# OIM Customization Installer

### Installing OIM Customization Installer Extension on JDeveloper

Before using OIM Customization Installer,

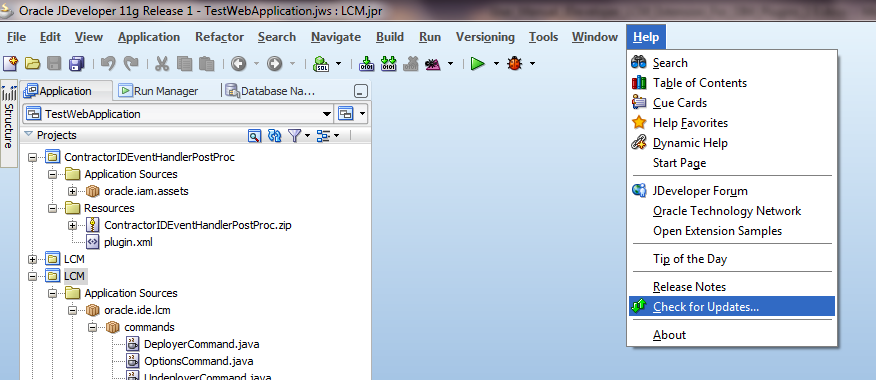
* JDeveloper 11.1.1.x should be installed on the system, which has been done already on the VM
* For any customizatiomn that has to be deployed using this framework, below folder structure should be followed to package its source as it should show up in the JDeveloper IDE. The package of all the customization artifacts that will be used in this training course, already follow this recommended structure.

|  |  |
| --- | --- |
| **Folder Structure** | **Description** |
| PROJECT-CONTEXT]/config | All .xml files |
| [PROJECT-CONTEXT]/resources | All the required resources files |
| [PROJECT-CONTEXT]/lib | All the required jar files |
| [PROJECT-CONTEXT]/src | All java source files |

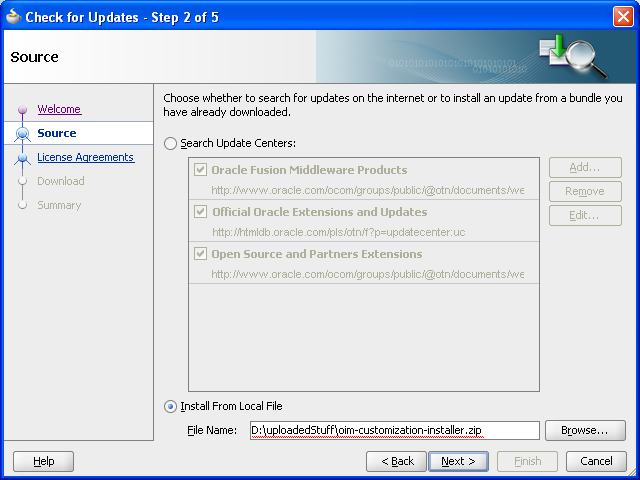
**Please note that the above folder structure is mandatory. A directory can be left empty, but should not be missed out from the folder structure.**

#### Install OIM Customization Installer Extension on JDeveloper

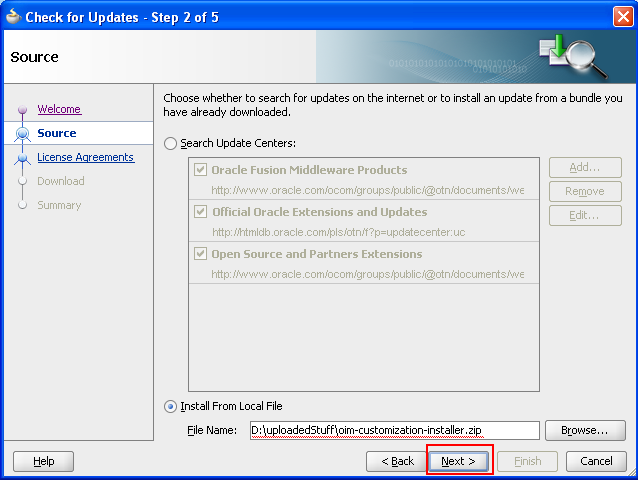
* + Select menu “Help” -> “Check for Updates”



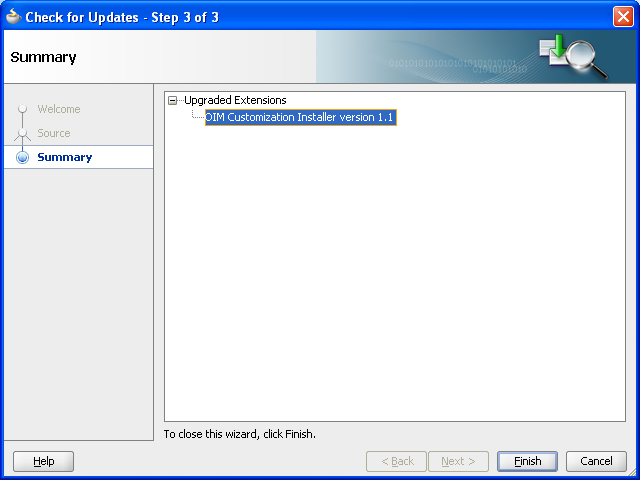
* + Select “Install from Local File” from the wizard.



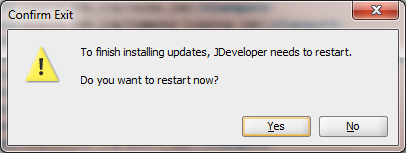
* + Browse and select the oim-customization-installer.zip package and click Next



* + Click on “Finish”.

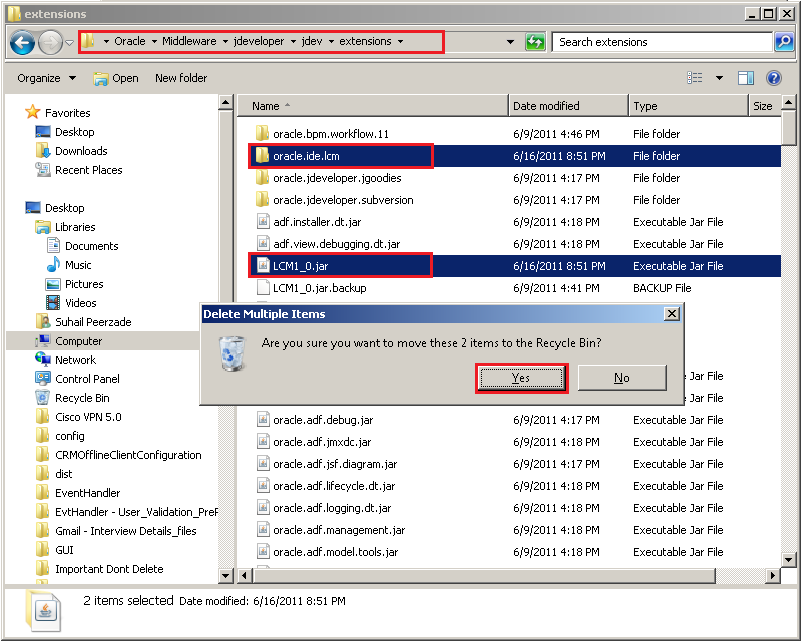


* + Restart JDeveloper

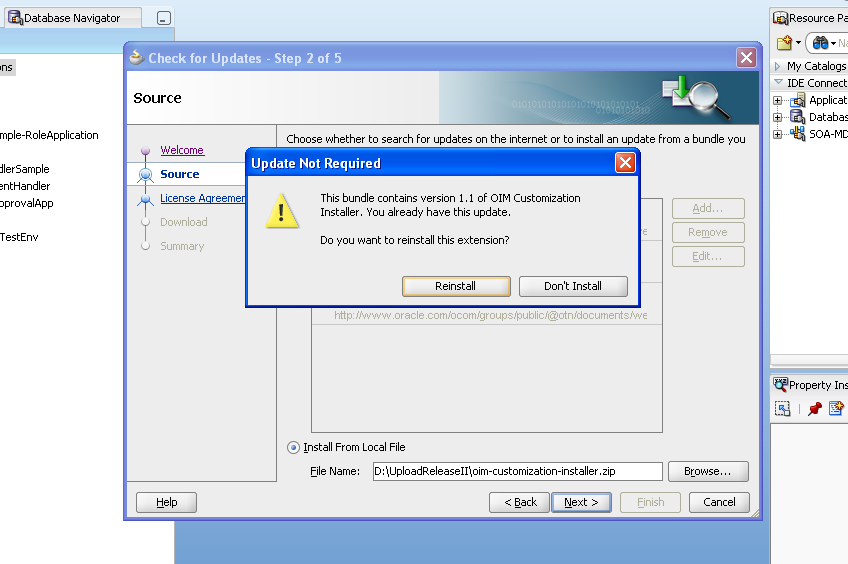


#### Reinstall OIM Customization Installer Extension on JDeveloper

* Manually delete all the files from the **{JDEVELOPER-HOME}/jdev/extensions** *folder (See the screenshots below)*.
  + **{JDEVELOPER-HOME}/jdev/extensions/OIMCustomizationInstaller.jar (JAR file to be deleted)**
  + **{JDEVELOPER-HOME}/jdev/extensions/oracle.ide.lcm (Folder to be deleted)**
  + Other backup files such as OIMCustomizationInstaller.jar.backup may also be deleted. However it is not necessary that you delete them**.**



