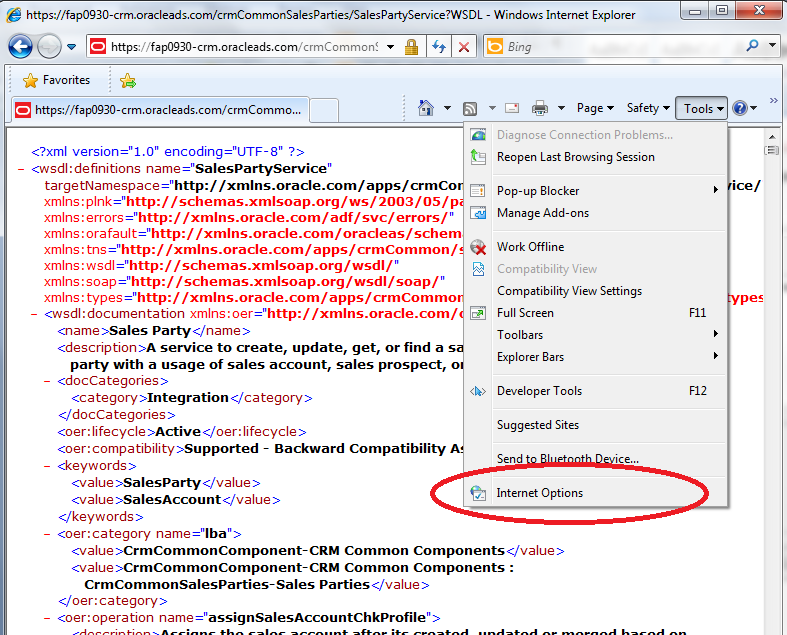
* Click Next



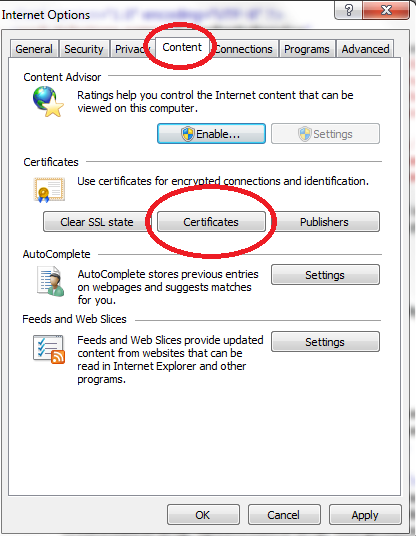
* Click Finish



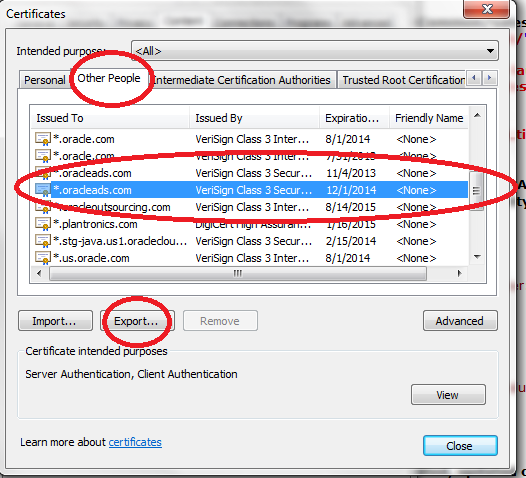
* Now you can export this cert
* Click Tools -> Internet Options



* Click Content -> Certificates



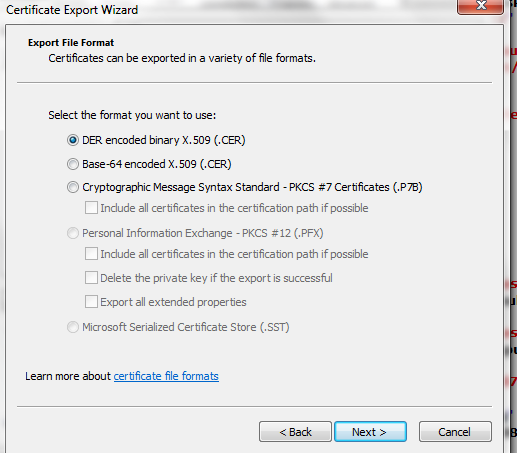
* Click Other People, select certificate you just imported and click Export



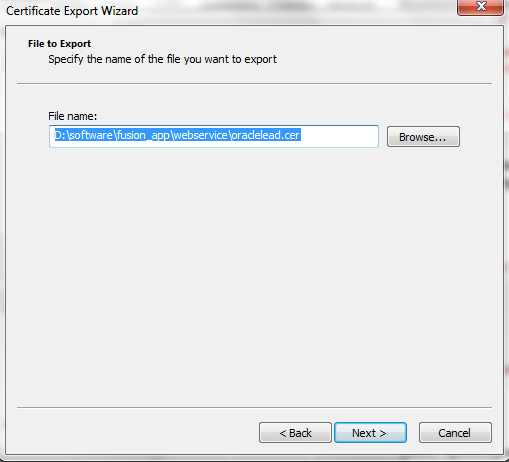
* Click Next



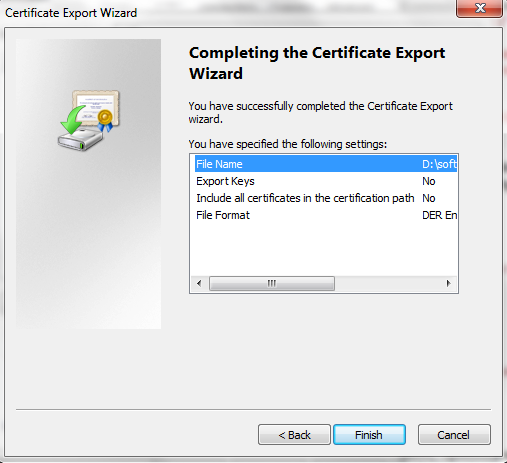
* Click Next



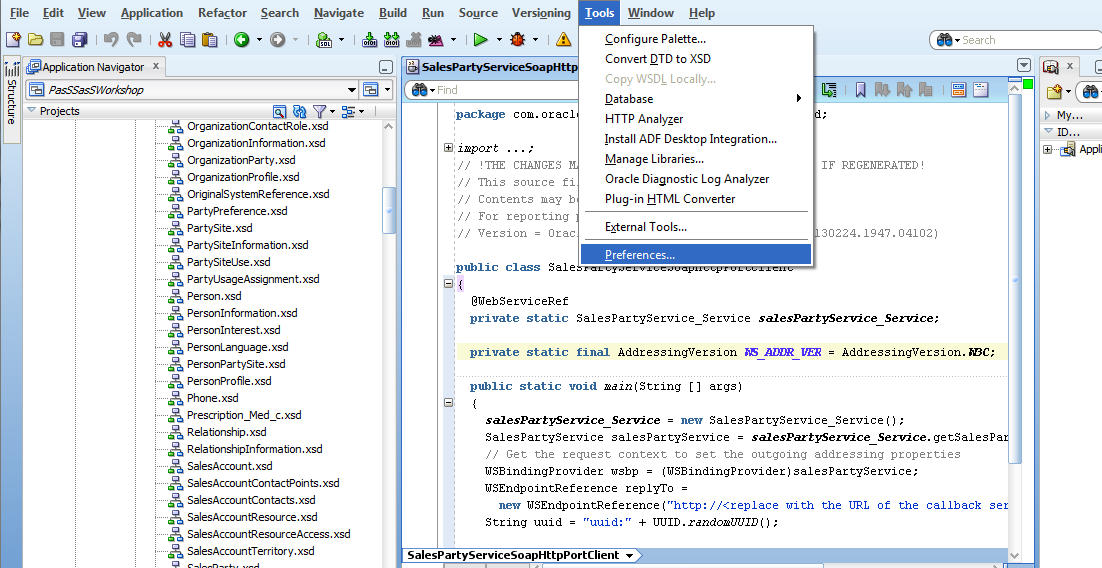
* Browser the location and type file name and click Next
* Sample: D:\software\fusion\_app\webservice\oraclelead.cer



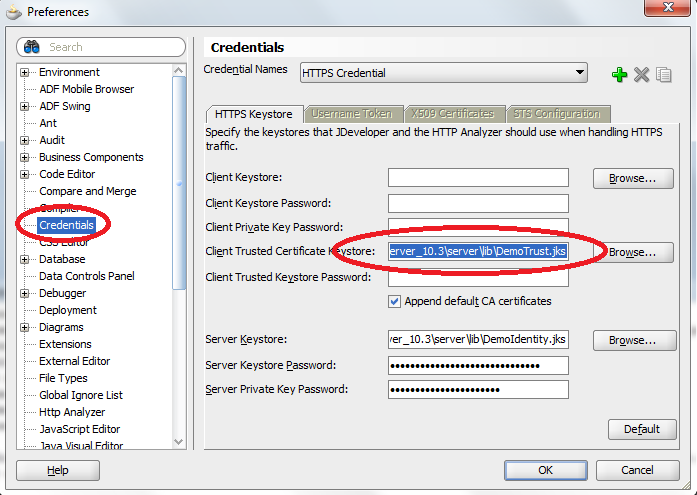
* Click Finish

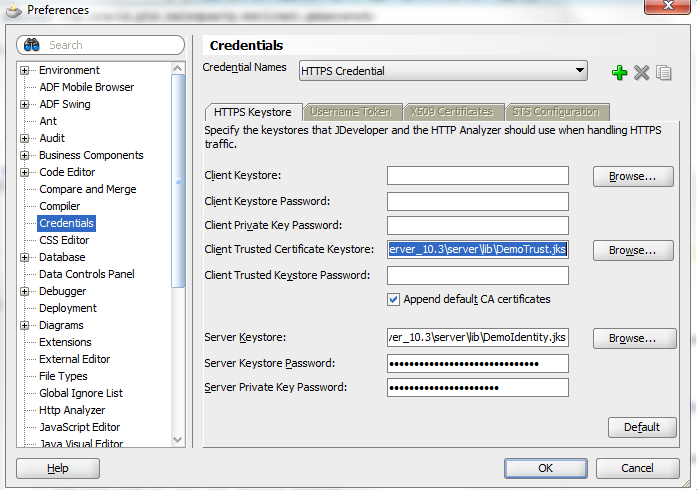


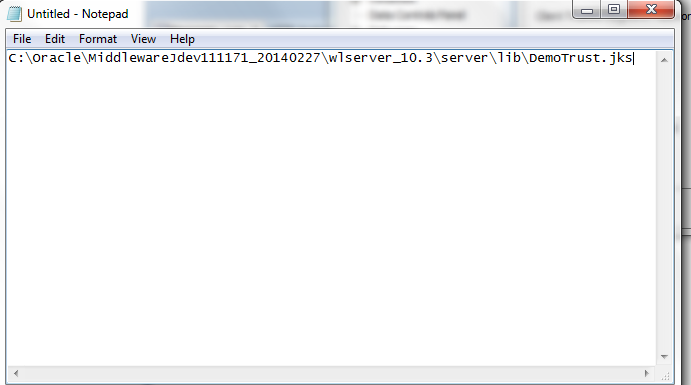
* Check Trusted Certificate Keystore location by clicking Tools -> Preferences



