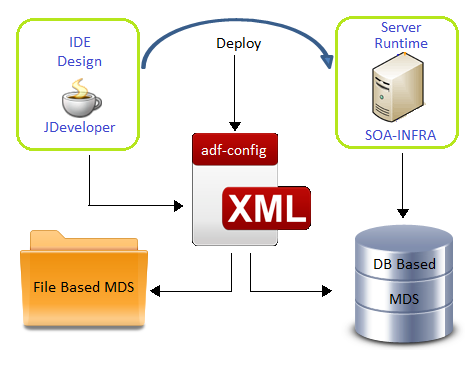
**Meta Data Store (MDS) in SOA 11g**

SOA Suite 11g has a provision for sharing SOA artifacts through MDS – Meta Data Store. This facilitates greater re-use of SOA artifacts such as XML Schemas, EBMs, WSDLs, Fault Policies, Rule repositories and Service Data Objects (SDOs) . You can configure MDS as file-based or database-based. File based MDS is used during design/deploy time where as DB based MDS is used at run-time by the SOA infra.

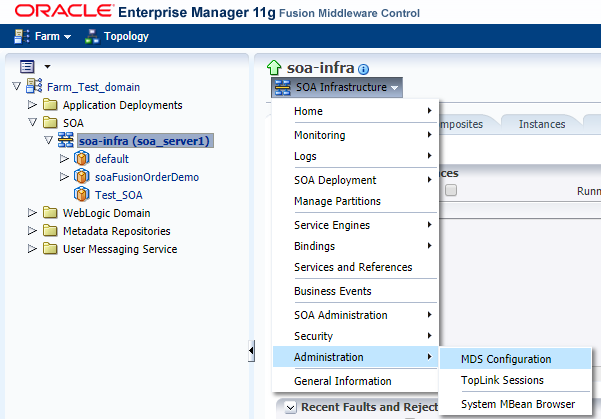
A configuration file named adf-config.xml that is part of every SOA project manages this partition mapping to be used by JDeveloper during design/deploy time.

All SOA projects that are deployed on the SOA infrastructure are stored in the MDS repository. If you create a connection from the JDeveloper 'Resource Palette' to the SOA MDS, there will be a 'deployed composites' folder which will contain all deployed artifacts.

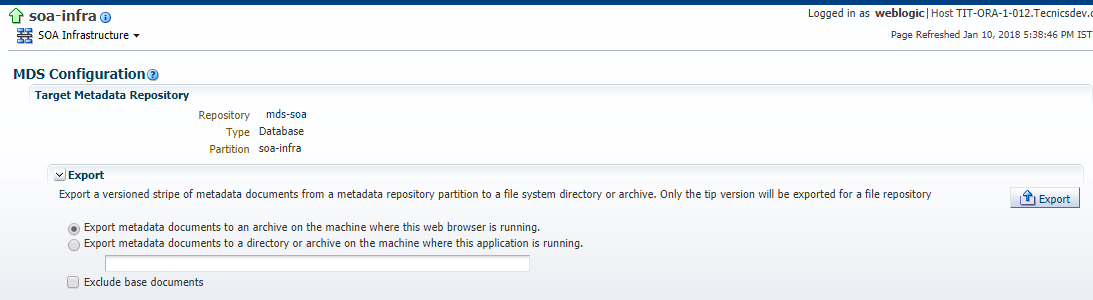


**Importing Meta Data Store(MDS) into Jdeveloper**

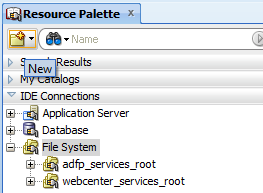
1. Opne EM
2. Click on soa-infra on Enterprise Manager.
3. Click on soa-infra 🡪 Administration 🡪 MDS Configuration

****

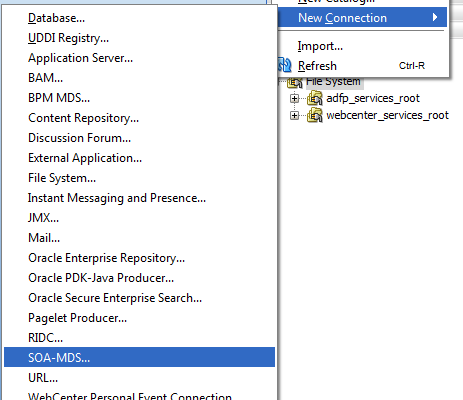
1. Click on Export button on the MDS Configuration page.

