

## Jbuilder+WebLogic+Oracle 实战 CMP

iceshape 2003 年 1 月 24 日

### 一、概念部分

- a) 本例子完全采用 J2EE 框架开发。使用 Web+EJB+DBMS 结构。其中 EJB 负责业务逻辑和数据库访问控制。EJB 部分实际上可以看作两层结构，Session Bean 和 Entity Bean。Session Bean 主要负责抽象表达业务逻辑，尽量避免在 Entity Bean 上使用关系来表达业务模型。Entity Bean 来实现数据库访问，执行单纯的数据读取和修改工作。下面列出 EJB 对应方法和数据库操作之间的关联：

数据库操作	EJB 方法
insert	ejbCreate()。使用该方法做数据插入时，必须将所有字段都列做参数，或者用一个包含所有列的 Java 类做参数（不包括自增长列）。该方法通过 Home 接口的 create 方法调用，返回 Remote 接口类型。
update	setFieldName(FieldType fieldValue)，使用该方法来修改一条纪录的某一字段值的时候，首先要使用 Home 接口的 find 方法将返回找到对应 EJB 的 Remote 接口，用该 Remote 接口的 setFieldName 方法修改字段的值。注意：必须把 setFieldName 也发布的 Remote 接口，使用时要十分小心。
delete	ejbRemove。该方法通过 Remote 方法的 remove 方法调用。该方法将删除对应的 EJB 本身的实例，容器会负责删除相应的数据纪录。所以删除一条记录时，要先用 Home 接口的 find 方法，找到该记录对应的 EJB 的 Remote 接口，在调用该接口的 remove 方法。
select	ejbFind，这是一个方法类型，包含很多 ejbFind 开头的方法。但至少有一个方法必须实现：ejbFindByPrimaryKey。这些方法都返回 Remote 接口或者它的集合。

注意：本例子使用 CMP 开发实体 Bean，所以 ejbFind 方法都不需要在 Bean 类里显示声明。但是必须在 ejb-jar.xml 文件中写出对应的 EJB\_QL 语句，让容器可以实现这些方法。

- b) 因为采用 CMP，所以会用很多地方涉及到 EJB QL 语言。这是专门为 CMP 而设记的数据库查询语言，建议大家看一下 j2ee tutorial 的 EJB QL 部分、《EJB2.0 企业级应用程序开发》的第 8 章 EJB 查询语言或者 EJB 2.0 规范第十一章 EJB QL:EJB Query Language for CMP。
- c) Session Bean 调用 Entity Bean 的方法，和客户端调用 Session Bean 一样，都必须根据 JNDI 名查找到 EJB 的 Home 接口，然后用 Home 接口生成符合要求的 Remote 接口。但是有一点需要注意：对于无状态的 Session Bean 引用 CMP 的 JNDI 查找过程不能放在 ejbActivate 方法中调用，因为无状态 Session Bean 没有 InActive 状态，同样它的 ejbPassitive 方法也没有实现的意义。虽然有状态 Session Bean 有 InActive

状态，但是建议大家把对于 EJB JNDI 查找实现在 `ejbCreate` 方法中。

d) 本例 CMP 对应 Table 的结构：

```
create table BM_RYXZ  
(  
  RYXZID NUMBER(15) not null,--Primary Key. Automatic Generate by Squence  
  RYXZ CHAR(8) not null,  
  BEIZHU VARCHAR2(30),  
  SCBZ CHAR(1) default '0' not null  
)
```

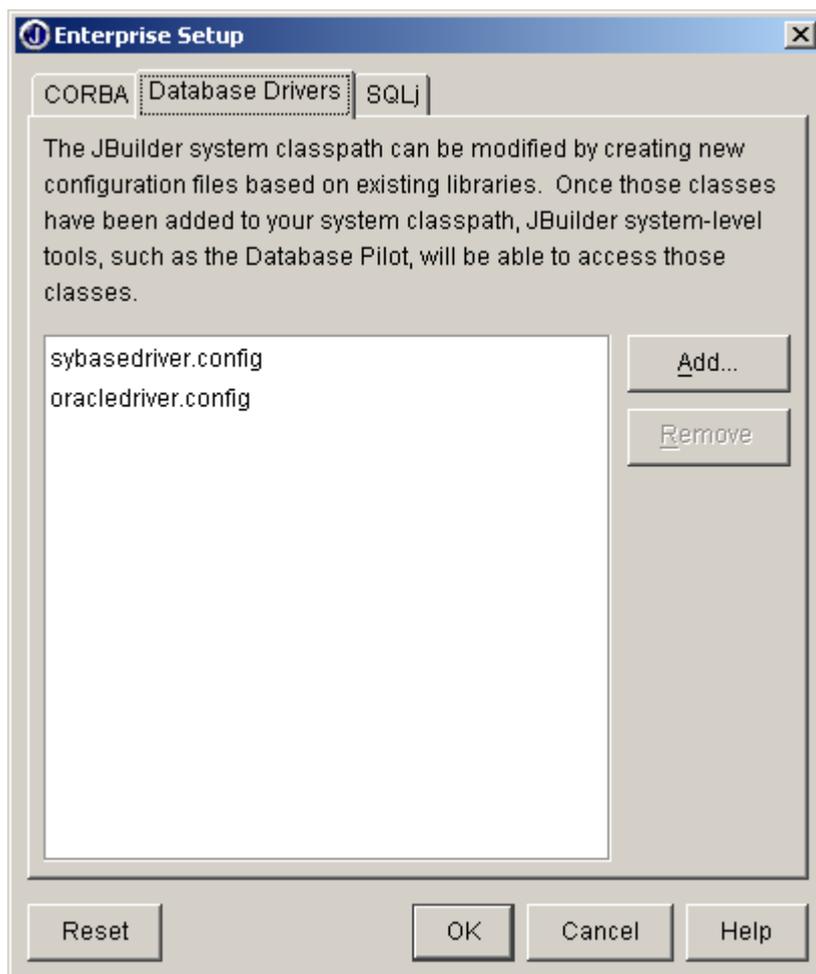
e)

## 二，工具使用

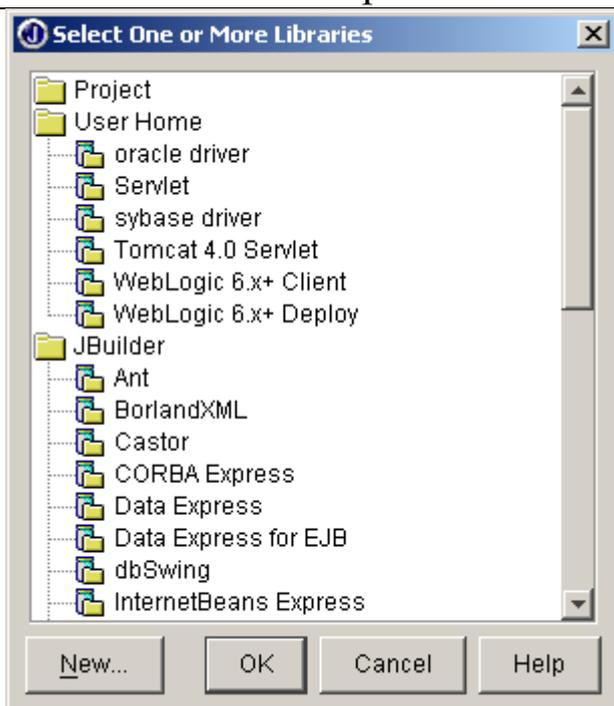
a) 本例数据库采用 Oracle 8.1.7。JDBC 驱动程序可以从网上下在。也可以使用 WebLogic 自带的驱动程序，放在 `bea\weblogic\server\ext\jdbc\oracle\817`目录下。