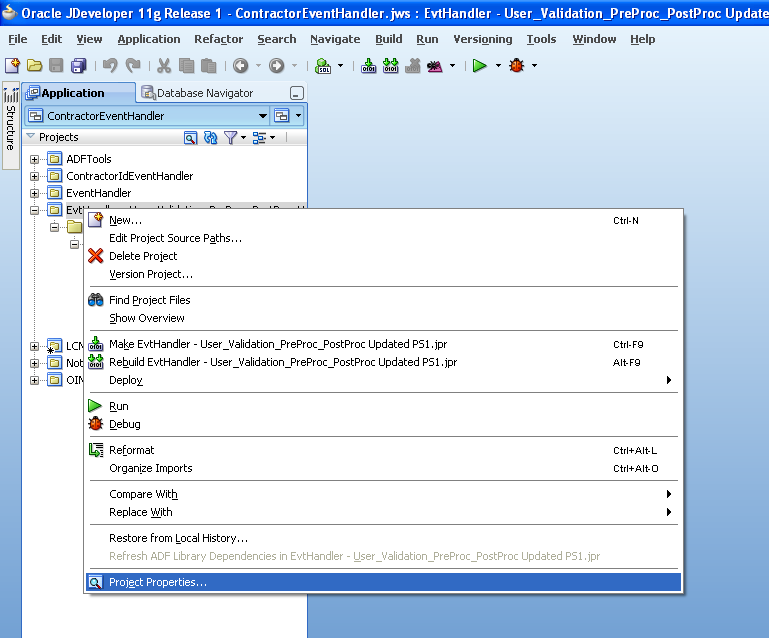
* + *To Install/Reinstall OIM Customization Installer,* the process of reinstallation of JDeveloper is similar as installation. If plug-in is already installed then JDeveloper will show below screen.

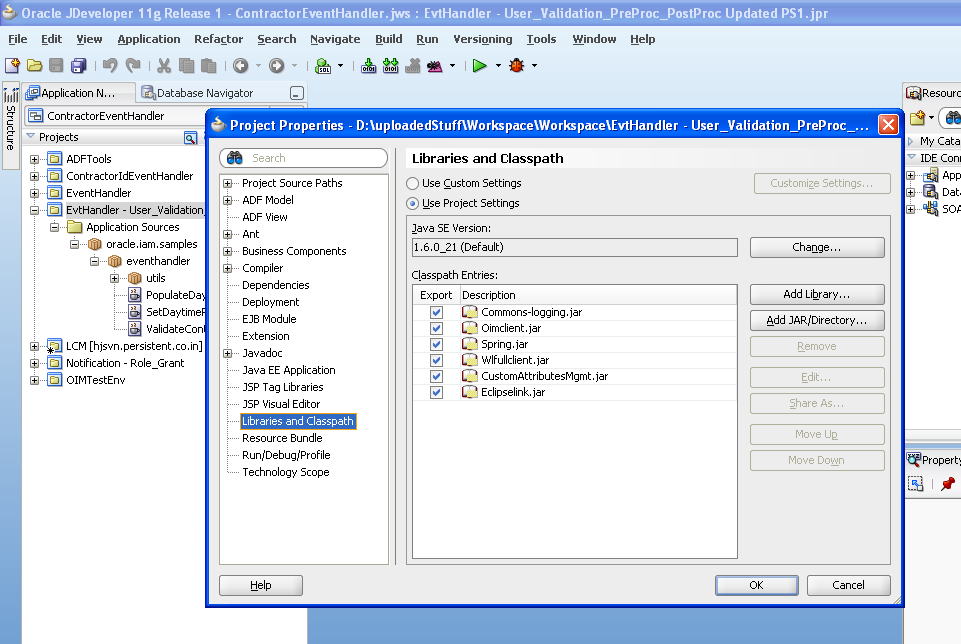
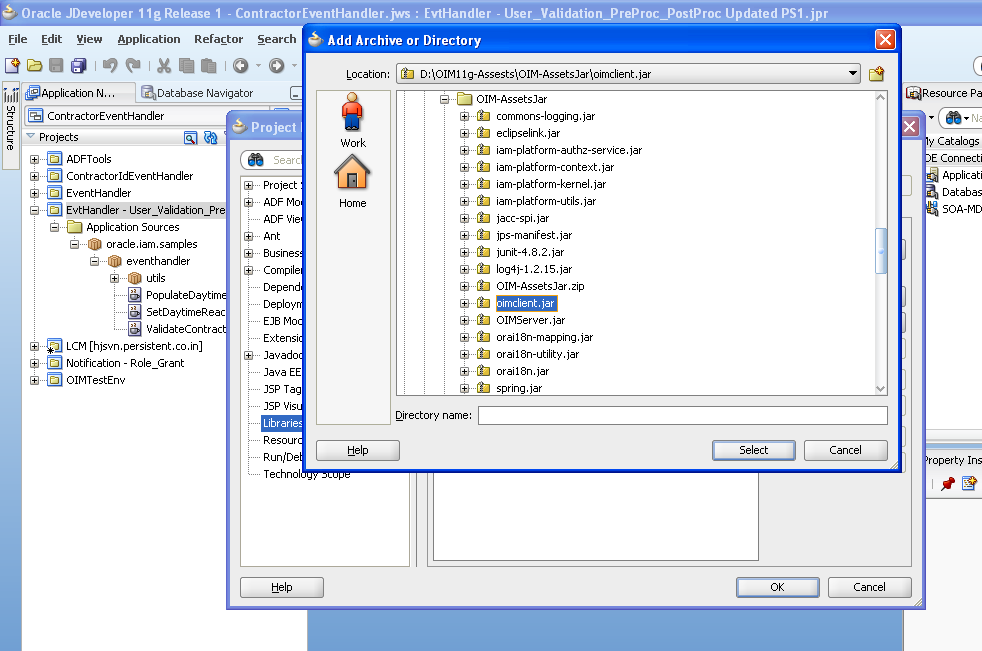


#### Java Project Build Path settings in JDeveloper

Please make sure you have added all required jar files for your plug-in java project in JDeveloper.

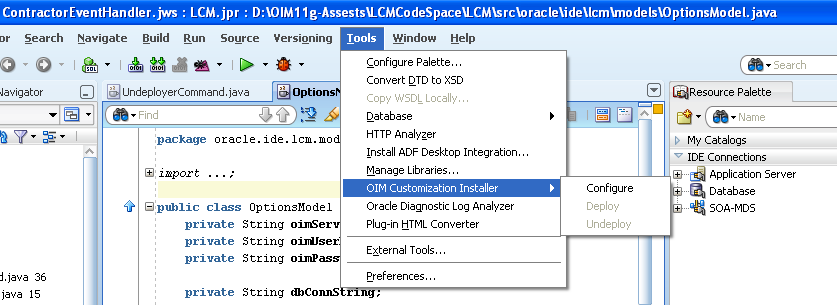
* + Select desired project and right click on “Project Properties”.



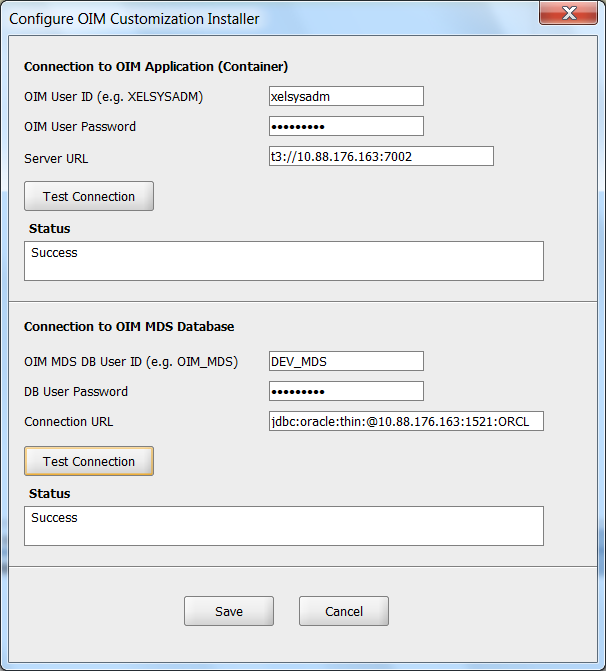
* + Select “Libraries and Classpath” and click on “Add JAR/Directory”.
  + Browse and add required JAR from local disk. 

#### Using OIM Customization Installer Extension for OIM Plugins

* + After restart, “OIM customization installer” Menu will be displayed under “Tools”.



* + Select “OIM customization installer” -> “Configure”

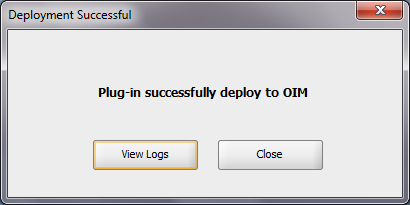


After saving the configuration details, user can verify connection using “Test connection” button.