* Copy Client Trusted Certificate Keystore location into Notepad. The certificate will be imported into this key store.
* Sample: C:\Oracle\MiddlewareJdev111171\_20140227\wlserver\_10.3\server\lib\DemoTrust.jks



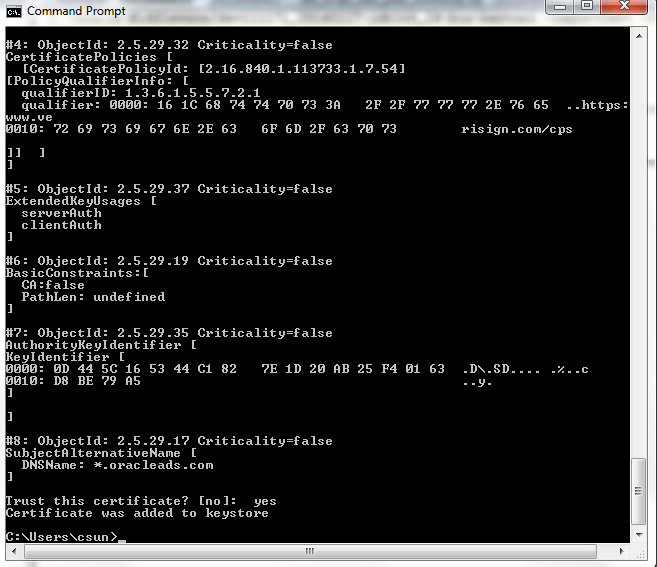




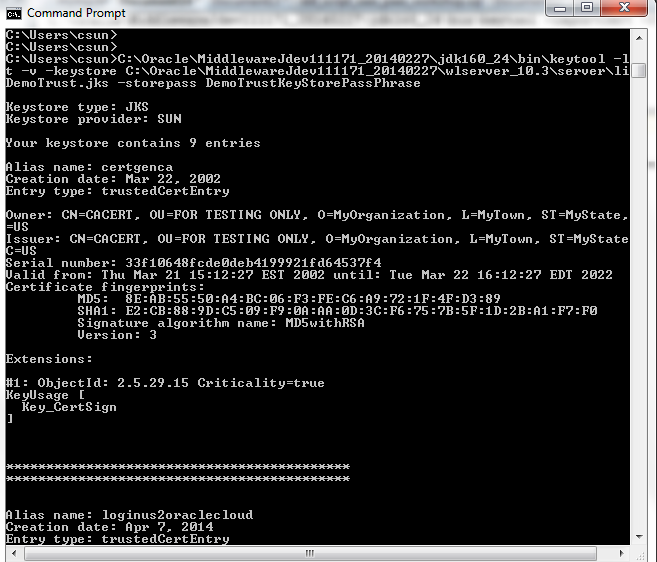
* Import Oracle certificate to a keystore file

1. Open an command prompt
2. Type command
3. Click Enter
4. Type ‘Yes’ for ‘Trust this certicate?’ and click enter

* Command: <JAVA\_HOME>\bin\keytool -importcert -file <Certificate\_Location>/oraclelead.cer -keystore <JDeveloper\_Location>\wlserver\_10.3\server\lib\DemoTrust.jks -alias oraclelead -storepass DemoTrustKeyStorePassPhrase
* Sample: C:\Oracle\MiddlewareJdev111171\_20140227\jdk160\_24\bin\keytool -importcert -file D:/software/fusion\_app/webservice/oraclelead.cer -keystore C:\Oracle\MiddlewareJdev111171\_20140227\wlserver\_10.3\server\lib\DemoTrust.jks -alias oraclelead -storepass DemoTrustKeyStorePassPhrase



* List the keystore contents
* Command: <JAVA\_HOME>\bin\keytool -list -v -keystore <JDeveloper\_Home>\wlserver\_10.3\server\lib\DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase
* Sample: C:\Oracle\MiddlewareJdev111171\_20140227\jdk160\_24\bin\keytool -list -v -keystore C:\Oracle\MiddlewareJdev111171\_20140227\wlserver\_10.3\server\lib\DemoTrust.jks -storepass DemoTrustKeyStorePassPhrase



# Import JCS certificate

Follow the same steps as Oracle Sales Certificate setup, but use JCS URL instead.

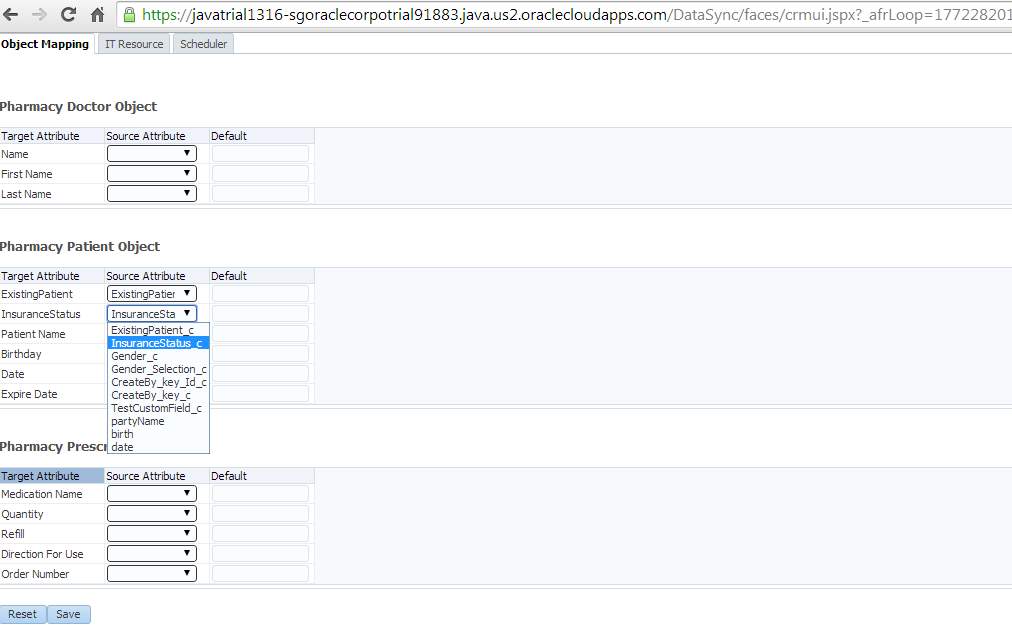


# 

# Test the data synchronization

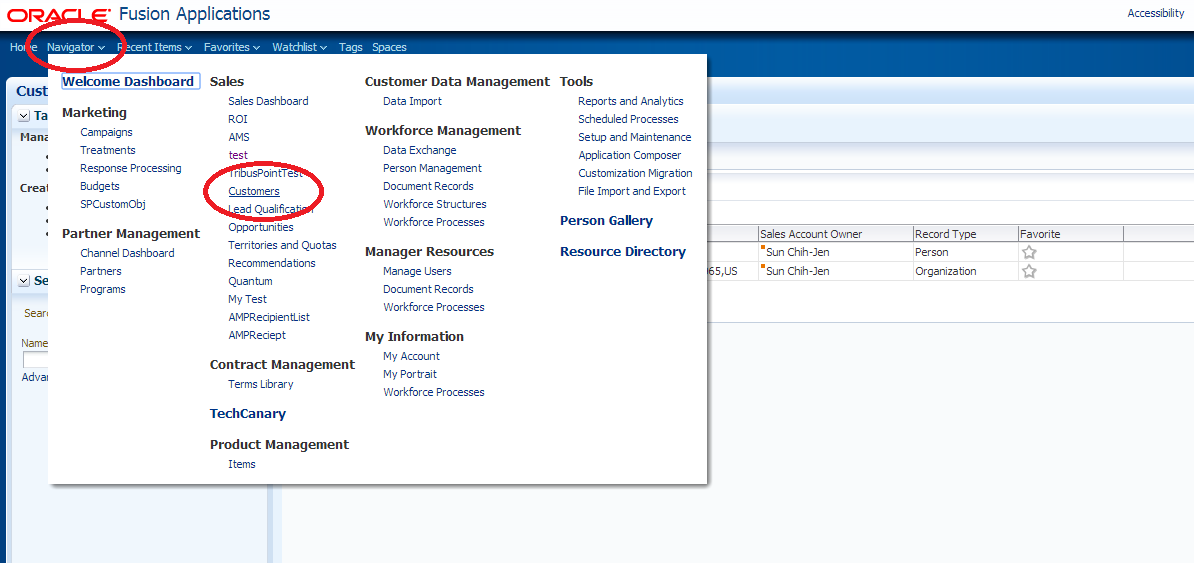
## Attribute Mapping

* Go to https://<JCS\_HOME>/DataSync/faces/crmui.jspx, set ExistingPatient and InsuranceStatus, then click Save