b) Jbuilder 配置 Oracle jdbc driver

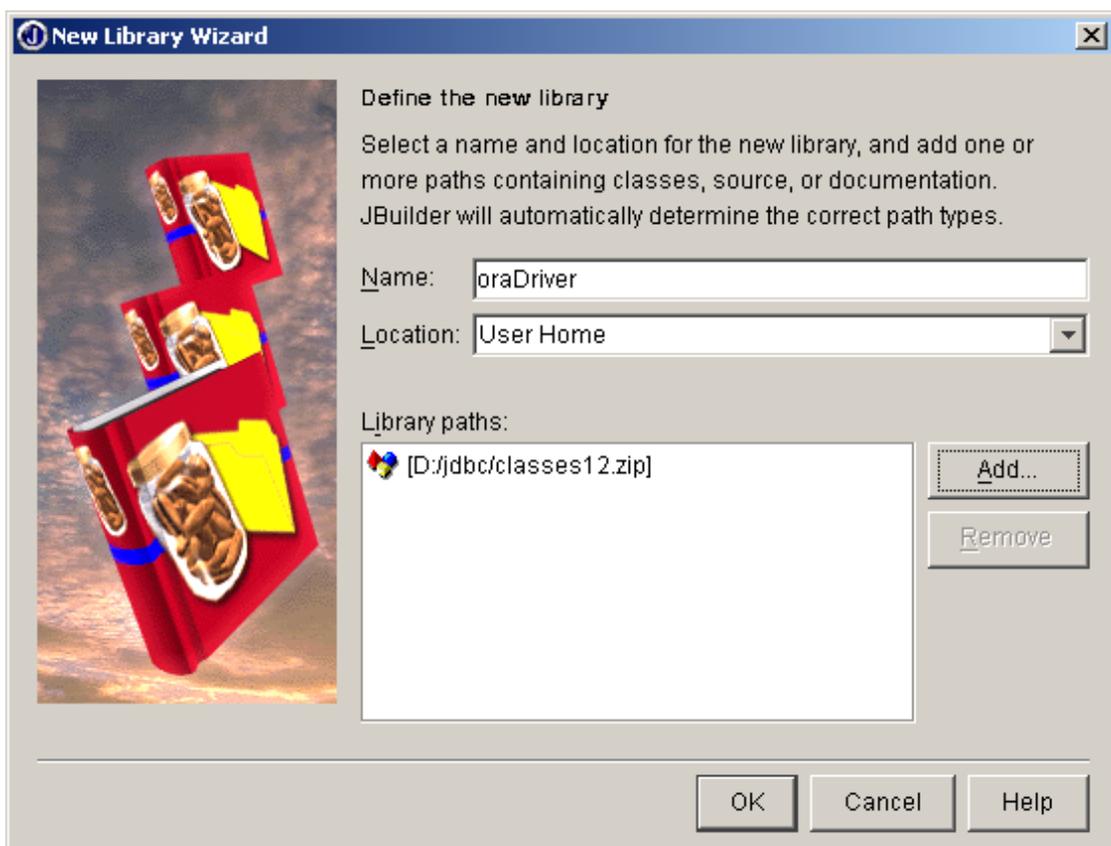
- i. 运行 Jbuilder
- ii. 选择 `tools\Enterprise Setup` 菜单，在弹出对话框中选择 `Database Drivers` 选项页，对话框如下



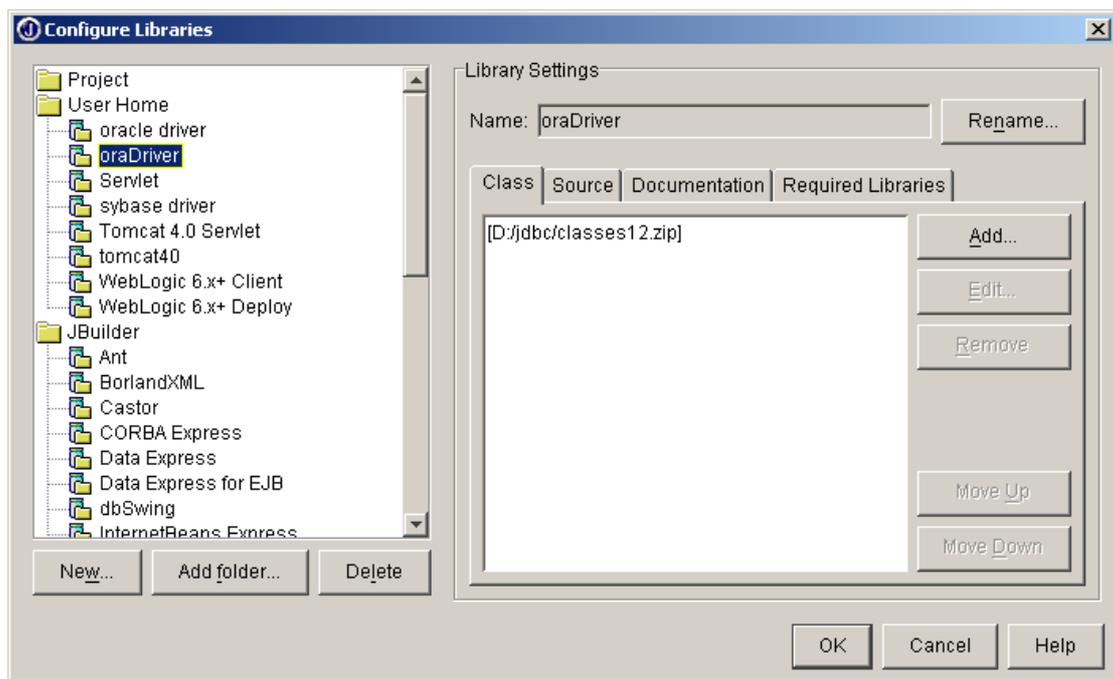
iii. 点击 `Add`，添加 Oracle JDBC Driver。对话框如下



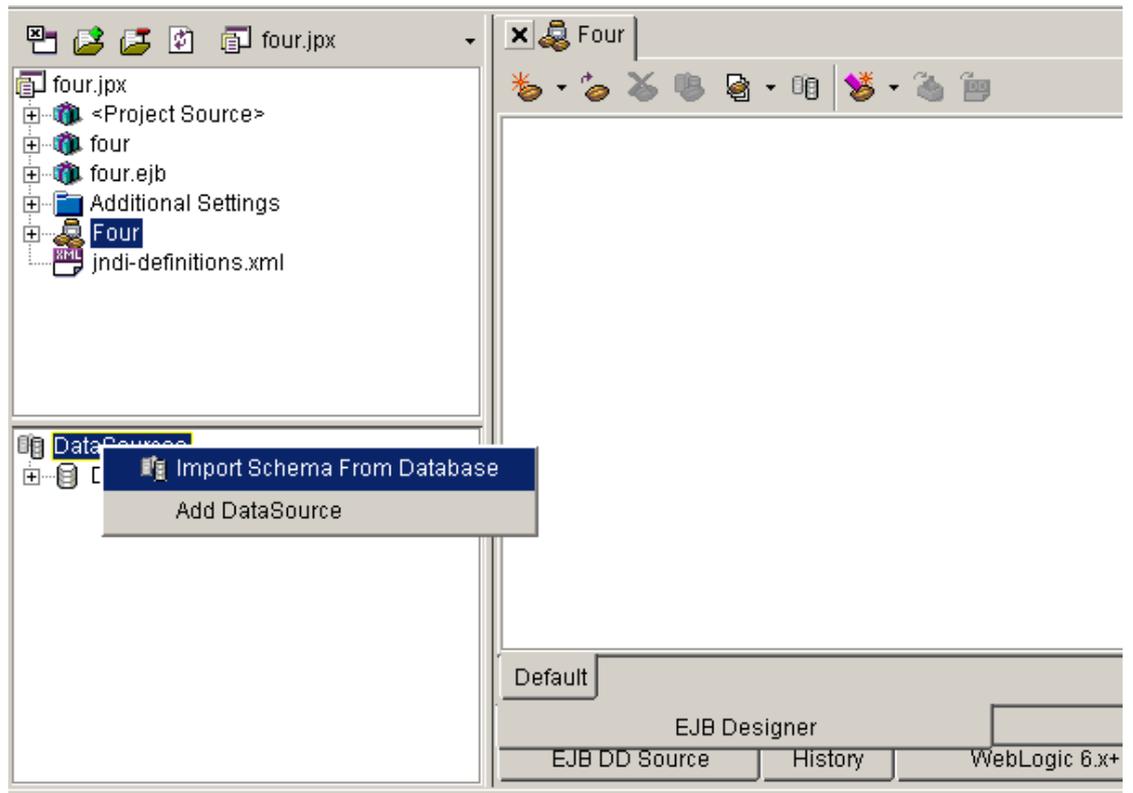
iv. 点击 New 按钮，出现对话框：



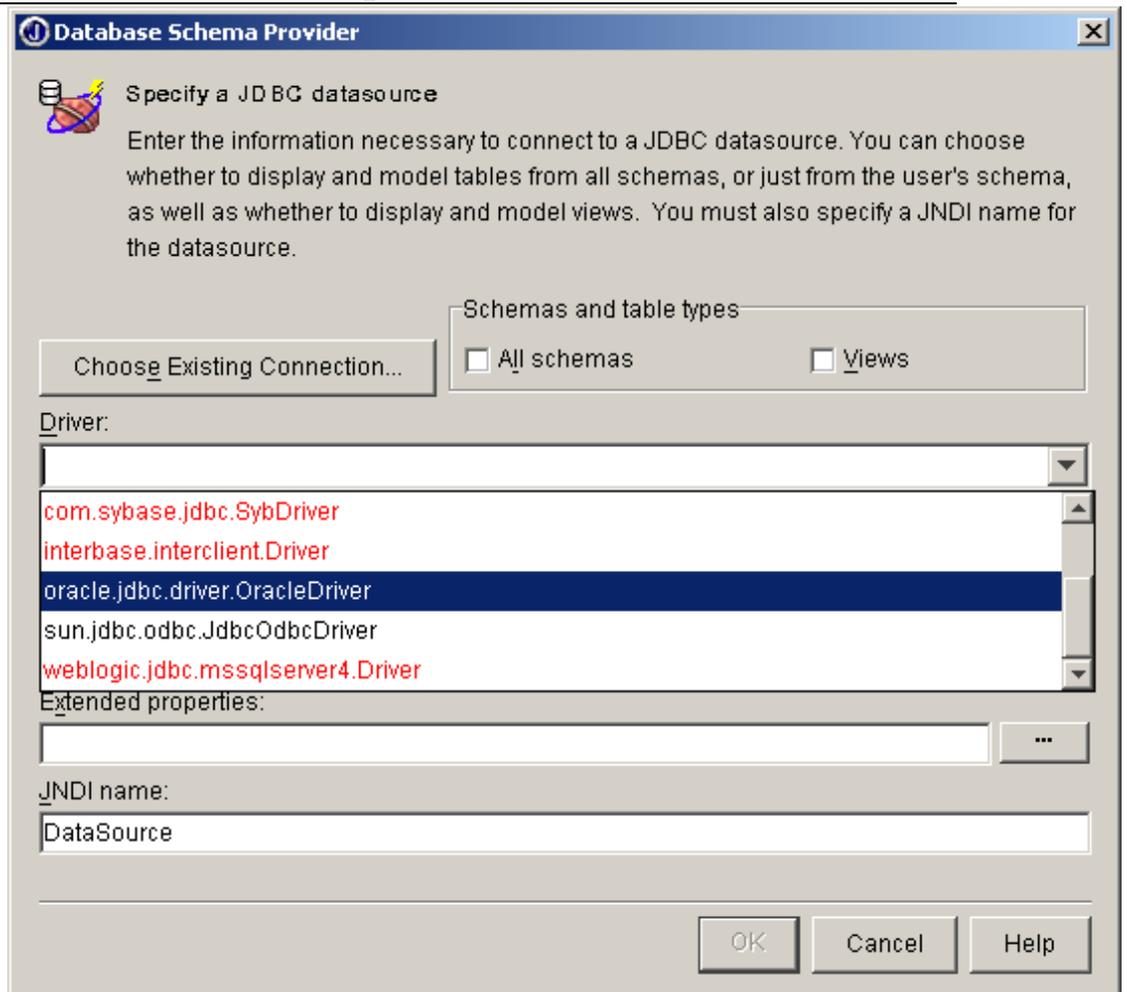
- v. 在 Name 输入框中输入 oraDriver。点击 Add，出现文件选择对话框。找到上述目录下的 classes12.zip 文件，并选中加入
- vi. 一路 OK 就完成了。
- vii. 如果你想修改你选择的文件，可以选择 Tools\Configure Libraries，在如下窗口