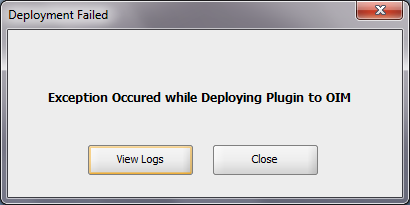
* + Before trying to deploy any project, please make sure you have added the dependent JAR files in the **project classpath** and have good package folder structure in place.
  + If you wish that these dependent JAR files should also be moved to OIM, then copy these files to **[PROJECT-CONTEXT]/lib** folder.
  + To perform single click deployment of implemented plug-in in JDeveloper,

“Tools” -> “OIM customization installer”-> “Deploy”

* + If the deployment is successful, a success dialog will be shown.



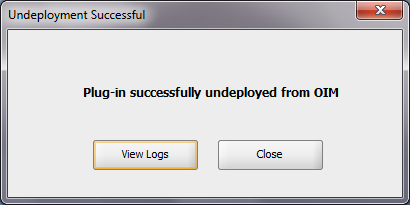
* + If the deployment fails, a failure dialog will be shown.



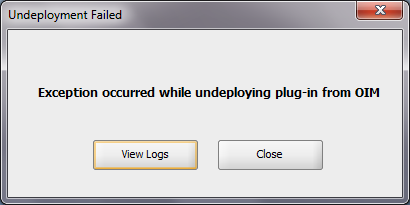
* + If a plug-in is not already installed, it will show up an error window.
  + To perform single click undeployment of deployed plug-in,

“Tools” -> “OIM customization installer”-> “undeploy”

* + If the undeployment is successful, a success dialog will be shown.



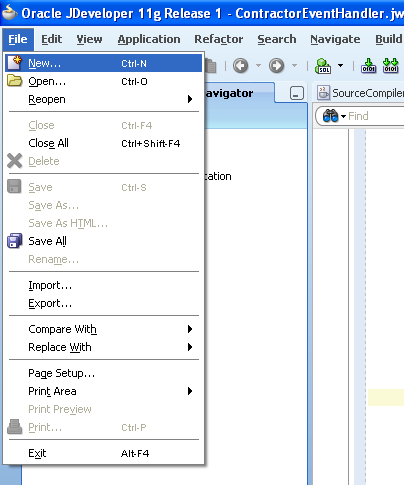
* + If the deployment fails, a failure dialog will be shown.



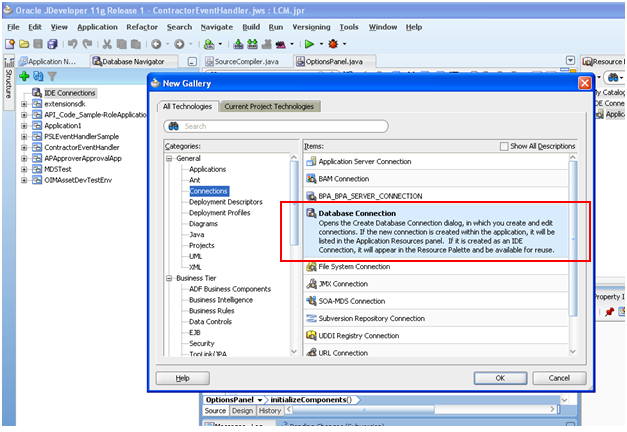
### Verification of plug-in deployment / undeployment from JDeveloper

#### Creating Database Connection in JDeveloper.

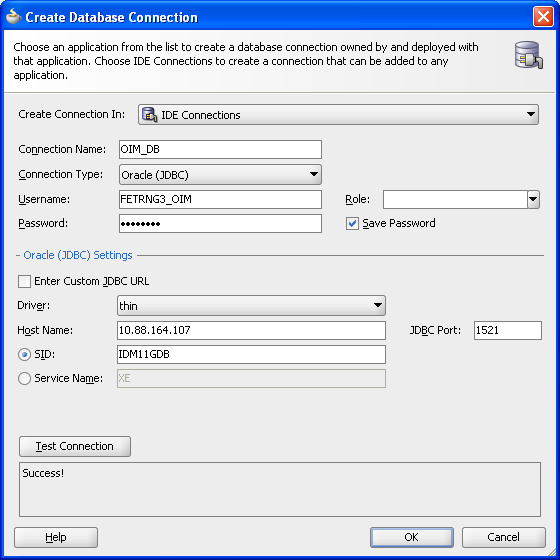
* + Open JDeveloper and select “File” -> “New”.



* + Select “General” -> “Connection” -> “ Database connection”

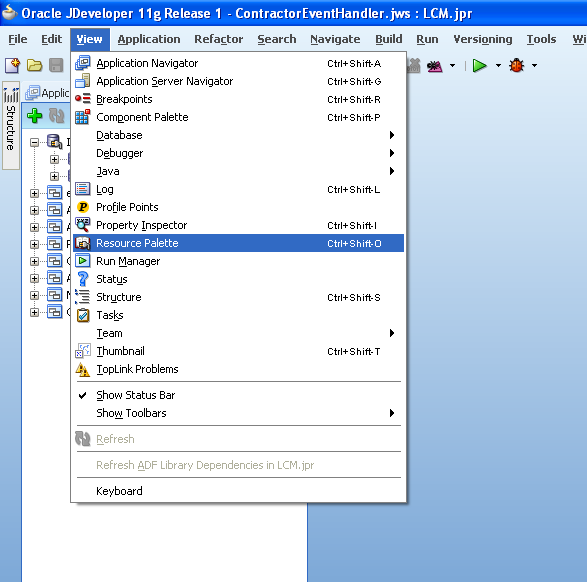


* + Fill all required information to create OIM Database.

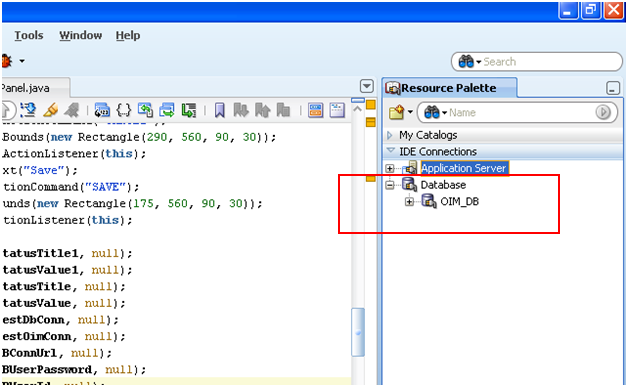


Verify connection status to OIM Database using “Test Connection”

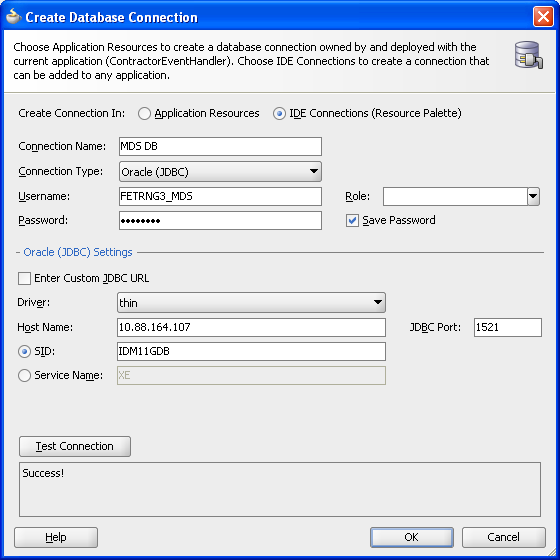
* + Open “Resource Palette” to navigate the created database connection



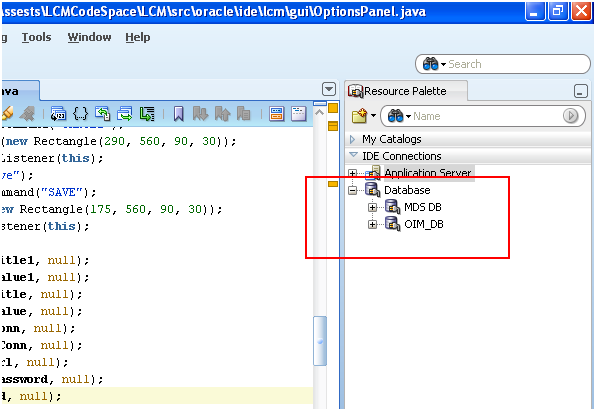
* + You can see created database connection in “Resource Palette”



* + Create Database connection to MDS and verify using “Test Connection”

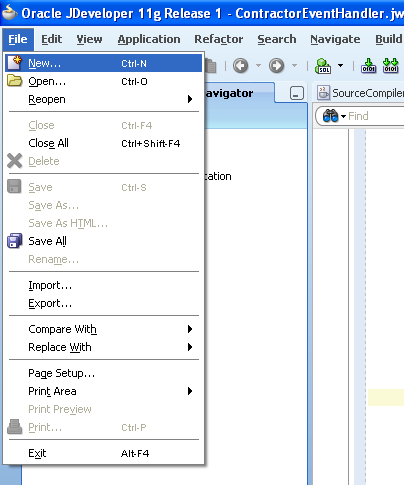


* + You can verify database connection under “Resource Palette”.