## Create Customer in Oracle Sales Cloud

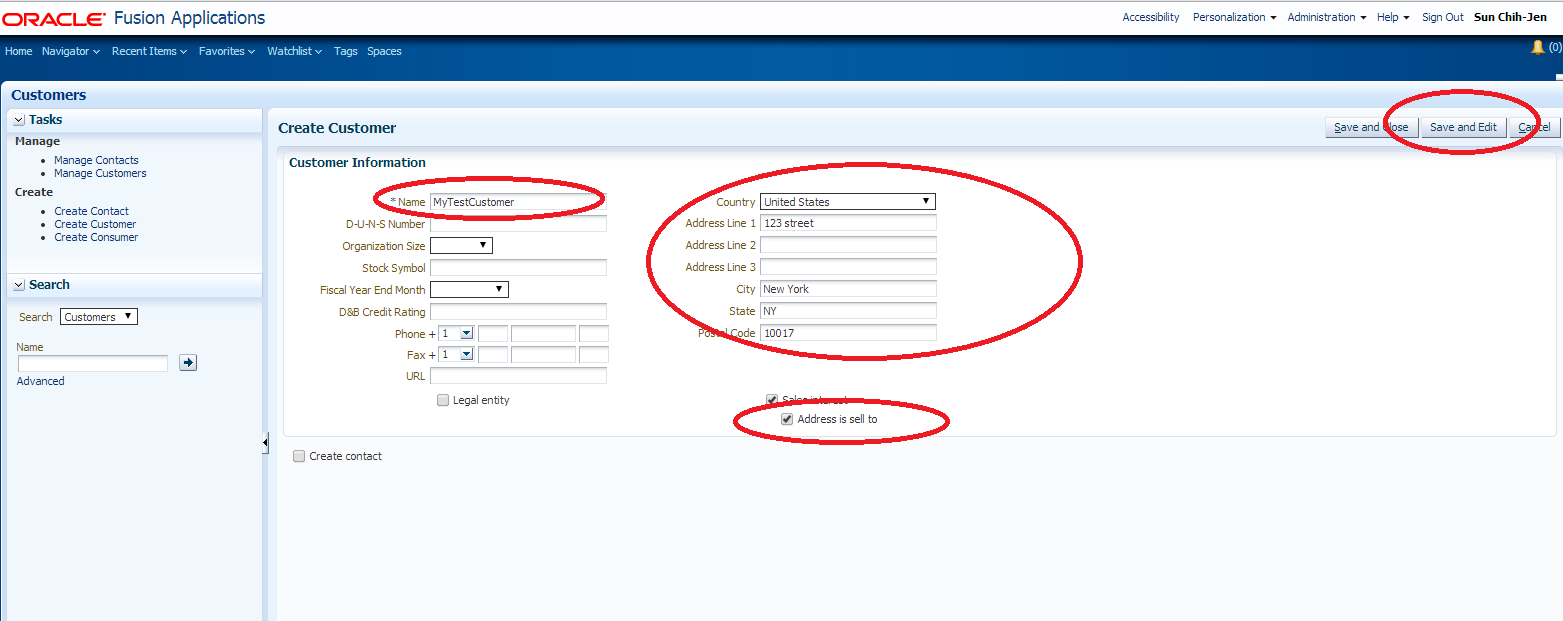
* Click Navigator -> Customers



* Click ‘+’ icon to create a customer

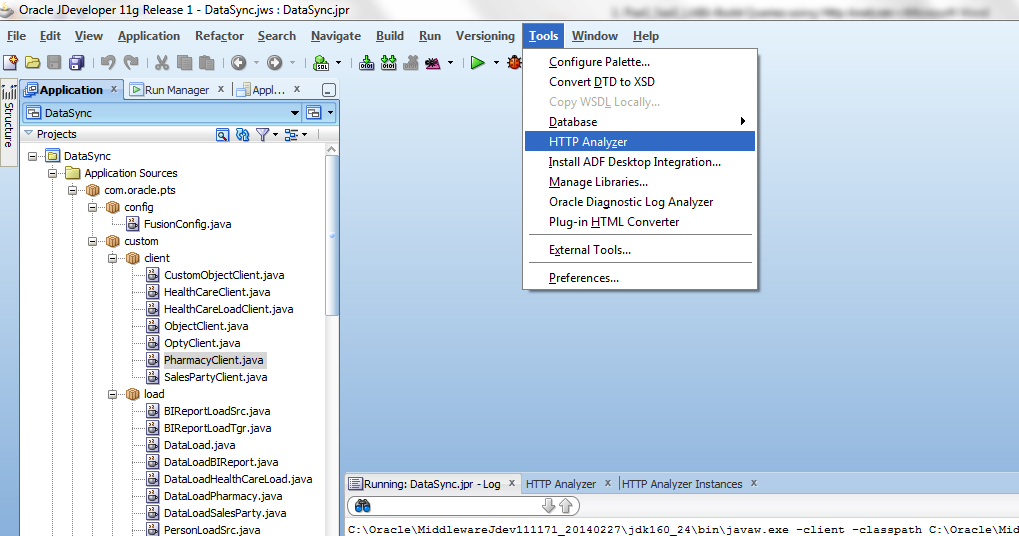


* Set Name as MyTestCustomer, an address and check Address is sell to checkbox, then click Save and Edit

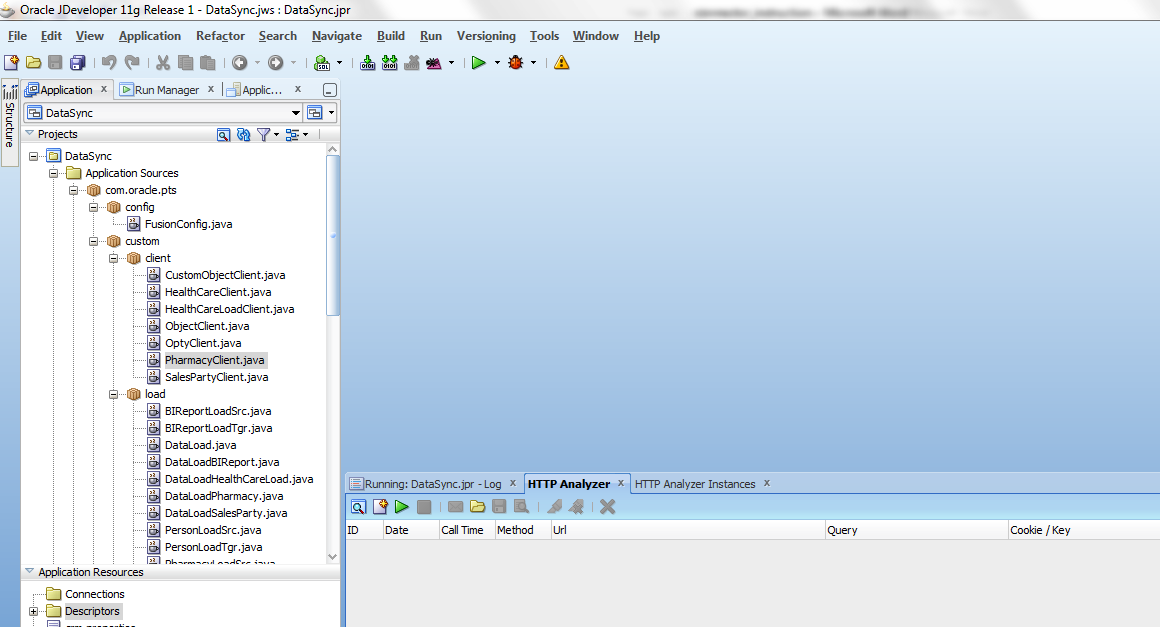


## Set Customer Id

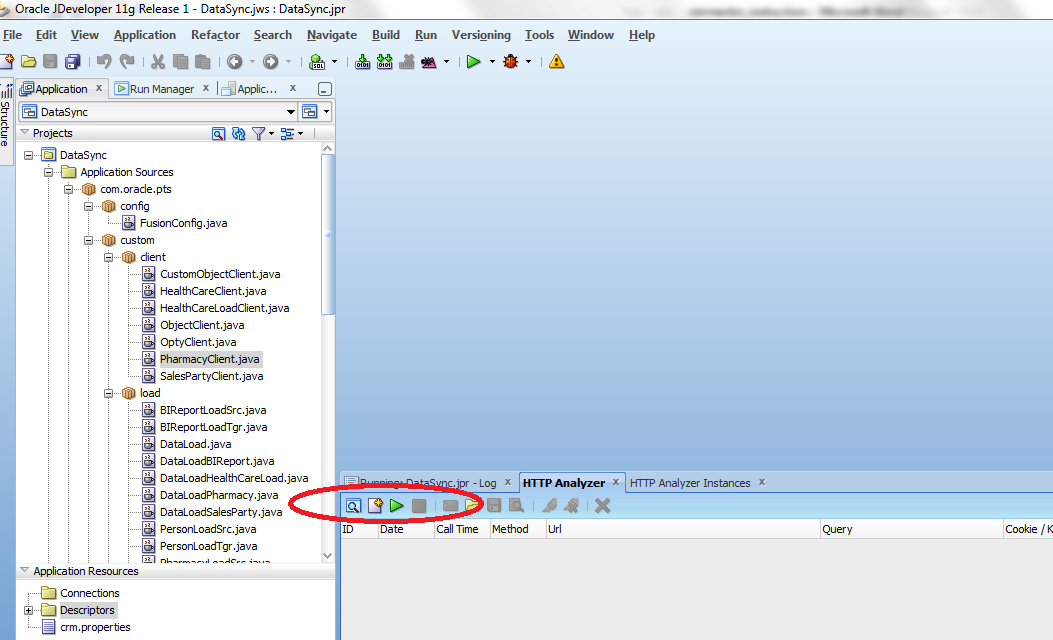
* In JDeveloper, Click Tooks -> HTTP Analyzer



