

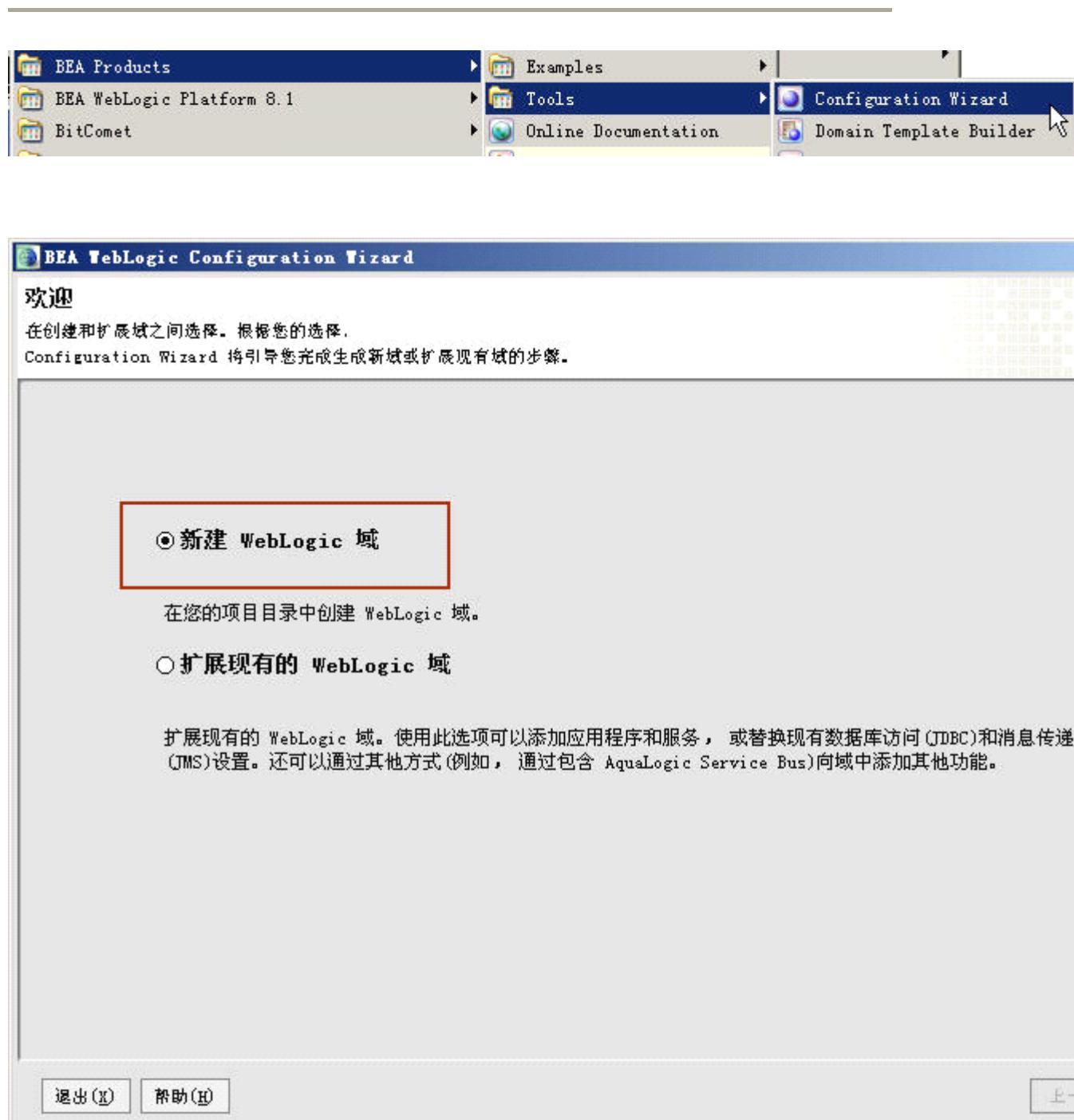
图解 eclipse+myelcipse 开发 EJB

在开发 ejb 之前，我们先得配置好服务器，我使用的是 Weblogic9.0 中文版，关于 Weblogic9.0 配置请看我的另一片文章。[配置 Weblogic9.0](#)