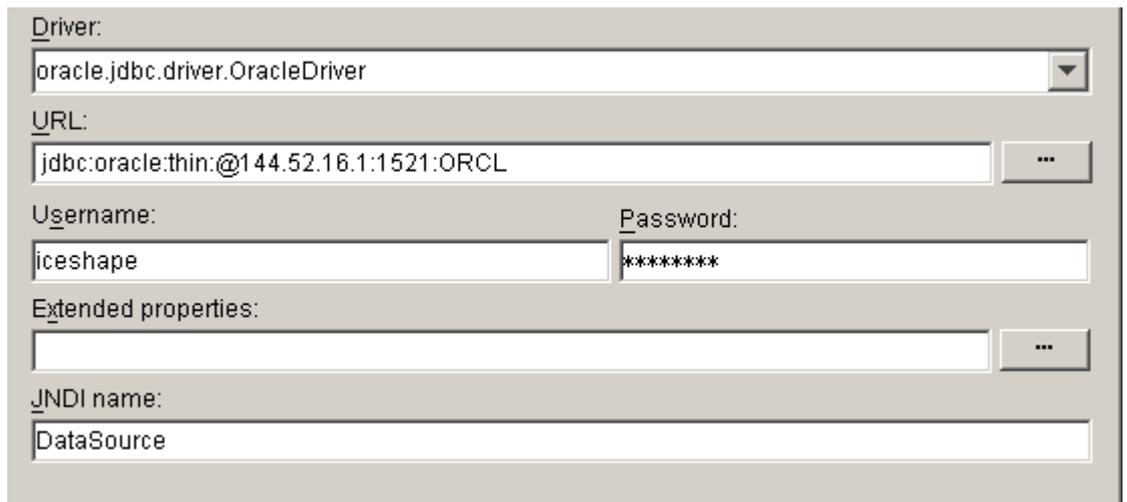
- viii. 在以后用到 JDBC Driver 的地方 Jbuilder 显示的类 `oracle.jdbc.driver.OracleDriver` 将变成黑色，表示可用。
- c) 结合 Sybase 数据库创建一个 CMP。首先声明一下，由于本人不知道的原因，在 Jbuilder 里面建立 CMP 需要先建立数据库连接。
  - i. 建立工程
  - ii. 建立 EJB Module，它是 EJB 的 jar 容器。相当于一个 EJB 应用。
  - iii. 建立好 EJB Module 后，双击树结构中新建的 EJB Module 接点，会看到如下界面：



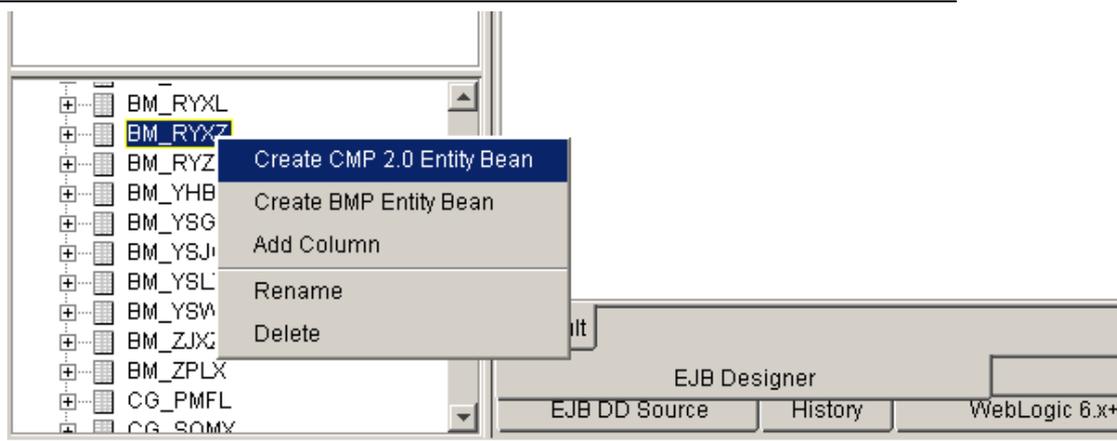
- iv. 选中 DataSources 点右键选择 Import Schema From Database，出现如下窗口



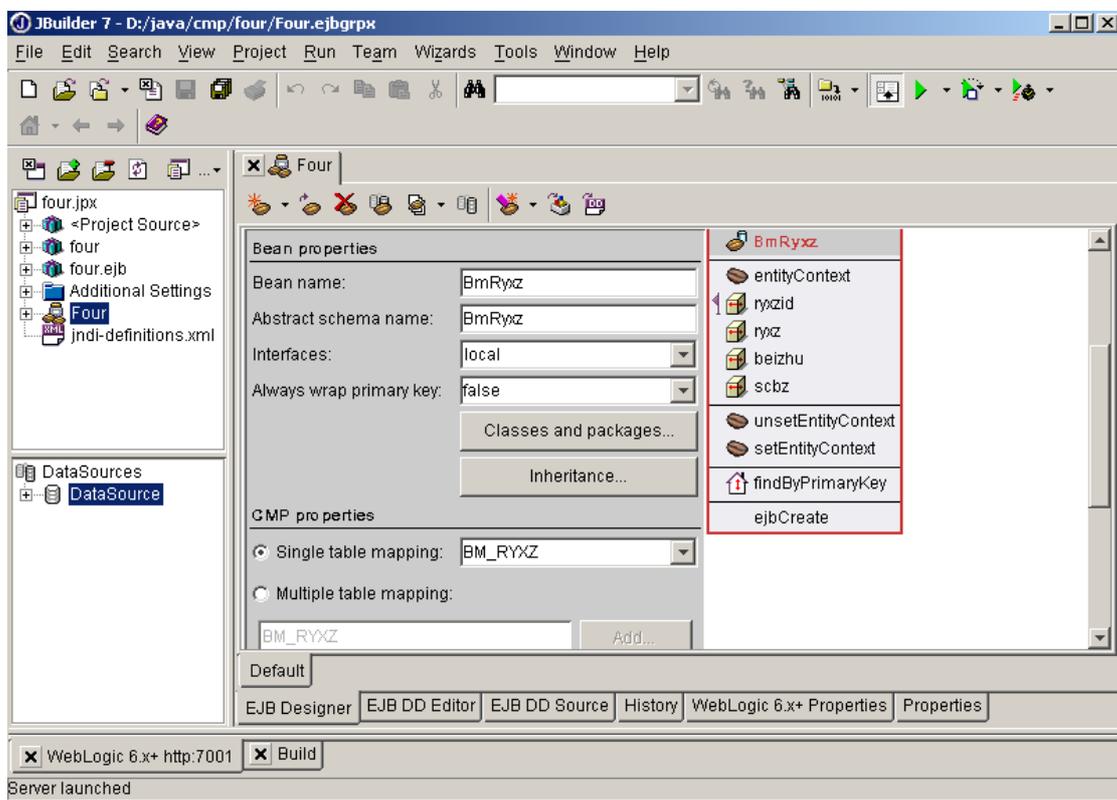
- v. Driver 下拉框中选择 oracle.jdbc.driver.OracleDriver。或者直接输入
- vi. 其余选项根据实际情况填写，参照如下：