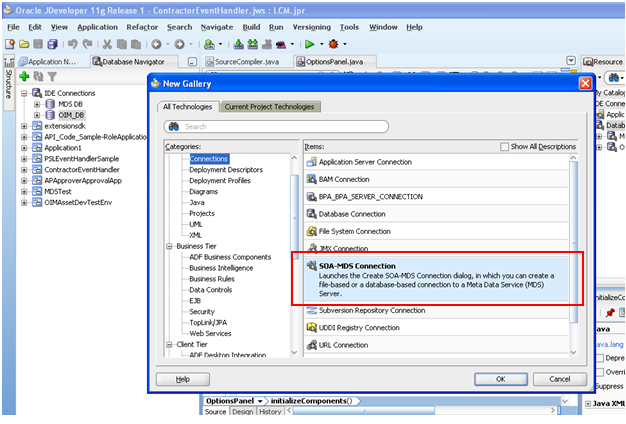


#### Creating SOA- MDS Connection

* Select “File” -> “New” from the File menu in JDeveloper window.



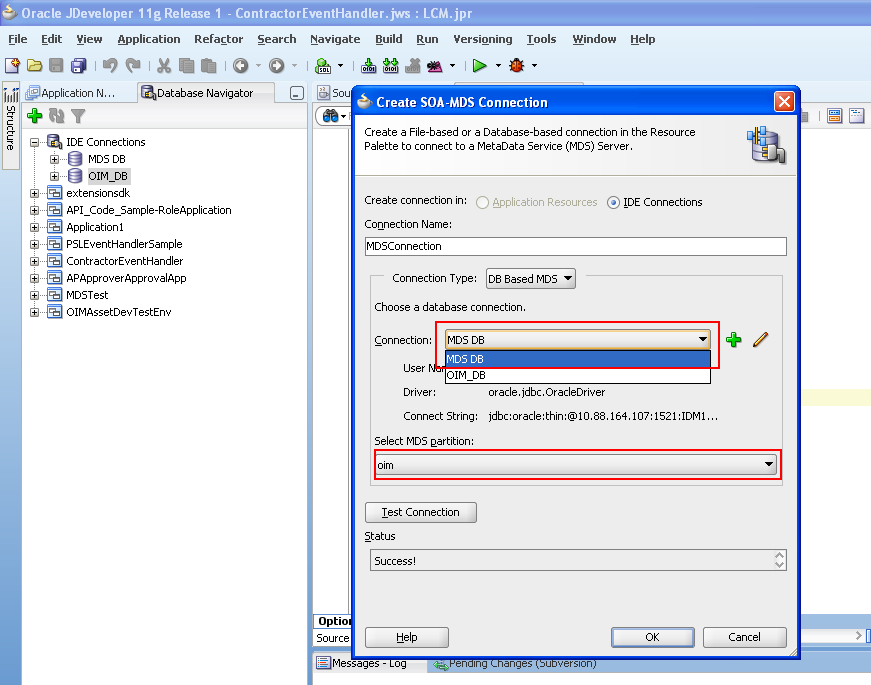
* Create SOA-MDS Connection. [This option will be available only if the “Oracle SOA Composite Editor” extension is installed on JDdeveloper. This can be downloaded from “[Oracle SOA Composite Editor](http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/156082.xml)”].



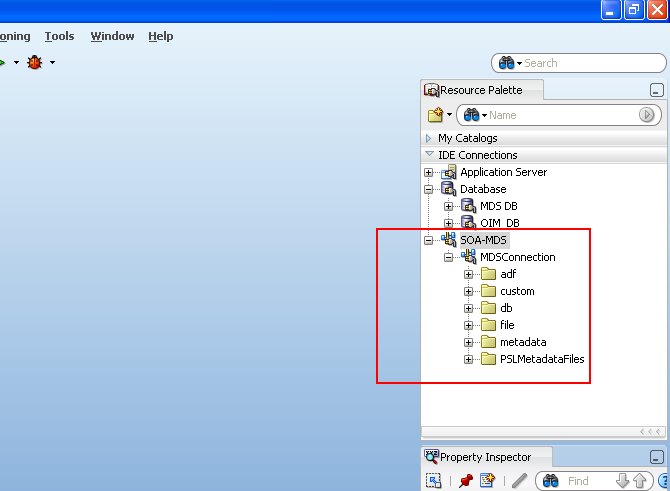
SOA-MDS connection is nothing but a user-friendly interface to view MDS database details. All the metadata files that have been imported in MDS can be viewed easily in a hierarchical tree structure.

* + Create SOA –MDS connection.

Please make sure you have selected “Connection” to MDS database and MDS partition as “oim”.



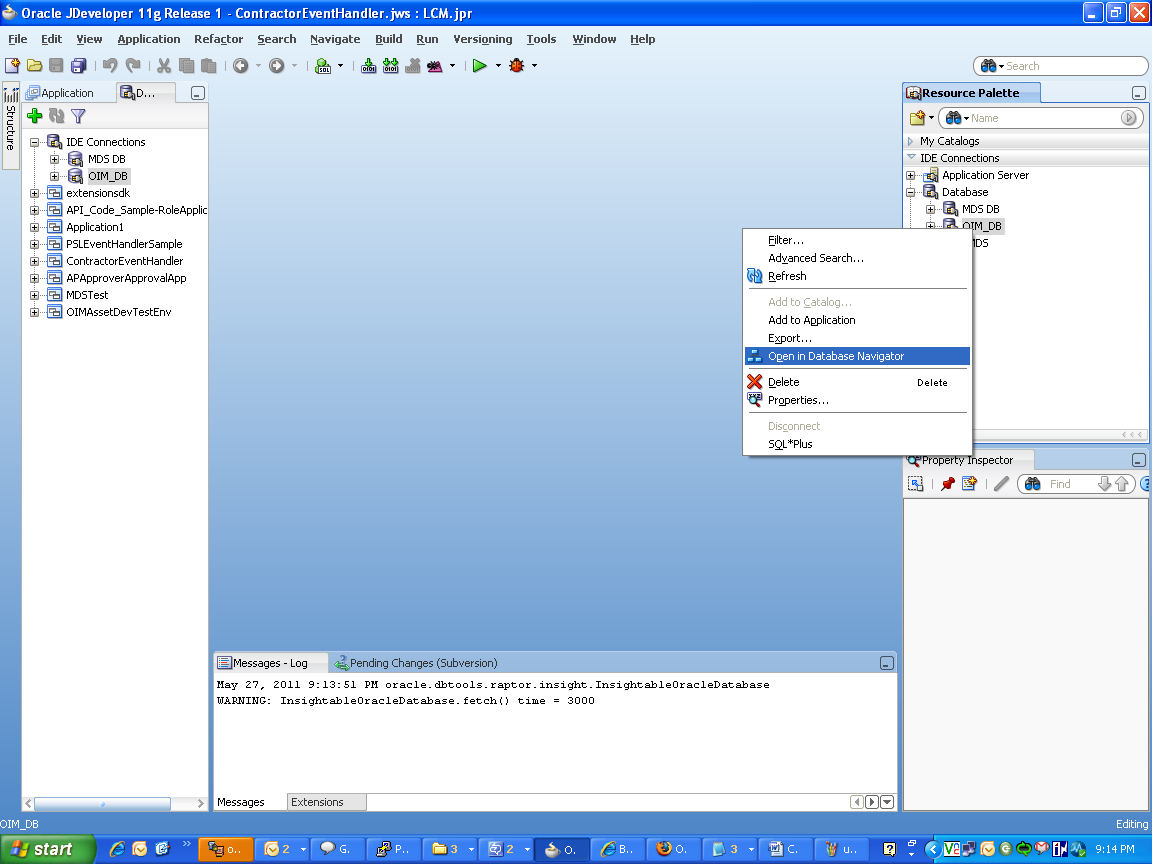
You can view all uploaded Metadata files using SOA-MDS connection in hierarchical tree structure.



