

# How to use Tomcat for ADF Application

<a href="#">Software Requirements.....</a>	<a href="#">1</a>
<a href="#">Application technology specification.....</a>	<a href="#">1</a>
<a href="#">Getting started.....</a>	<a href="#">2</a>
<a href="#">Create Model.....</a>	<a href="#">5</a>
<a href="#">Create View.....</a>	<a href="#">11</a>
<a href="#">Deploy on Tomcat.....</a>	<a href="#">21</a>
<a href="#">Setup tomcat.....</a>	<a href="#">22</a>

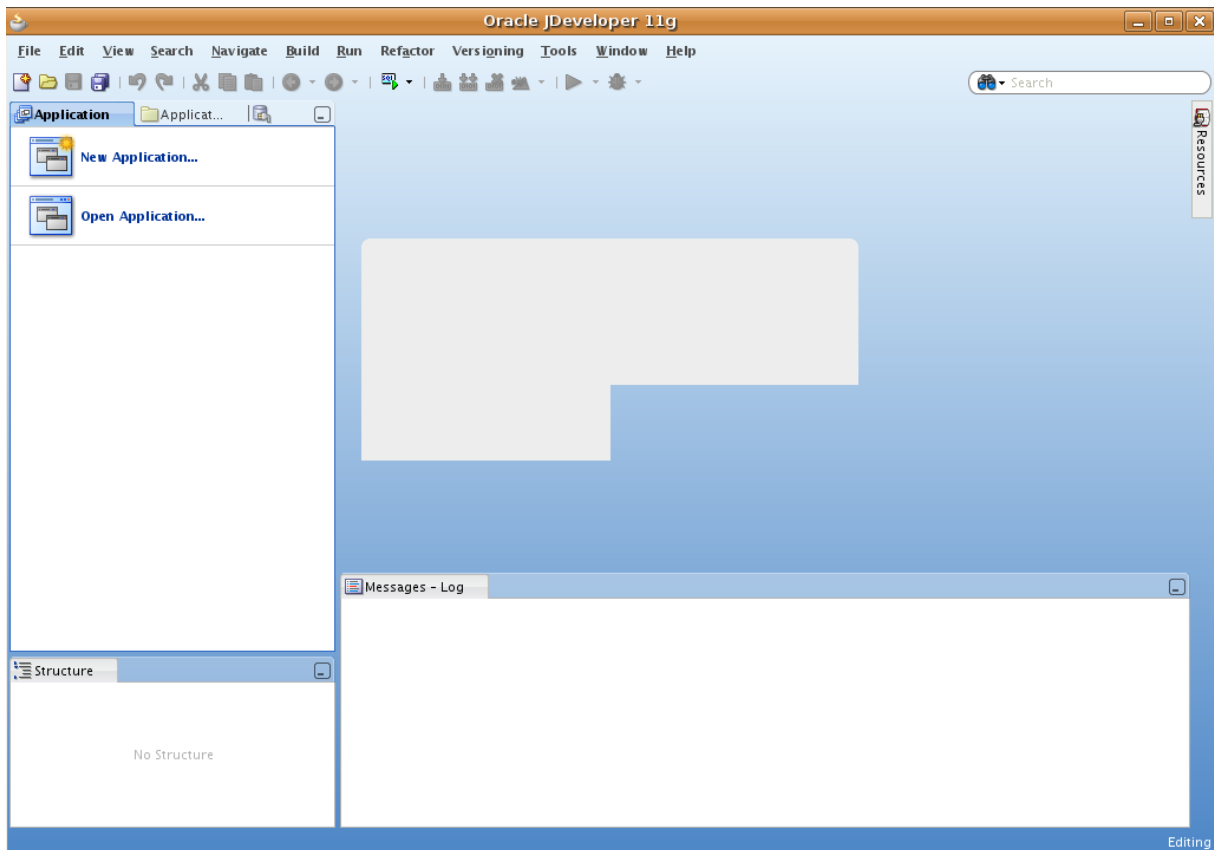
## Software Requirements

Oracle JDeveloper 11g  
Java 1.6.xxx  
Apache Tomcat 6.0.18

Based on HR Demo databáze (hr/hr)