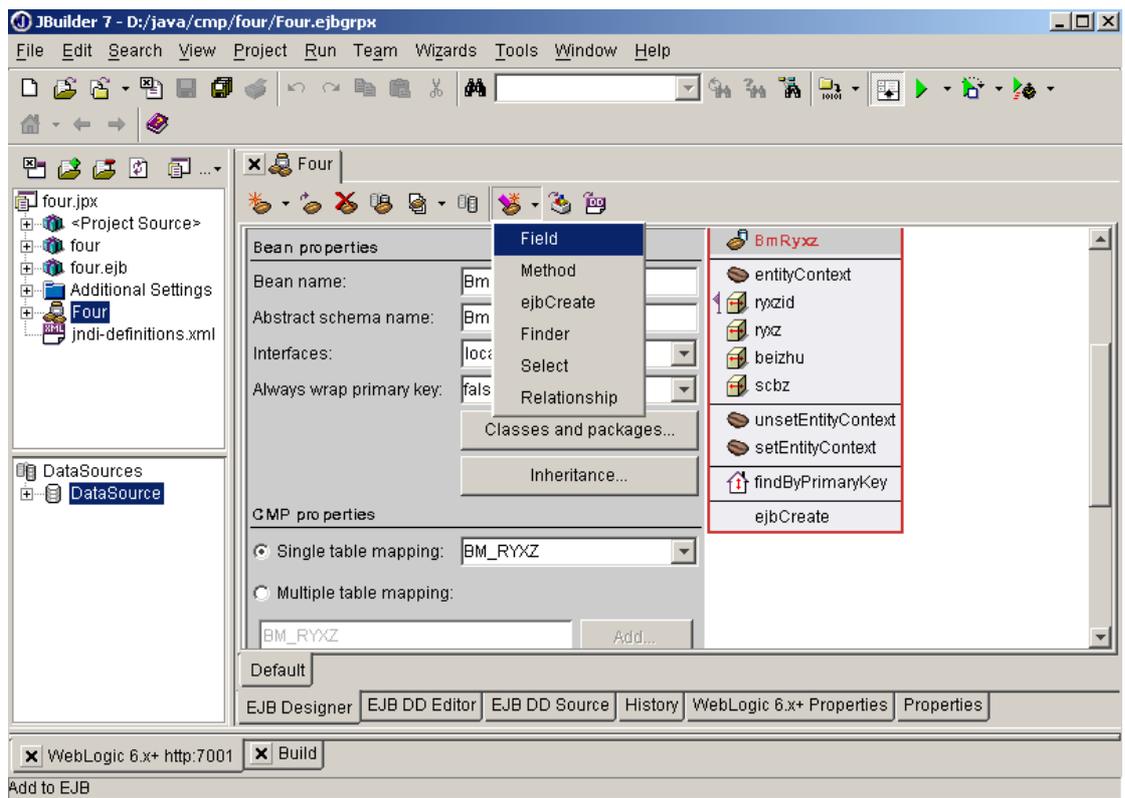
- vii. 注意 URL 的书写，1521 是端口号，ORCL 是数据库名，thin 后面一定要有冒号，笔者曾经两次因为忽略这个冒号而浪费很多时间。



- viii. 选中其中的一个表，右键选择 Create CMP 2.0 Entity Bean，创建一个 CMP。  
在右边的工作区中可以看到 CMP 的信息

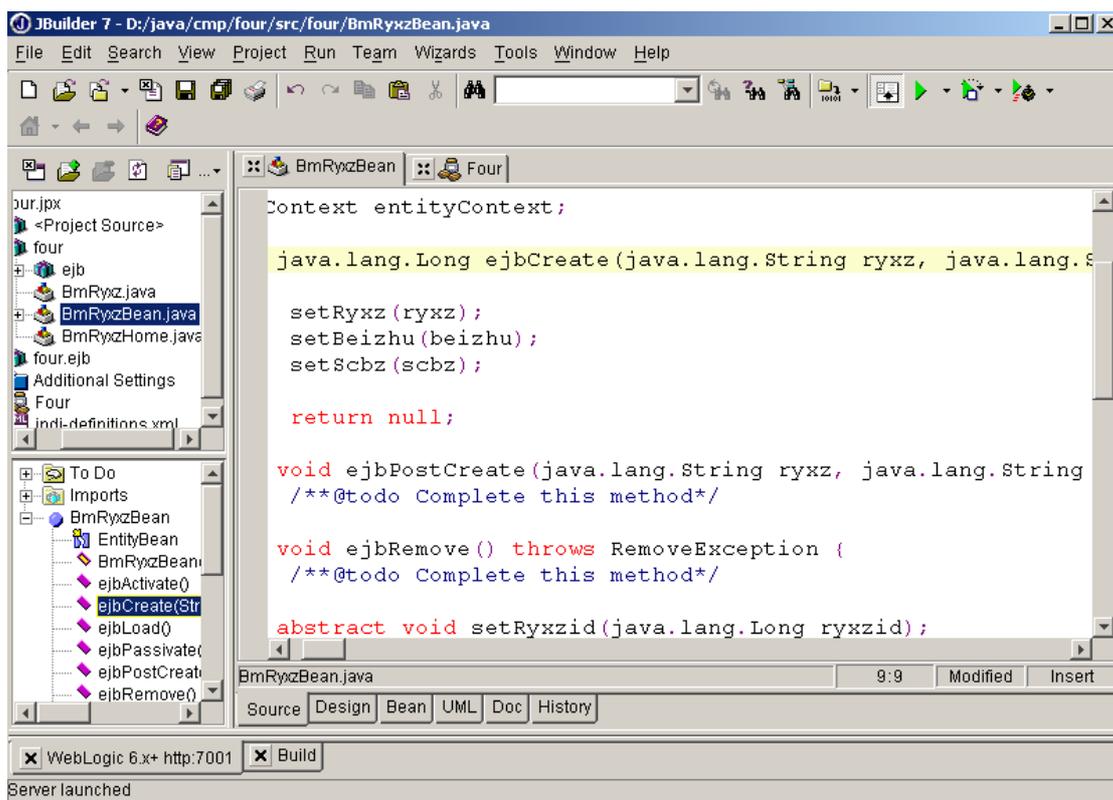


- ix. 可以用如下图的菜单添加字段，方法和关系。注意这里的字段添加不会写入数据库！

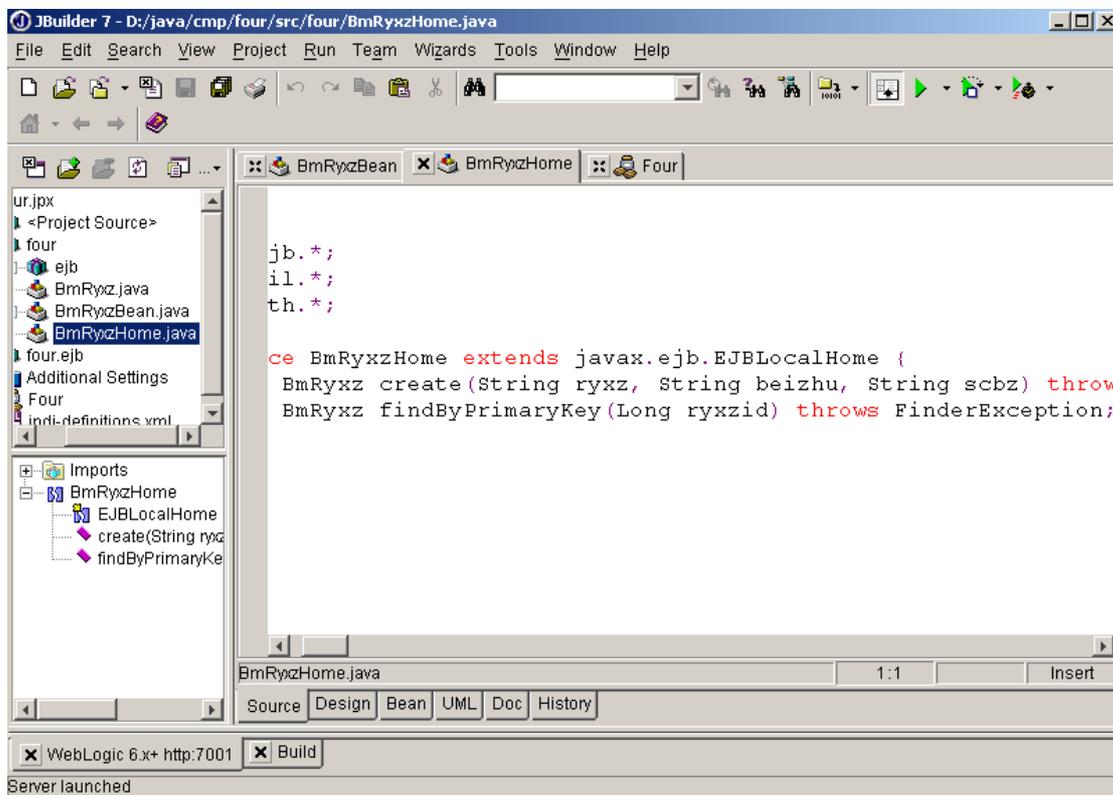


- x. 修改 `ejbCreate` 方法，以便可以添加一条纪录。同时要将对应的 `ejbPostCreate` 方法的参数列表做对应的修改。（任何时候 `ejbCreate` 方法都要对应一个参数列表相同的 `ejbPostCreate` 方法）。在 `ejbCreate` 方法中将数据字段都加入参数列表，却省 Jbuilder 只将 `PrimaryKey` 加入参数列表。注意，如果 `PrimaryKey` 是自增型字段，则不能加入 `ejbCreate` 的参数列表！怎么处理，下面会提到。本例中需要将 Jbuilder 在 `ejbCreate` 方法中加入的 `ryzxid` 参数删掉。也可以另外新建一个 `ejbCreate` 方法将所有字段作为参数，但实际上只有 `PrimaryKey` 做参数的 `ejbCreate` 方法是没有用的。如果你新建 `ejbCreate` 方法的话，注意重载该方法不要引起二义性，同时要实现相应的 `ejbPostCreate` 方法。参数列表修改完之后，在 `ejbCreate` 方法的 `return null;` 语句之前加入如下语句（参数列表中添加的参数的 `set` 方法）：

```
setRyxz(ryxz);
setBeizhu(beizhu);
setScbz(scbz);
```