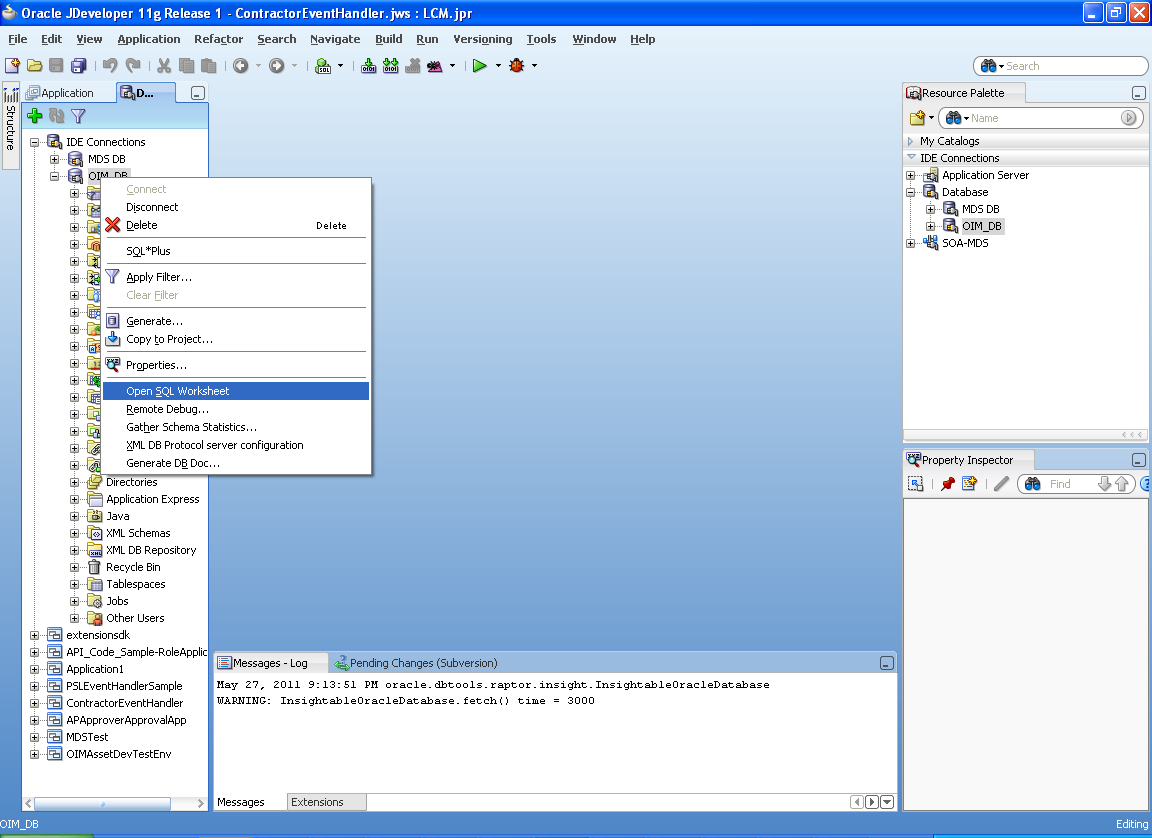
#### Verification of deployed plug-in

* + Verify plug-in table in OIM DB

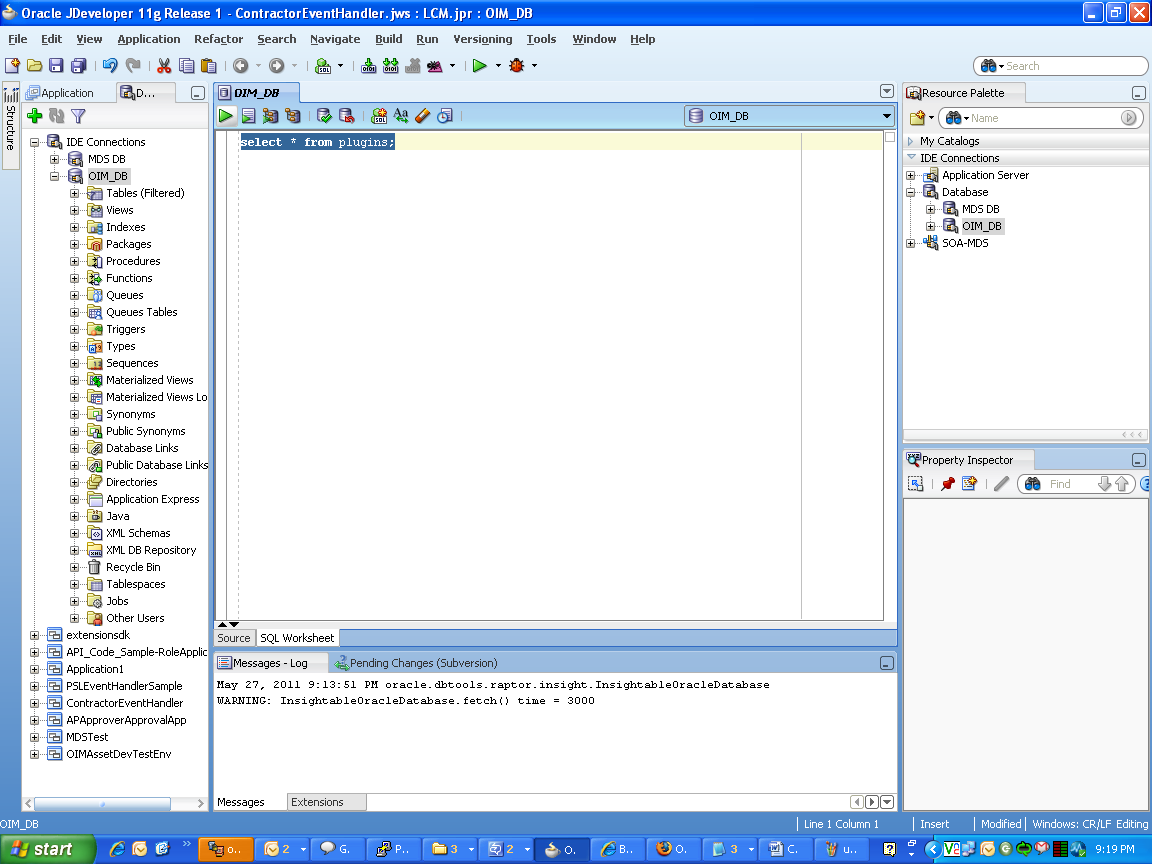
1. Open database navigator.



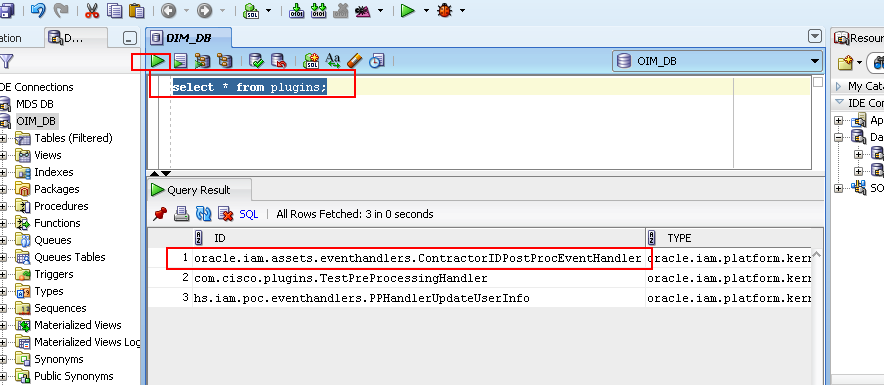
”Open SQL worksheet”



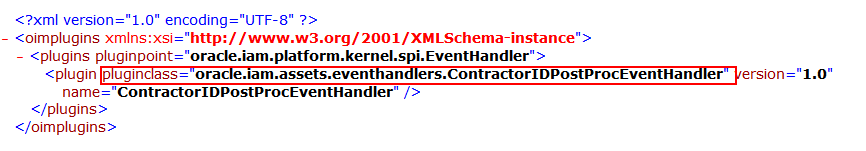
1. Fire “select \* from plugins;” query in database SQL worksheet and verify plug-in ids which is given in plugin.xml while deployment.



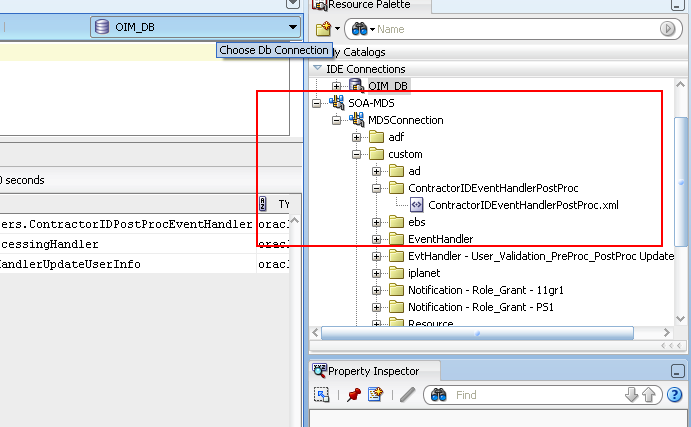
1. You can verify the ID of plug-in in result query.



1. The same id was provided in plugin.xml.

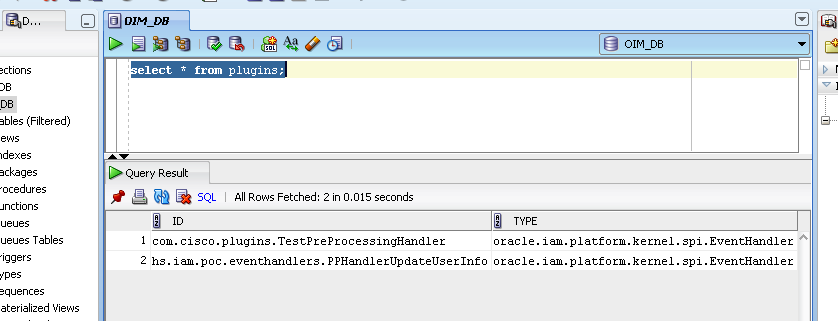


1. Verify SOA-MDS partition.

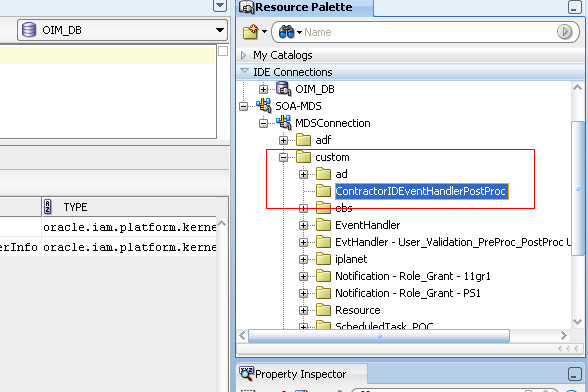


#### Verification of undeployed plug-in

* Verify plug-in table in OIM DB



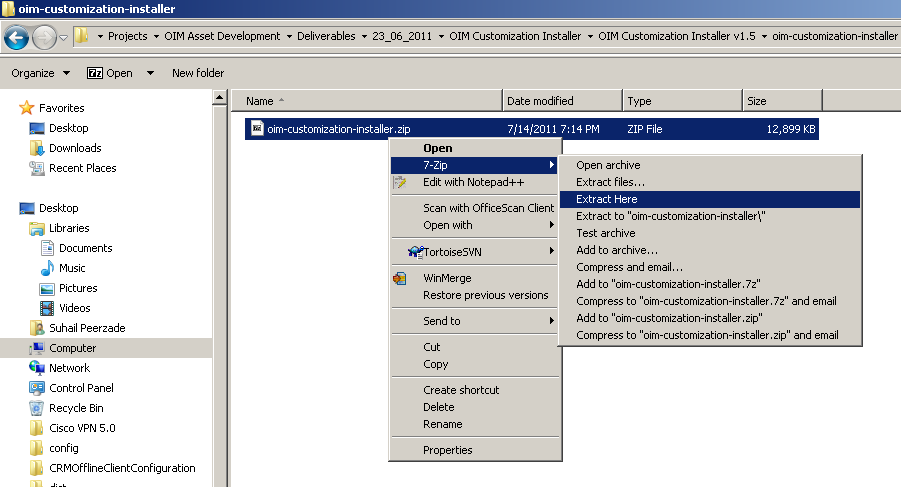
#### Verify SOA-MDS partition.