****

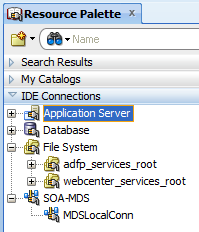
1. [soa-infra\_metadata.zip](http://localhost:7001/em/faces/as/j2eeApplication/admin/configureMDSConnections?_adf.ctrl-state=tp28haf1b_9&target=%2FFarm_Test_domain%2FTest_domain%2Fsoa_server1%2Fsoa-infra&type=oracle_soainfra&ipMsgID=c8111222) file will be downloaded.
2. Open JDevloper
3. Go to Resource Palette.
4. Click on new 🡪 New Connection

****

1. Select SOA-MDS from the list.

****

1. Give the connection name.
2. Select File Based MDS option.
3. Select the downloaded file by using browse button.
4. Click on test connection to check.
5. If test success then click OK.
6. Connection is created to SOA-MDS.



**Meta Data Store(MDS) Setup in SOA 11g**

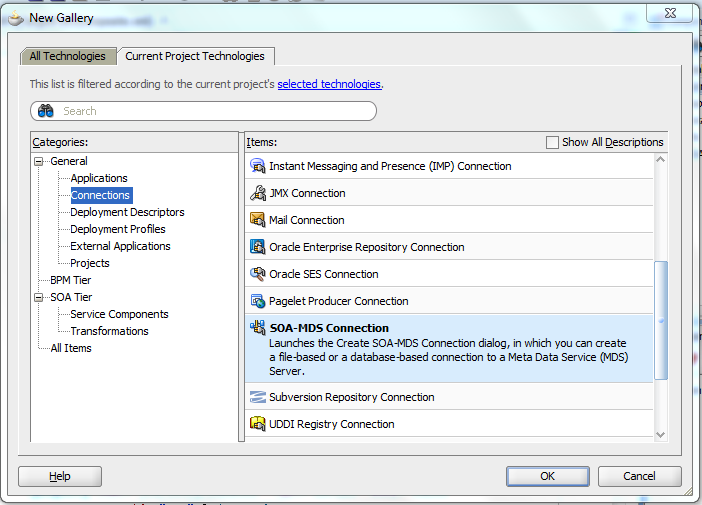
Create directory structure under apps folder. In my case, I’ve created folder structure **JDEV\_HOME/integration/seed/apps/myproject/fault/** to store fault-policies and fault-bindings fault handling files.

1. First you go to your local 11g Jdeveloper installation folder. In my case it is C:\Jdev\Middleware\jdeveloper\integration folder .You can see **seed**folder , If not create a seed folder.

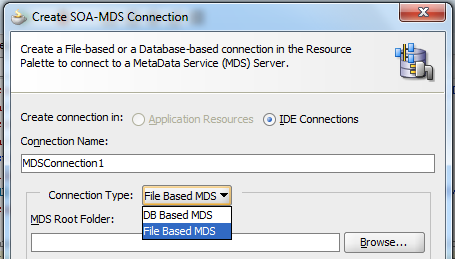
2. Now create folder with name apps under seed.

3. Under apps you can create your folder structure to place all your XSD, WSDL files like   apps/ **myproject** / **fault**/ fault-policies.xml etc.