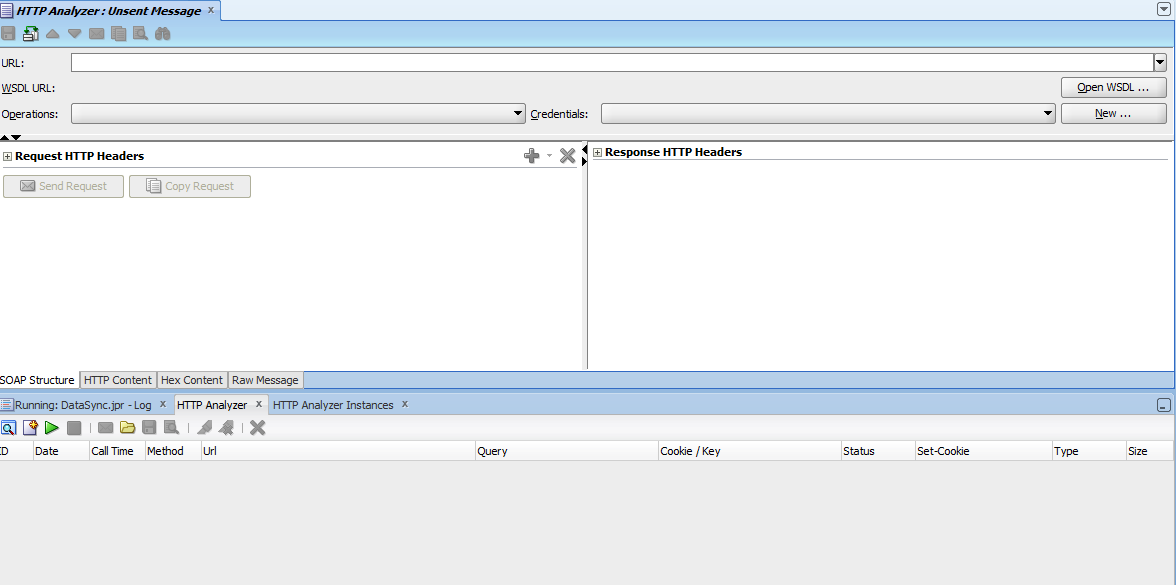
* HTTP Analyzer shows up in the bottom of the IDE.



* Click “create a new request” icon on the bottom of the IDE

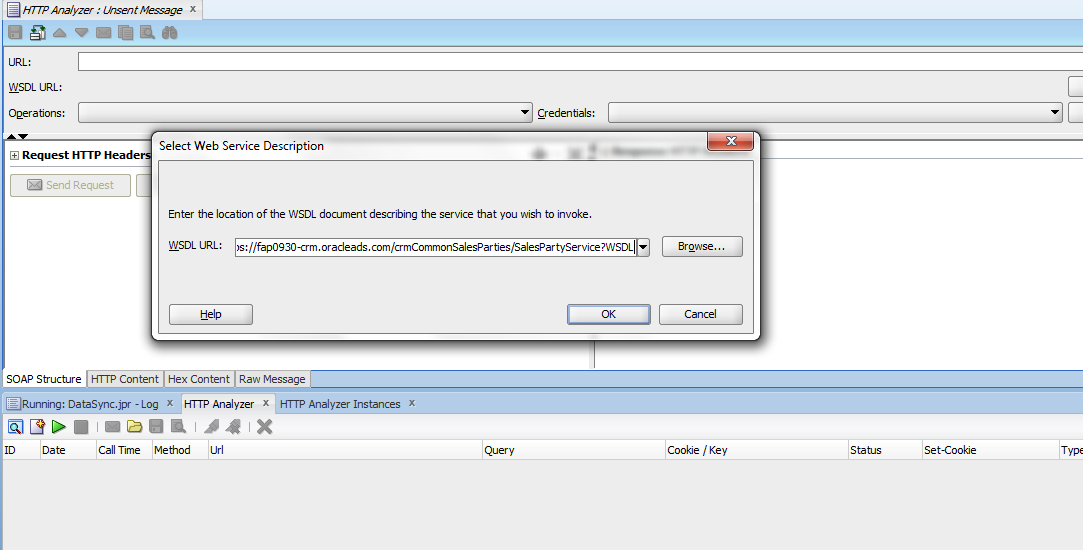


* HTTP Analyzer page shows up

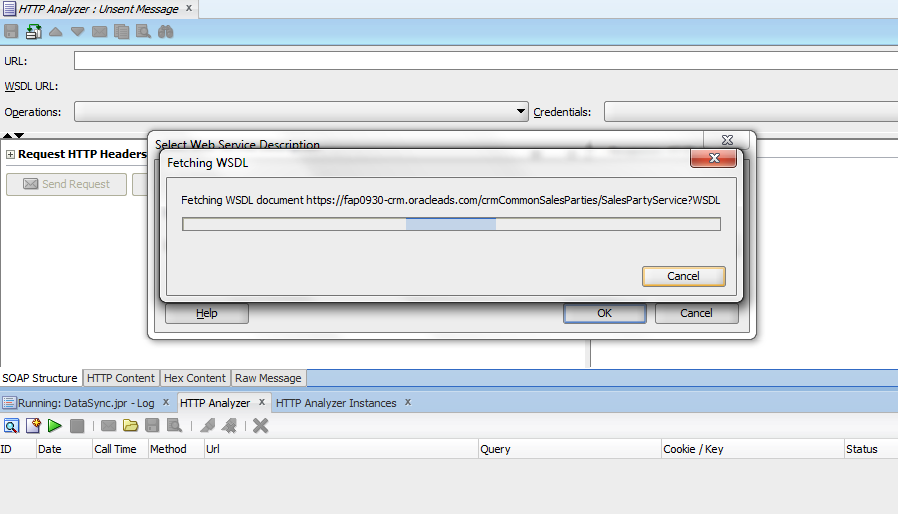


* Click Open WSDL button on the top right side, enter WSDL URL: https://<OSC\_HOME>/ crmCommonSalesParties/SalesPartyService?WSDL and click OK

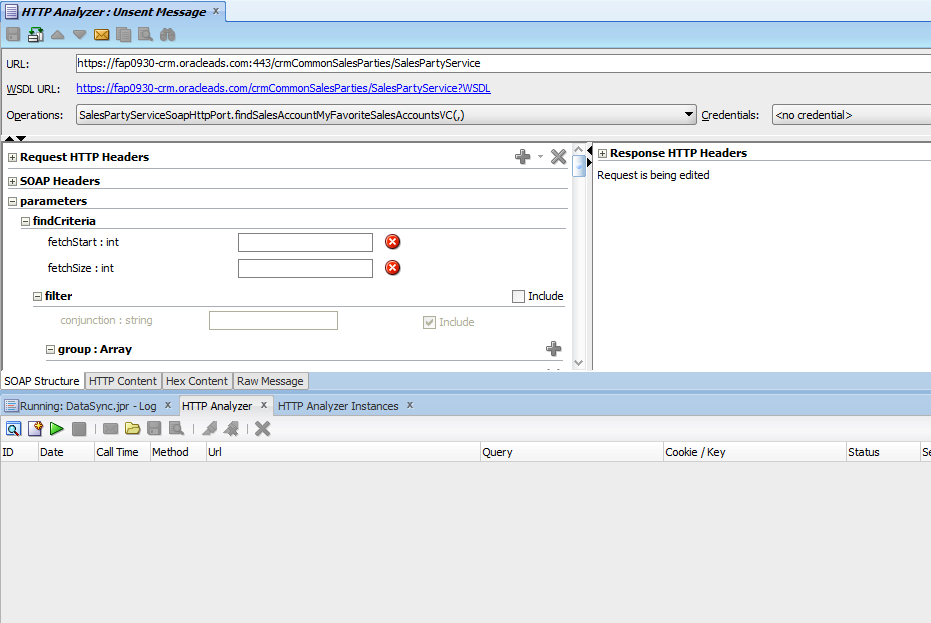
Where OSC\_HOME is Oracle Sales Cloud hostname



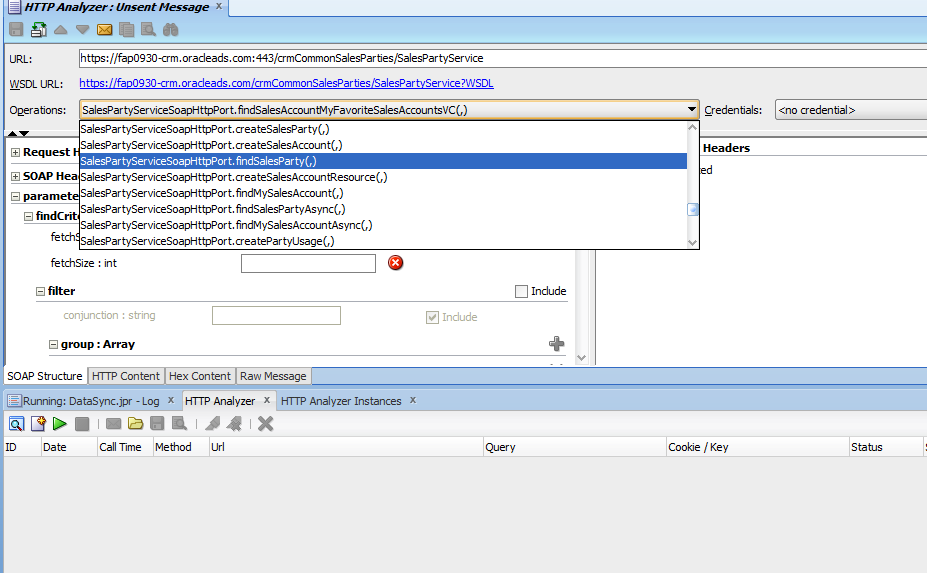
* Wait for fetching WSDL documentation



* Request Template shows up



* Select SalesPartyServiceSoapHttpPort.findSalesParty(,)



* Click on HTTP Content tab



* Copy payload and paste it to HTTP Content

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://xmlns.oracle.com/apps/crmCommon/salesParties/salesPartiesService/ns1es/" xmlns:ns2="http://xmlns.oracle.com/adf/svc/ns1es/">

