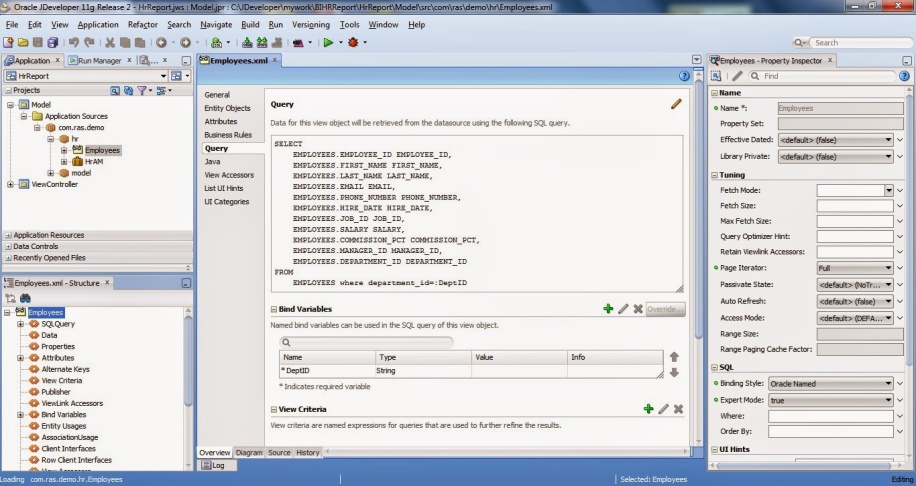
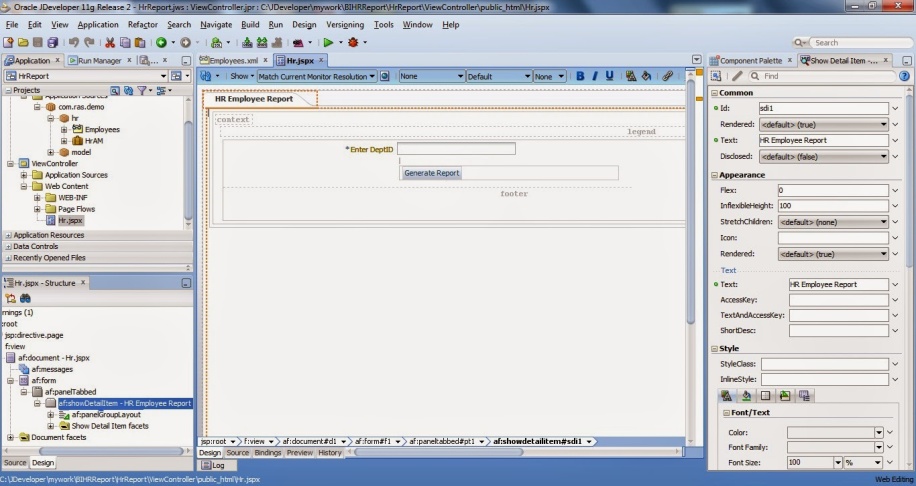
**Integrating ADF BI Publisher In JDeveloper 11g to Generate Report Programmatically.**

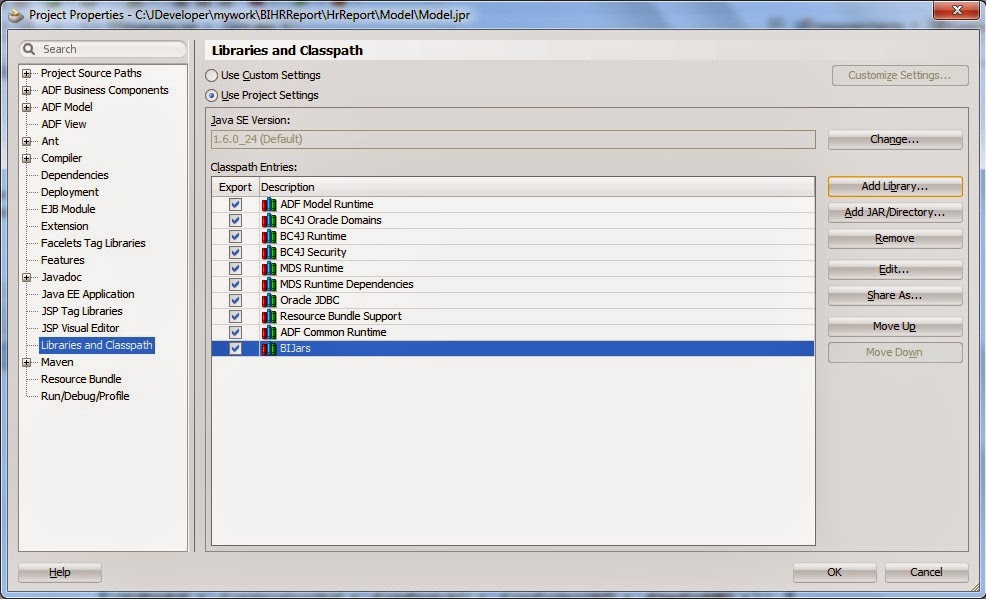
Oracle JDeveloper 11.1.2.3  
ADF Faces RC  
Business Components  
Oracle BI Publisher 10g  
Oracle BI Publisher Desktop 10g  
  