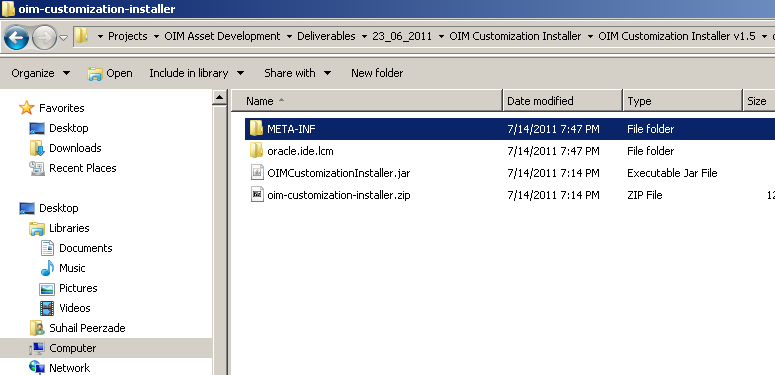
### Appendix - Adding support for newer JDeveloper versions

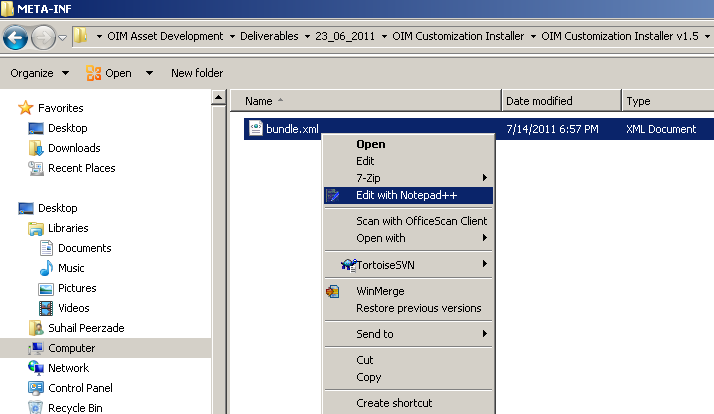
Not for this training but later if you want to use the OIM Customization installer on a later upcoming version of JDeveloper, follow the procedure mentioned below to update this JDeveloper plugin.

* 1. Extract the contents of oim-customization-installer.zip

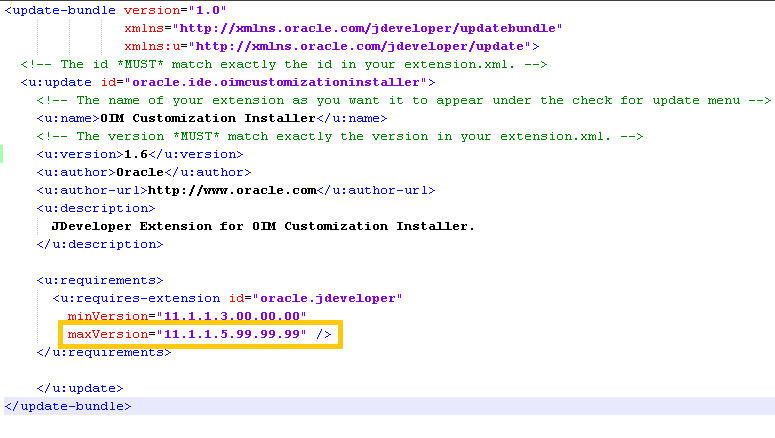
**

* 1. Navigate inside the META-INF directory and open bundle.xml in a text editor

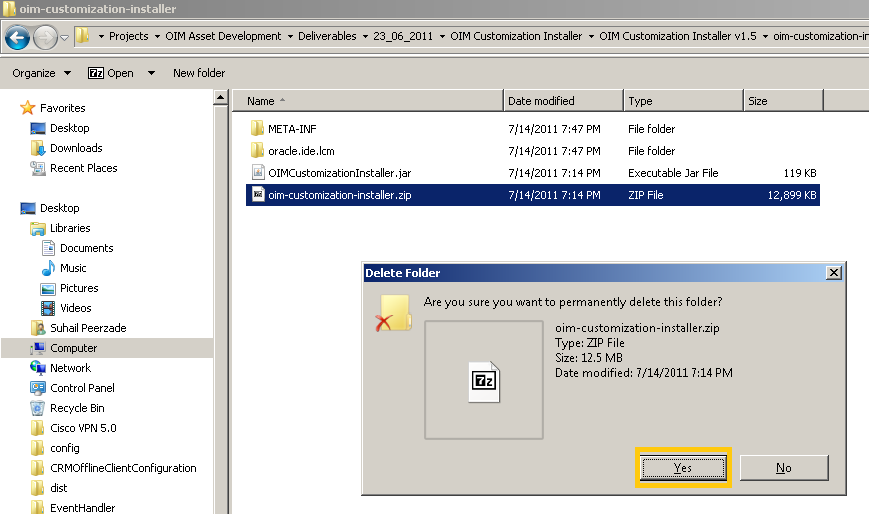
**

**

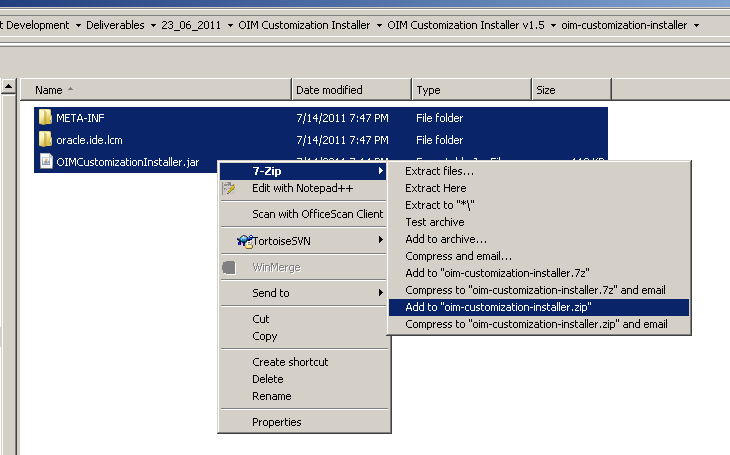
* 1. Update the highlighted line in the below screenshot to specify the latest JDeveloper version for which the support needs to be added.

**

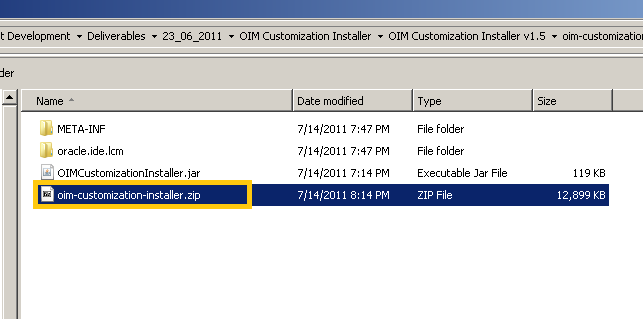
* 1. Repackage oim-customization-installer.zip.
     1. Delete existing oim-customization-installer.zip file.



* + 1. Create new oim-customization-installer.zip file.