<soapenv:Header/>

<soapenv:Body>

<ns1:findSalesParty>

<ns1:findCriteria>

<ns2:fetchStart>0</ns2:fetchStart>

<ns2:fetchSize>-1</ns2:fetchSize>

<ns2:filter>

<ns2:conjunction>And</ns2:conjunction>

<ns2:group>

<ns2:conjunction>And</ns2:conjunction>

<ns2:item>

<ns2:conjunction>And</ns2:conjunction>

<ns2:upperCaseCompare>true</ns2:upperCaseCompare>

<ns2:attribute>PartyName</ns2:attribute>

<ns2:operator>=</ns2:operator>

<ns2:value>MyTestCustomer</ns2:value>

</ns2:item>

</ns2:group>

</ns2:filter>

</ns1:findCriteria>

<ns1:findControl>

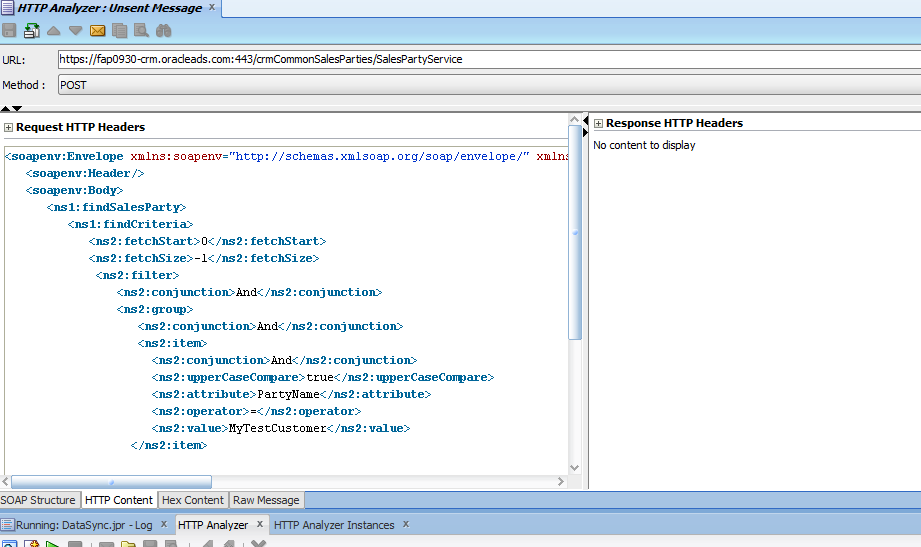
<ns2:retrieveAllTranslations>false</ns2:retrieveAllTranslations>