This post would explain how to Integrate BI Publisher with ADF Fusion Application to generate report programmatically. The overview of BI Publisher & BI Publisher Desktop is not discussed here.  
  
Sample workspace is based on Oracle HR schema. BI Publisher Desktop is used for designing RTF template. This should present as menu in MS-Word after installation. However, designing of template is also not covered in this post. Please [read](http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/biee/r1013/bipub/intro/gettingstarted.htm#t3) the Oracle Official Document to create RTF Template based on XML Data.  
  
BI Publisher JAR Files -  
1. i18nAPI\_v3  
2. versioninfo  
3. xdocore  
4. xmlparserv2-904  
  
  
The jars can get from BI Publisher Desktop installation directory.  
The Location is - **C:\Program Files\Oracle\BI Publisher\BI Publisher Desktop\TemplateViewer\lib**  
The jars are also included with Sample Workspace under **/HrReport/jars** folder.  
  
The ViewObject rows will be converted to XML data. This XML data is then merged with RTF Template to generate XSL Template. Finally this XSL Template will return outputstream of data. This stream can be written to desired report output like **PDF, HTML, EXCEL** etc.  
  
The ViewObject Query will return all the employees in particular Department based on parameter DeptID. The DeptID is Bind Variable parameter.

[](http://3.bp.blogspot.com/-F8ZQC1rKieM/U5shDYqdTRI/AAAAAAAAANw/JRB4AbFRr_U/s1600/BIRep1.jpg)

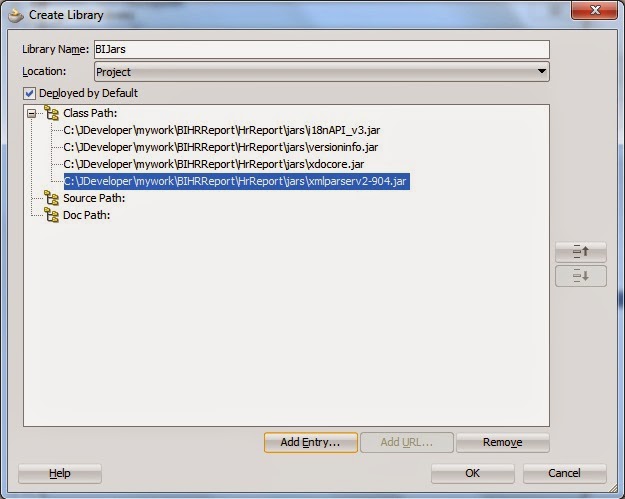
The HR.jspx page has inputText & Generate Report Button. The inputText is wired to **Binding** property to submit value in DeptID bind variable.  
  
Please note that there is 2nd hidden button, which is actually doing the report generation process. The reason for using two button is to check some validation & then start process. Furthermore, **af:fileDownloadActionListener** always download a file irrespective of any valid or invalid input.  
To avoid such situation, first button will check validation, if it passes, queqe an action event to 2nd hidden button.