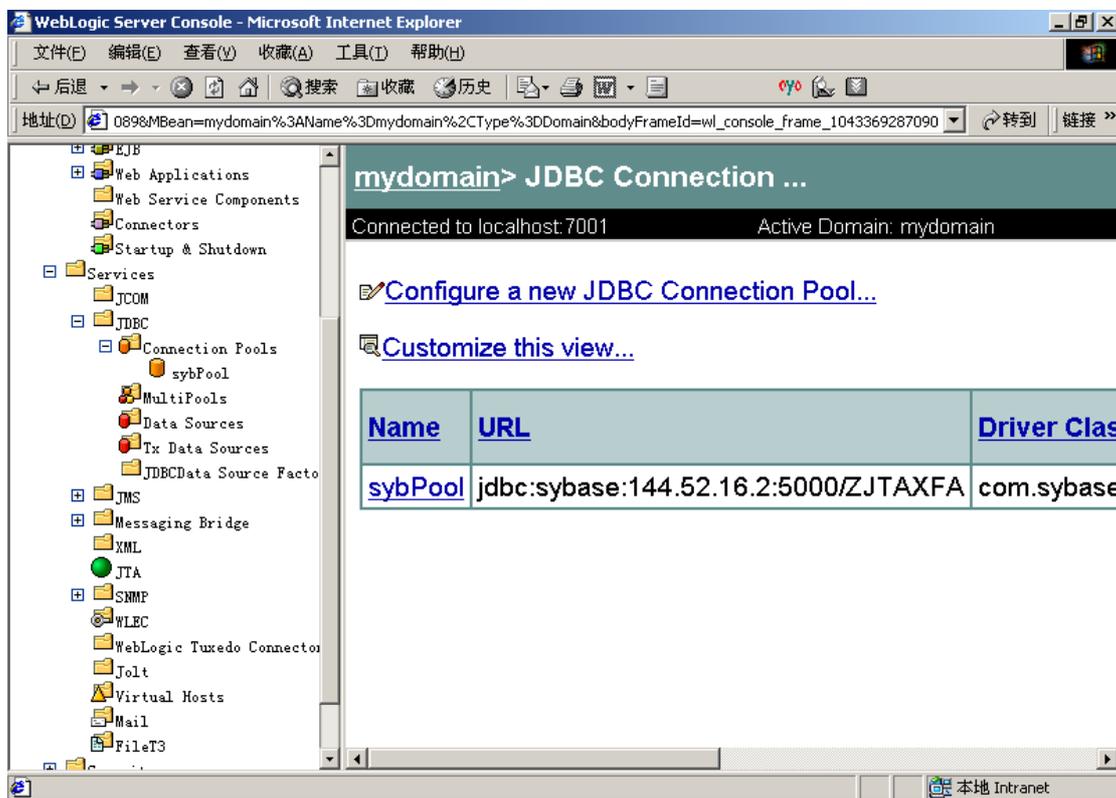
- xi. 修改 ejbCreate 方法对应得该 EJB 的 Home 接口中的 create 方法（这里同样通过参数列表对应方法）的参数列表。



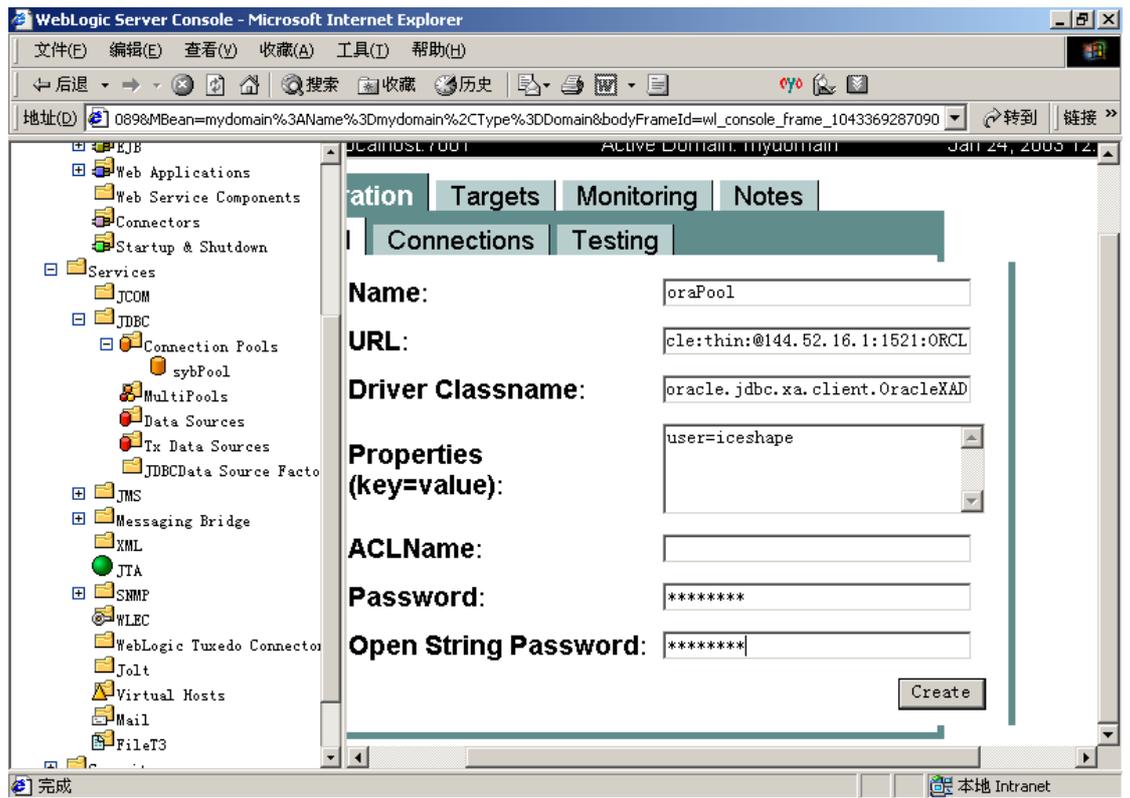
- xii. 需要注意的是，在字段类型选择的时候不要用原始数据类型，如 int,double 等，

而是用相应的封装类 Integer,Double 来代替,因为这些字段都要在网络上进行远程传输,而传输数据都必须实现 java.io.Serializable 接口,否则无法正常在网络间传输。

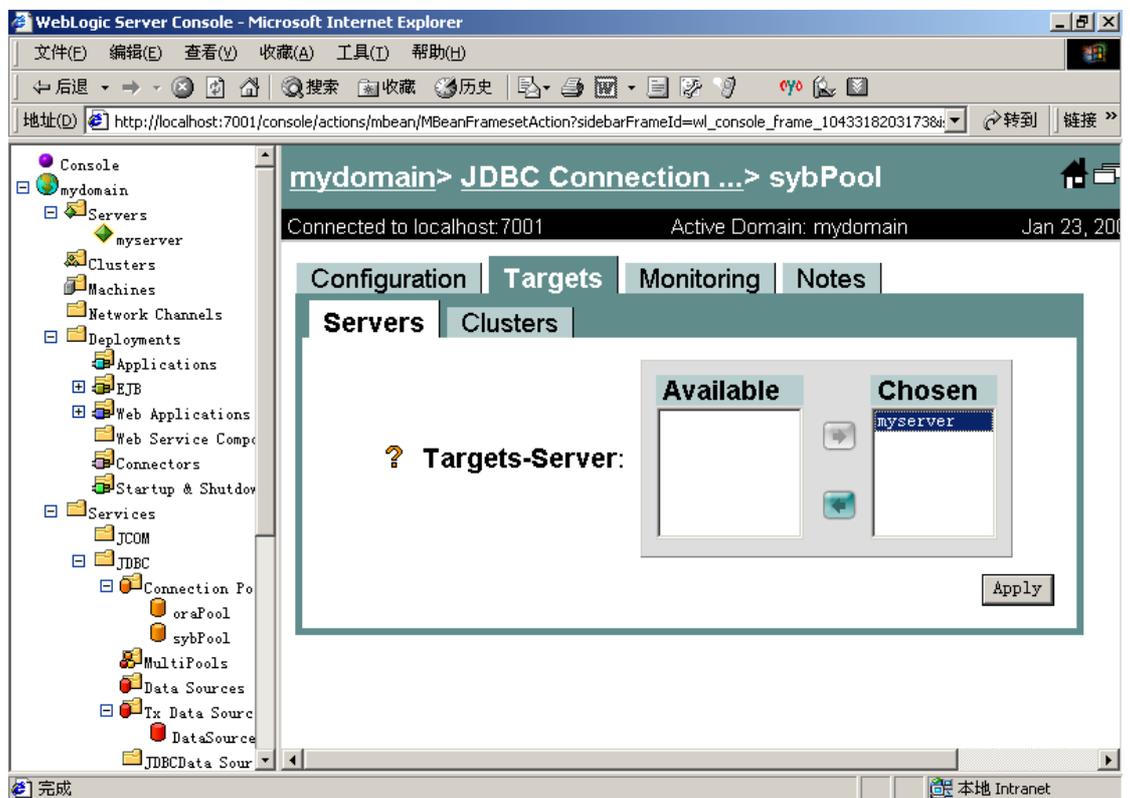
- xiii. 现在已经实现了一个简单的具有数据添加功能的 CMP 了。下面配置相应的 WebLogic 服务器的数据库信息。