## Application technology specification

Fussion Web Application  
ADF Bussines Components

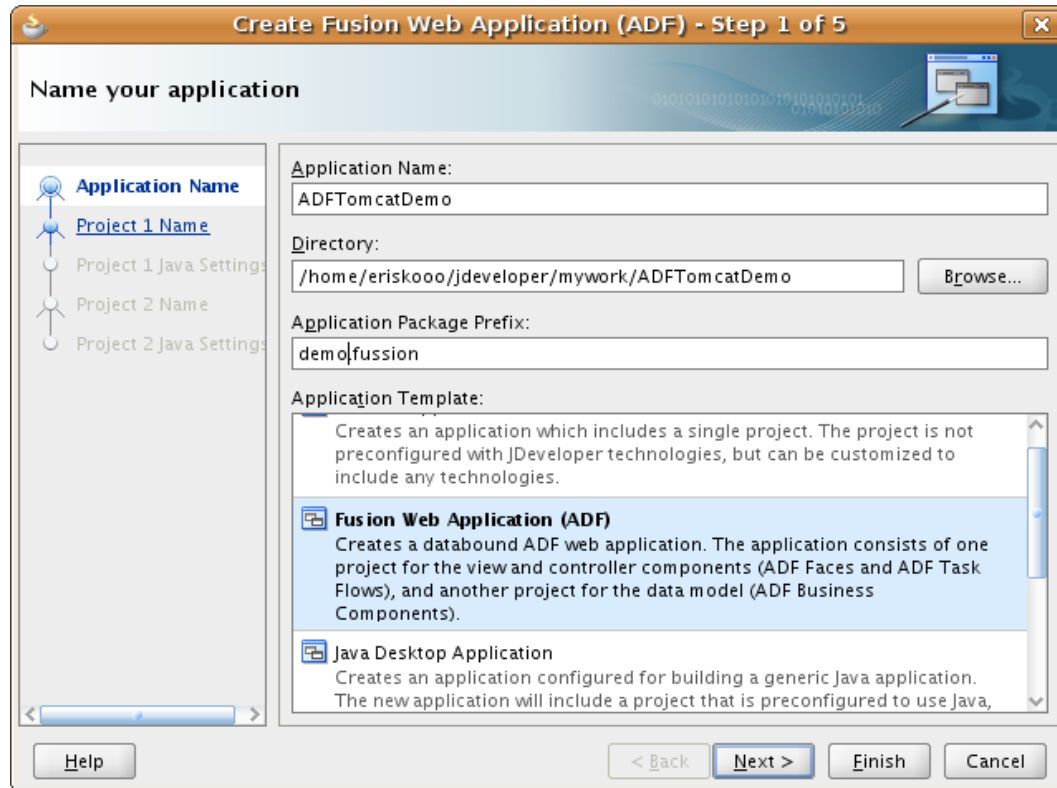


## Getting started

Click on New application, Choose Fusion Web Application

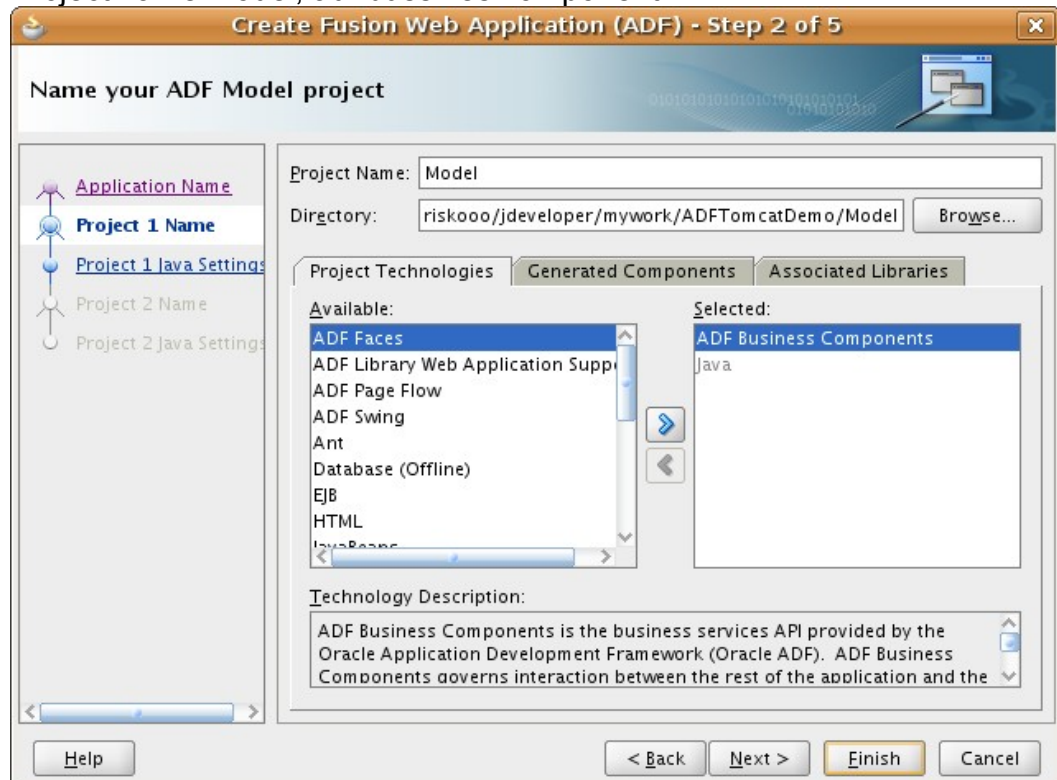
Application name -> ADFTomcatDemo

Prefix -> demo.fussion



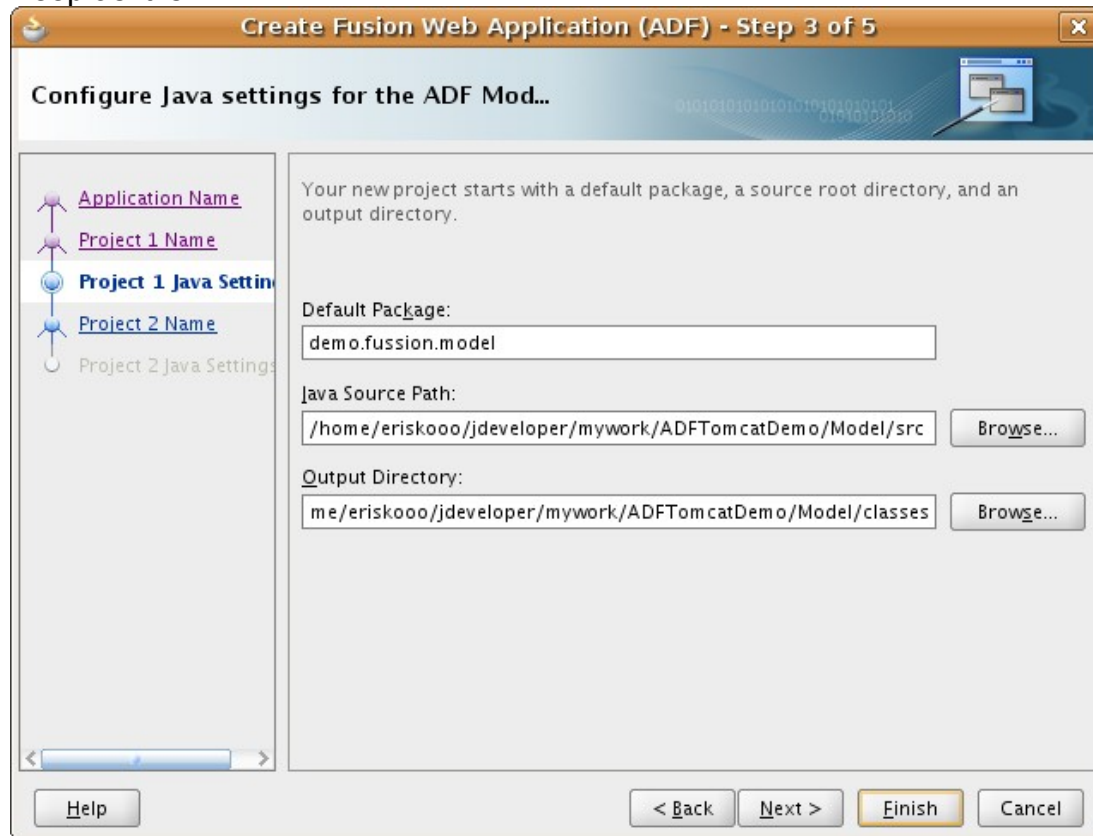
<next>

Project name model, adf bussines komponent



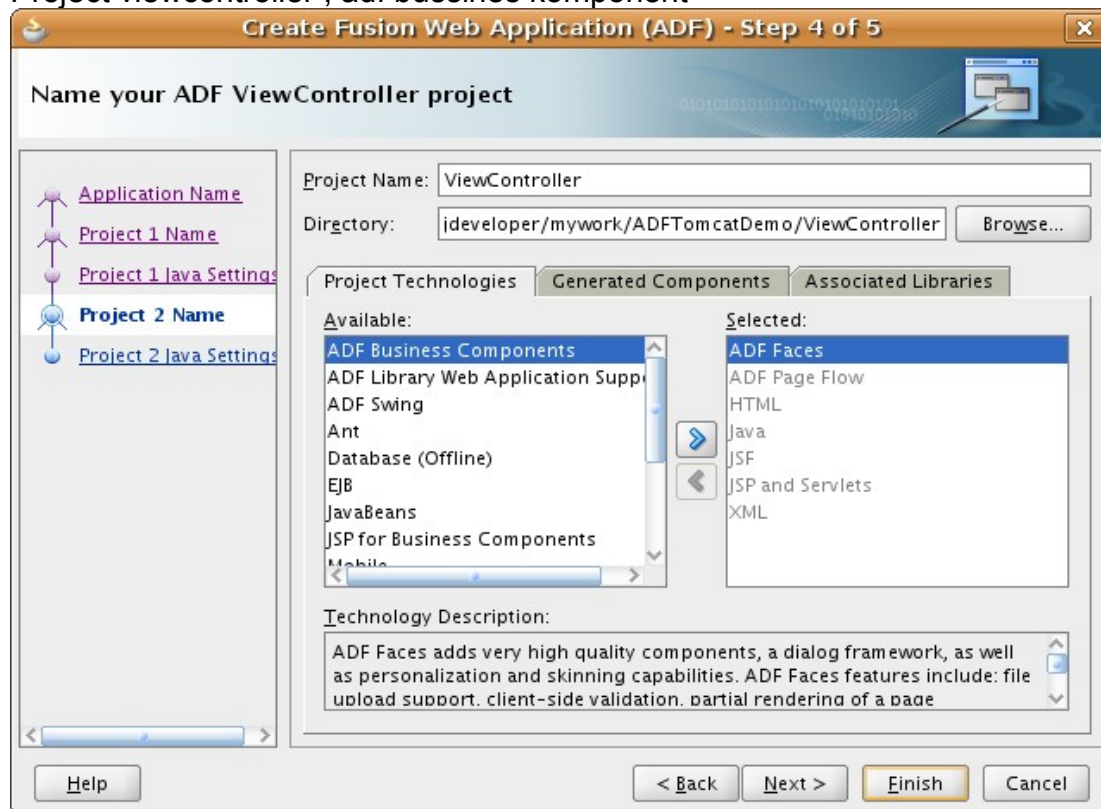
<next>

Keep as it is



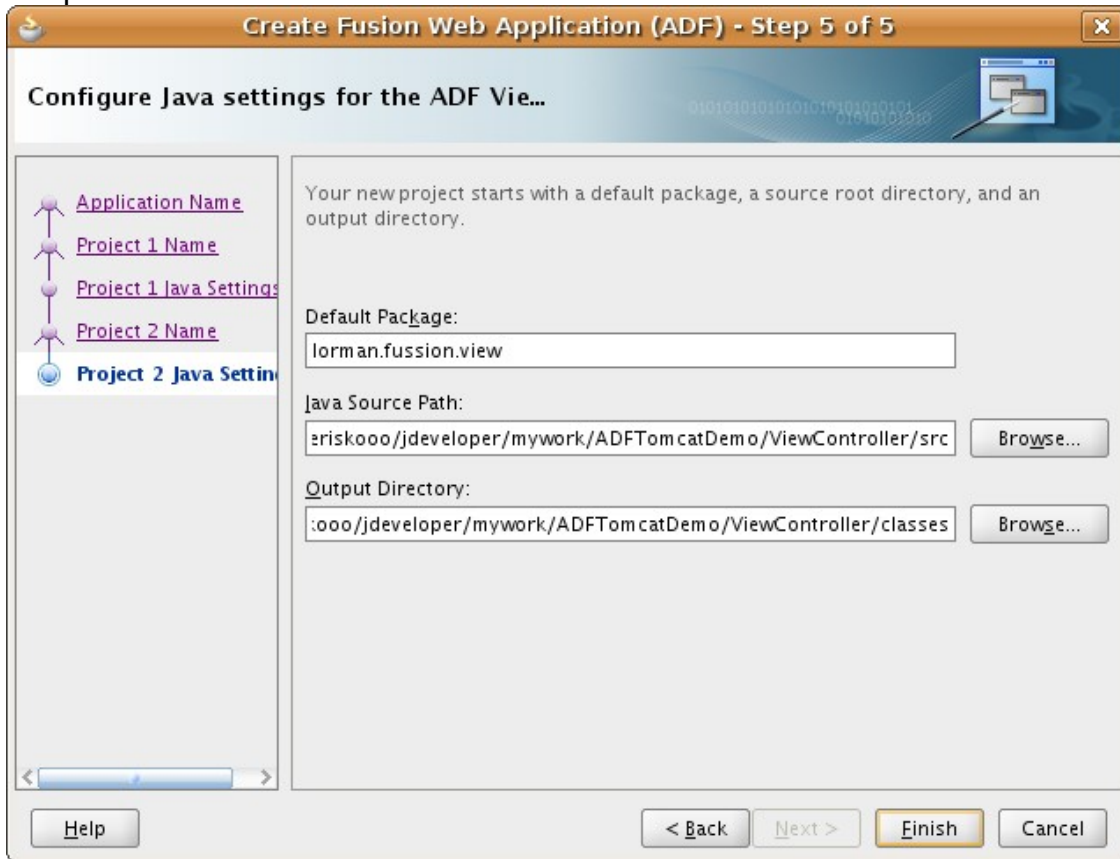
<next>

Project viewcontroller , adf bussines komponent



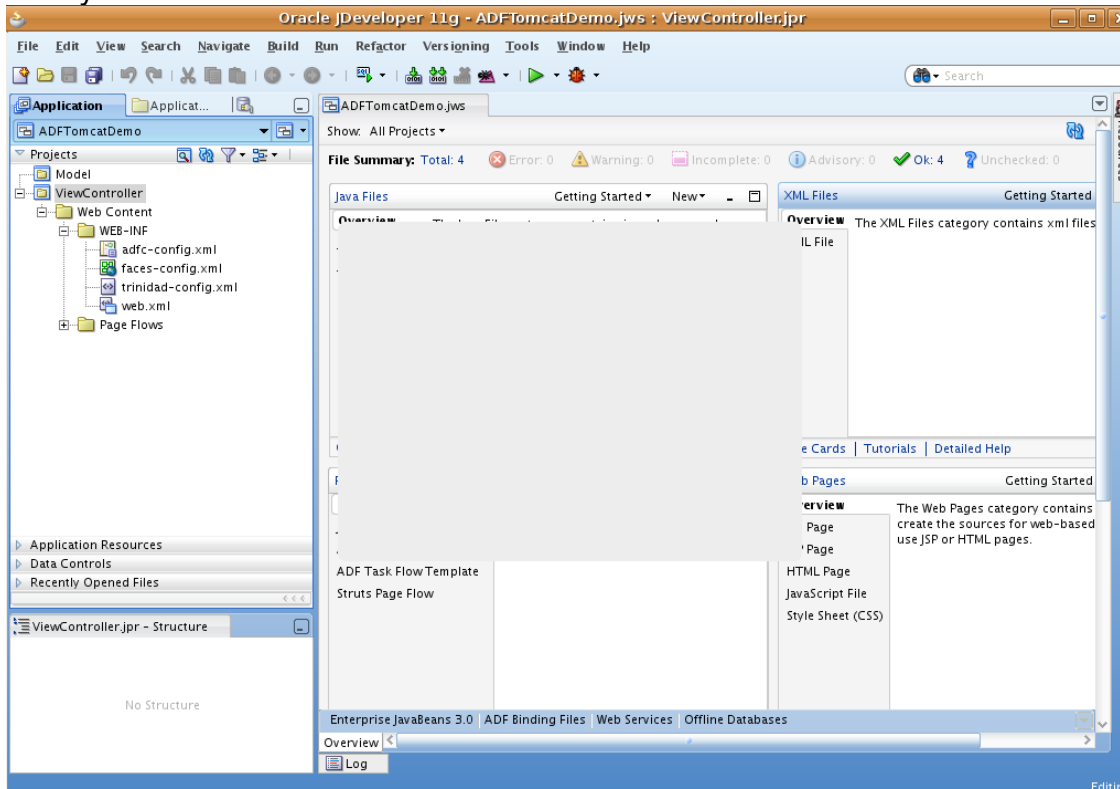
<next>

Keep as it is



<finish>

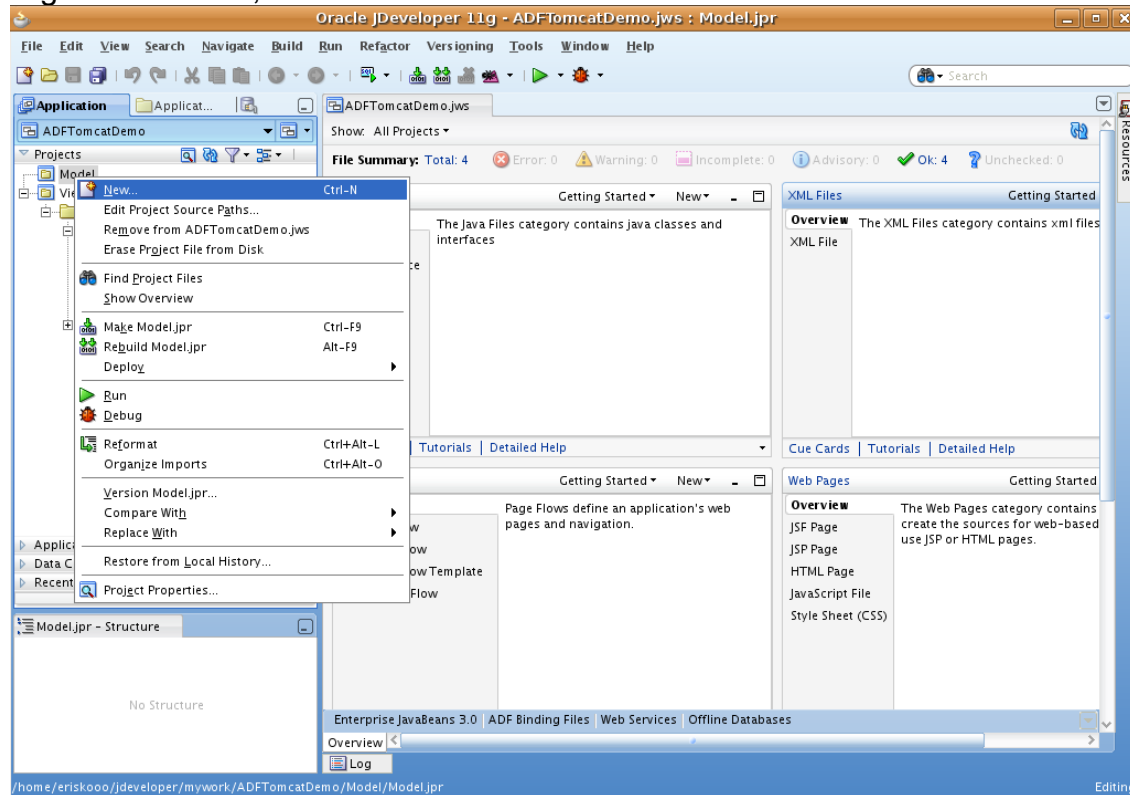
Now your screen should look like this



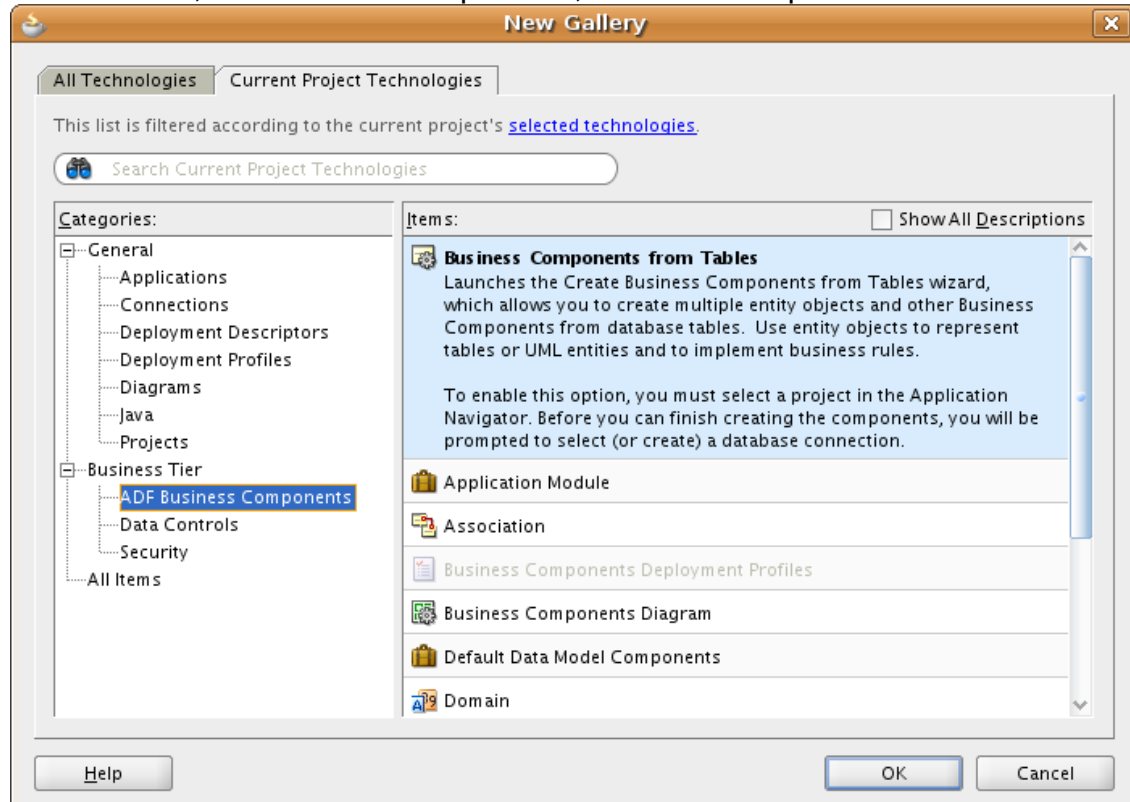
## Create Model

Add Entity

Right click Model, new



Bussines tier, adf bussines components, bussines components from tables



<ok>

## Connection, + (create databáze connection)

Setup as on screen

Configure a new database connection and add it to the current application (ADFTomcatDemo).

Create Connection In:  Application Resources  IDE Connections

Connection Name:

Connection Type:

Username:  Role:

Password:   Save Password  Deploy Password

- Oracle (JDBC) Settings -

Enter Custom JDBC URL

Driver:

Host Name:  JDBC Port:

SID:

Service Name:

Success!

<ok>

## Back on screen initialize bussines components object

This project has not yet been initialized for Business Components. After specifying the following information for your Business Components Project (jpx file), you will be prompted to create your Business Component(s).

Specify the database connection that lets you create Business Components from existing database objects.

Connection:

User Name:

Driver:

Connect String:

Choose the proper SQL flavor and type map that fits your application.

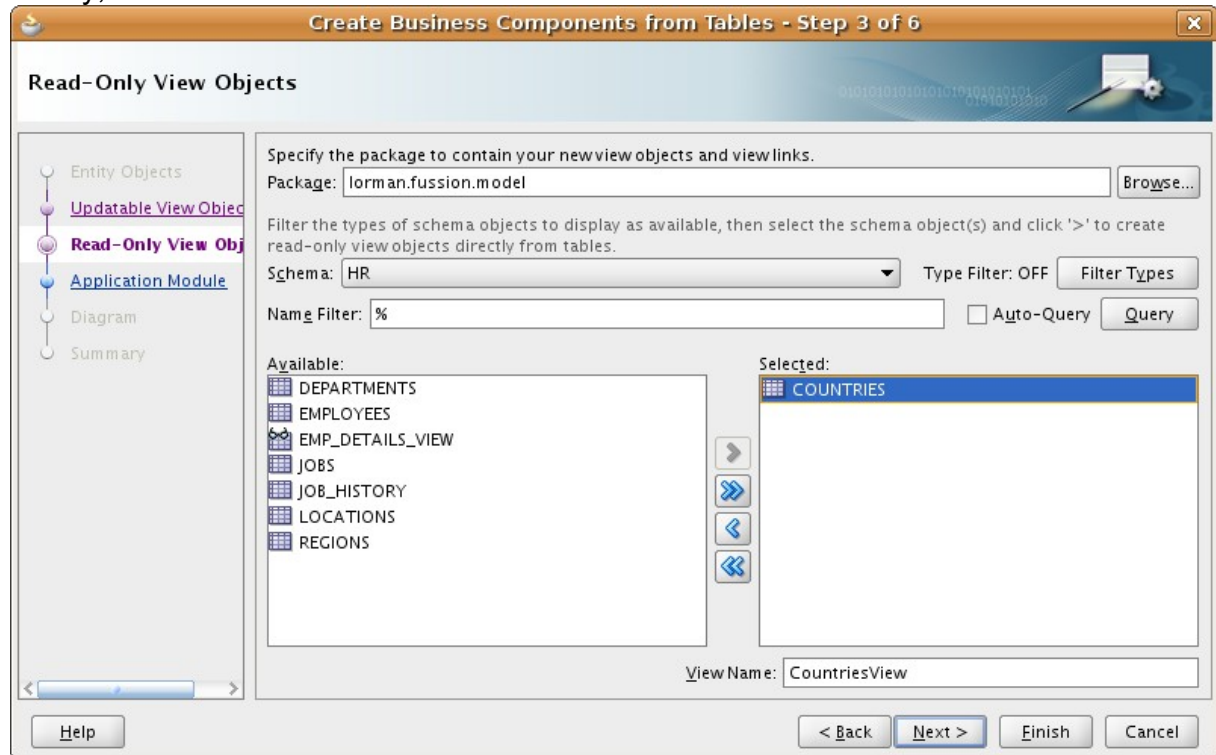
SQL Flavor:

Type Map:

<ok>

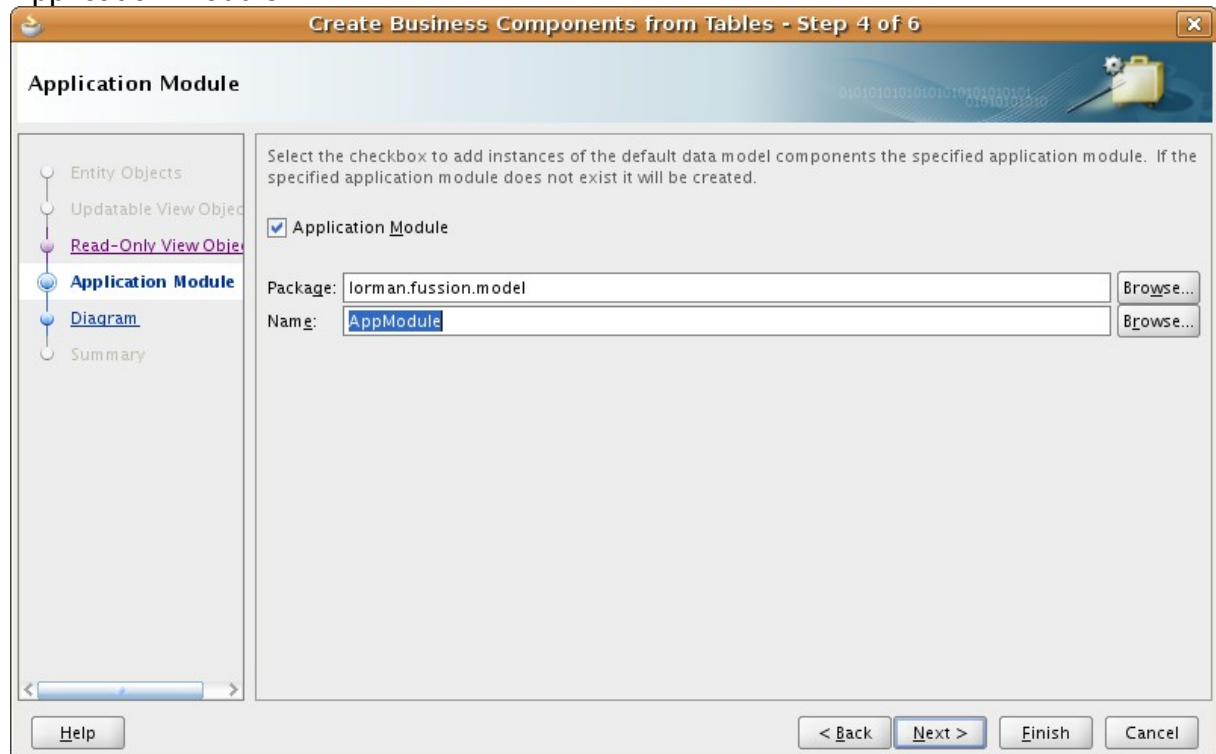
Entity objects, next  
Updatable view object, next

Read-only view objects  
Query, select Countries



<next>

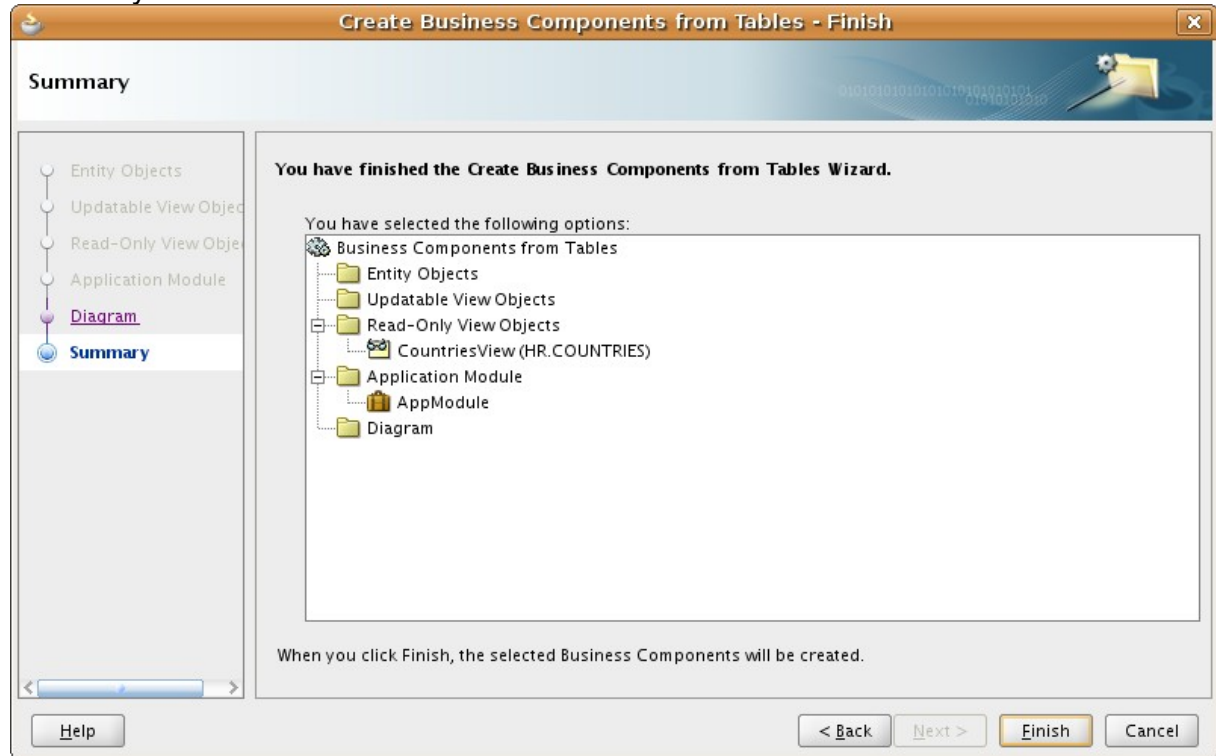
Application module



<next>

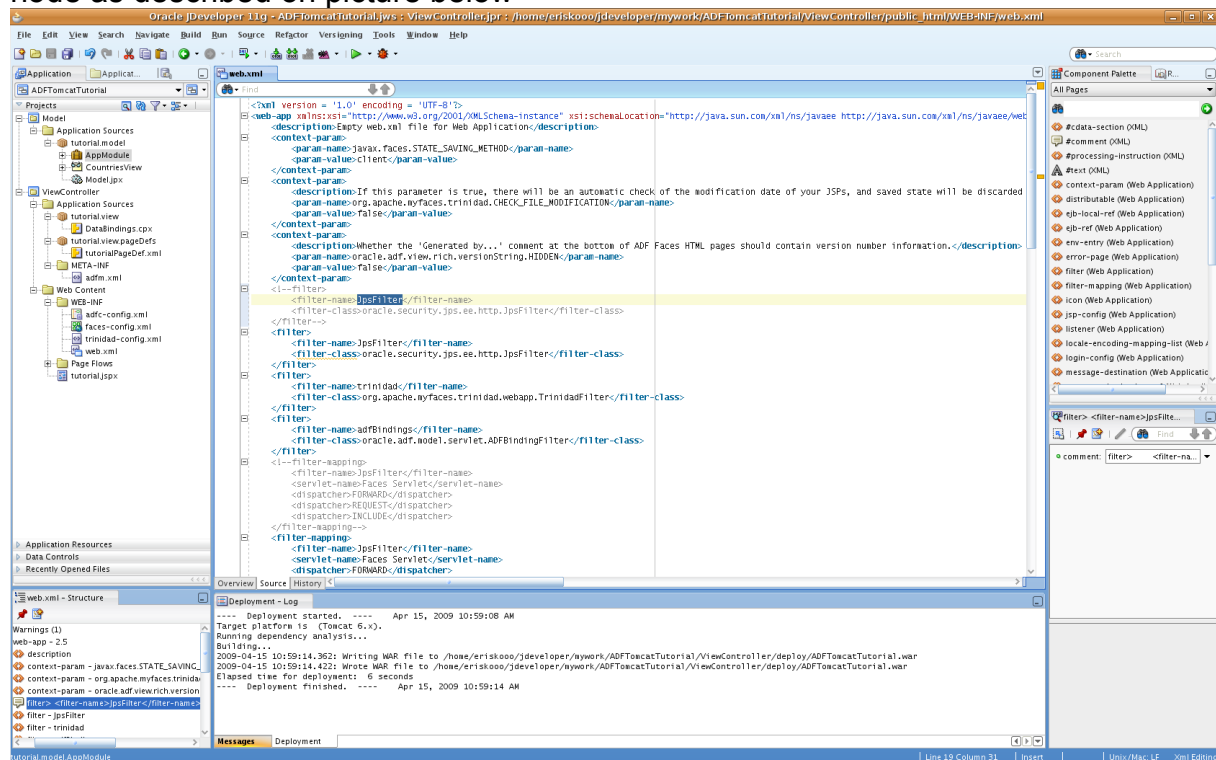
Diagram  
<next>

Summary

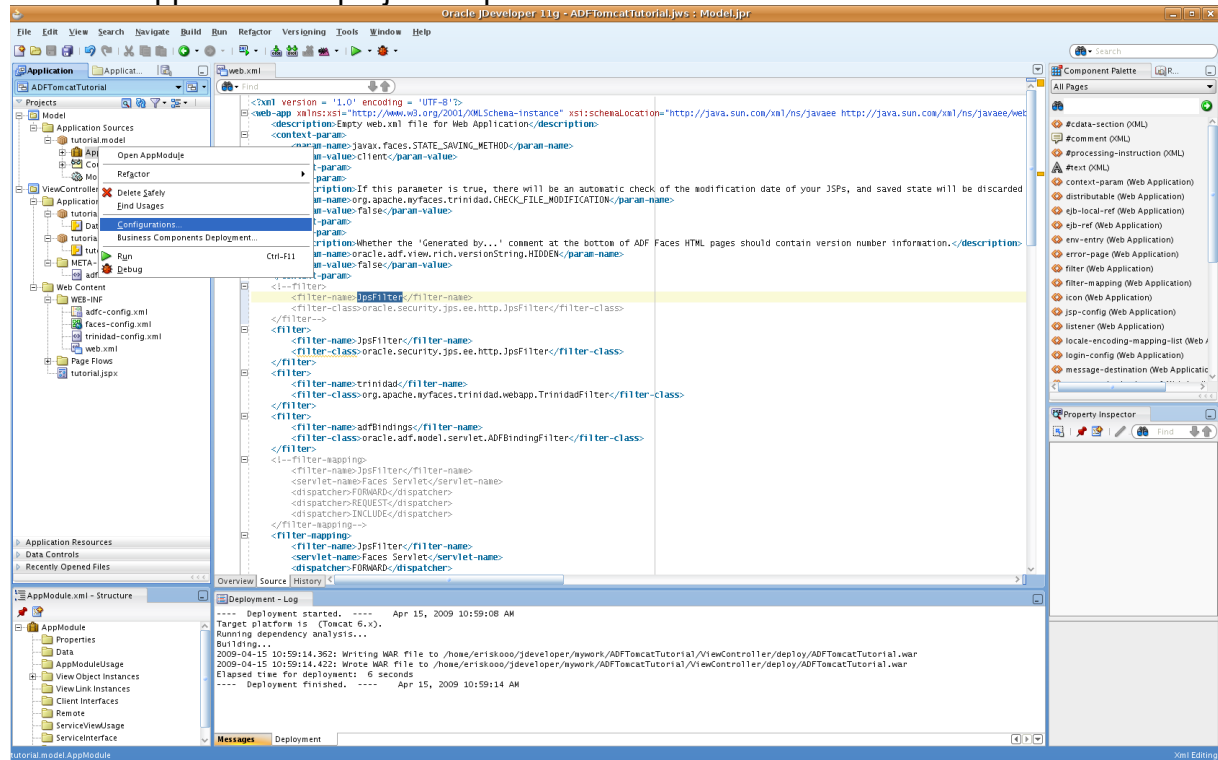


<finish>

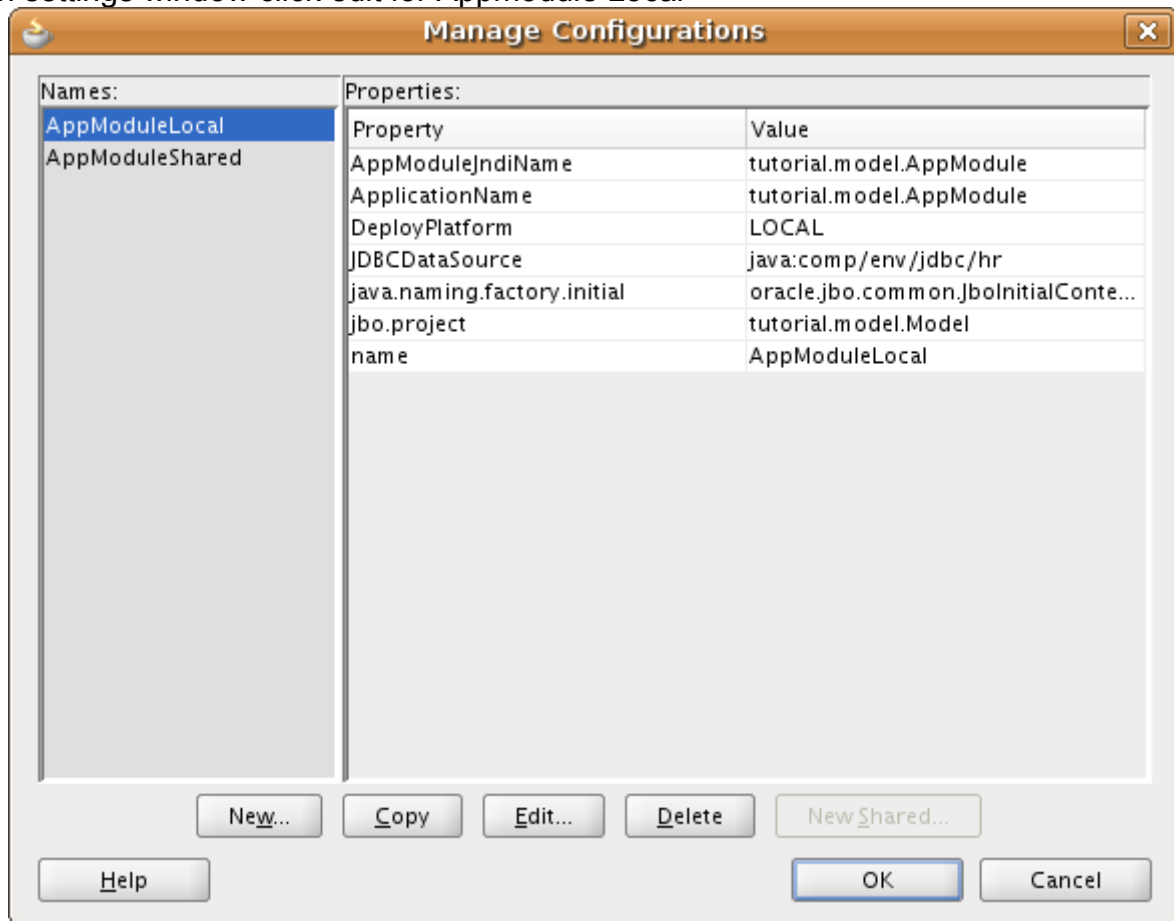
Click on Web.xml, open source view and comment out JpsFilter filter and mapping node as described on picture below