</ns1:findControl>

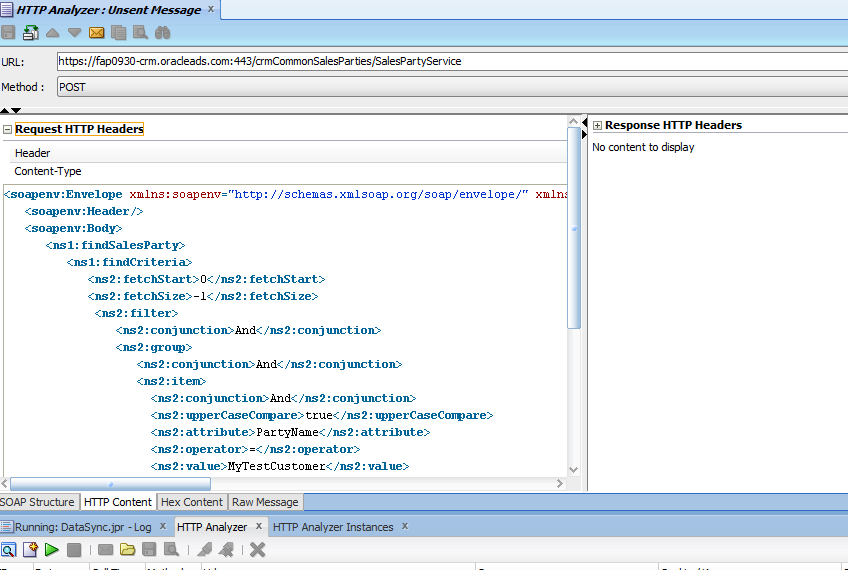
</ns1:findSalesParty>

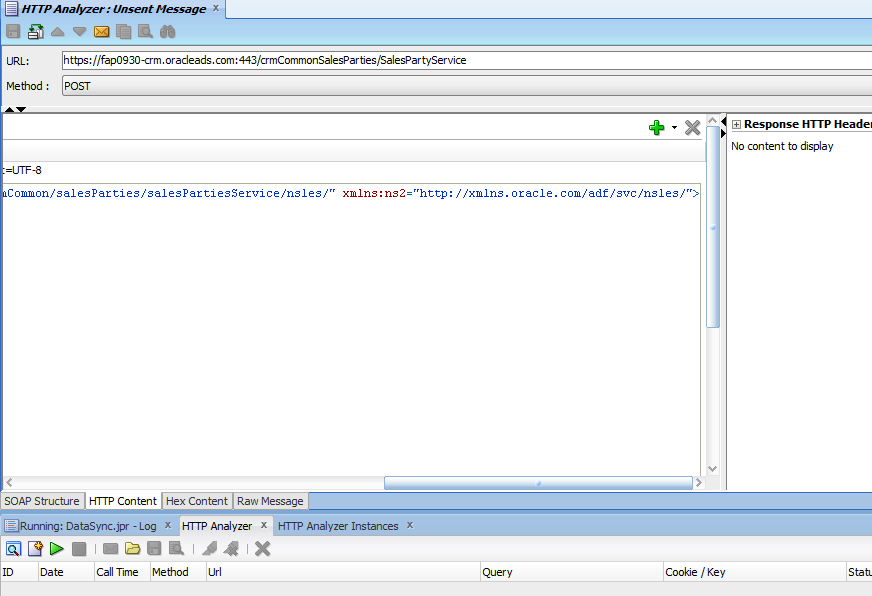
</soapenv:Body>

</soapenv:Envelope>

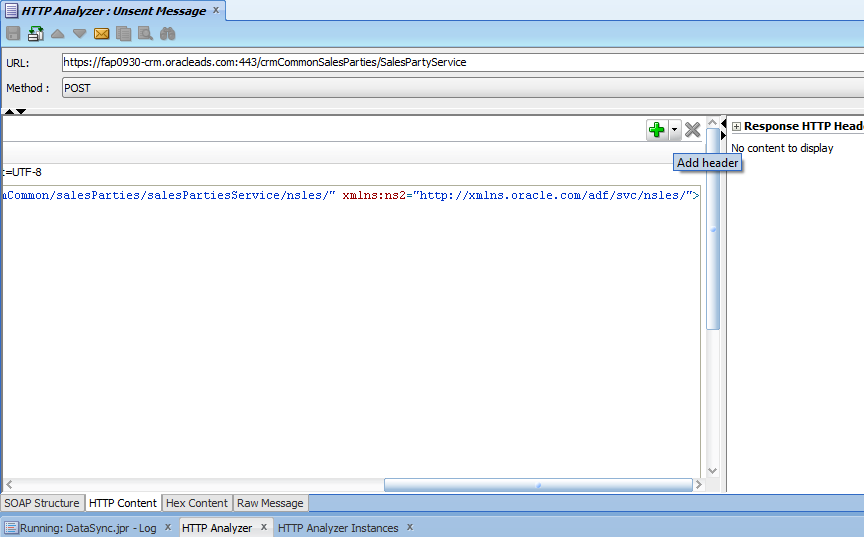


* Click SOAP Structure tab and click “Request HTTP Headers

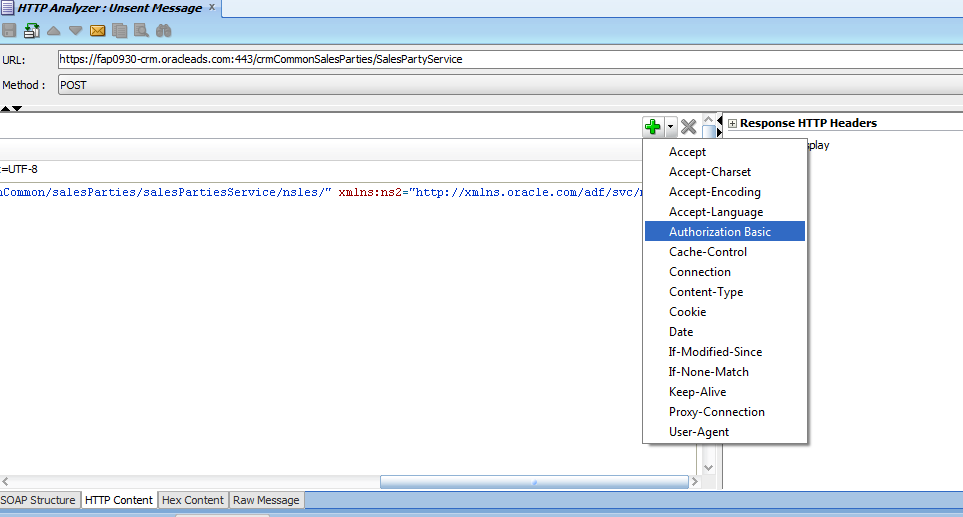




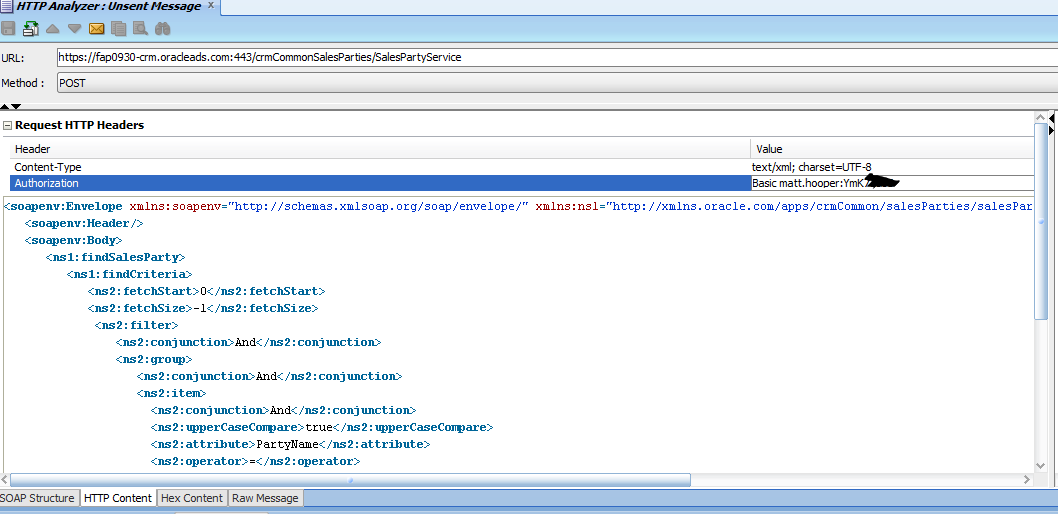
* Click “+” Add headers icon



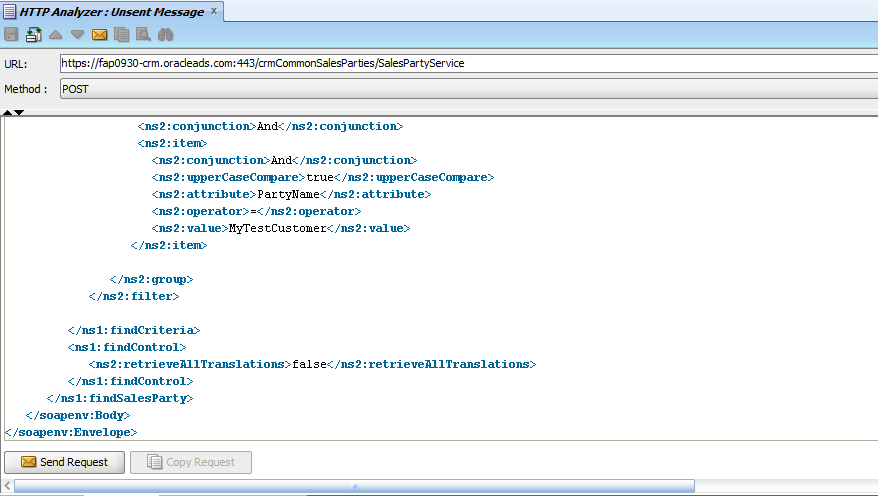
* Choose “Authorization basic”



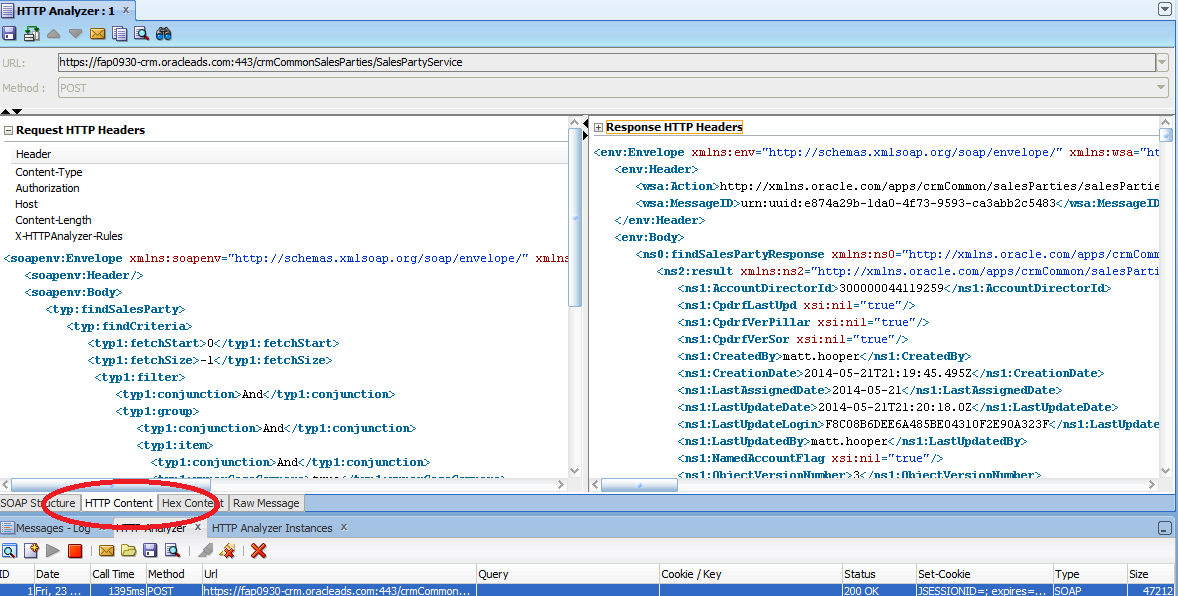
* Fill up username and password



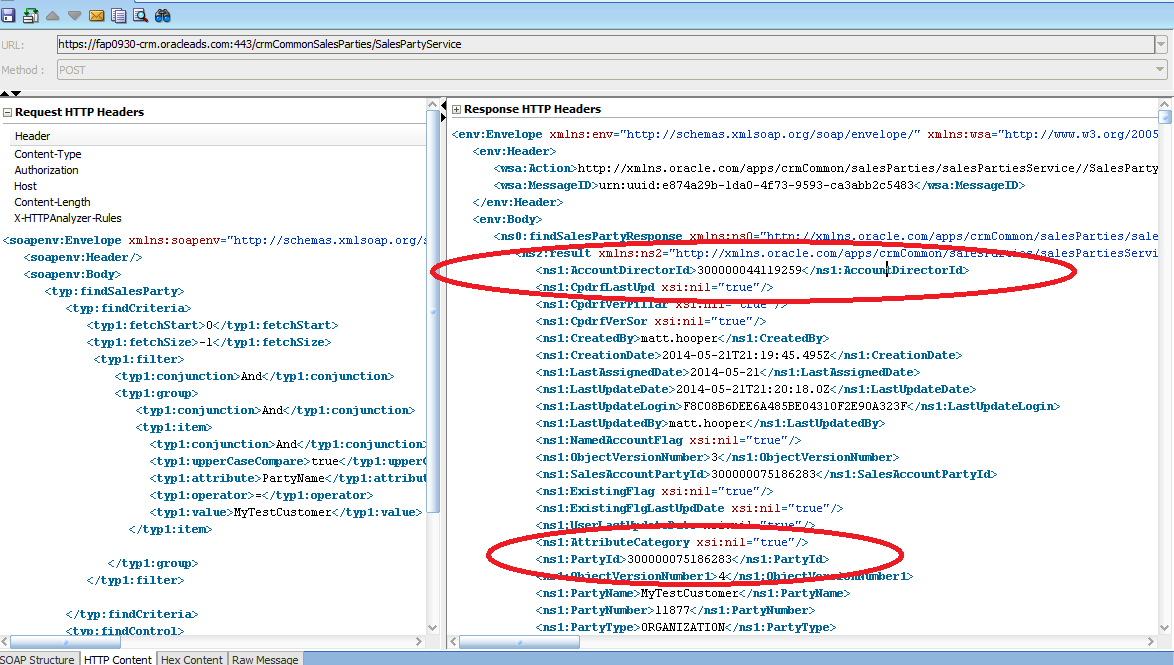
* Click “Send Request” button



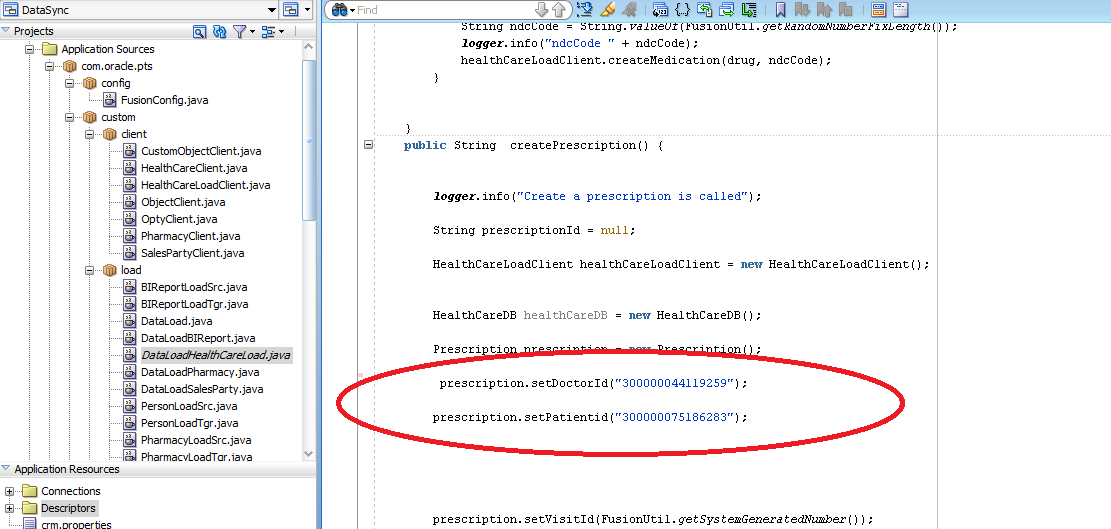
* Get the result back
* Click HTTP Content



* Copy AccountDirectorId and PartyId to a text editor and we will use these Ids to search OSC SalesParty from DataSync app in JCS