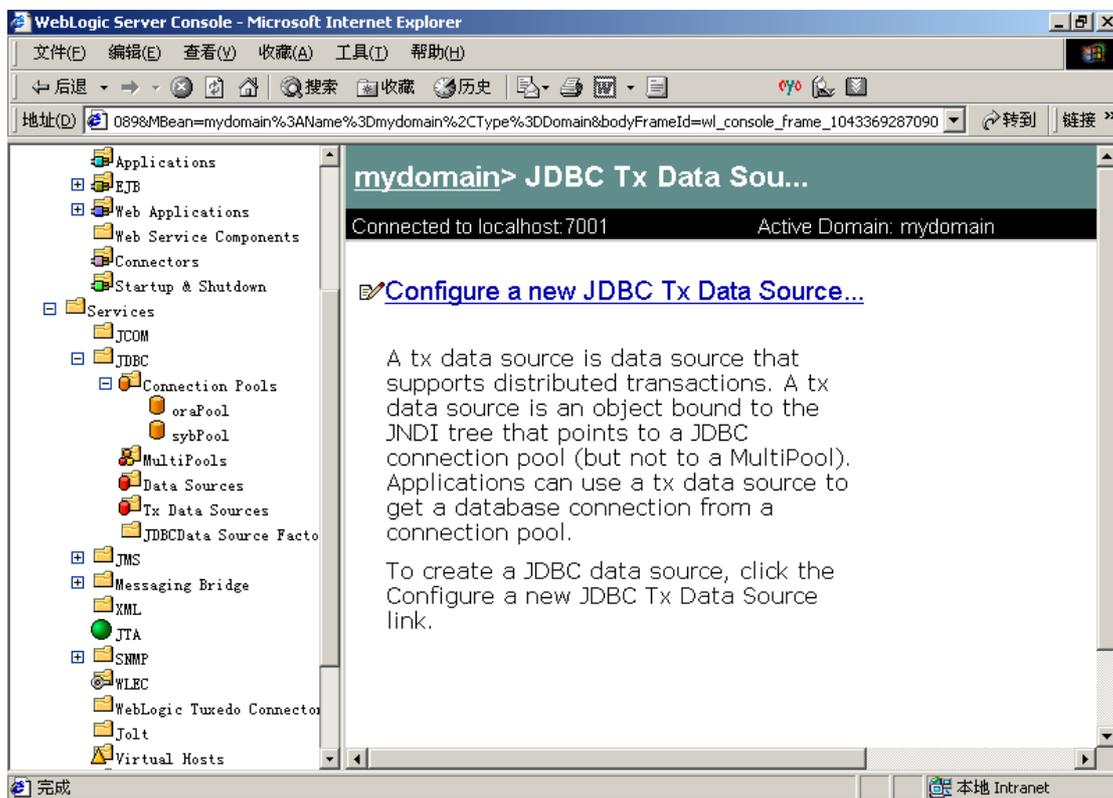
- xiv. 进入 WebLogic 管理控制台,在树型结构中找到 JDBC\Connection Pool 接点。右边的工作区如上图。选择 Configure a new JDBC Connection Pool..., 新建一个新的数据库连接池。注意,这里 Oracle 的驱动程序类跟 Jbuilder 中使用的不一样,因为 J2EE 服务器要控制 TX 分布式事务协议的资源管理器,所以需要对应的支持分布式事务的数据库驱动程序类: oracle.jdbc.xa.client.OracleXADataSource。



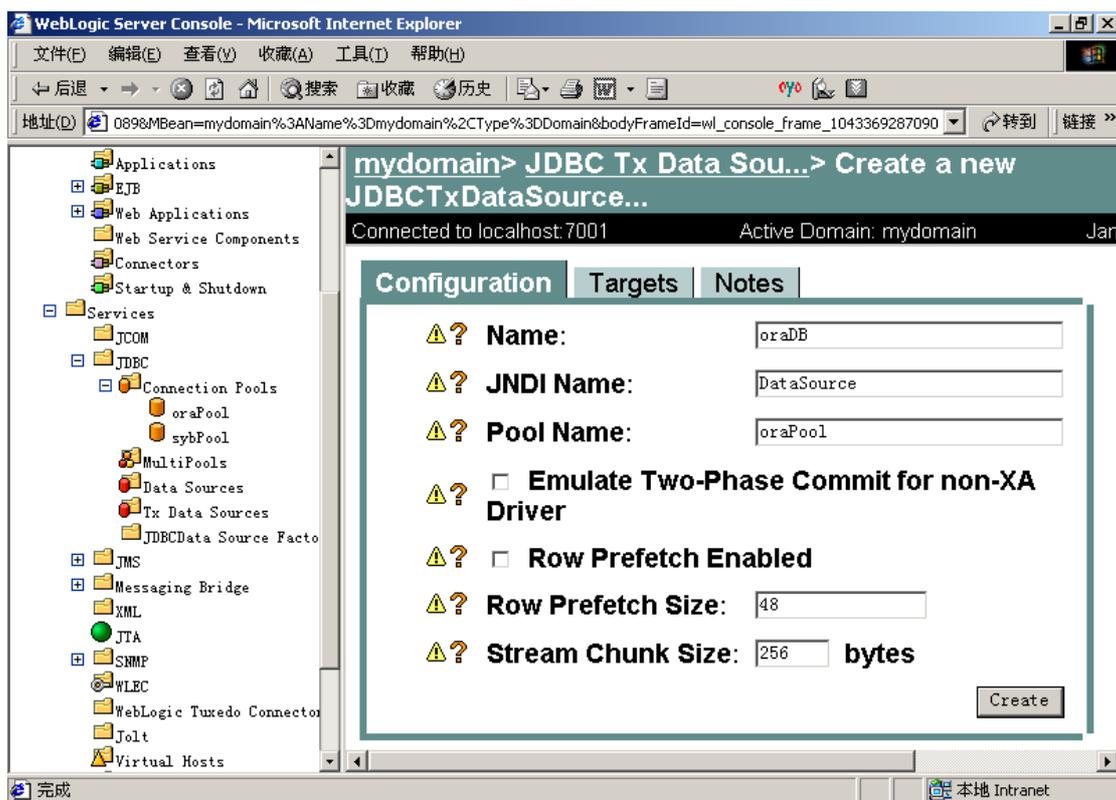
- xv. 输入如上图的信息，URL、Driver Classname 和 Jbuilder 中间里 DataSource 是的一样。Properties 中输入 user,password 等。也可以在下面的 password 输入框中输入。按 Create 确定。然后转到 Targets 选项页。



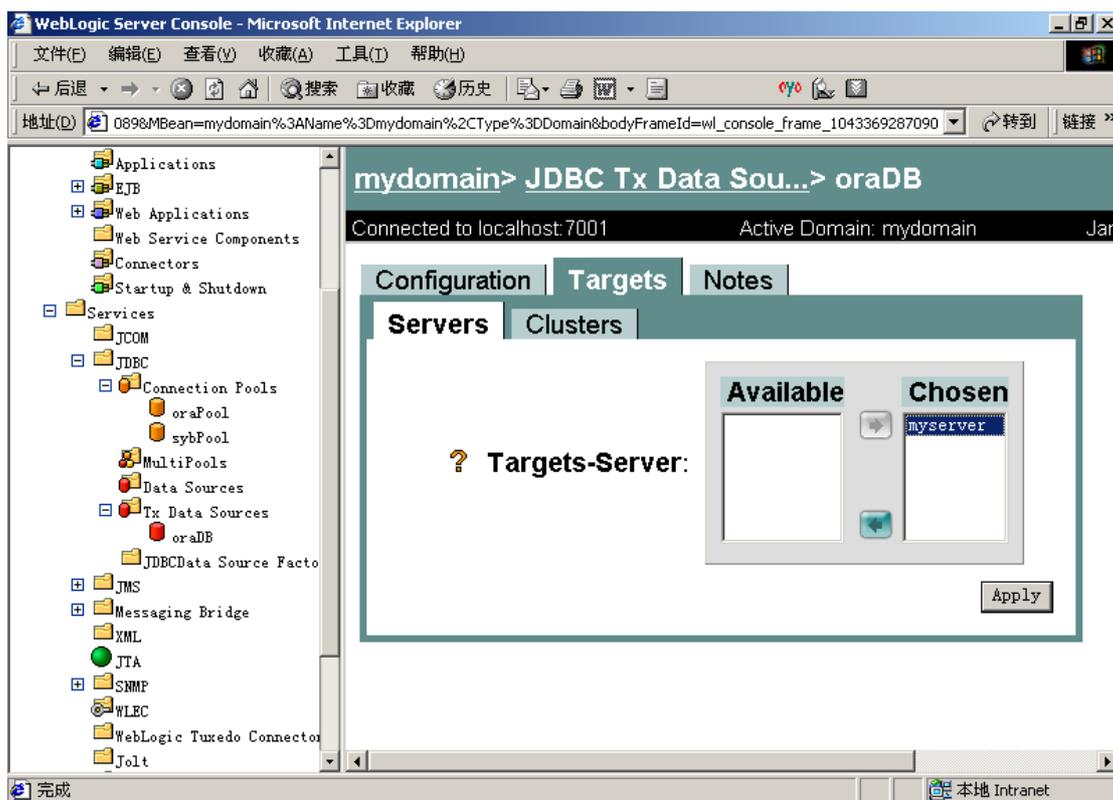
- xvi. 在上图的 Available 框中选中使用该 Pool 的 Server，点击右箭头按钮，假如到 chosen 框中。点击 Apply 按钮确定。
- xvii. 下面配置 Tx Data Source。因为 EJB 一般情况下需要控制多资源管理器的分布式事务，所以需要支持 Tx 协议数据源支持，所以我们不需要配置 Data Source。找到树型结构的 JDBC\Tx Data Source 接点。