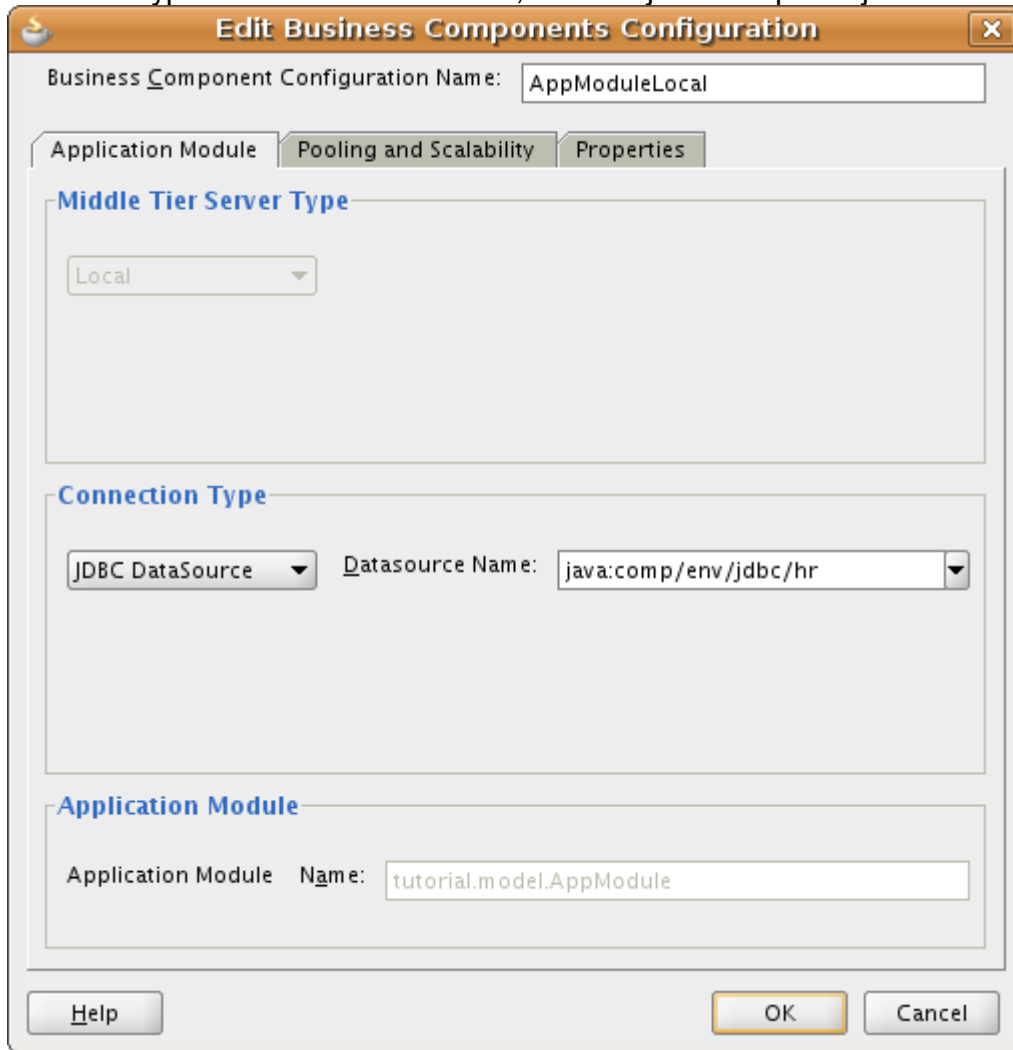
Click on AppModule in project inspector



in settings window click edit for AppModule Local



Set connection type as JDBC DataSource, name it java:comp/env/jdbc/hr

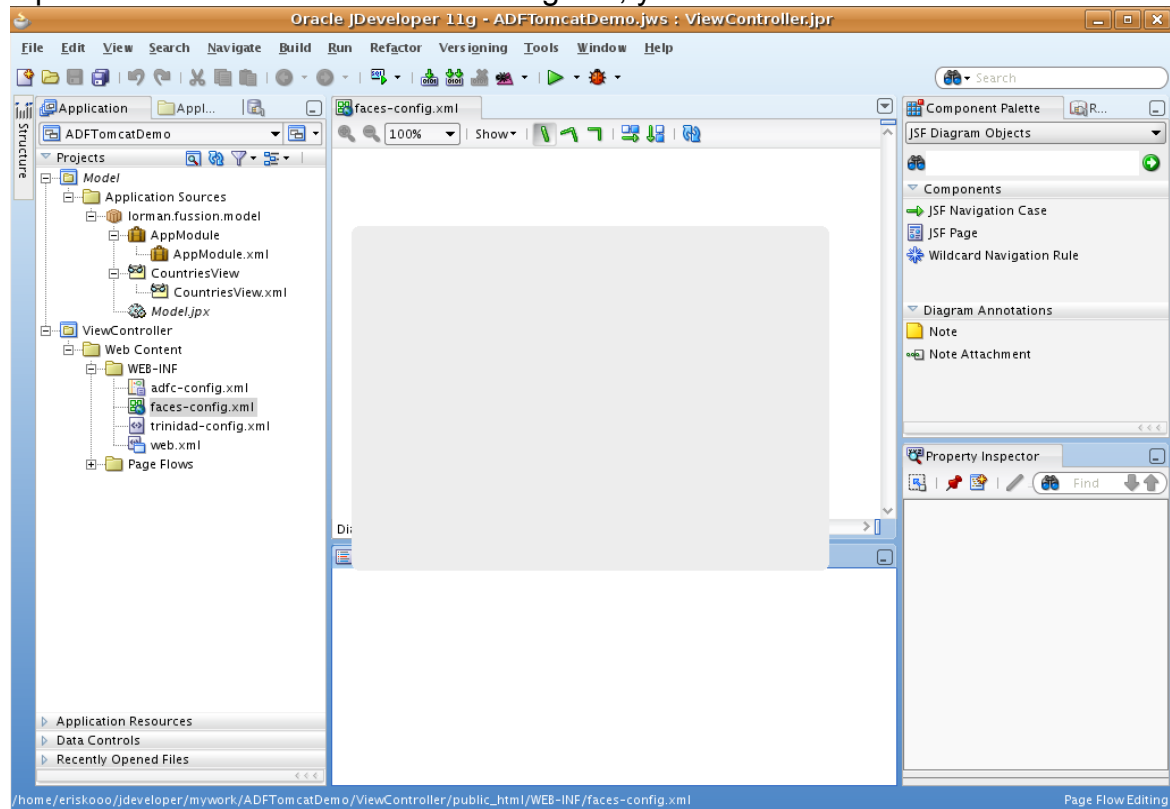


Proceed same steps for AppModuleShared, then click OK

Model part of our application is now ready.