4. Create MDS connection in jdeveloper as below.  
New -> Connections -> SOA-MDS Connection.

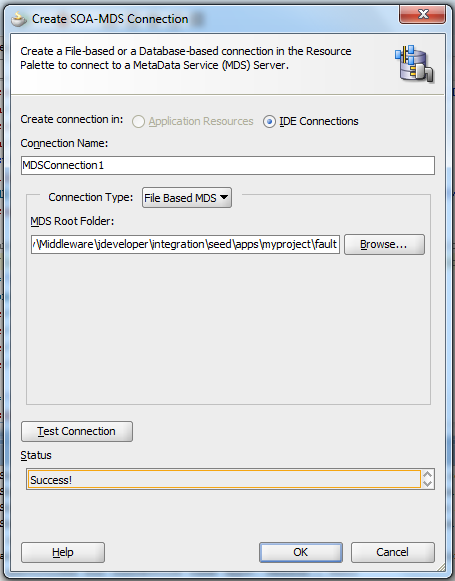


5. The SOA-MDS connection can be created in two ways. One is File Based MDS and other is Data Based MDS.



**File Based MDS creation**

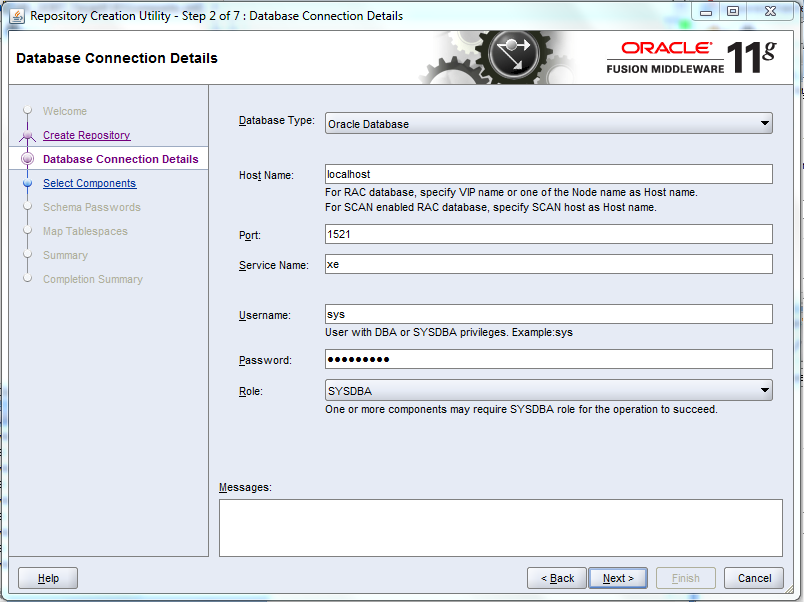
In **Create SOA\_MDS Connection** window, select File Based MDS as Connection Type for file based MDS connection, we will point to the local folder in our system   where all the WSDL and XSD files are placed as shown below. So when we use the wsdl and xsd files in the project, they will be referred from the local system.

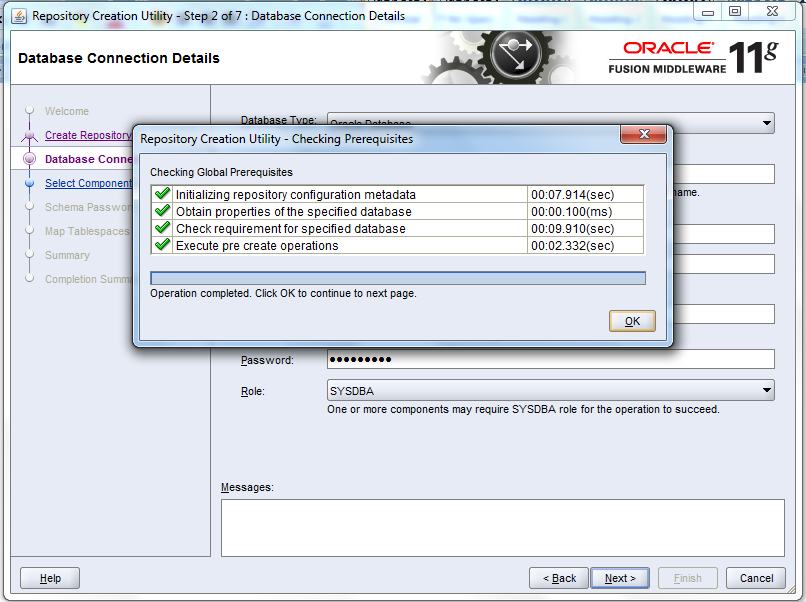


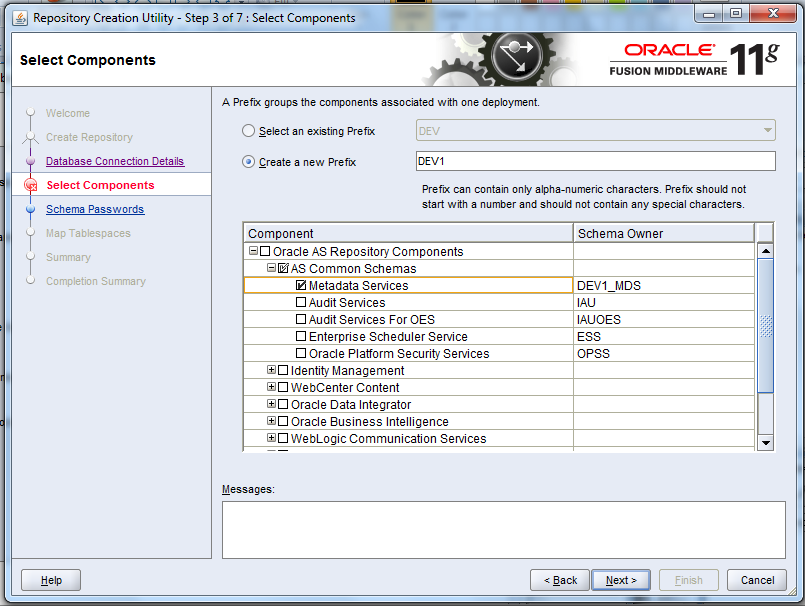
Click OK. Now the File Based MDS connection is created.