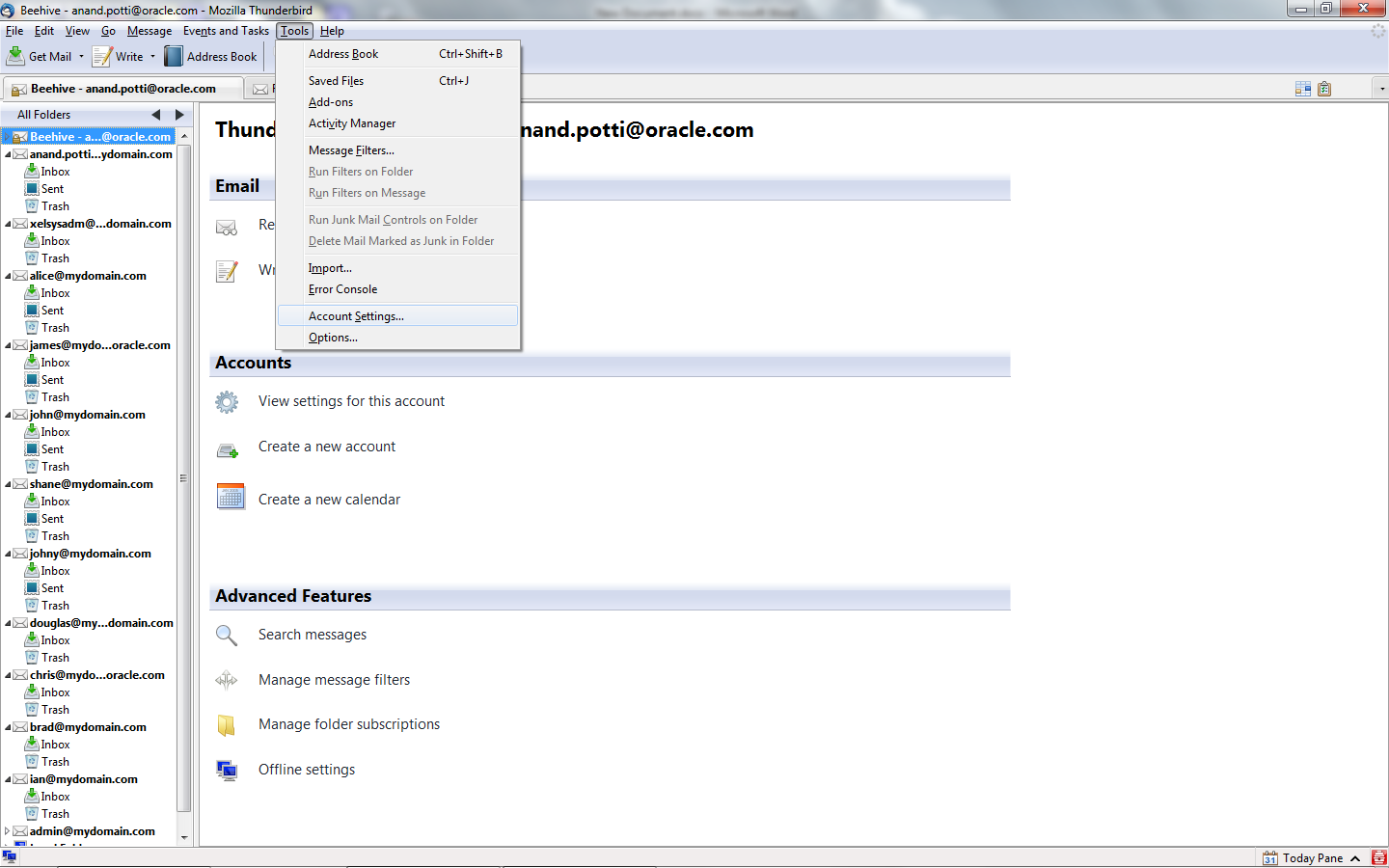
* + 1. The below highlighted will be the latest oim-customization-installer.zip file with the support for the specified version of JDeveloper.



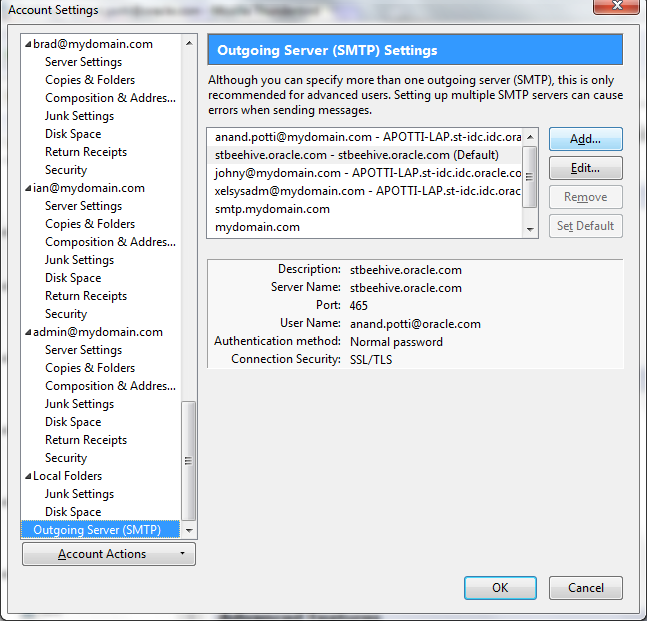
### Appendix – Accessing JES based Email server INBOX

The section below and document illustrates the steps to be taken to configure Thunderbird client for the email ids created in the local email server (JES).

1. Open Thunderbird client. Click on ‘Tools’-> ‘Account Settings’

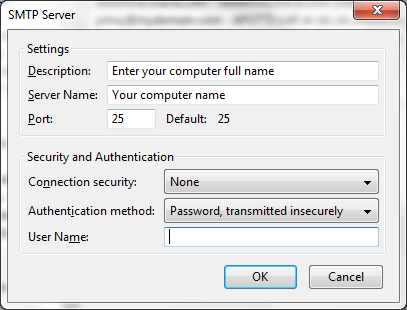


1. Click on ‘Outgoing Server (SMTP)’ -> ‘Add’

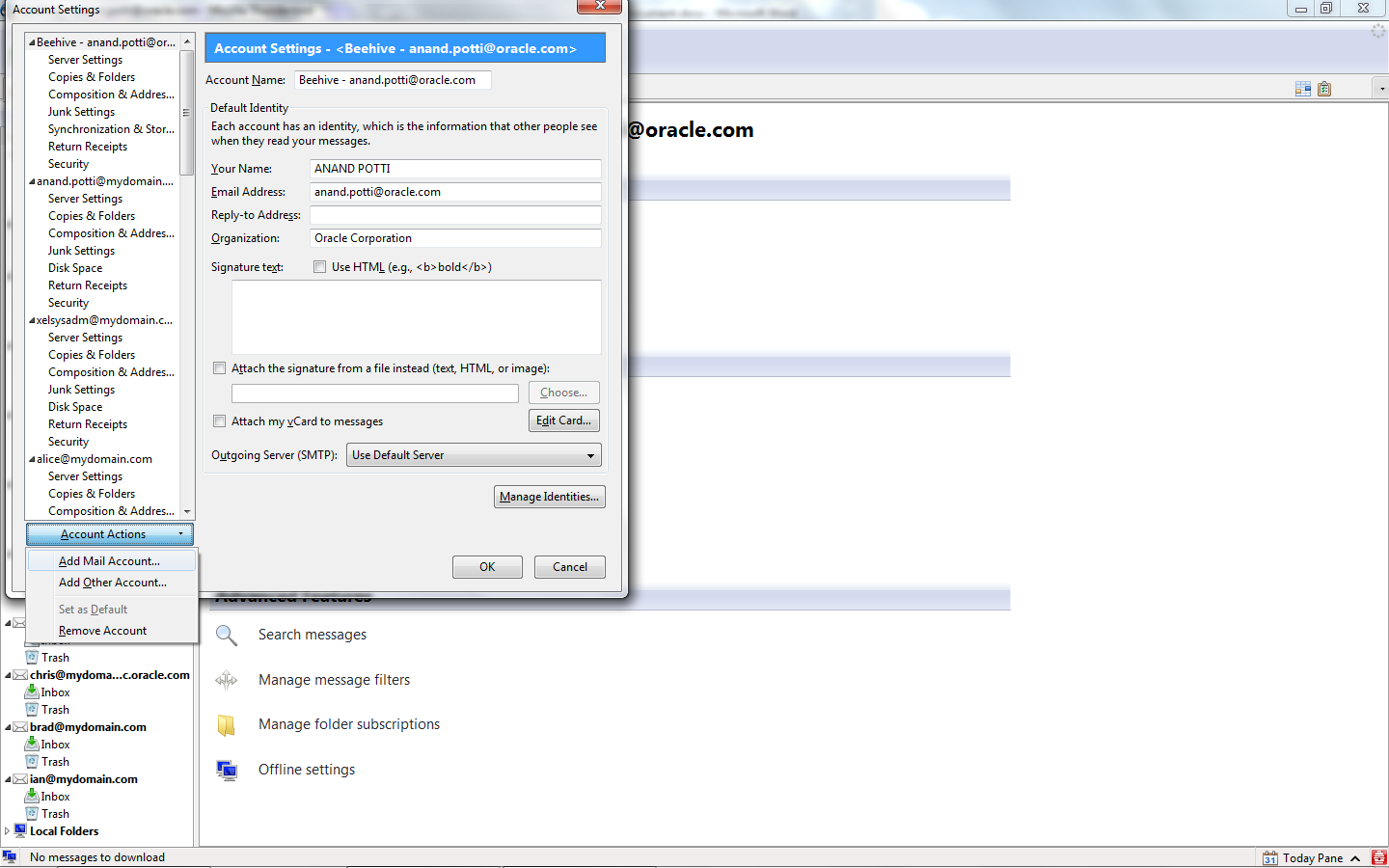


1. In the pop-up window,

* Enter the server name (which should be **your full computer name** in which you have installed the mail server (like JES)).
* Give the outgoing port number as configured in the mail server (the default is 25).
* Click ‘OK’.