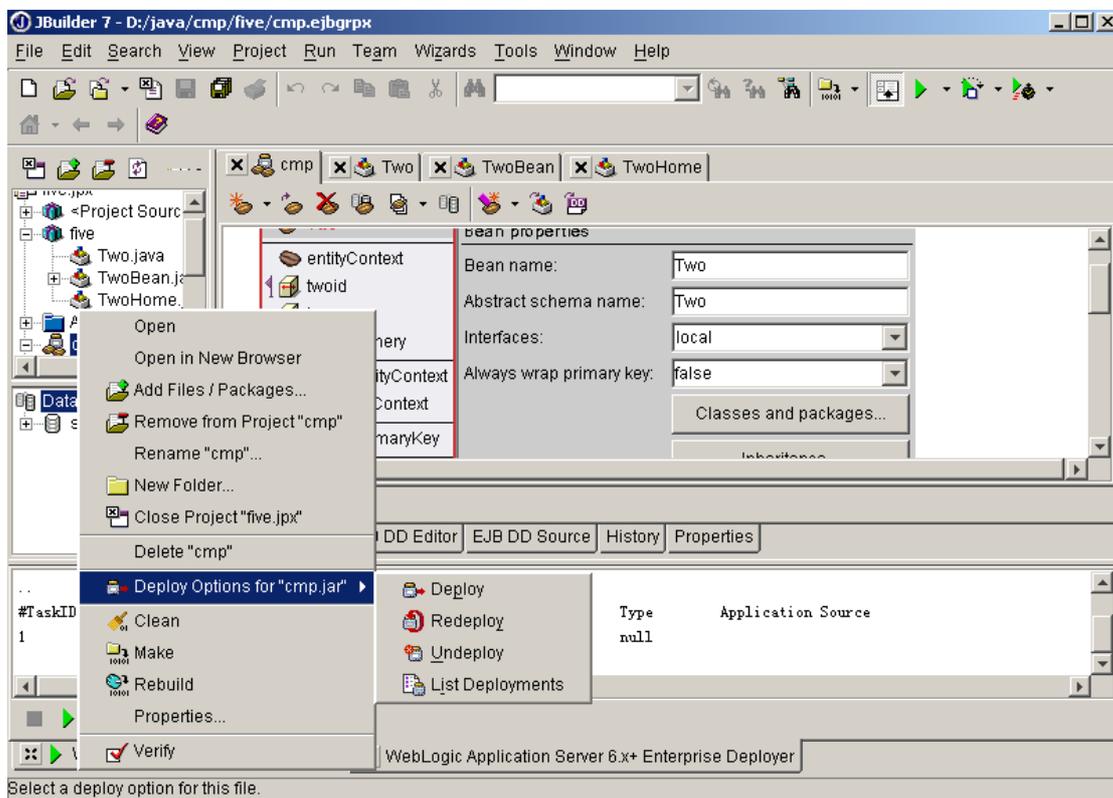
- xviii. 选择 Configure a new JDBC Tx Data Source，新建一个 DataSource。



- xix. 按照上图输入 DataSource 信息。其中 JNDI Name 要跟在 Jbuilder 中的 DataSource 的名字对应。在 Jbuilder 的 DataSources 下。
- xx. 和 Pool 一样在 Targets 选项页中选择一个 Server 然后 Apply。



- xxi. 重新启动服务器。确定一切都已经配置好了，就可以开始发布你的 EJB 了。
- xxii. 部署 CMP。如下图，如果是第一次部署选择 Deploy，如果不是选择 Redeploy 菜单。

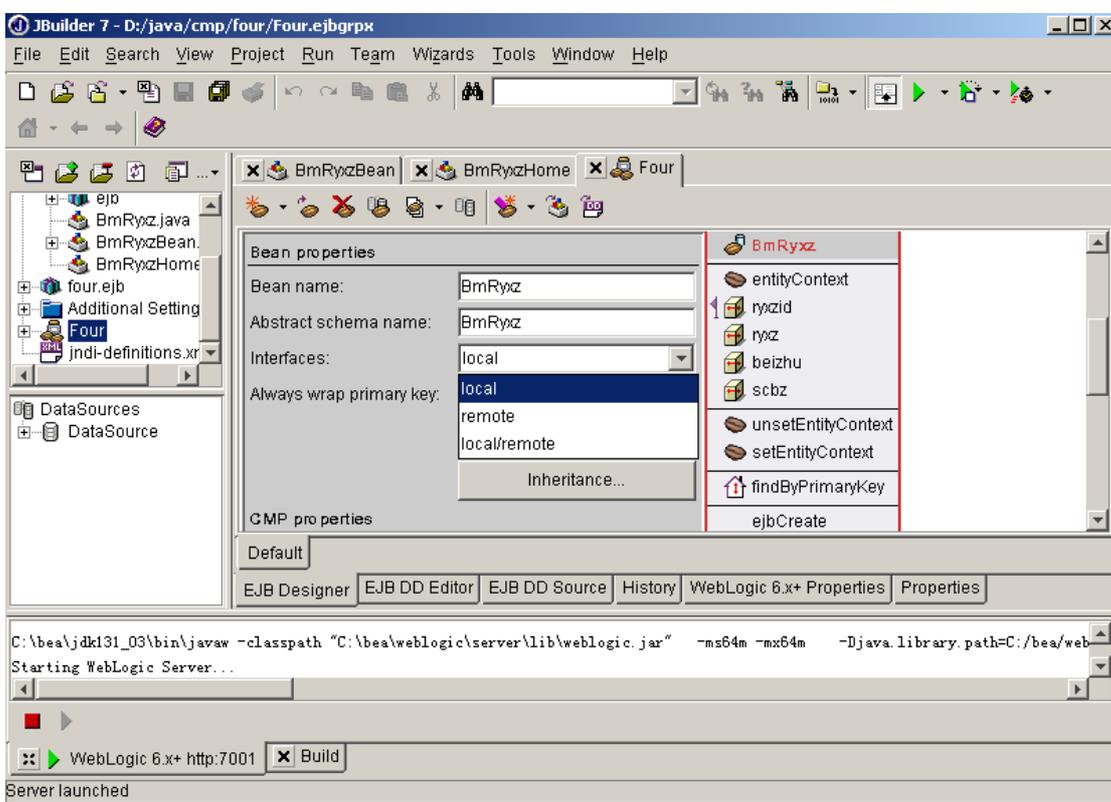


xxiii. 如果部署成功。会在 Jbuilder 的 Debug 窗口中显示如下的成功信息：

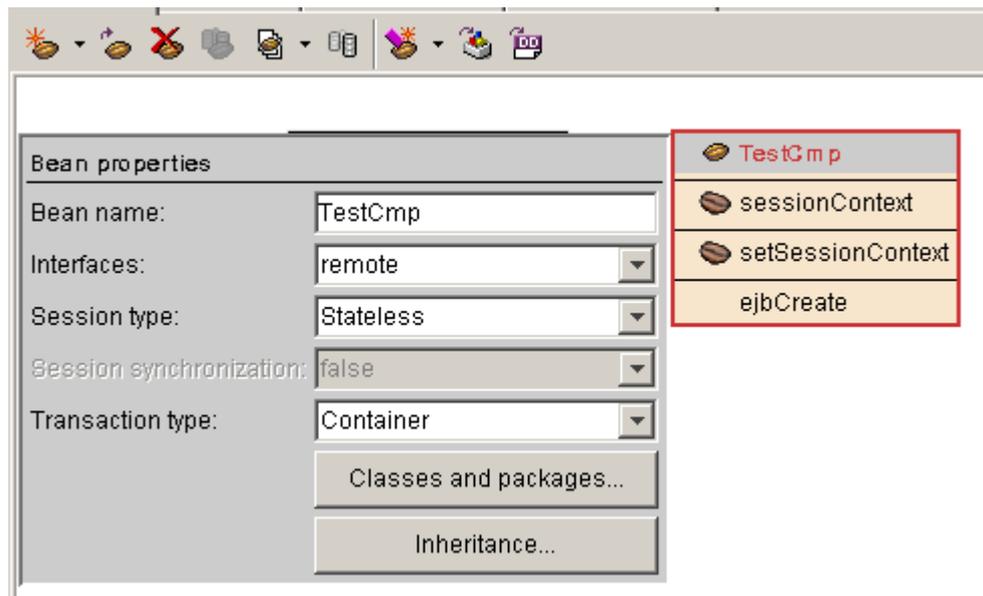
```

Operation started, waiting for notifications...
....
#TaskID      Action      Status      Target      Type      Application Source
2            Activate   Success     myserver    Server    Four        C:\bea\user_projects\mydomain\myserver\upload
    
```