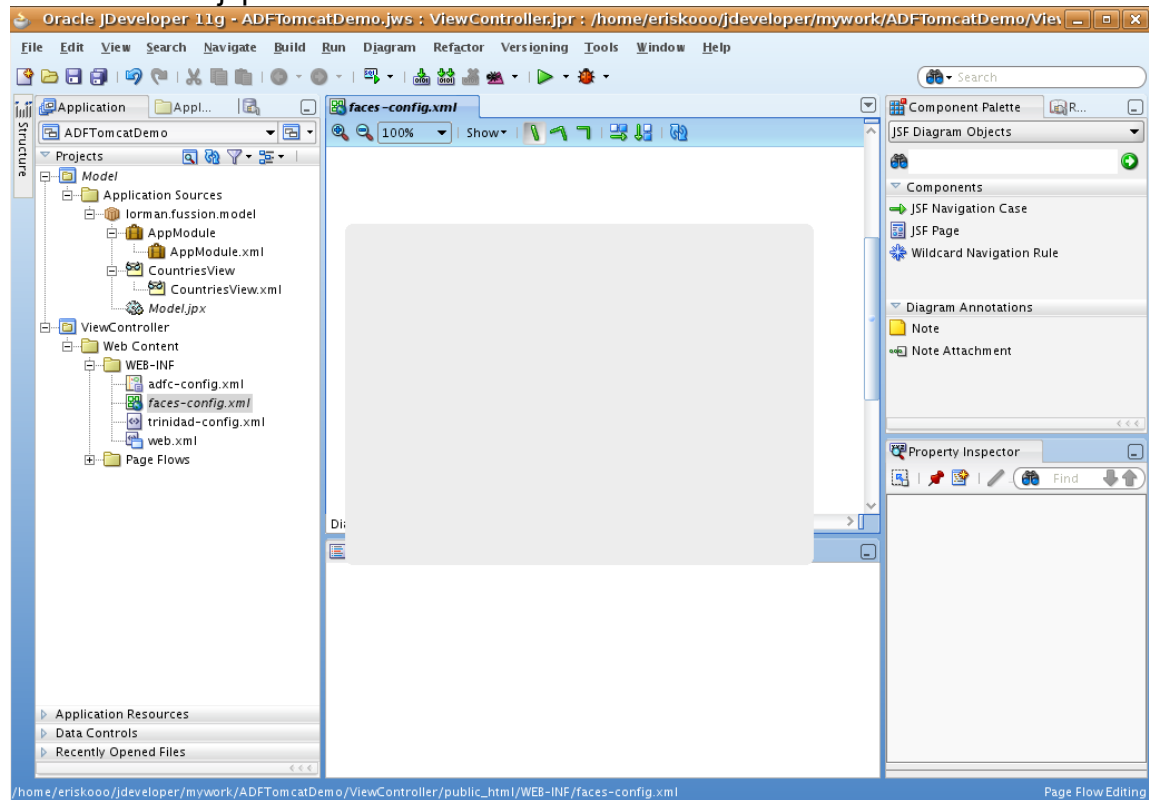
## Create View

Open from ViewController faces-config.xml, your screen should look like this

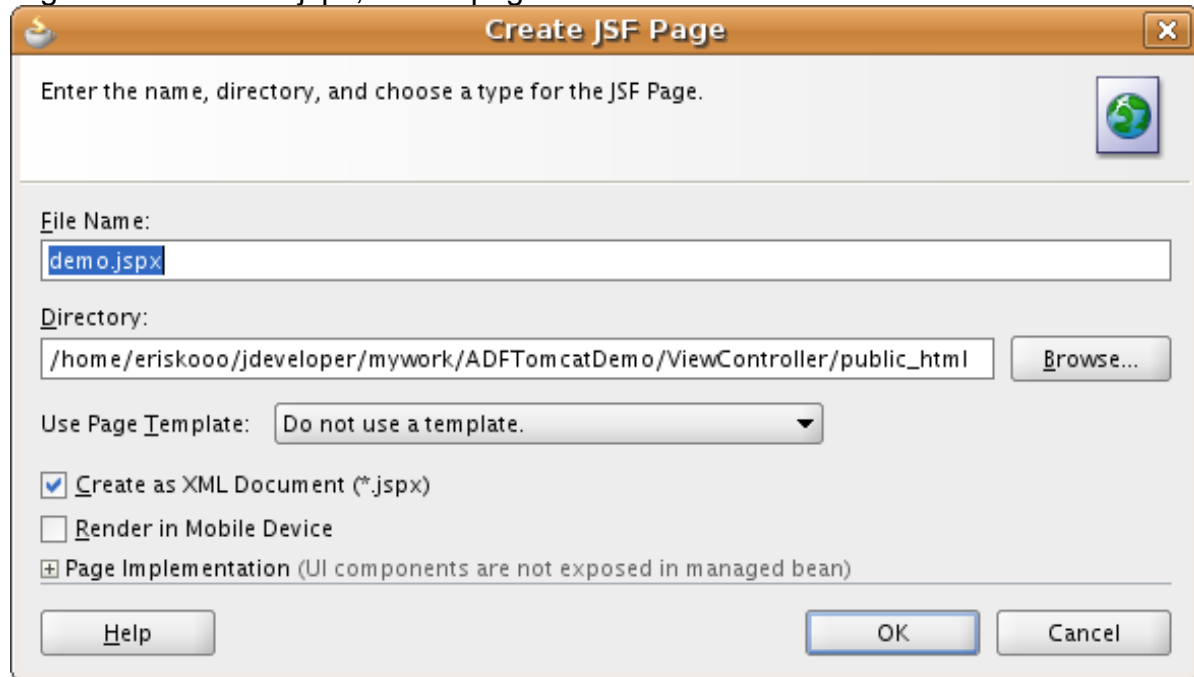


From komponent patlete / components / JSF Page grap to view

Name it demo.jspx

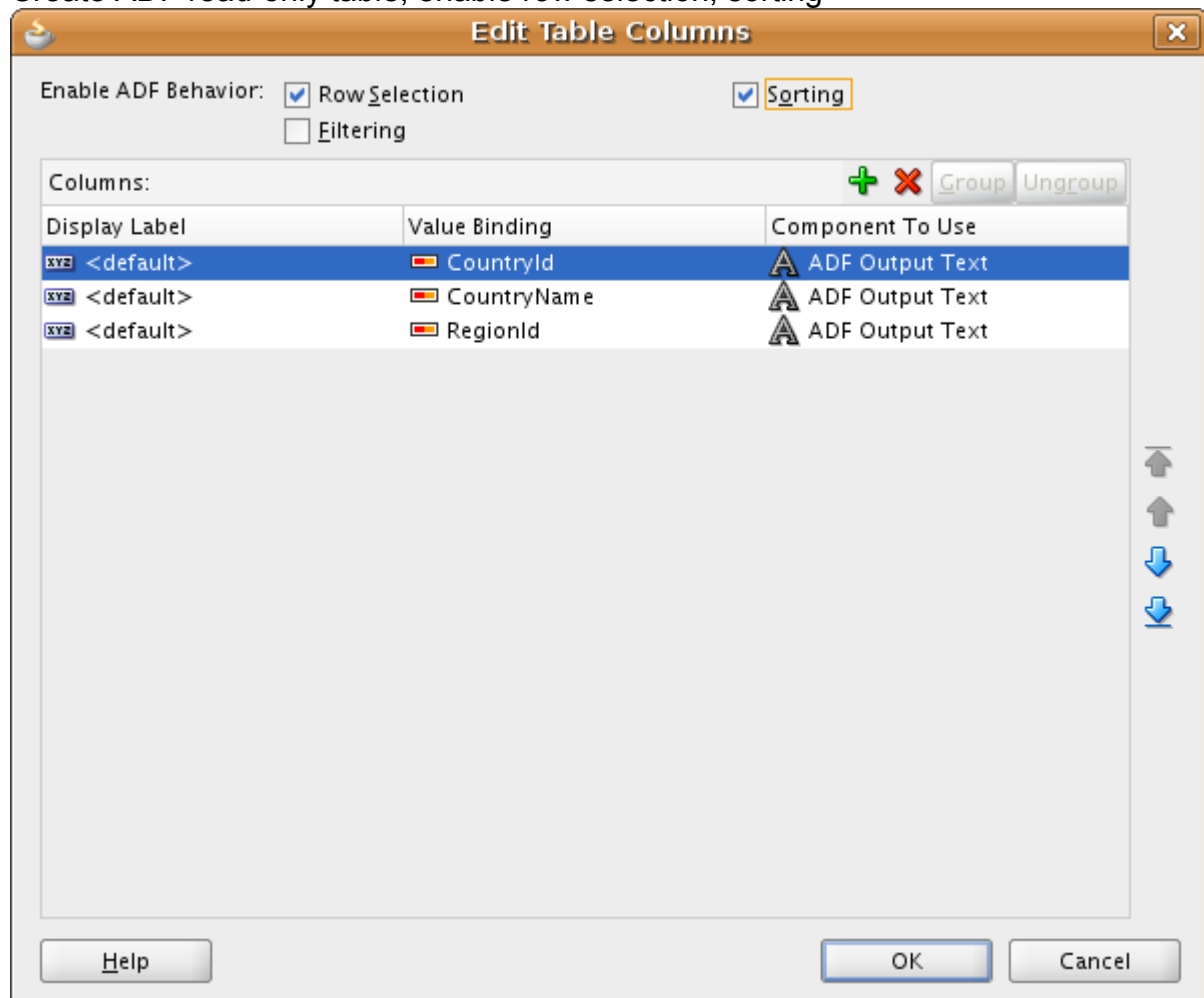


Right click on demo.jspx, create page



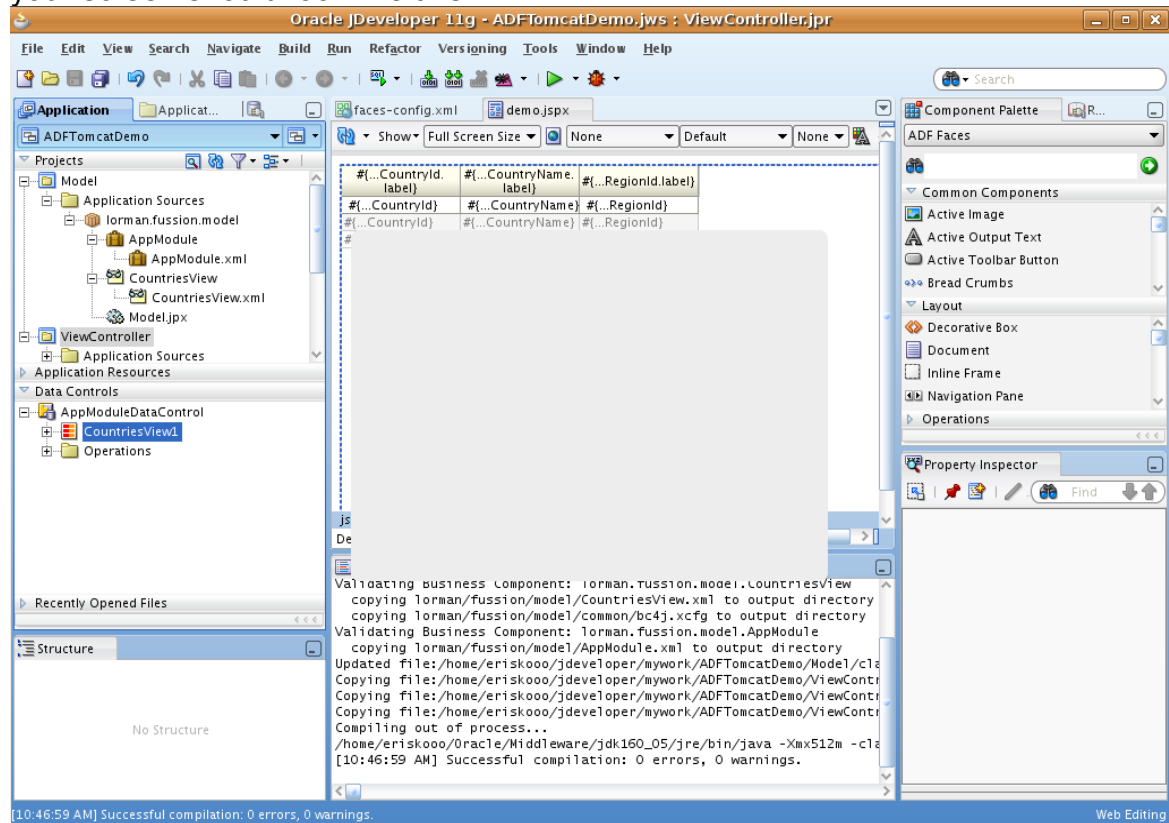
<ok>

From data controls / AppModuleDataControl / CountriesView1 drag on center screen  
Create ADF read-only table, enable row selection, sorting