[](http://3.bp.blogspot.com/-jQQ2ubu56XA/U5shLzGHBsI/AAAAAAAAAN4/Zeobxm2Dzw4/s1600/BIRep2.jpg)

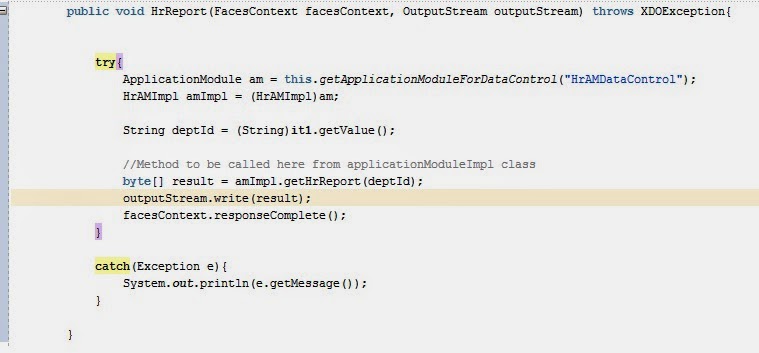
Since, actual process will take place in **HrAMImpl.java** class, so jars should be present in Model Project.  
Add all the jars as Library in **Model Project**. The advantage of adding as Library is that path references will resolve easily as far as if they exist in project source files. All jars are added as **BIJars** in **Libraries & Classpath** section of **Model Project** Properties.

[](http://1.bp.blogspot.com/-R4rmvKXseiU/U5res6IOsKI/AAAAAAAAAMM/OUQec4CChLQ/s1600/BIRep3.jpg)

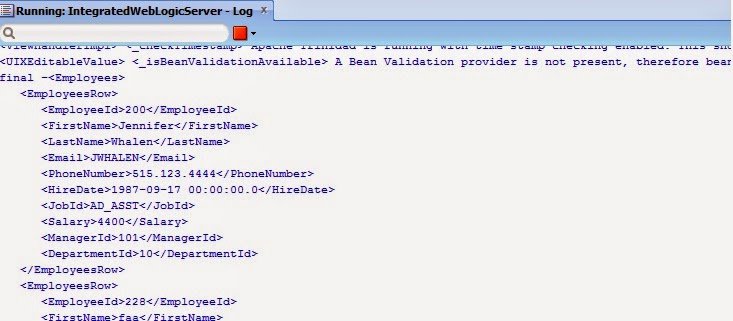
To add the jars as Library, click **Add Library** then click **New** to open **Create Library** dialog.  
Click **Add Entry** to add jar files. Press OK to close all dialogs.

[](http://1.bp.blogspot.com/-I22phShwL6Y/U5rgNsU-sQI/AAAAAAAAAMY/fqoFXI6Wqpw/s1600/BIRep4.jpg)

The 2nd hidden button will invoke **HrReport()** method from HR managed bean. Which inturn invokes a method **getHrReport()** in **HrAMImpl** class.

[](http://2.bp.blogspot.com/-op2Gv4JvB-U/U5rsVVWWqvI/AAAAAAAAAMo/Qw7m1aQYtLw/s1600/BIRep5.jpg)

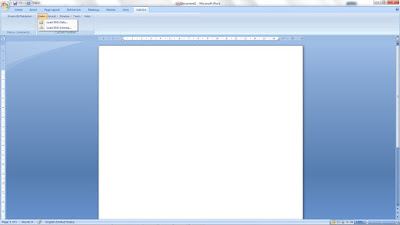
The conversion of ViewObject rows to XML tag is done in **getXMLDataHr()** method. **vo.writeXML()** method is used for generating XML Nodes.  
The XML Data can be printed in Log Window by uncommenting the line, the last one in method - **System.out.println("final -" + finalXMLData)**.

[](http://2.bp.blogspot.com/-ul483Kzim_I/U5r3q5eHkcI/AAAAAAAAAM4/2kRw0-3iNUQ/s1600/BIRep6.jpg)