1. Weblogic9 配置

配置 Weblogic9.0

初次使用 Weblogic，需要对其进行域的配置。现在我就图解说明怎样配置 Weblogic9.0



BEA WebLogic Configuration Wizard

选择域源

选择要从中创建域的源。您可以通过单击所需的组件或者从现有域模板列表中选择来创建域。

◎ 生成一个自动配置的域以支持下列 BEA 产品：

- WebLogic Server (Required)
 Apache Beamiva

○ 将此域基于一个现有模板

模板位置:

<input type="button" value="上一</input>

BEA WebLogic Configuration Wizard

配置管理员用户名和密码

创建一个要分配到管理员角色的用户。
此用户是用于启动开发模式服务器的默认管理员。

放弃更改(C)

| | |
|-------------------------|---|
| *User name: | weblogic |
| *User password: | ***** |
| *Confirm user password: | ***** |
| Description: | This user is the default administrator. |

填用户名和密码

退出(X) 帮助(H) 上一

BEA WebLogic Configuration Wizard

配置服务器启动模式和 JDK

选择 WebLogic 域启动模式和 J2SE Development Kit (JDK)
以便用于此域。

要在生产中使用 WebLogic JRockit, BEA 建议在项目周期的早期使用 WebLogic JRockit 开发和测试您的应用程序。有关 WebLogic JRockit 的信息, 请参阅 WebLogic JRockit 文档。

WebLogic 域启动模式

开发模式
对用户名和密码使用 boot.properties, 并对应用程序使用轮询以进行部署。
在迭代开发过程中建议使用 Sun JDK 以获得更好的启动性能。

生产模式
需要输入用户名和密码, 对应用程序不需要轮询以进行部署。
建议使用 WebLogic JRockit JDK 以获得更好的运行时性能和管理。

JDK 选择

选择 JDK, Weblogic 要 JDK1.5 的版本。

BEA 提供的 JDK

| |
|--|
| JRockit SDK 1.5.0_03 @ E:\bea\jrockit |
| JRockit SDK 1.4.2_04-9120 @ E:\bea\jrockit |
| Sun SDK 1.5.0_03 @ E:\bea\jdk150_03 |
| Sun SDK 1.4.2_04 @ E:\bea\jdk142_04 |

其他 JDK
位置:

退出 (X) 帮助 (H) 上一

BEA WebLogic Configuration Wizard

自定义环境和服务设置

如果需要，则可以通过更改域源中已经定义的一些配置设置来自定义您的域。

是否希望自定义下列任意选项？

- 管理服务器的配置，包括监听地址和监听端口
- 受管服务器的数量和特性，包括监听端口
- 群集的数量和群集特性
- 计算机的数量和计算机特性
- 到群集和计算机的服务器分配
- JDBC 数据源（如果已经定义）
- JMS 文件存储目录（如果已经定义）

要保持域源中定义的这些设置，并直接继续创建您的域，则请保持选中“否”。

如果需要集群设置，就选择“是”，我选择的是“否”。

否

是

退出 (X)

帮助 (H)

上一





2. Weblogic8 配置：

- 以前一直使用免费简单的 Tomcat，近来因为项目需要改用 WebLogic，安装配置过程中走了不少弯路。下面就简单介绍一下 WebLogic 8 安装，并配置一个 DefaultWebApp 目录，以使象我一样的兄弟少走点弯路。

安装过程就不多说了，执行 platform812_zh_CN_win32.exe 文件，开始安装。安装目录我选在 d:\bea，选择完全安装，其他按默认值就行。

- 下面主要看看配置过程，以下步骤按照 WebLogic 配置出现界面，就不每步抓图了：）
 - “开始->程序->Bea WebLogic Platform 8.1-> Configuration Wizard”，启动 Configuration Wizard 工具。
 - 选择“新建 WebLogic 配置”，单击“下一步”。
 - 选择“Basic WebLogic Server Domain”，单击“下一步”。
 - 选择“自定义”，单击“下一步”。
 - 更改 Name，我这儿改为 hgknight，单击“下一步”。
 - 选择“否”，单击“下一步”。
 - 选择“否”，单击“下一步”。
 - 选择“否”，单击“下一步”。



- 输入“User name”和“passWord”，单击“下一步”，需要 8 位，简单一点我就都输入 hgknight.