xxiv. 好了。下面该测试一下你的 EJB 是否能工作了。由于 CMP 大部分是被本地(同一个 JVM 中)的 SessionBean 调用的,所以 Jbuilder 的 CMP 缺省实现它的 Local 接口。没有 Remote 接口,在 Jbuilder 中无法用 EJB Test Client 来测试。如果你一定想要用 Jbuilder 中建立的 EJB Test Client 来测试,可以在生成 CMP 后将其接口类型改为 Remote。如下图:



xxv. 这里我们用更符合本例子情况的方法来测试。就是另外在建立一个 SessionBean 来测试这个 CMP。注意,这里同样由于 CMP 是实现的 Local 型的接口,请大家尽量把 SessionBean 和它需要引用的 CMP 建在同一个包中,至少要建立在同一个 EJB Module 中(一个 EJB Module 将开包成一个.jar 文件),一面在部署的时候需要考虑到包的依赖性而不能多服务器分布大量的 EJB 包。新建一个 SessionBean 命名为 TestCmp:

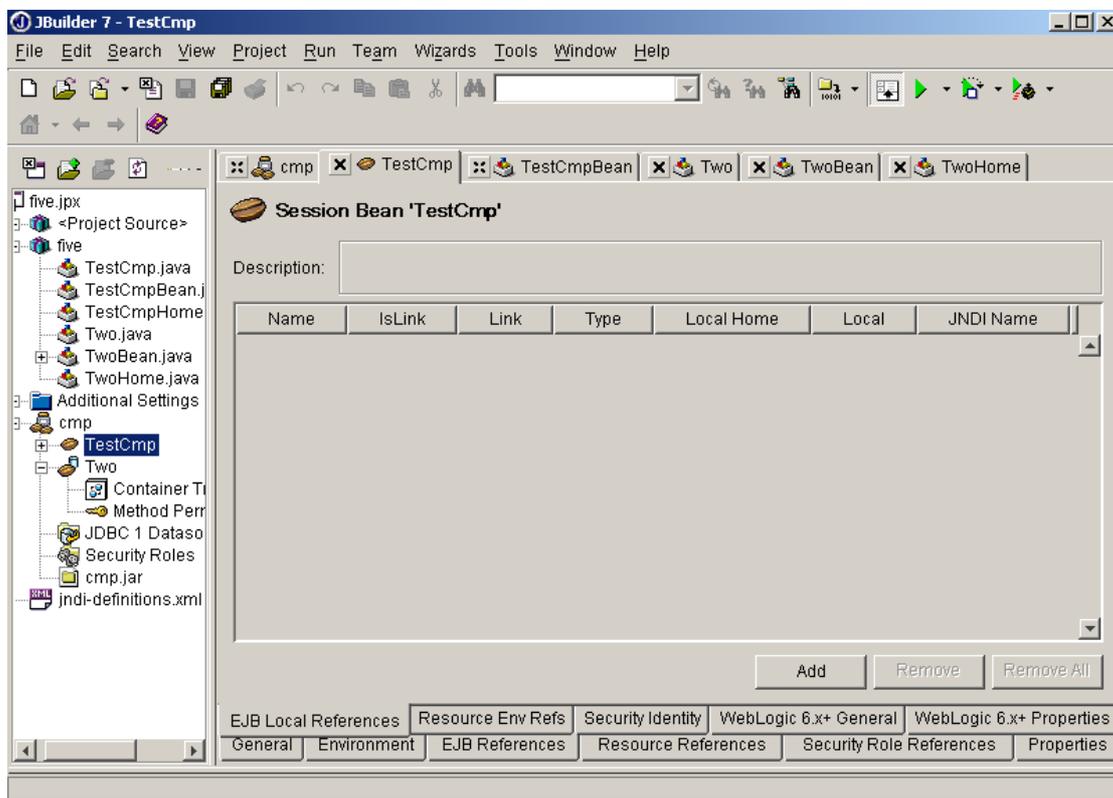


- xxvi. 在 TestCmpBean 类中手动引入下列包（如果你的 CMP 和 SessionBean 不再同一个包中,则需要引入 CMP 所在的包,因为需要使用 CMP 的 Home 和 Remote 接口）
- ```
import javax.naming.*;
import javax.rmi.PortableRemoteObject;
```
- xxvii. 在 SessionContext sessionContext; 语句后加入如下声明：  
BmRyxxHome brh;//上面例子中 CMP 的 Home 接口  
BmRyxx br;//上面例子中 CMP 的 Remote 接口
- xxviii. 在 ejbCreate 方法中加入如下语句（实现 CMP JNDI 查找）：
- ```
try{
    Context ctx = new InitialContext();
    Object ref = ctx.lookup("BmRyxx");// BmRyxx 是上面例子中 CMP 的 JNDI 名
    brh = (TwoHome)PortableRemoteObject.narrow(ref, BmRyxxHome.class);
}catch(Exception ex){
    throws new CreateException("CMP find error in session bean create!");
}
```
- xxix. 添加一个方法插入数据。
- ```
public void testCmpInsertData(){
    try{
        br = brh.create("this is a test record", "zxr 2003-1-23", "0");
        System.out.println("Test Success!");//在控制台输出成功信息
    }catch(Exception ex){
        ex.printStackTrace();//打印出错点的堆栈信息
    }
}
```
- xxx. 在该 SessionBean 的 Remote 接口中发布上面的方法。在 TestCmp.java 中加入

如下方法声明:

```
public void testCmpInsertData() throws RemoteException;
```

- xxxii. 下面需要做一小点部署工作, 可以让 SessionBean 引用 CMP。在 EJB Module (你建立的 EJB Module 的名字) 接点下双击建立的 SessionBean 接点, 这里是 TestCmp, 如下图:



- xxxiii. 找到 EJB Local References 选项页。点 Add 按钮添加一个 EJB 引用。参照下图, Name 项的输入可以随便, link 项是要引用的 EJB 的名字, 如本例中的 Two。

| Name   | IsLink                              | Link                         | Type                            | Local Home   | Local    | JNDI Name |
|--------|-------------------------------------|------------------------------|---------------------------------|--------------|----------|-----------|
| cmpref | <input checked="" type="checkbox"/> | Two <input type="checkbox"/> | Entity <input type="checkbox"/> | five.TwoHome | five.Two |           |

- xxxiiii. 现在 Rebuild 一下整个 EJB Module。上面我们提到自增字段的问题, 本你中的 ryxid 就需要解决这个问题, 在 WebLogic 中有一个部署描述文件描述数据库字段信息 weblogic-cmp-rdbms-jar.xml。在该文件中加入如下一段描述信息, 然后 make 这个 EJB Module 就会把部署描述文件修改信息打包发布。

```

</table-map>
<automatic-key-generation>
  <generator-type>ORACLE</generator-type>
  <generator-name>test_sequence</generator-name>
  <key-cache-size>10</key-cache-size>
</automatic-key-generation>
eblogic-rdbms-bean>
gic-rdbms-jar>

```

weblogic-cmp-rdbms-jar.xml 30:8

weblogic-ejb-jar.xml    ejb-modeler-schema.xml    weblogic-cmp-rdbms-jar.xml

ejb-jar.xml    ejb-borland.xml    ejb-modeler-layout.xml

EJB Designer   EJB DD Editor   EJB DD Source   History   WebLogic 6.x+ Properties   Properties

xxxiv. 然后再 Redeploy 一下。成功信息如下

```

Operation started, waiting for notifications...
.....
#TaskID    Action    Status    Target    Type    Application Source
0          Activate    Success    myserver    Server    Four    C:\bea\user_projects\mydomain\myserver\upload\

```

xxxv. 下面用 Jbuilder 生成一个 SessionBean 的 TestClient 来测试我们的整个 EJB Module。生成一个 TestCmpTestClient 类，在该 java 文件中加入如下声明和方法：

```

private TestCmp testcmp = null;
.....
//test insert
public void testInsertData(){
    try{
        testcmp = testCmpHome.create();
        testcmp.testCmpInsertData();//调用 SessionBean
    }catch(Exception ex){
        ex.printStackTrace();
    }
}

```

xxxvi. 在该 java 文件的主方法 TestCmpTestClient client = new TestCmpTestClient(); 语句后加入如下语句：

```
client.testInsertData();
```

xxxvii. 好了，运行这个类，如果在 WebLogic 的控制台输出 Create Success 你的 CMP 就完全成功了，察看一下你的数据库，是不是多了一条 test record ☺

后记：

本文是我在做用 sybase 数据库的 ejb 项目的时候用 Oracle 做的一个测试程序，主



<http://www.ChinaJavaWorld.com>

---

要目的是熟悉 Jbuilder 和 WebLogic 的应用。匆忙之中可能有很多错误，如果有什么问题，请 email 到：[iceshape@hotmail.com](mailto:iceshape@hotmail.com)。