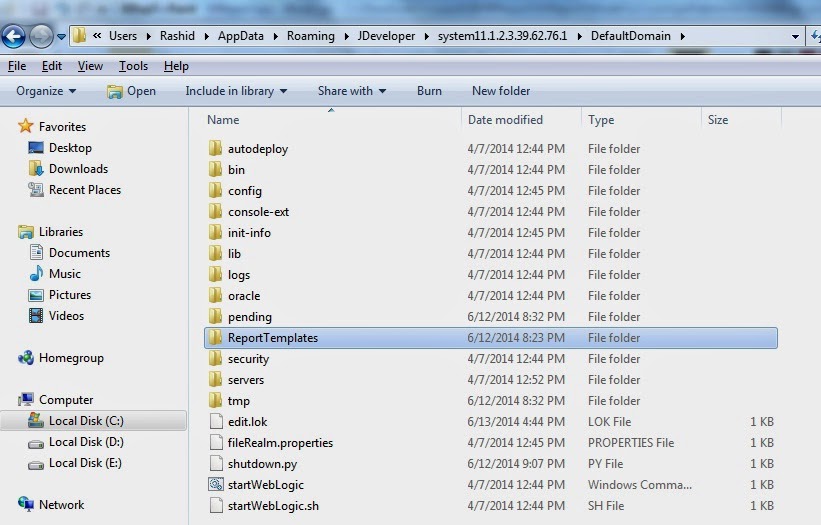
If the Query does not return any row, then corrupted output will be generated of report. To avoid this, we have to return a blank XML output for template. This will generate a blank report output when ViewObject does not return any row.

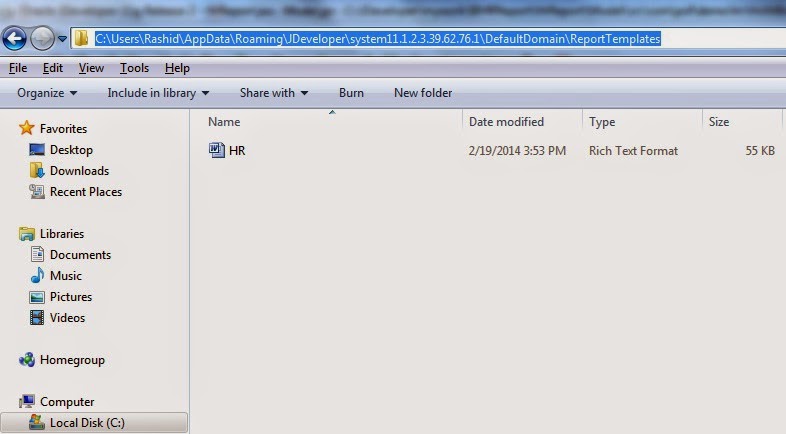
[](http://3.bp.blogspot.com/-cd2qZtrj4xE/U5sN0GpccTI/AAAAAAAAANc/2_OKsCsroW8/s1600/BIRep9.jpg)

Create a XML file anywhere in FileSystem. Copy-Paste generated XML Data in file & save as **Employee.xml** (in this case). A sample XML file is shipped with Sample Source.  
  
Start BI Publisher Server (This step is required only when RTF is going to upload in Report Server.This is only applicable when report generation takes place from BI Server.This step can be skipped & start with MS-Word.When this step is skipped, there is no any need to make connection with BI Server from MS-Word).  
  
Open MS-Word. Load **Employee.xml** file from BI Publisher menu **Load XML Data** & design the Layout of template. Save the template with **HR.rtf**(in this case). A **HR.rtf** is shipped with Sample Source. Visit Oracle site mentioned above to learn designing the template.

[](http://3.bp.blogspot.com/-YXGGAmCS-bQ/VZDuD3MXGQI/AAAAAAAAAco/x-uCmz3csmA/s1600/XMLData.jpg)

Finally, navigate to **DefaultDomain** directory in **IntegratedWebLogicServer** or **ProdWeblogicServer.**  
In my Windows 7 the path is-**C:\Users\Rashid\AppData\Roaming\JDeveloper\system11.1.2.3.39.62.76.1\DefaultDomain** .  
Create a directory **ReportTemplates** here & paste **HR.rtf** under it. And its done.

[](http://4.bp.blogspot.com/-Xm6bOlJ_jcc/U5r6oOYog3I/AAAAAAAAANE/5kQwZCV9bc4/s1600/BIRep8.jpg)

[](http://4.bp.blogspot.com/-cNzeMJxgIrI/U5r6wCGDGTI/AAAAAAAAANM/DFyhPUvtlYs/s1600/BIRep7.jpg)

Sample can be downloaded from [**here**](https://drive.google.com/open?id=0B7wAhdi54nbqM2JDLUlmbnRETkU). Sample includes Template file **HR.rtf**, a XML File **Employee.xml** for employee data. To run this sample, **HR.rtf**should placed under above mentioned WebLogic Server Directory Location.