10. 选择作为 Window 服务，单击“下一步”。

11. 默认设置，单击“下一步”。



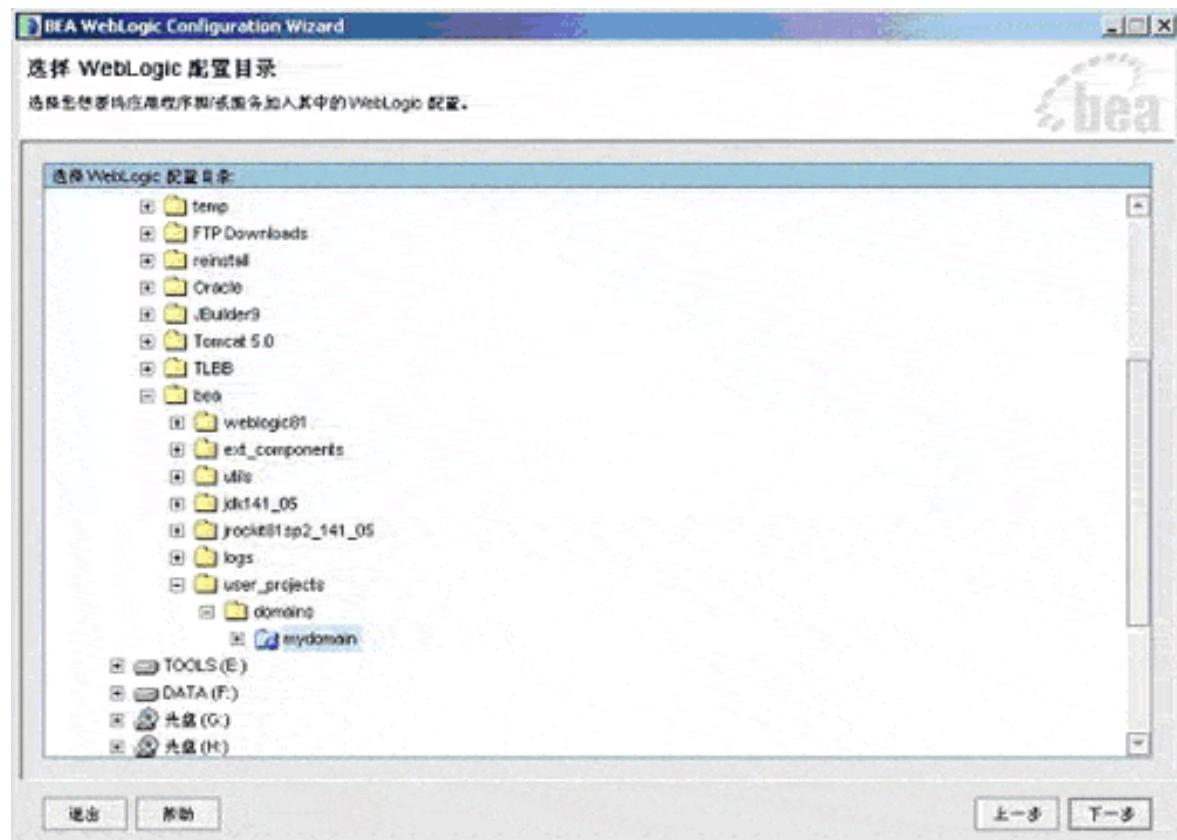
12. Bea WebLogic 默认安装了 JDK1.4.1，我要使用 JDK 1.4.2。选择“其他 Java SDK”，点击“浏览”指定 J2SDK 1.4.2 安装目录。单击“下一步”。