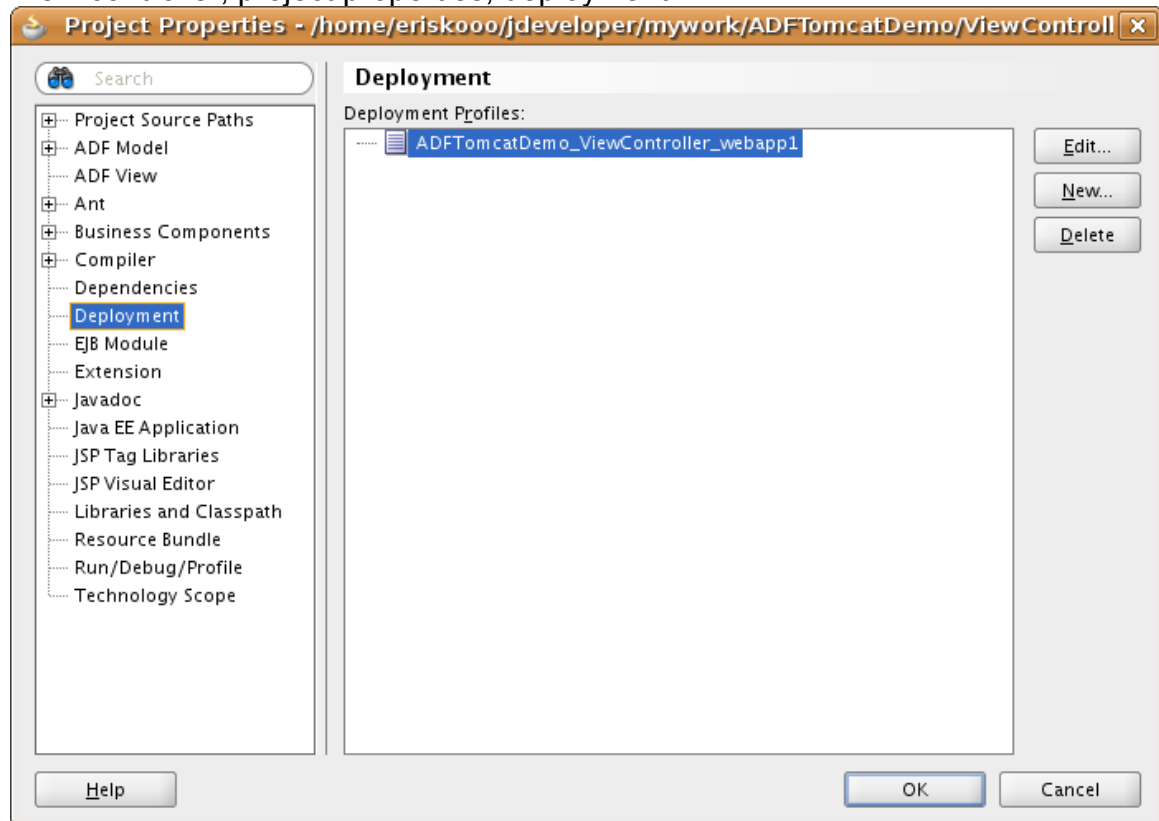


<ok>

Build project,  
your screen should look like this

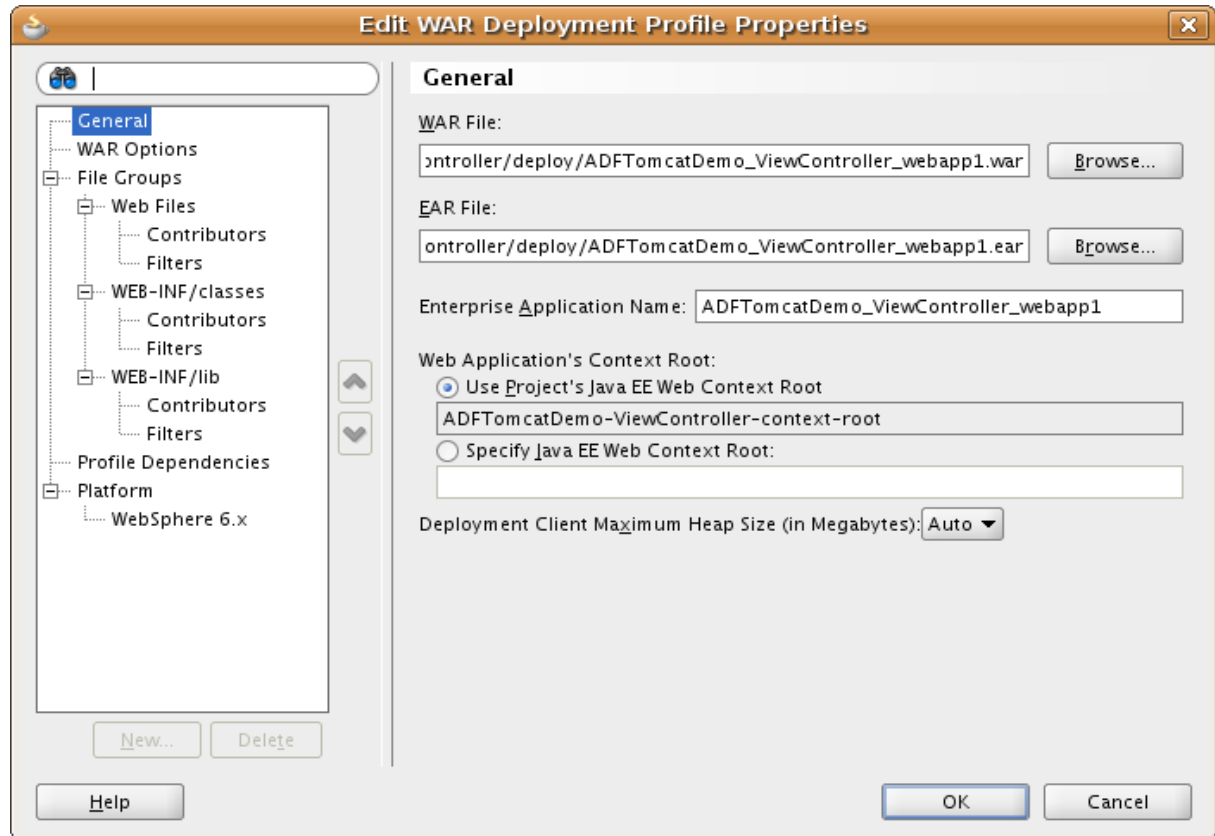


Setup Application  
View controller, project properties, deployment

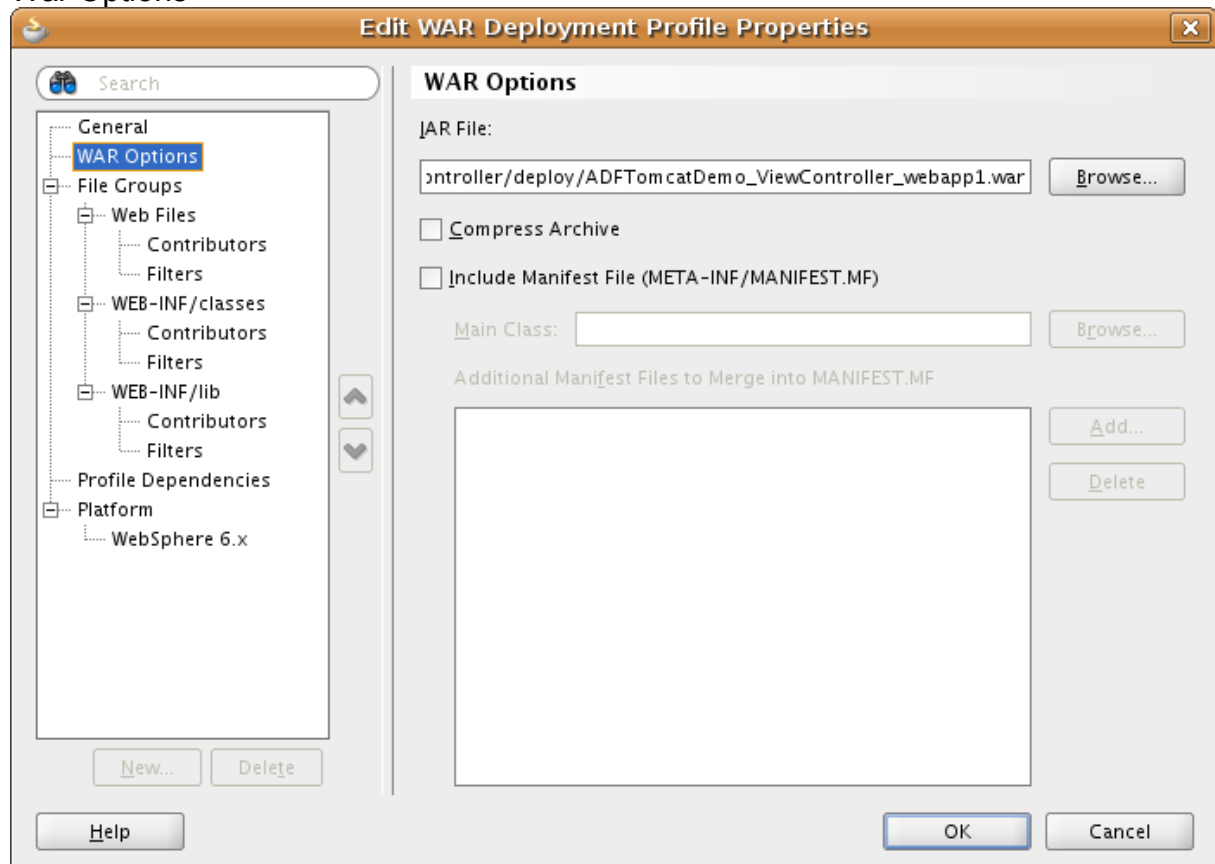


Edit

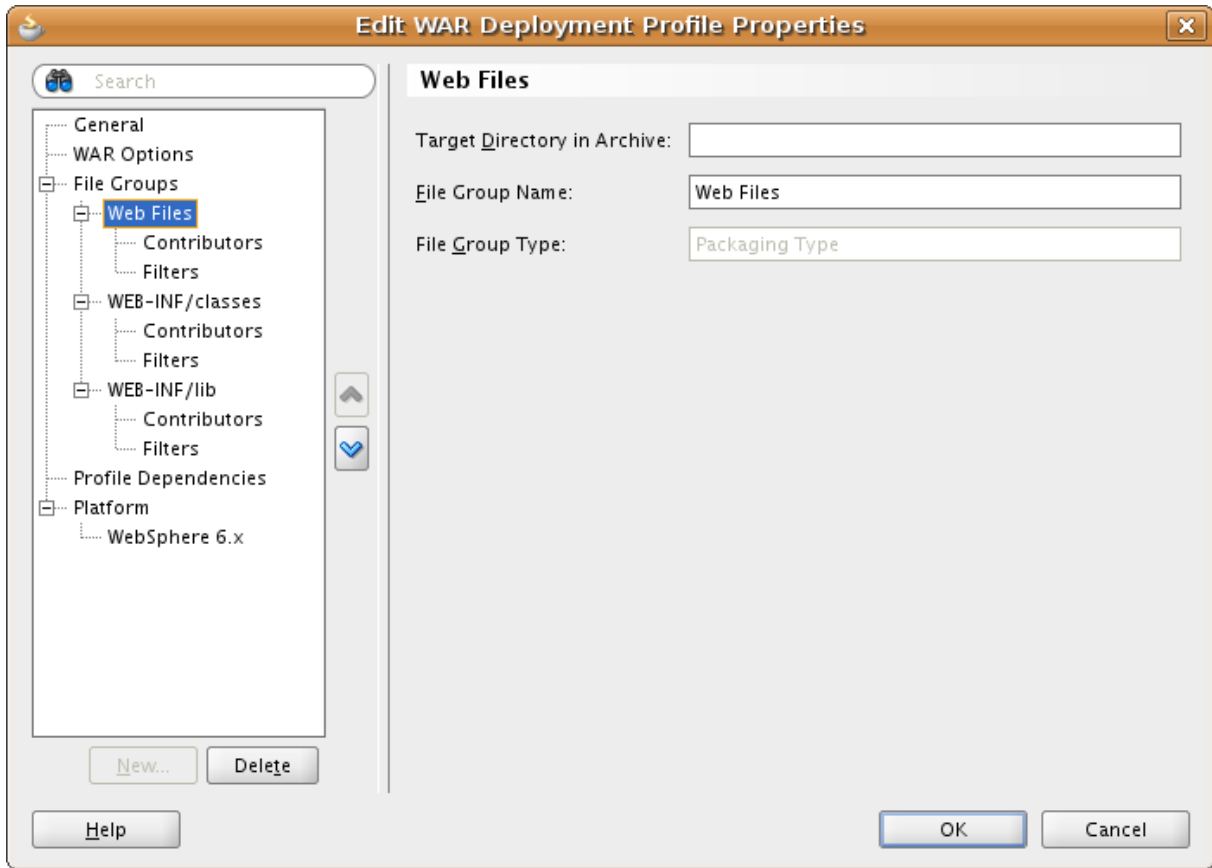
## General



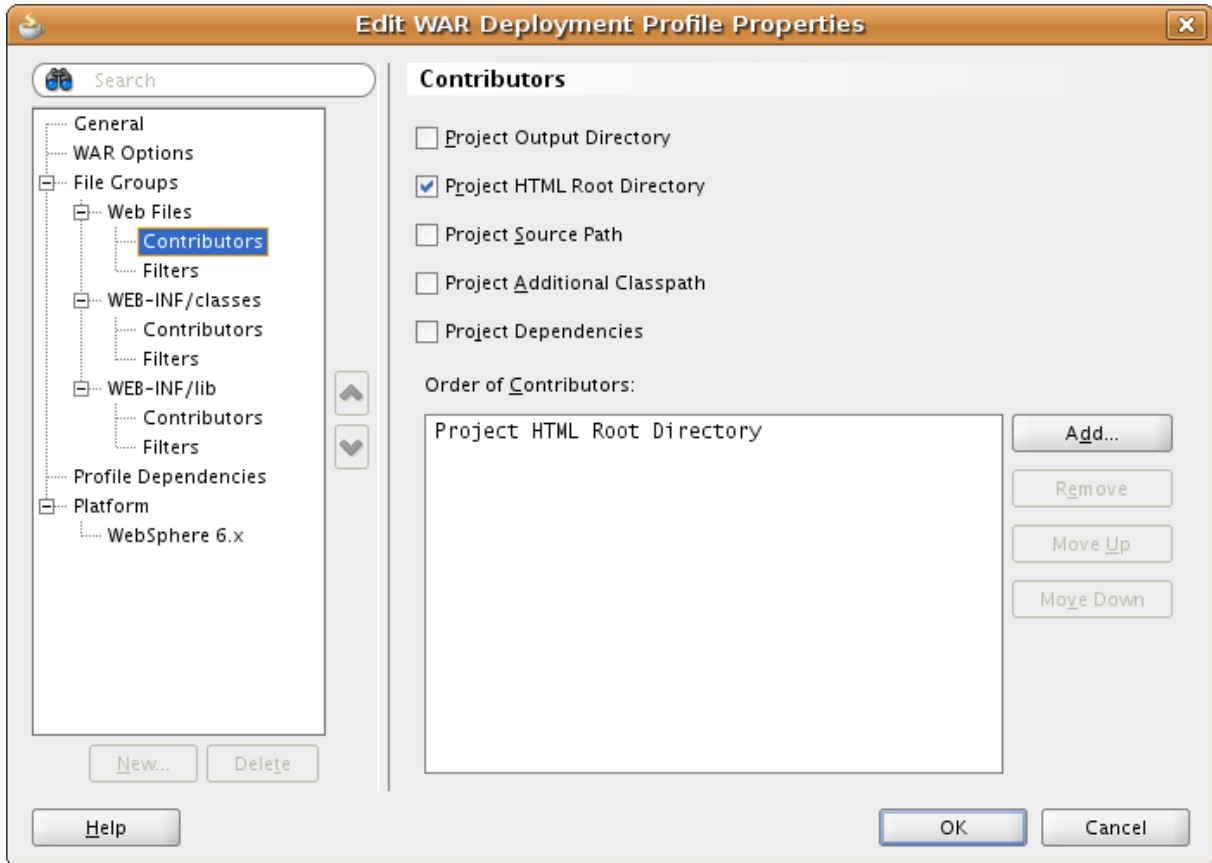
## War Options



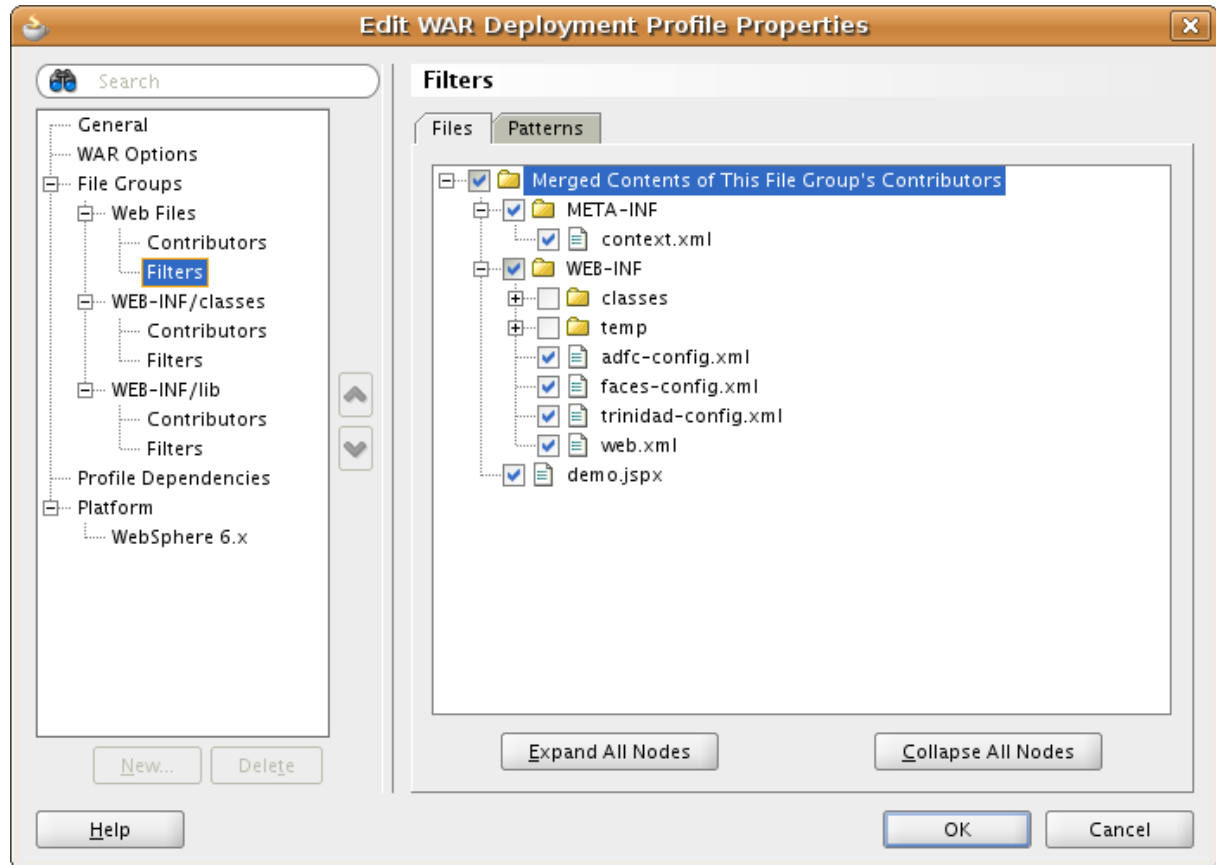
### web files



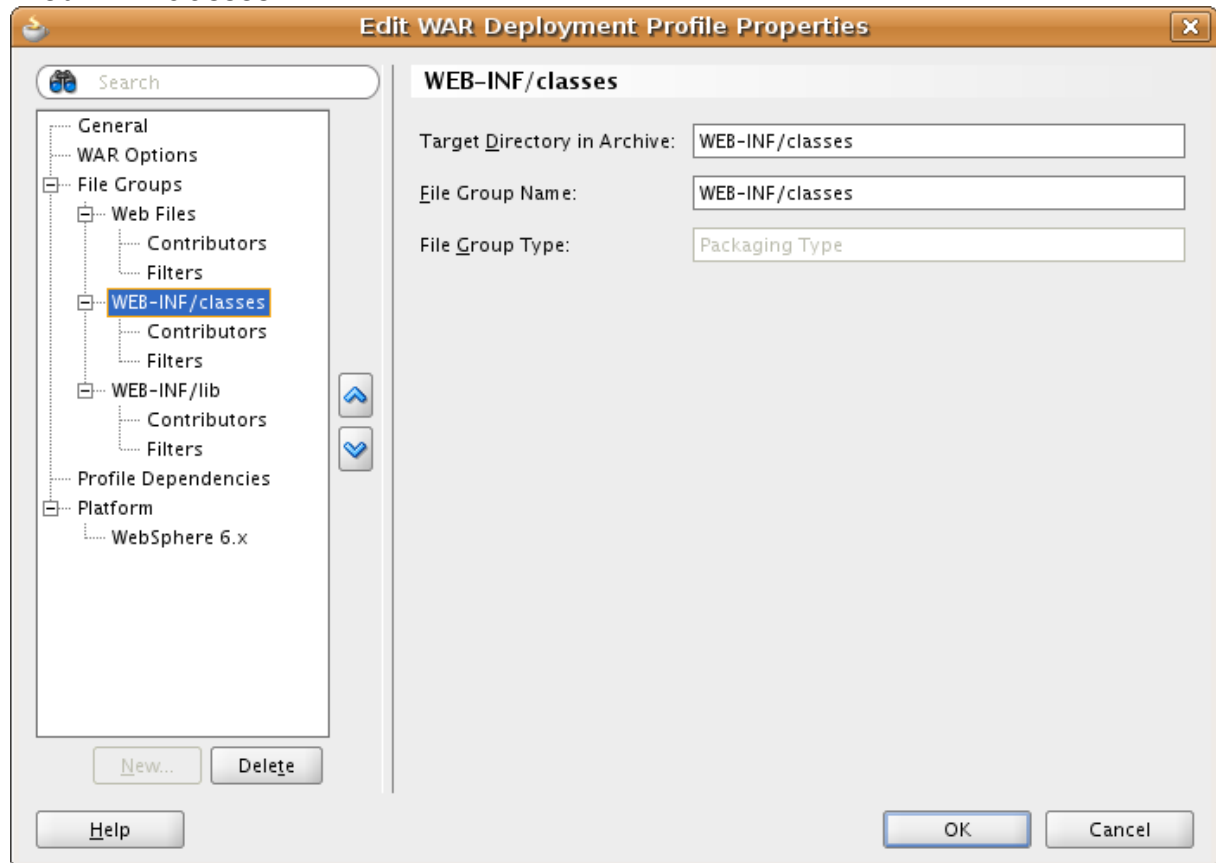