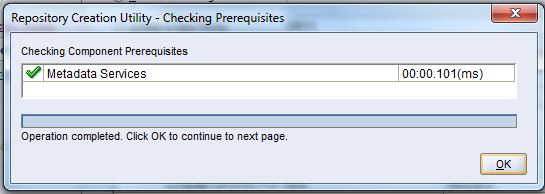
**DB Based MDS creation**

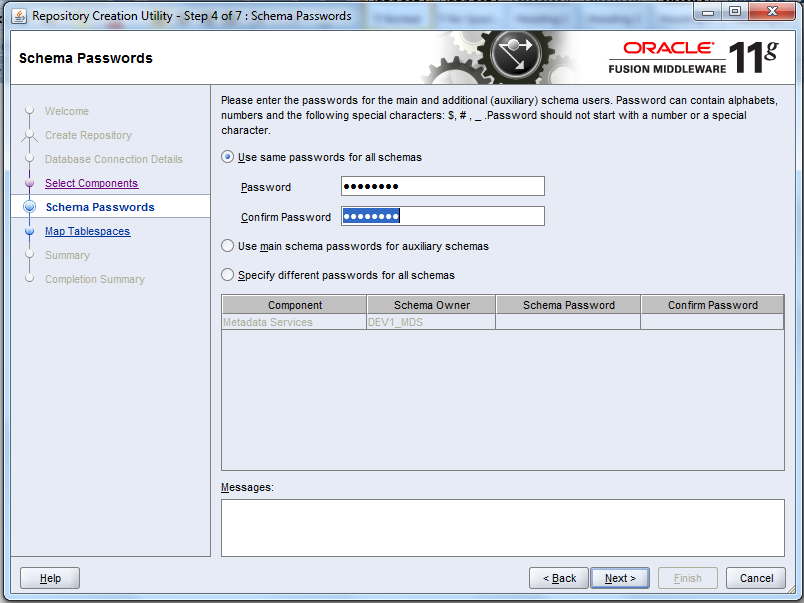
Let’s see how to create DB based MDS connection. In Create SOA\_MDS Connection window, select DB Based MDS as Connection Type for file based MDS connection.