13. 默认值设置，单击“创建”。

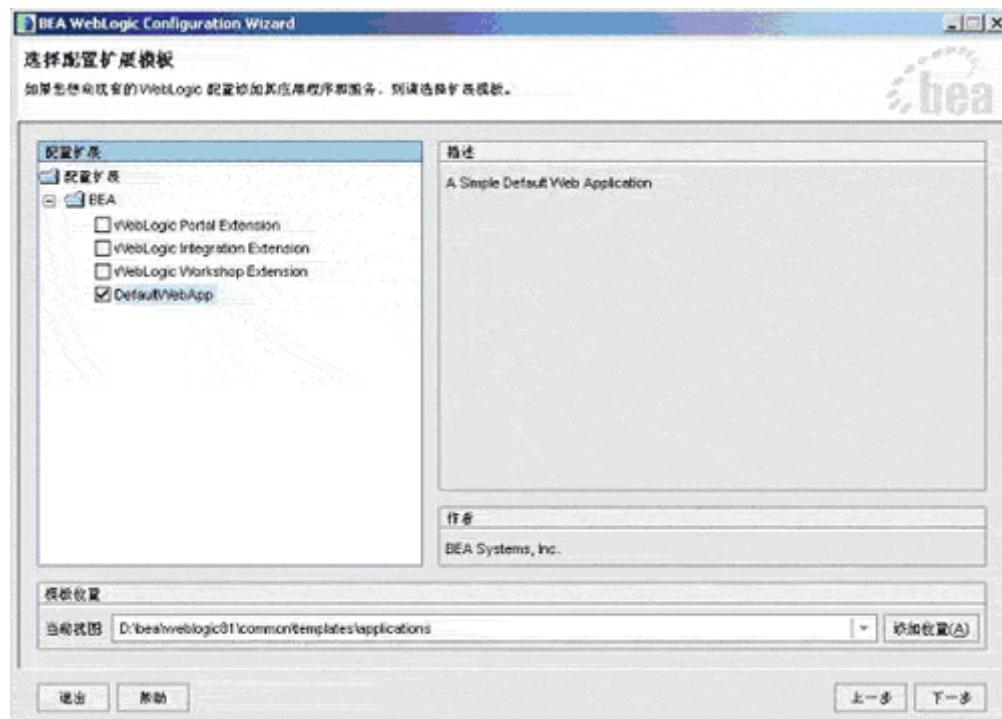
14. 选择“启动管理服务器”，单击“完成”。

15. 再次启动 Configuration Wizard 工具。

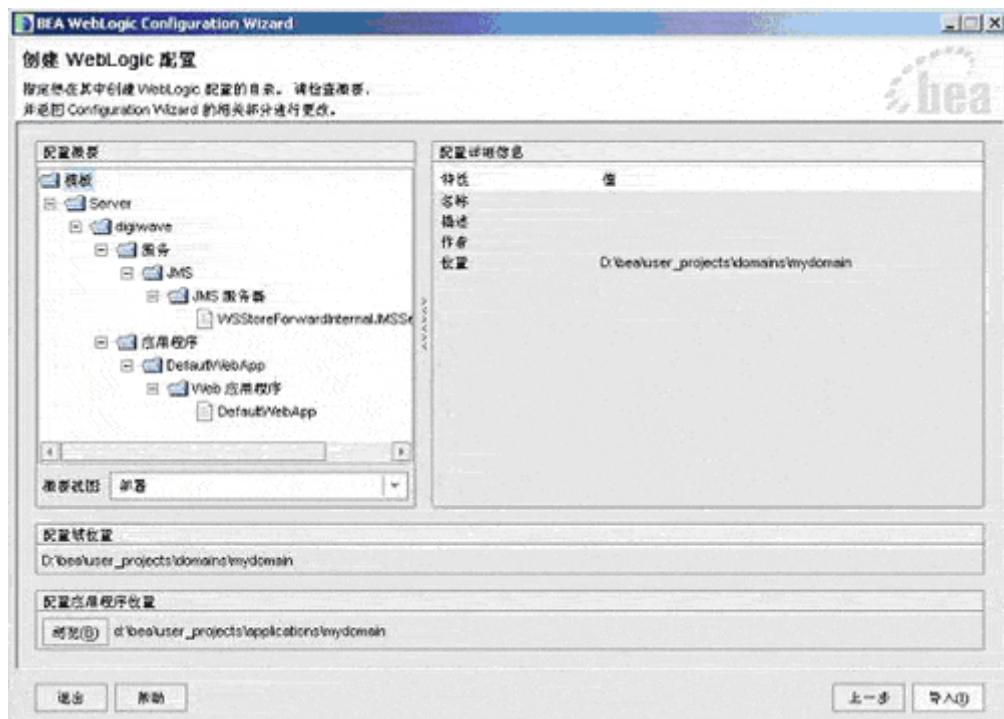
16. 选择“扩展现有的 WebLogic 配置”，单击“下一步”。



17. 选择“\bea\user_projects\domains\mydomain”，单击“下一步”。



18. 在“DefaultWebApp”前打上勾选中,单击“下一步”。
19. 默认设置, 单击“下一步”。
20. 默认设置, 单击“下一步”。
21. 默认设置, 单击“下一步”。
22. 默认设置, 单击“下一步”。