### web files - contributors



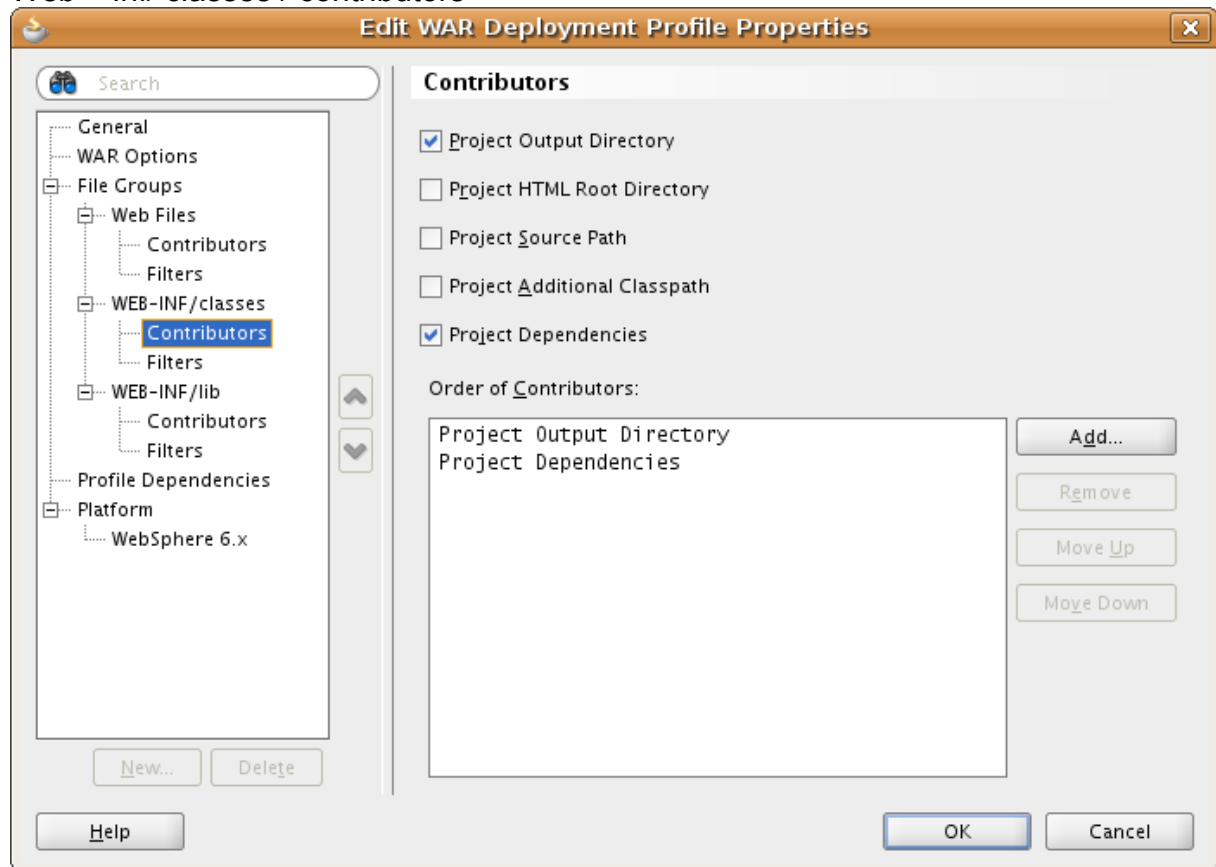
### web files - filters



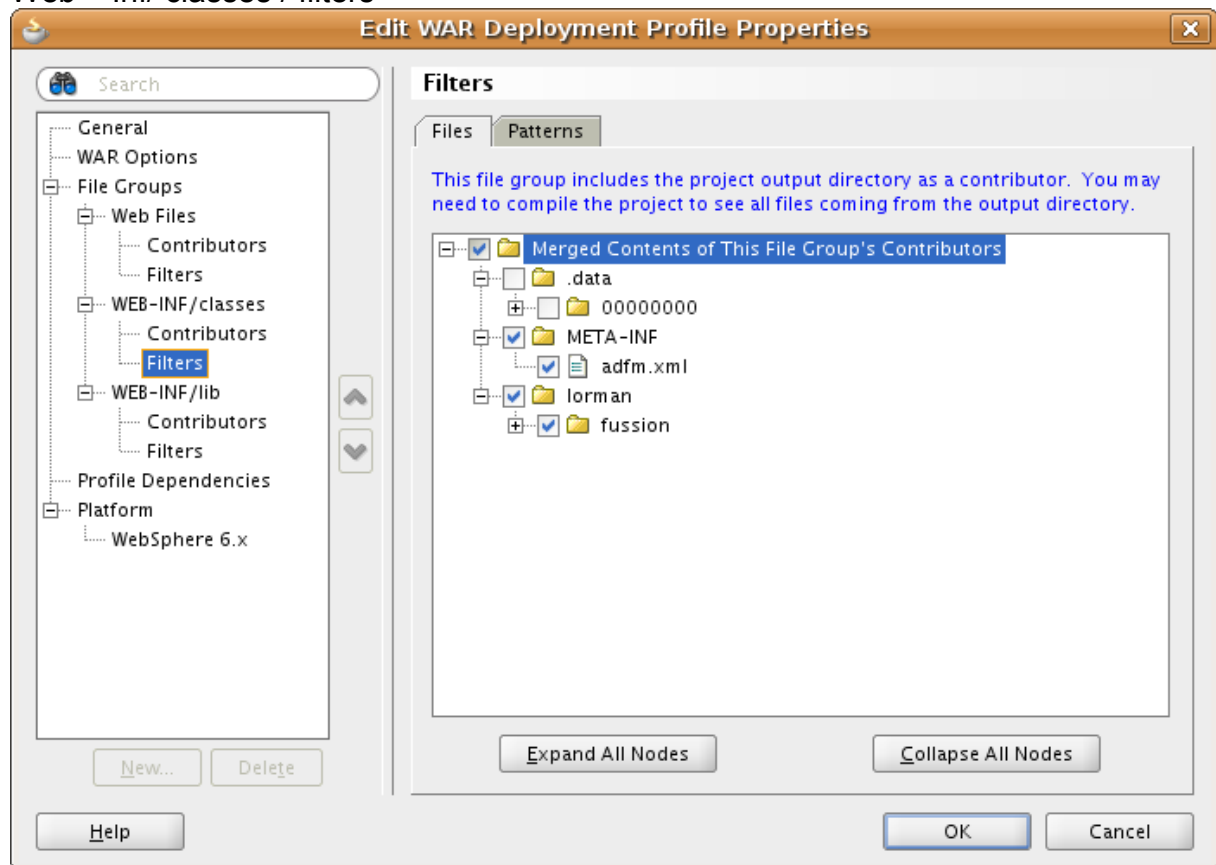
### Web - Inf/ classes



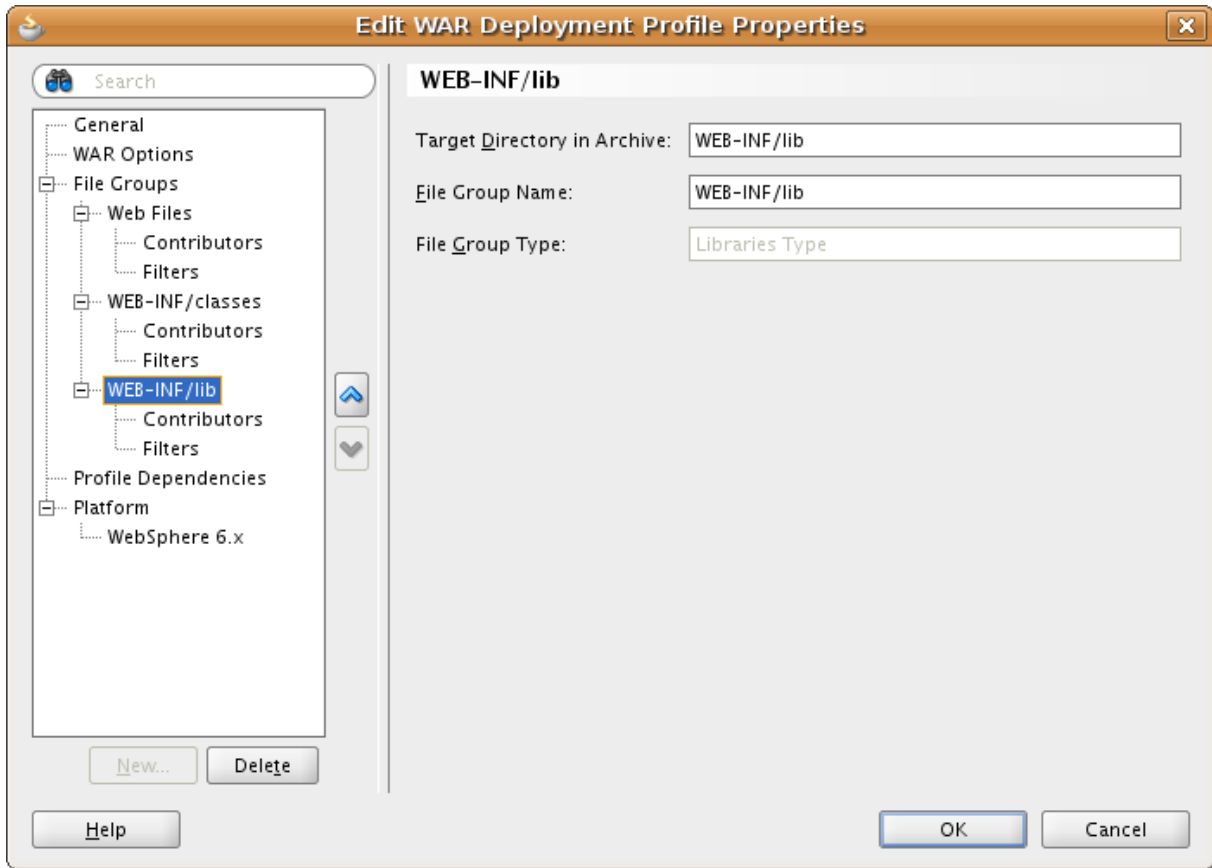
### Web – Inf/ classes / contributors



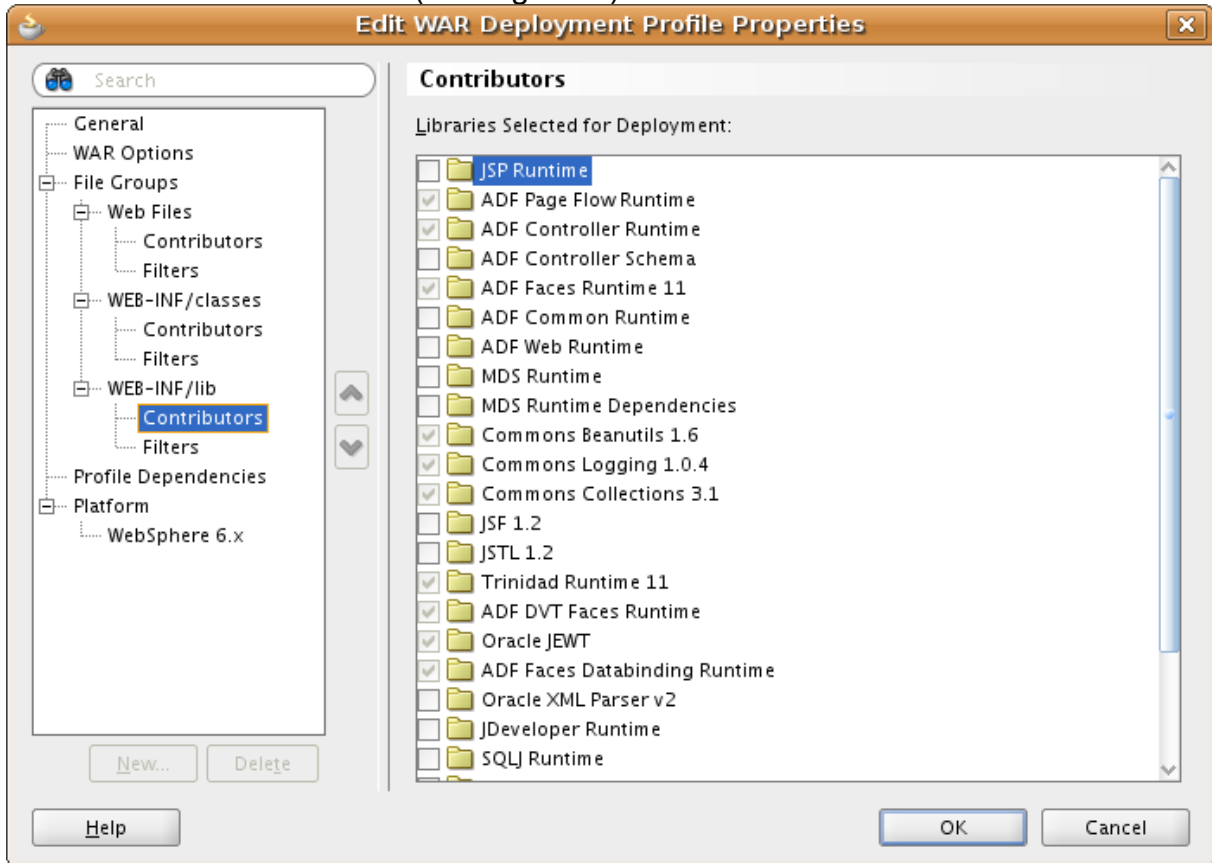
### Web – Inf/ classes / filters



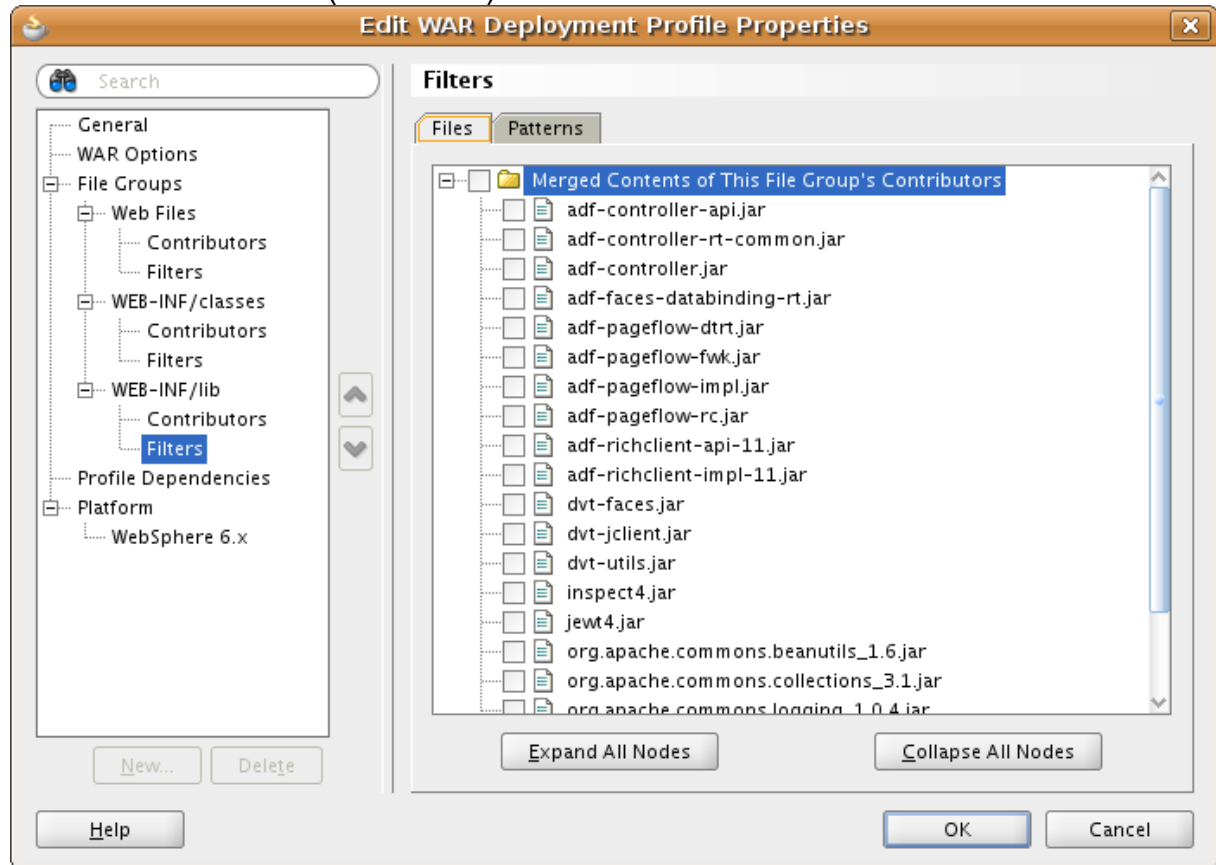
### Web – inf / Lib



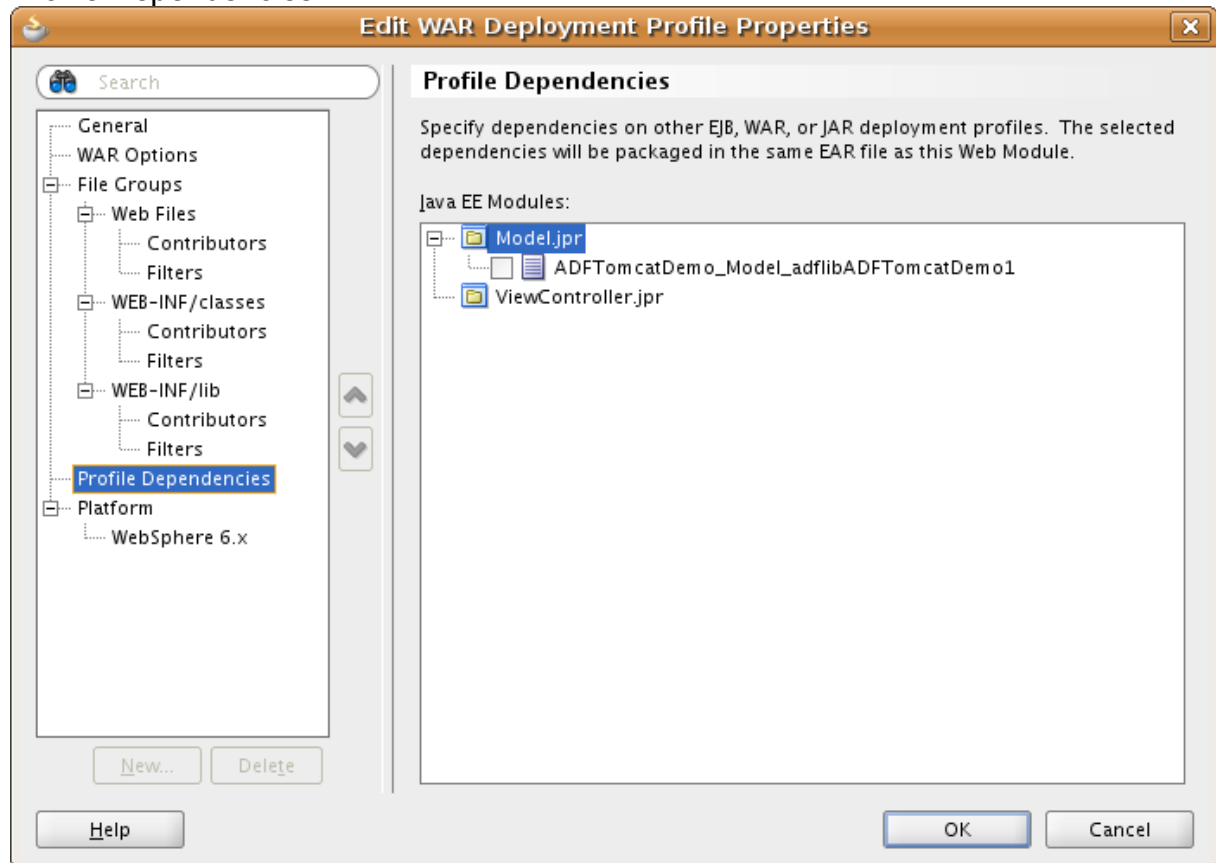
### Web – inf / Lib / contributors (nothing to do)