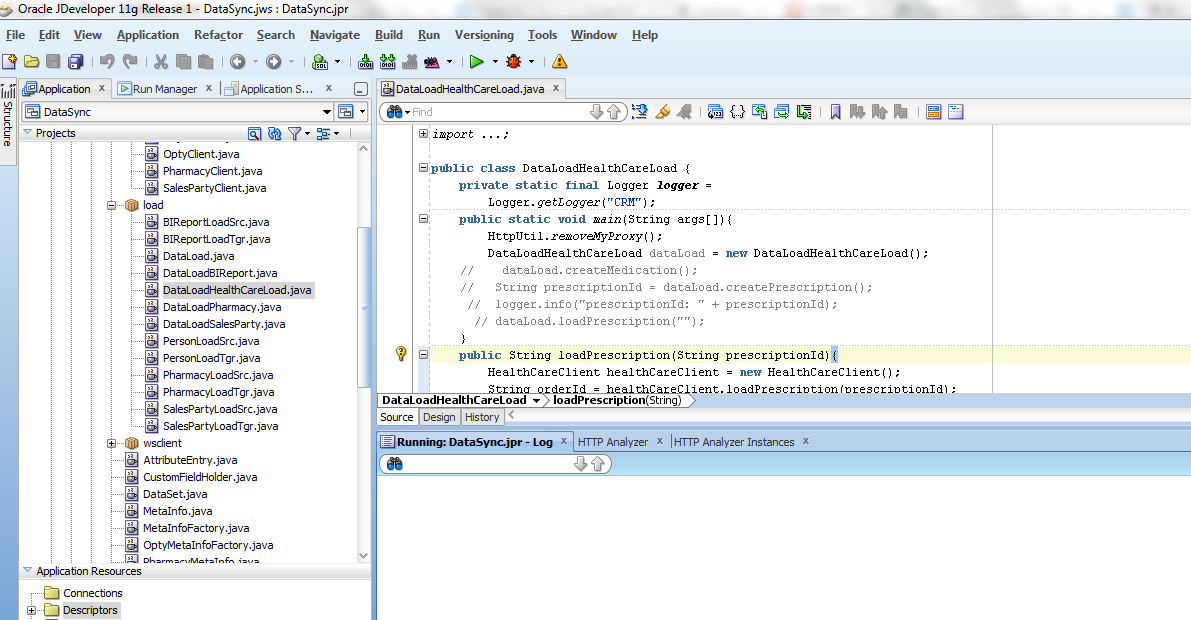


* Open DataLoadHealthCareLoad.java and set DoctorId and Patientid in createPrescription() method where DoctorId is AccountDirectorId and Patientid is PartyId from the precious step



## Create a prescription in JCS

* Open DataLoadHealthCareLoad.java



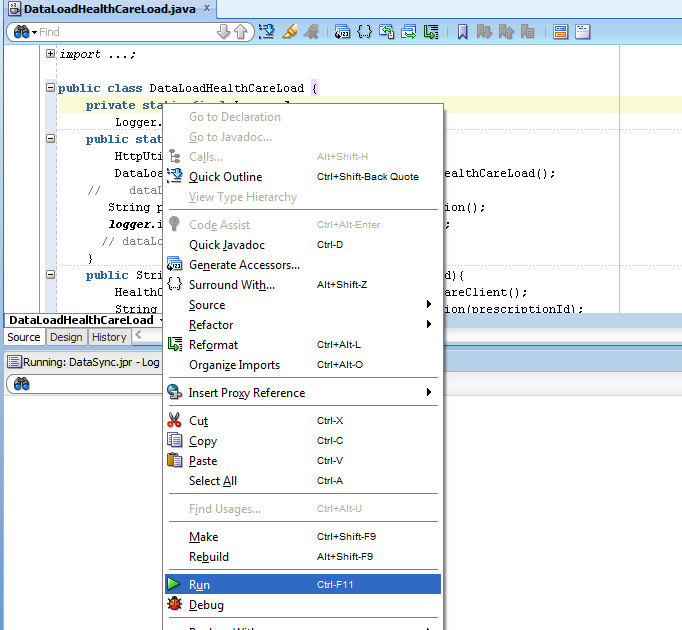
* Modify the code as below

This will call JCS web service to create a prescription in JCS

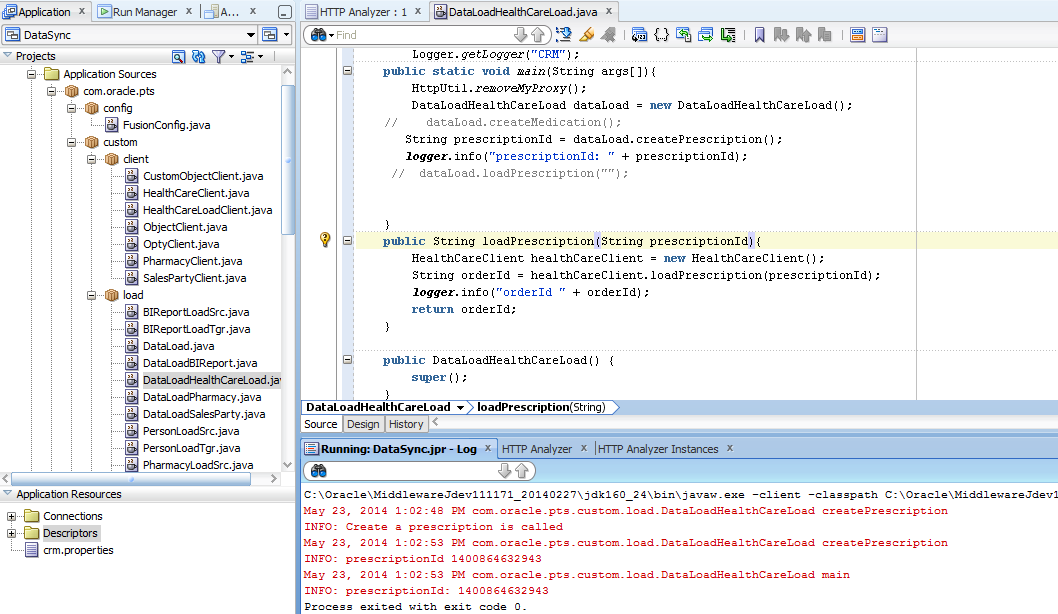
|  |
| --- |
| String prescriptionId = dataLoad.createPrescription();  logger.info("prescriptionId: " + prescriptionId); |



* Right click DataLoadHealthCareLoad .java file and click Run



* Get the prescriptionId back from JCS



## Send a prescription to the pharmacy

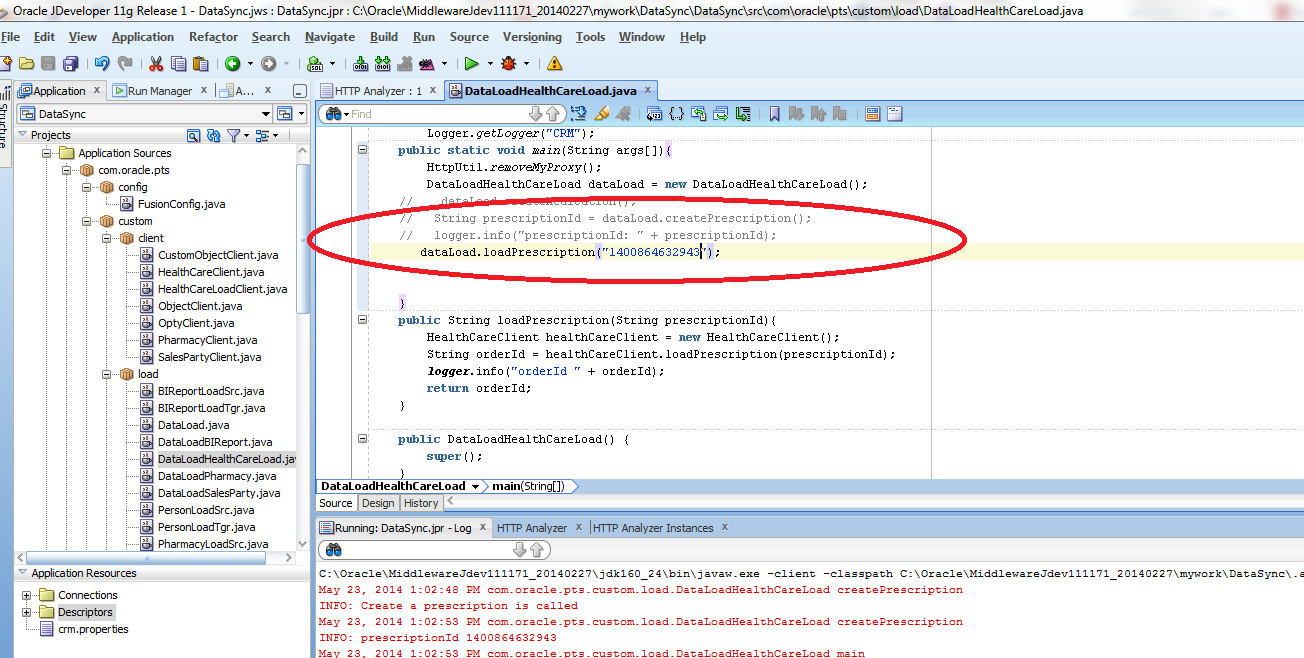
* Modify the code as below and pass prescriptionId got from the previous step to loadPrescription method

This will call JCS web service, and JCS will get the patient and doctor data from Oracle Sales Could and prescription from JCS and send the prescription to the pharmacy by calling the pharmacy web service.