23. 可更改配置应用程序位置, 默认为“`\bea\user_projects\applications\mydomain`”, 单击“导入”。导入完成后, 系统会在应用程序位置下创建一个 `DefaultWebApp` 目录, 把你的 jsp 就放在这儿吧:)

24. 单击“完成”结束配置。
25. “控制面板->管理工具->服务”, 启动服务“`beasvc mydomain_digiwave`”。
26. 此时就可以打开 `http://127.0.0.1:7001/console`, 使用用户名和密码 (我这儿是 hgknight) 登录管理页面了。

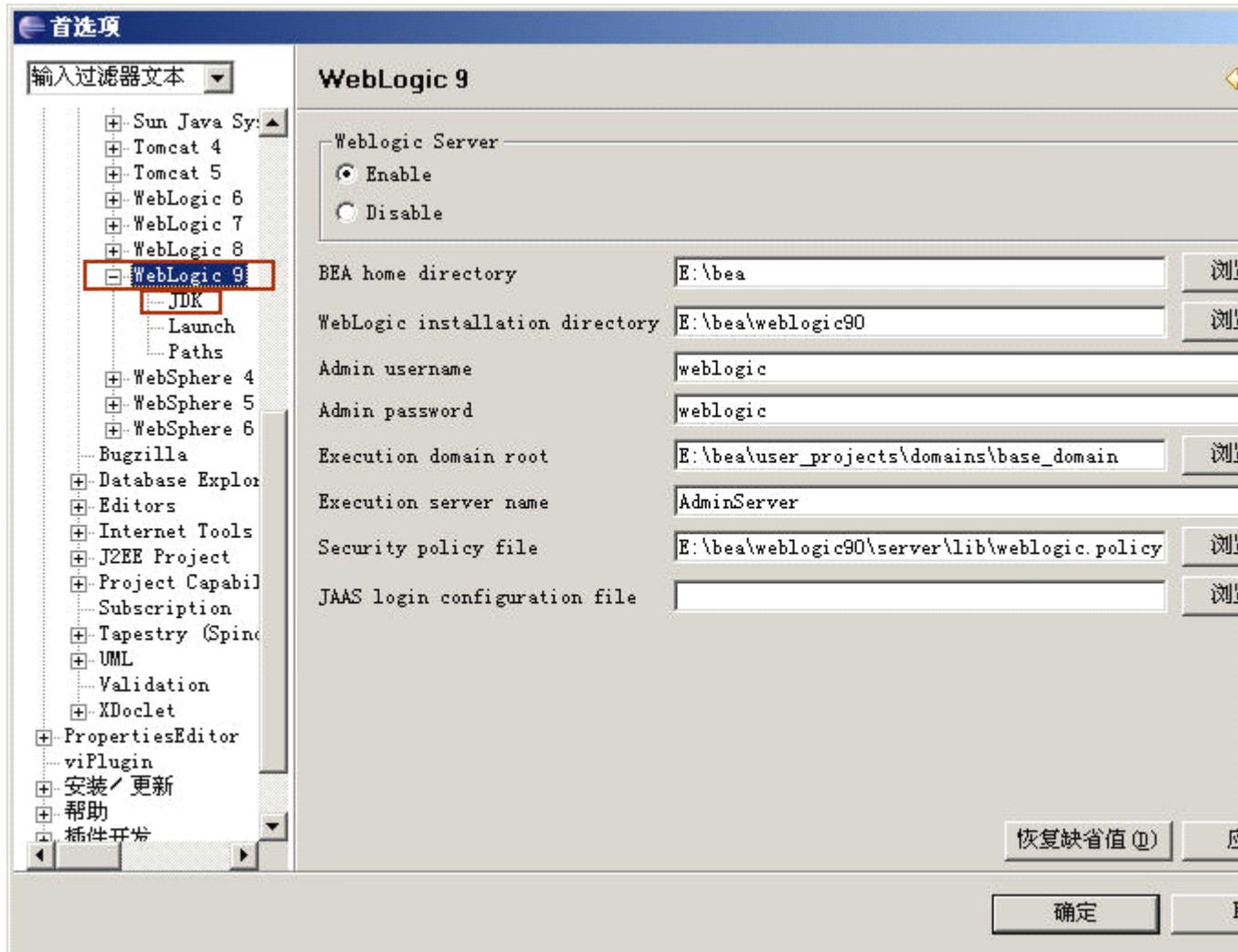
使用 Servlet, 就在 WEB-INF/web.xml 里定义就行

```
<web-app>
<servlet>
<servlet-name>ConfigServlet</servlet-name>
<servlet-class>ConfigServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>ConfigServlet</servlet-name>
<url-pattern>/servlet/ConfigServlet</url-pattern>
</servlet-mapping>
</web-app>
```