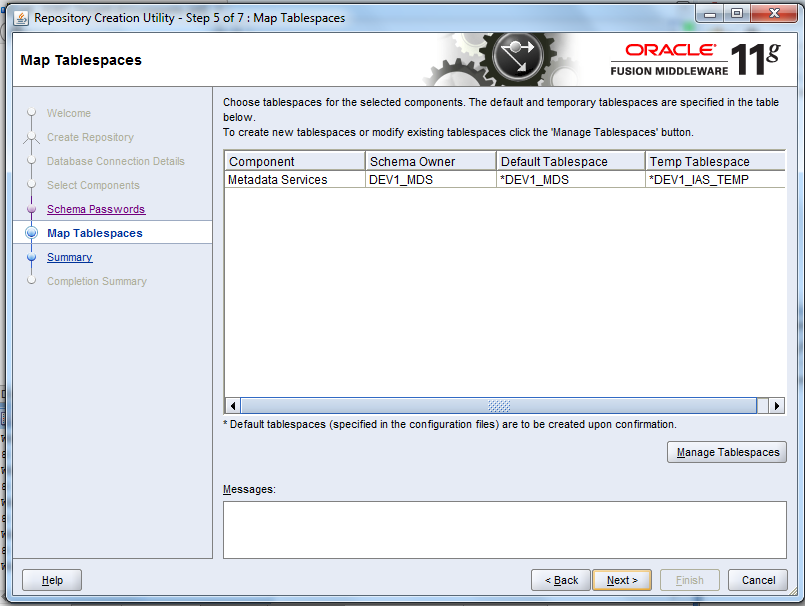


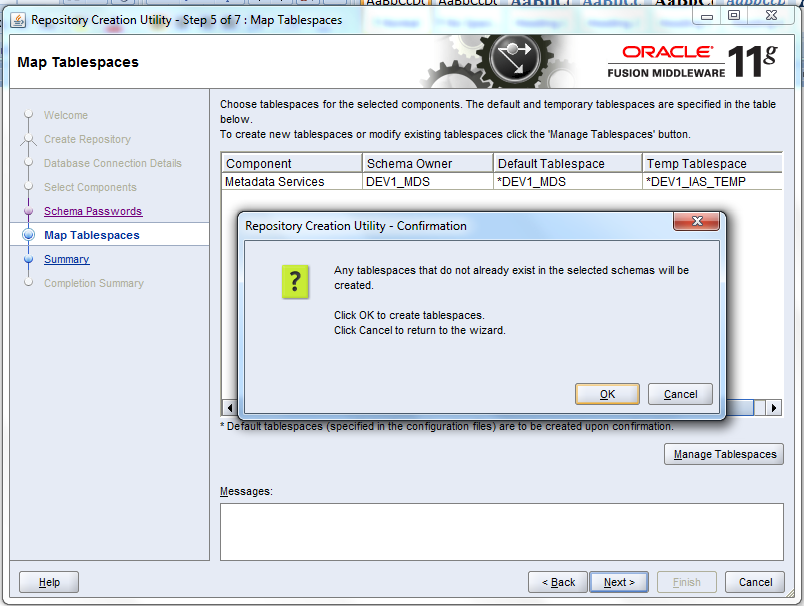


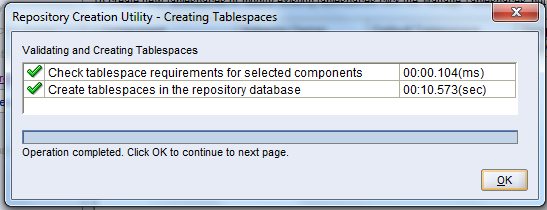


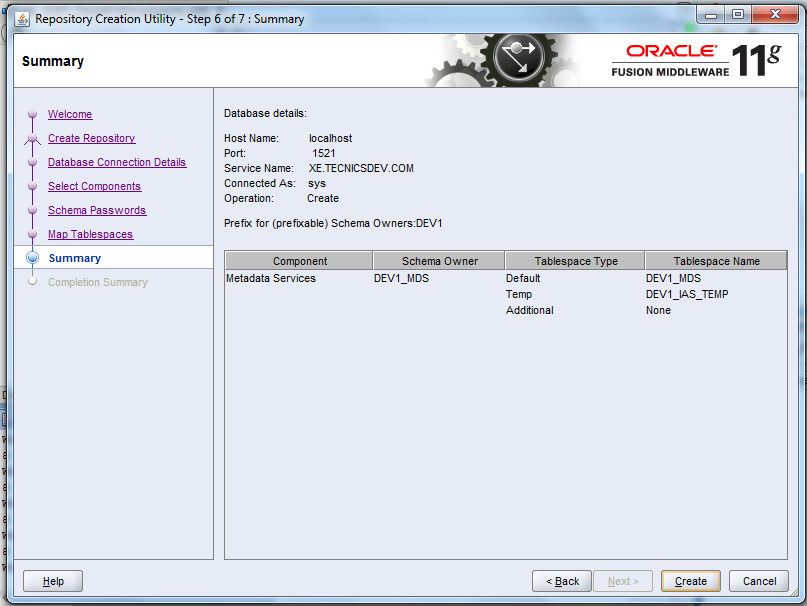


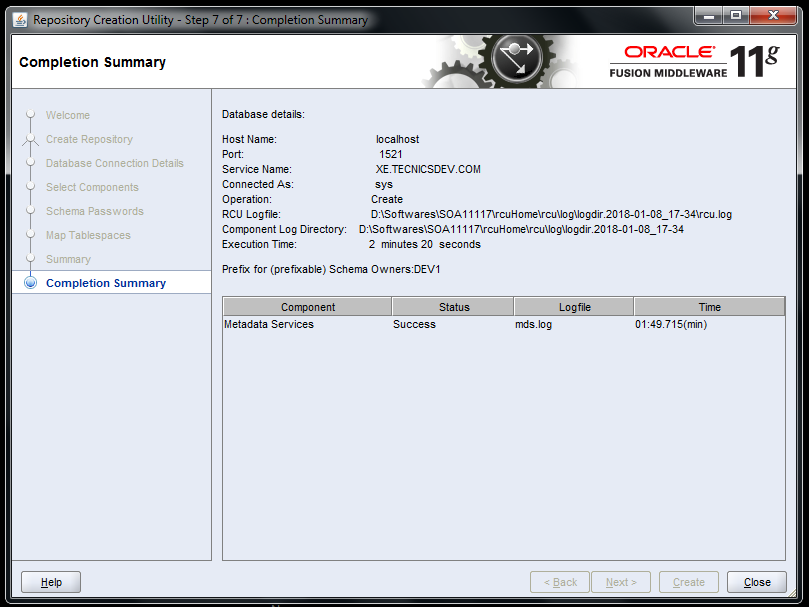


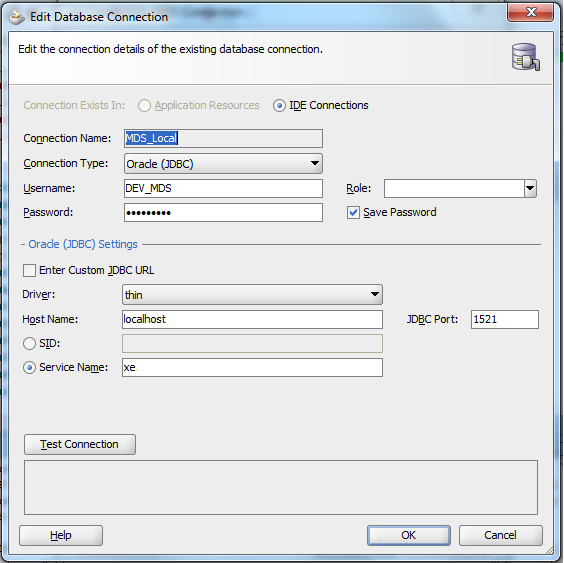


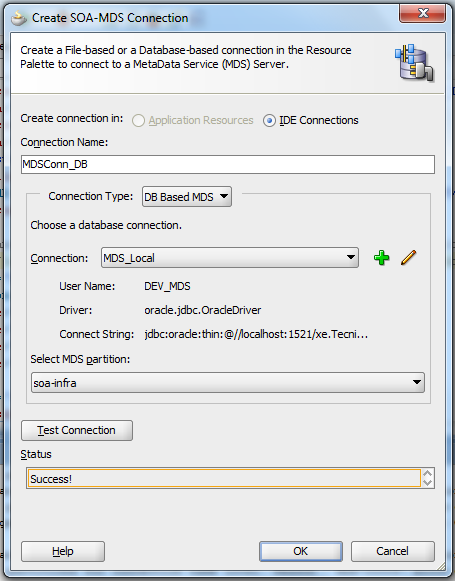






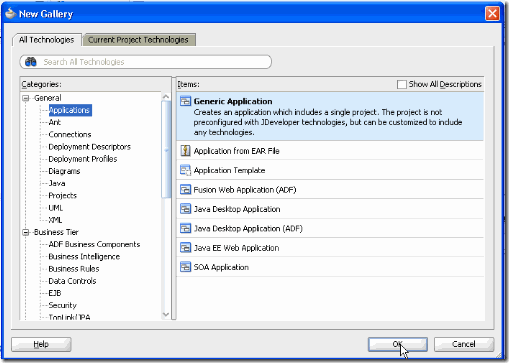


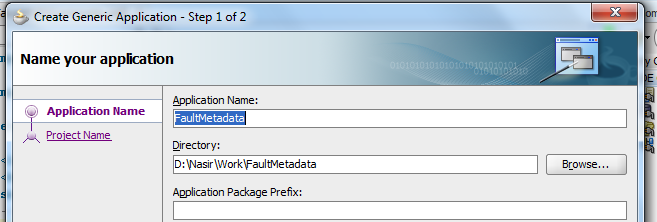




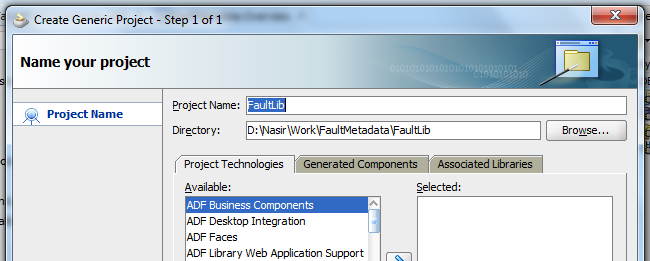
For DB Based MDS connection, we should have created all the schema for the weblogic in the database by running the Repository Creation Utility (RCU) wizard. After RCU execution, the DEV\_MDS schema will get created in the database. We need to use the **DEV\_MDS**schema connection and **soa-infra** as MDS Partition while creating the DB Based MDS connection as shown in the above screen shot.( Get in touch with your DBA for MDS schema credentials)  
  
Click OK. Now the DB Based MDS connection is created.

**Deploy MDS repository to Weblogic Server**  
First, we need to create JAR file bundle to include schemas,WSDL's  we need. Then we need to include this jar file into a SOA bundle that can be deployed to the SOA Server.  
  