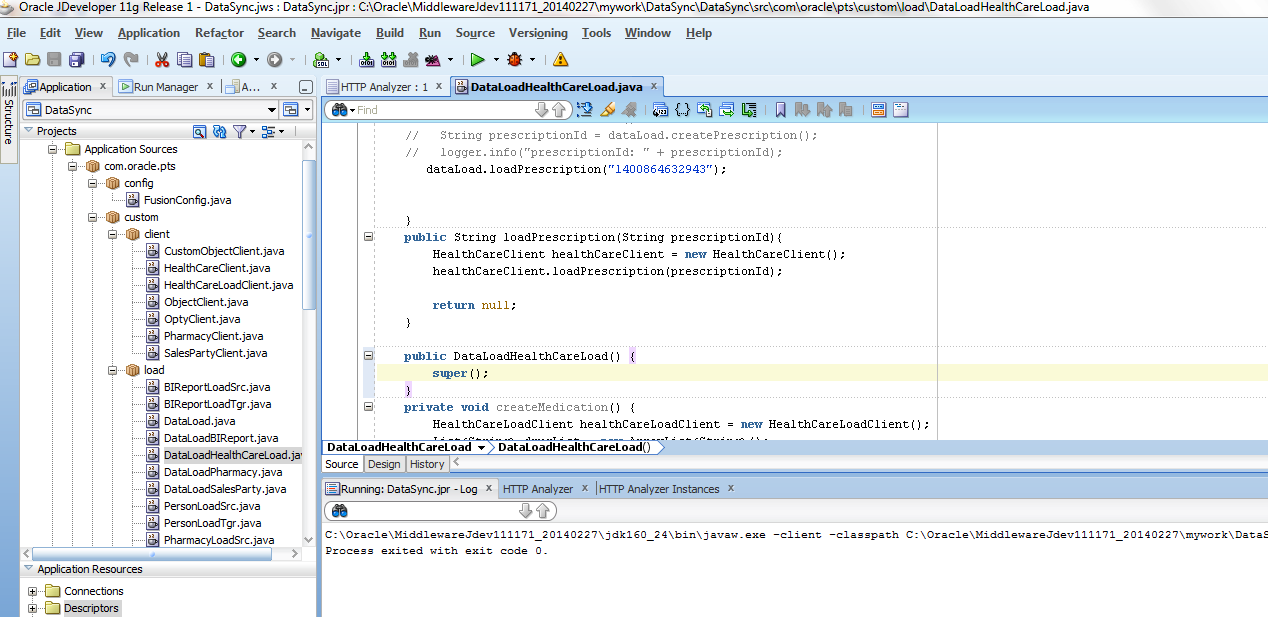
|  |
| --- |
| // String prescriptionId = dataLoad.createPrescription();  //logger.info("prescriptionId: " + prescriptionId);  dataLoad.loadPrescription("1400792058946"); |

This will call JCS web service to create a prescription in JCS

|  |
| --- |
| // String prescriptionId = dataLoad.createPrescription();  // logger.info("prescriptionId: " + prescriptionId); |

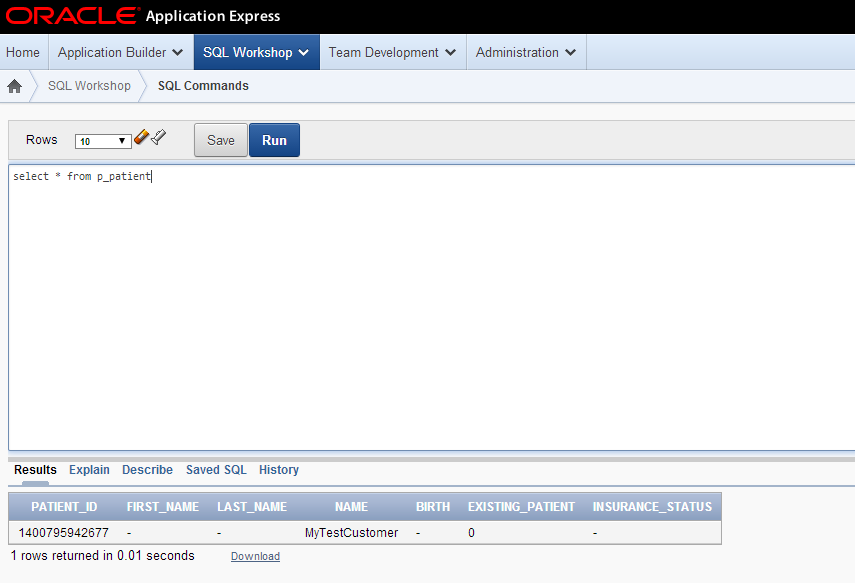


* Right click DataLoadHealthCareLoad .java file and click Run



* Go to DBCS admin console and Run SQL command, select \* from p\_patient

p\_patient is the database from the Pharmacy application. By running this dataSync from previous step, patient is created in the Pharmacy site. Patient data is Customer data in Oracle Sales Cloud created in the previous step.



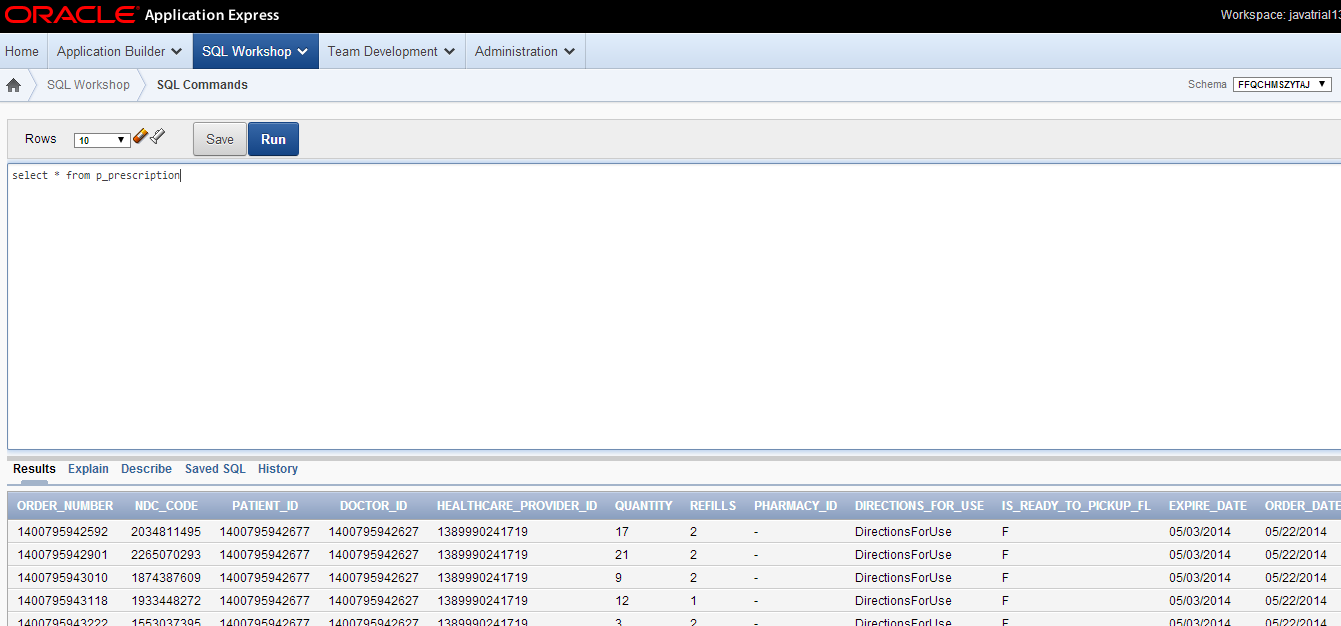
* Run SQL command, select \* from p\_doctor

p\_doctor is the database from the Pharmacy application. By running this dataSync from previous step, doctor is created in the Pharmacy site. Doctor is owner for the Customer in Oracle Sales Cloud created in the previous step.

# 

* Run SQL command, select \* from p\_prescription

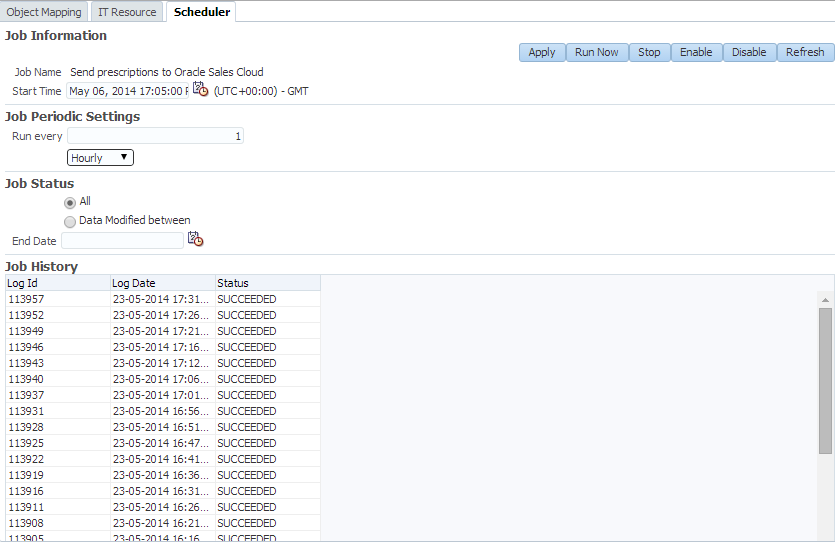
p\_prescription is the database from the Pharmacy application. By running this dataSync from previous step, prescription is created in the Pharmacy site. Prescription is from the DataSync app in JCS created from the previous step.



## Scheduler

* Go to https://<JCS\_HOME>/DataSync/faces/crmui.jspx, click Scheduler tab.

By running this scheduler, data is synchronized from JCS health care database table to Pres\_c, Pres\_med\_c and Med\_c custom object in Oracle Sales Cloud



# Customization

## Mapping

* Create a PartnerAppMetaInfoFactory that extends MetaInfoFactory java class
* Set target objects and attributes

attributeEntry =

getAttributeEntry("Expire Date", "Expire Date");

attributeEntryList.add(attributeEntry);

targetObjectList.put(objectName, metaInfo);

* Set source objects and attributes

attributeEntry = getAttributeEntry("lastName", "lastName");

attributeEntryList.add(attributeEntry);

sourceObjectList.put(objectName, metaInfo);

* Modify or extend CrmMapping java classMapping UI
* Set PartyAppMetaInfoFactory as a MetaInfoFactory instance to set target objects/attributes and source objects/attributues

MetaInfo targetObjectMetaInfo =

metaInfoFactory.getTargetObjectMetaInfo(targetObjectName);

MetaInfo sourceObjectMetaInfo =

metaInfoFactory.getSourceObjectMetaInfo(sourceObjectName);

## Data Synchronization

* Create PartyAppLoad java class
* Get opbject/attribute mapping by calling MappingObjDB java class

MappingObj mappingObj = mappingObjDB.getMappingObj("PartneApp");

* Call OSC web service proxy to get OSC data and push data to partner app
* Or call partner app to get third party’s data and push data to OSC(another direction)

# 