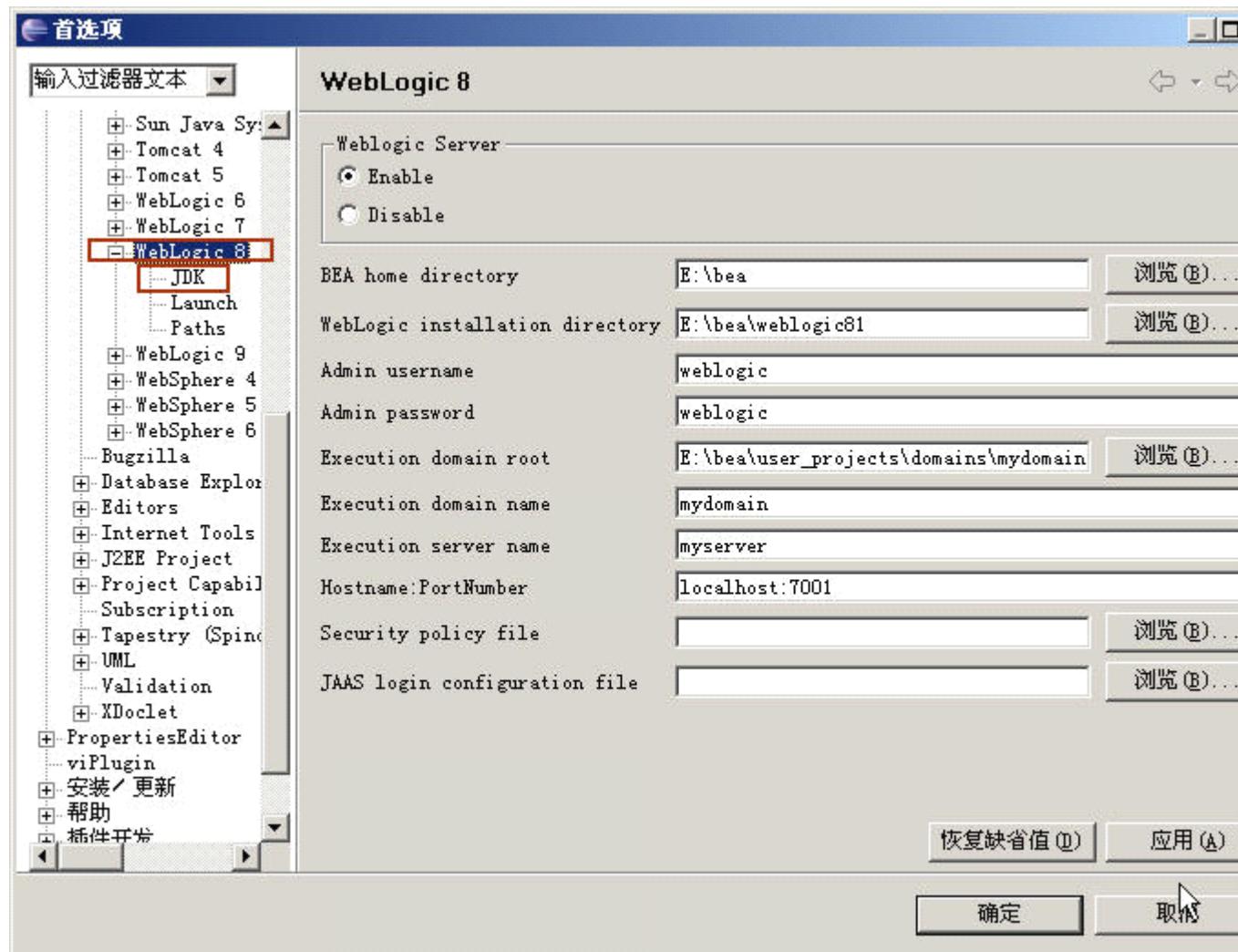
3. Eclipse 中配置 weblogic

首先需要配置好 eclipse。我这里主要说明 weblogic 的配置。



注意 JDK 选择 JDK5.0 的版本。