In JDeveloper Create Generic Application – FaultMetaData

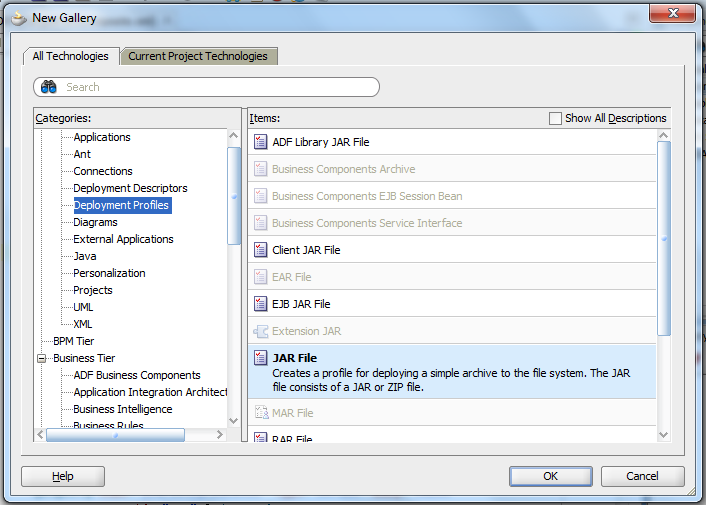




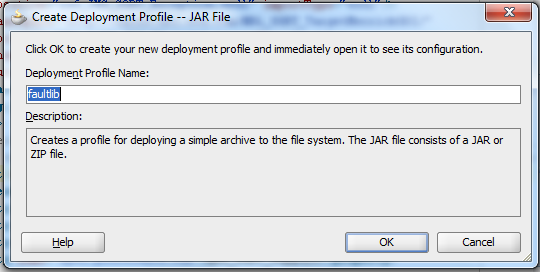
Create a project – MediatorLib



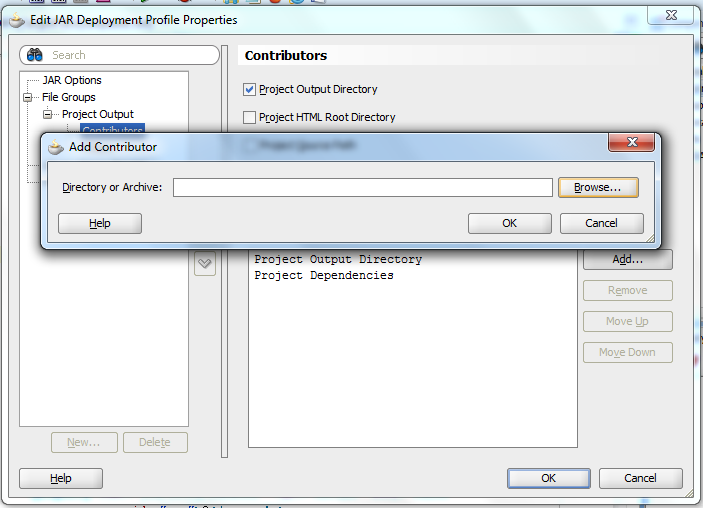
Right click on MediatorLib and select New –> Deployment Profile –> JAR File

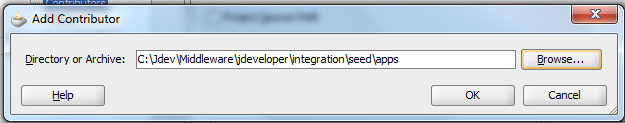


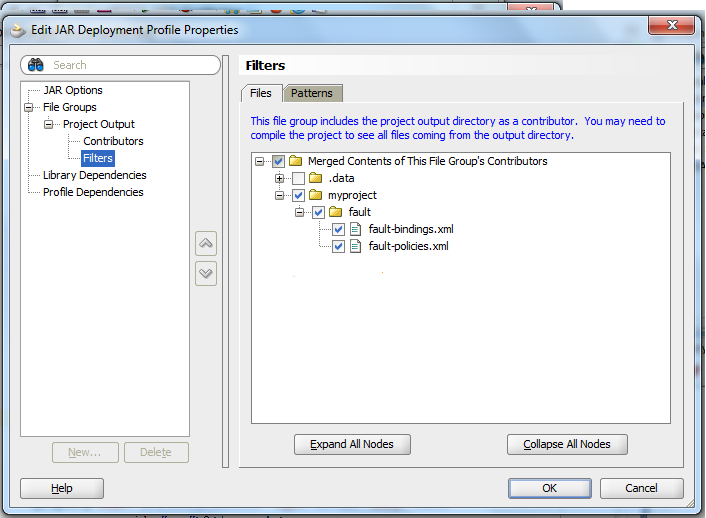
Enter **faultlib**as the deployment profile name.