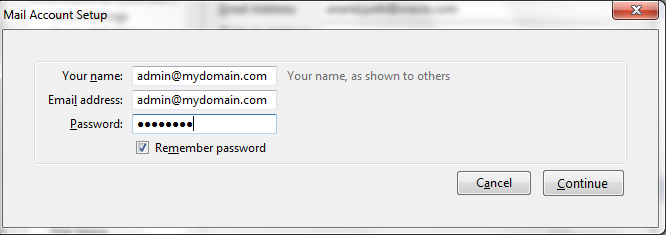


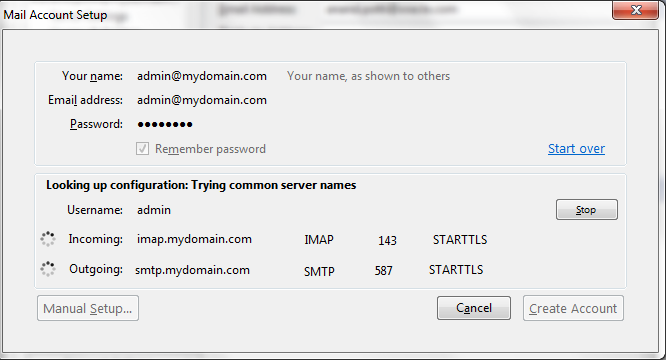
1. Click on ‘Account Settings’ -> ‘Add Mail Account’



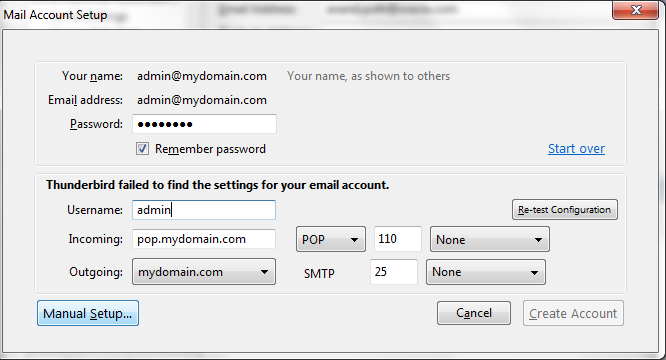
1. In the pop-up window,

* Enter the email address which needs to be configured and the account password.
* Click ‘Continue’.
* Allow the client to identify the server details.

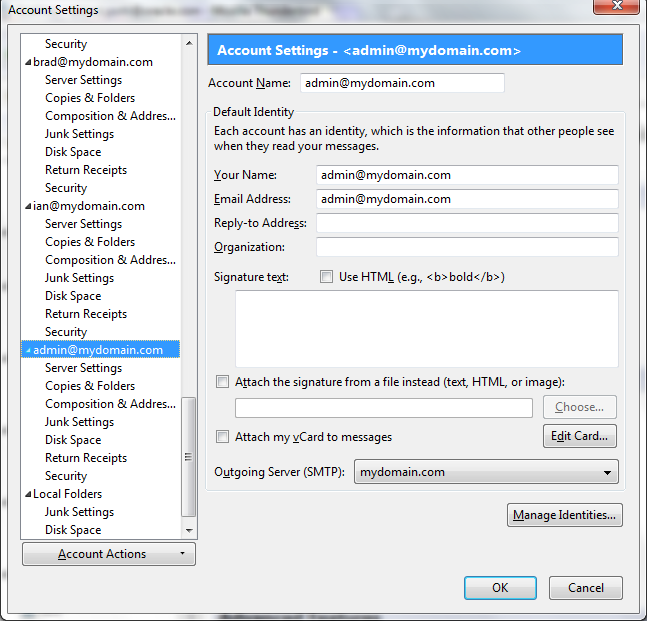




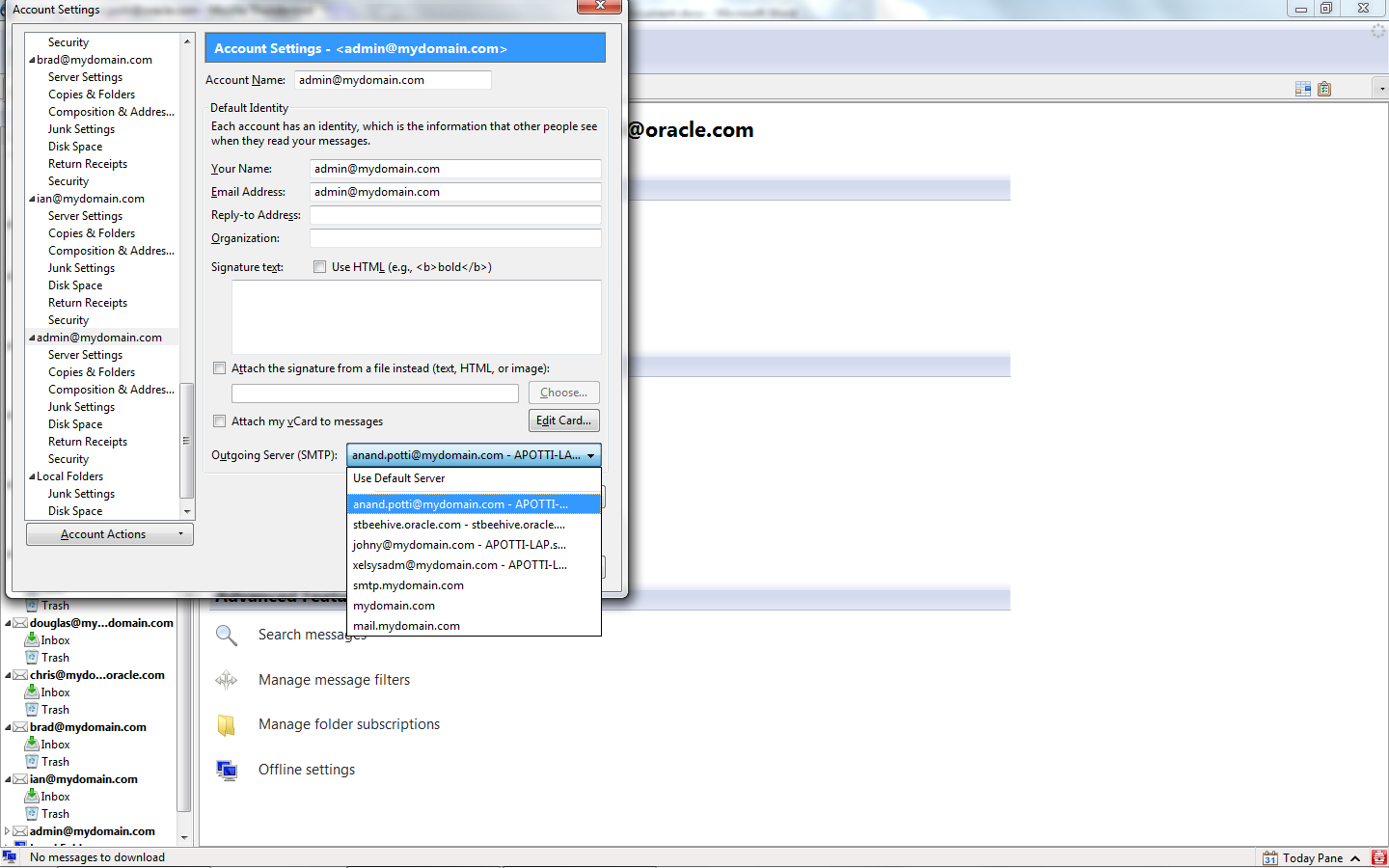
1. Click on ‘Manual Setup’.



1. In the manual setup wizard give the email id in ‘Your name’ and ‘Email Address’ fields.

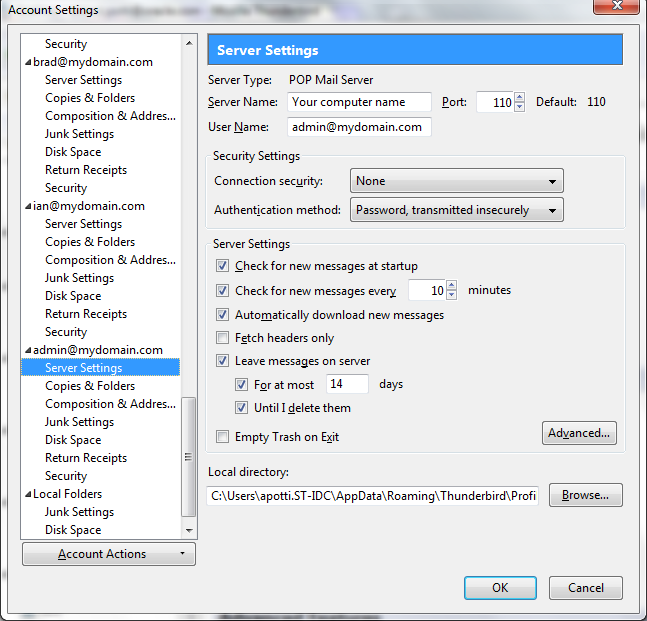


1. Select the ‘Outgoing Server (SMTP)’ as the one you created in step 2.



1. Click on ‘Server Settings’ option.

* Give you full computer name in ‘Server Name’ field.
* Enter the email address in ‘User Name’ field.
* Enter the port as defined in the email server for Incoming. (The default value is 110).
* Ensure that the server Type is POP Mail Server.



1. The above steps complete the configuration of the email account in Thunderbird. Verify the setup by sending an email on the same account to the same account as shown below.