### Web – inf / Lib / filters (remove all)

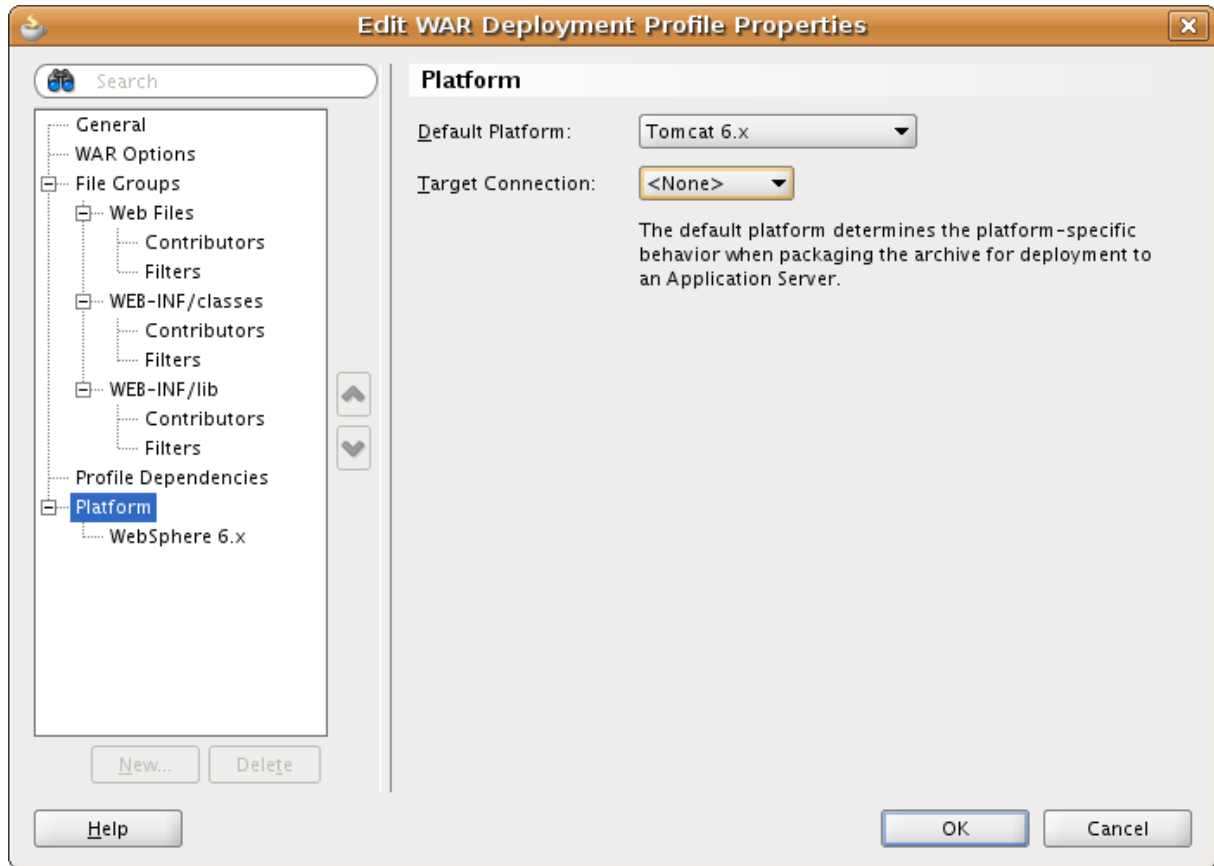


### Profile Dependencies



## Platform

### Tomcat 6.x



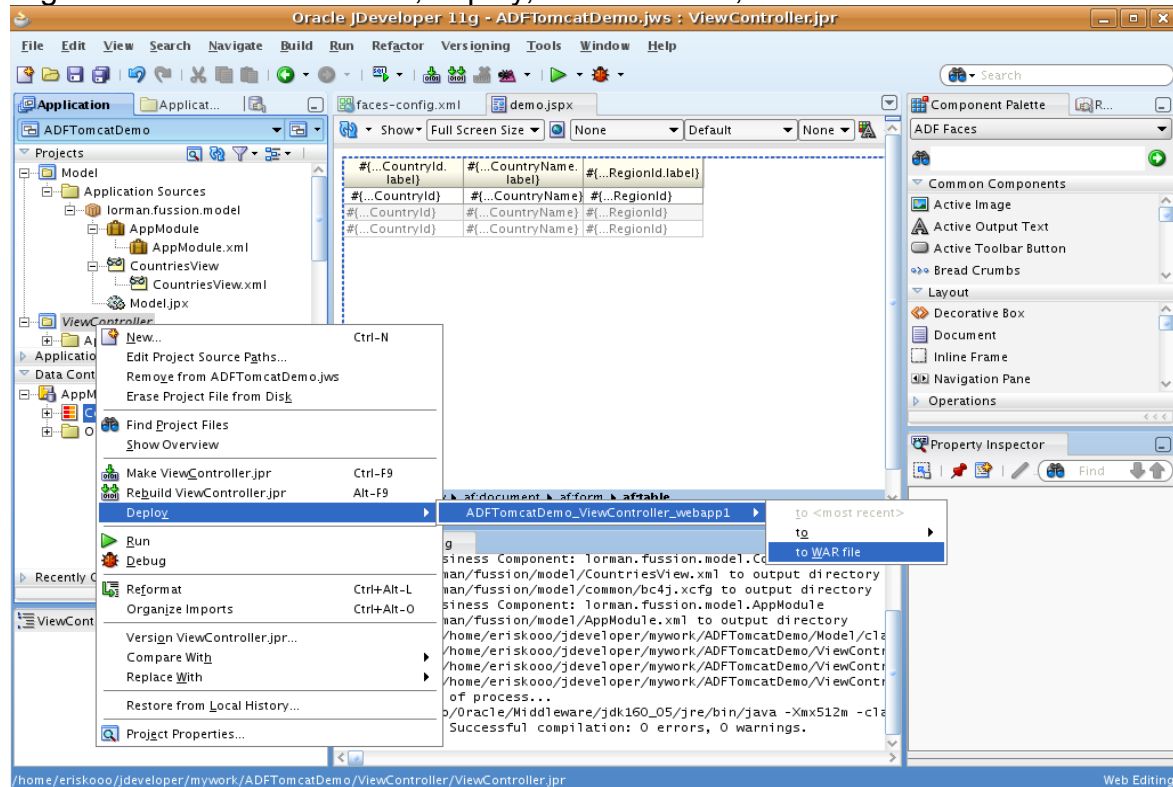
<ok>

<ok>

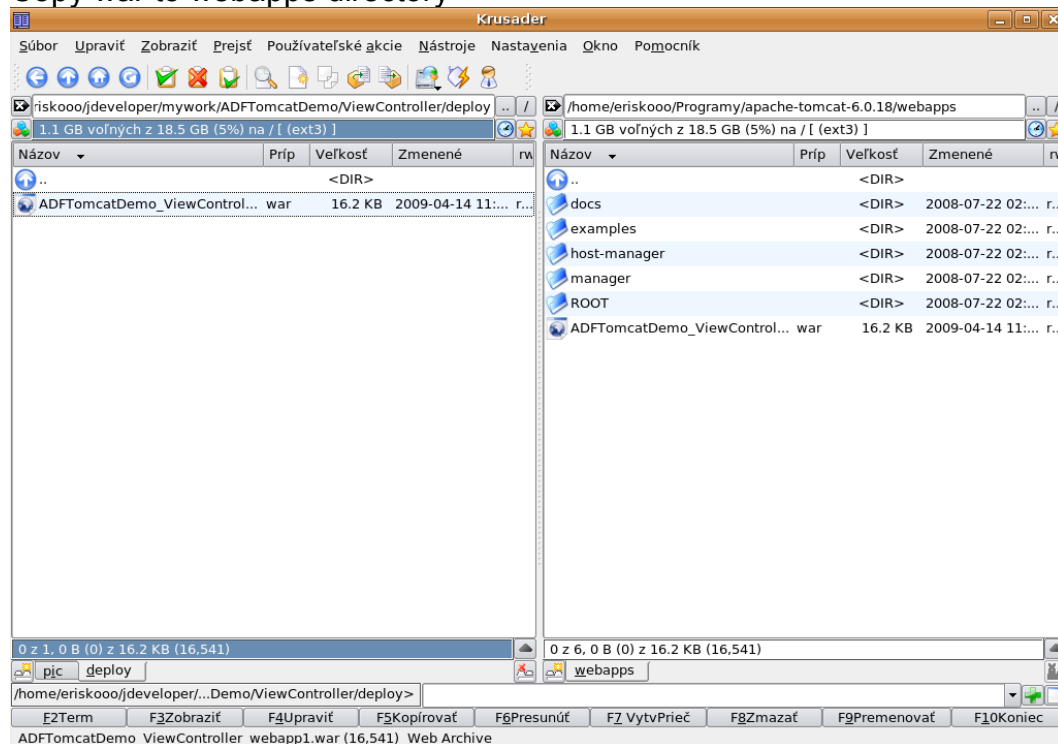
## Deploy on Tomcat

- Add context.xml to /adftomcatdemo/viewcontroller/public\_html/META-INF/
- Setup in context.xml connection
- url, user, password, databasename, port
- Create war

Right click on view controller, deploy, adftomcat ... , to War file



## Copy war to webapps directory



## Setup tomcat

Copy libraries from list to Tomcat lib directory

adf-controller-api.jar  
adf-controller.jar  
adf-controller-rt-common.jar  
adf-faces-databinding-rt.jar  
adflibfilter.jar  
adflogginghandler.jar  
adfm.jar  
adfmweb.jar  
adf-pageflow-dtrt.jar  
adf-pageflow-fwk.jar  
adf-pageflow-impl.jar  
adf-pageflow-rc.jar  
adf-richclient-api-11.jar  
adf-richclient-impl-11.jar  
adf-share-base.jar  
adf-share-ca.jar  
adf-share-support.jar  
annotations-api.jar  
cache.jar  
catalina-ant.jar  
catalina-ha.jar  
catalina.jar  
catalina-tribes.jar  
commons-el.jar  
db-ca.jar  
dms.jar  
dvt-faces.jar  
dvt-jclient.jar  
dvt-utils.jar  
el-api.jar  
fmw\_audit.jar  
identitystore.jar  
inspect4.jar  
jasper-el.jar  
jasper.jar  
jasper-jdt.jar  
javatools-nodeps.jar  
javax.management.j2ee\_1.0.jar  
jwt4.jar  
jmxframework.jar  
jmxspi.jar  
jps-api.jar  
jps-common.jar  
jps-ee.jar  
jps-internal.jar  
jps-unsupported-api.jar  
jsf-api.jar  
jsf-ri.jar

jsp-api.jar  
jstl.jar  
kniznice.txt  
mdsrt.jar  
ojdbc6.jar  
oracle-el.jar  
oraclepki.jar  
oracle.xdb\_11.1.0.6.0.jar  
org.apache.commons.beanutils\_1.6.jar  
org.apache.commons.collections\_3.1.jar  
org.apache.commons.logging\_1.0.4.jar  
osdt\_cert.jar  
osdt\_core.jar  
servlet-api.jar  
share.jar  
standard.jar  
tomcat-coyote.jar  
tomcat-dbcj.jar  
tomcat-i18n-es.jar  
tomcat-i18n-fr.jar  
tomcat-i18n-ja.jar  
trinidad-api.jar  
trinidad-impl.jar  
wls-api.jar  
xercesImpl.jar  
xmlef.jar  
xml.jar  
xmlparserv2.jar

Edit /conf/server.xml

Add data resource node

```
<GlobalNamingResources>

    <Resource name="UserDatabase" auth="Container"
type="org.apache.catalina.UserDatabase" description="User database that can be
updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
pathname="conf/tomcat-users.xml"/>
    <Resource name="jdbc/hr" auth="Container"
type="oracle.jdbc.pool.OracleDataSource"
driverClassName="oracle.jdbc.driver.OracleDriver"
factory="oracle.jdbc.pool.OracleDataSourceFactory"
url="jdbc:oracle:thin:@localhost:1521:xe" user="hr" password="hr" maxActive="20"
maxIdle="10" maxWait="-1"/>

</GlobalNamingResources>
```

Created by Erich, Lorman 04 / 2009, Contact : elorman@dezadata.cz

Edit /conf/context.xml

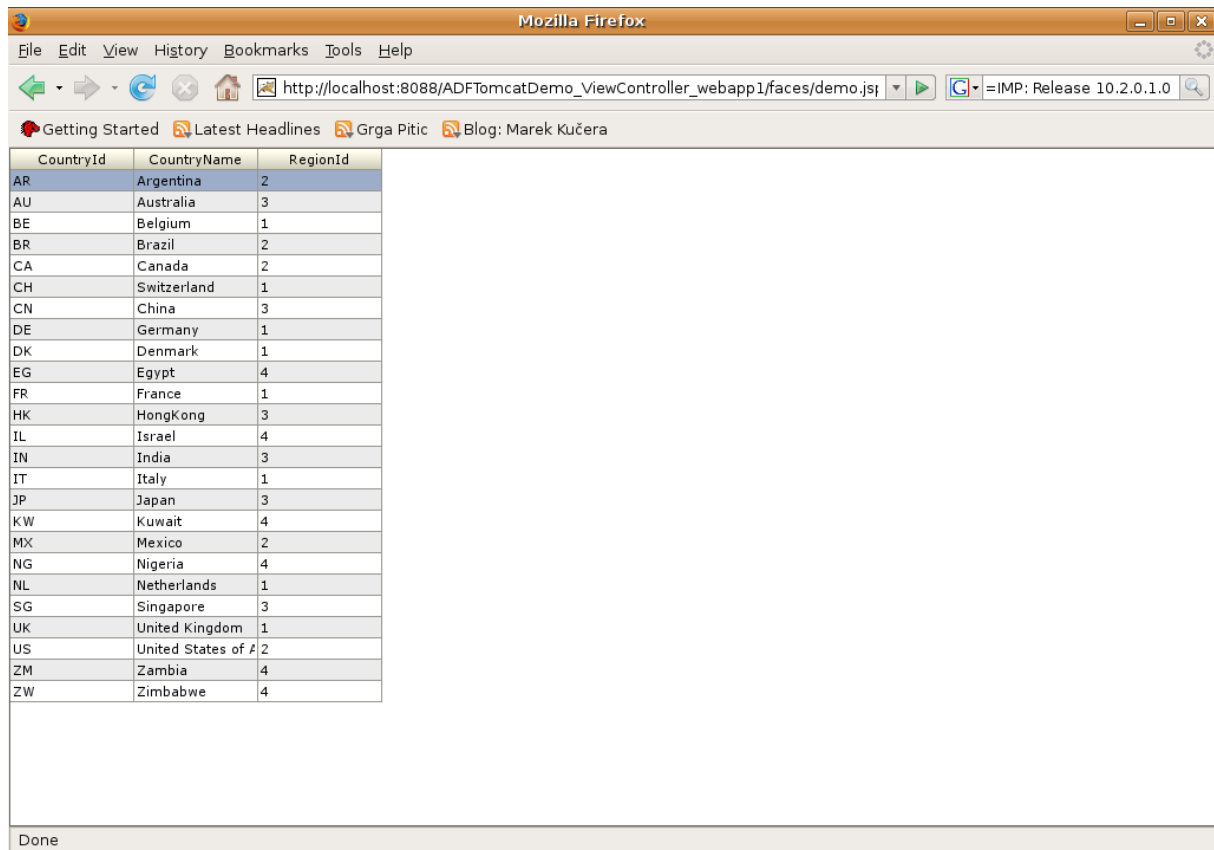
add resource link

```
<ResourceLink global="jdbc/hr" name="jdbc/hr" type="oracle.jdbc.pool.OracleDataSource"/>
```

Start tomcat

```
/bin/startup.sh
```

Application should be visible at <<warname>>/faces/demo.jspx



The screenshot shows a Mozilla Firefox browser window displaying a table of country data. The browser's address bar shows the URL `http://localhost:8088/ADFTomcatDemo_ViewController_webapp1/faces/demo.jspx`. The table has three columns: CountryId, CountryName, and RegionId. The data is as follows:

CountryId	CountryName	RegionId
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3
DE	Germany	1
DK	Denmark	1
EG	Egypt	4
FR	France	1
HK	HongKong	3
IL	Israel	4
IN	India	3
IT	Italy	1
JP	Japan	3
KW	Kuwait	4
MX	Mexico	2
NG	Nigeria	4
NL	Netherlands	1
SG	Singapore	3
UK	United Kingdom	1
US	United States of A	2
ZM	Zambia	4
ZW	Zimbabwe	4