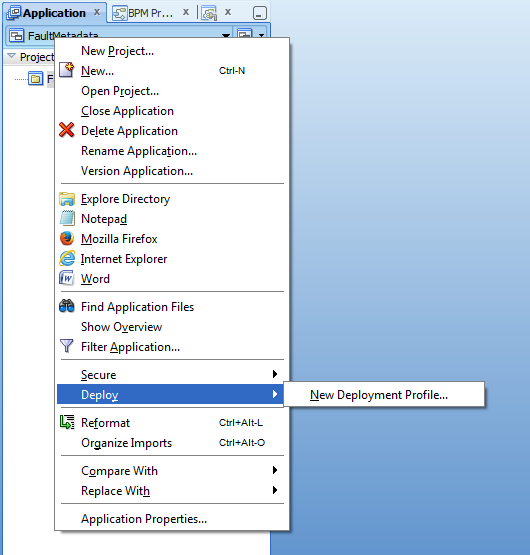
In the JAR deployment properties dialog, click on **Contributors**, click on **Add**button and select **apps**directory from your JDeveloper seed location – JDEV\_HOME/integration/seed/apps As for below screenshots



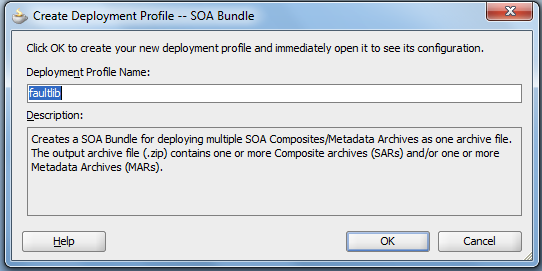


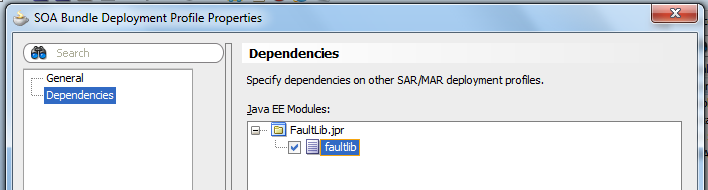


Click on **OK**. This completes creation of JAR file. We now need to create an application deployment profile to install these schemas in SOA Server.  
  
Right click on FaultMetaData application and select Deploy –> New Deployment Profile. Select “SOA Bundle” as Profile type.

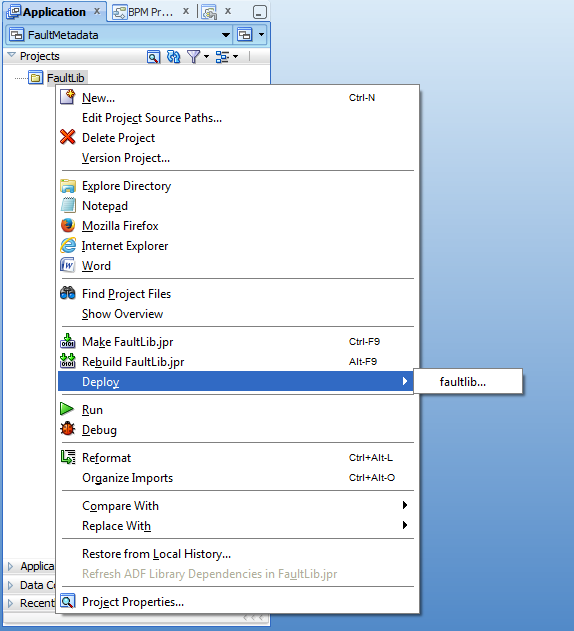


Enter faultlib for Deployment Profile Name. From the properties dialog, open Dependencies and select faultlib.

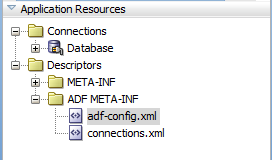




To deploy FaultLib to SOA Server, right click on FaultMetaData and select Deploy –> mediatorlib.



You can verify the same from the SOA MDS connection created under the 'Resource Palette'. This should now reflect the 'apps' folder at the same level as 'soa' metadata store.  
  
Now, if you try to deploy the SOA project which references MDS you may hit a deployment error complaining that the resource referenced is not found. This is because the JDeveloper consults adf-config.xml configuration file during design & deploy time (refer back to the initial architecture diagram). Since there is no information about the DB based MDS in the adf-config the deployer complains.



To fix this, go back to the adf-config.xml file and add the DB metadata store information so that the deployer can read & locate the referenced resources from MDS.