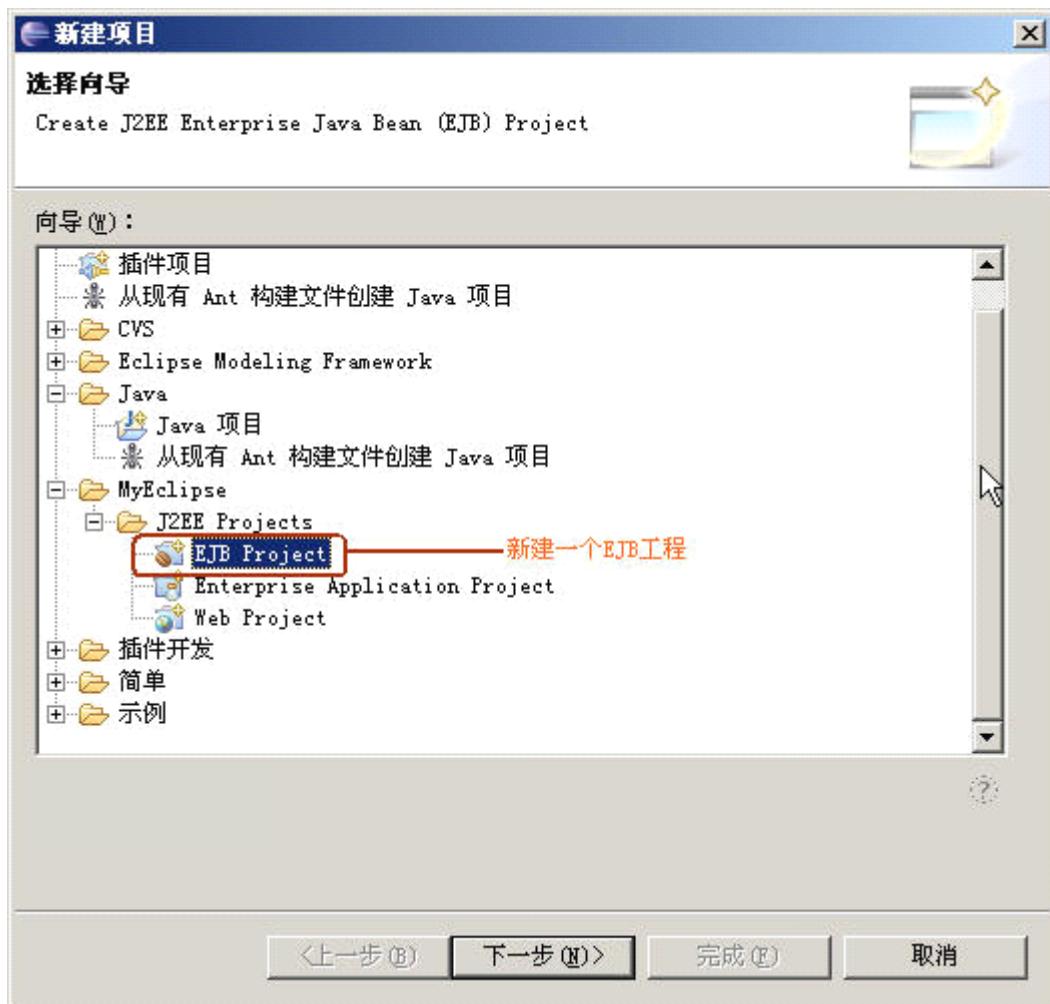
顺便将 weblogic8 的配置也贴上来，供大家比较参考

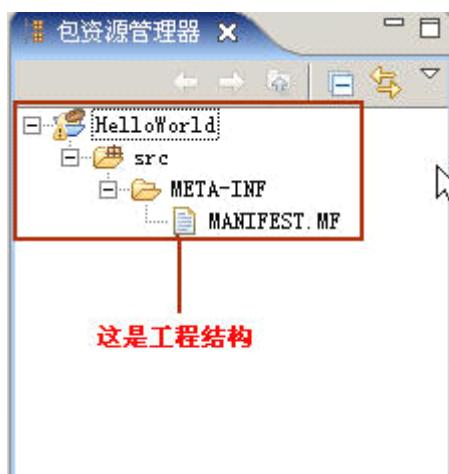
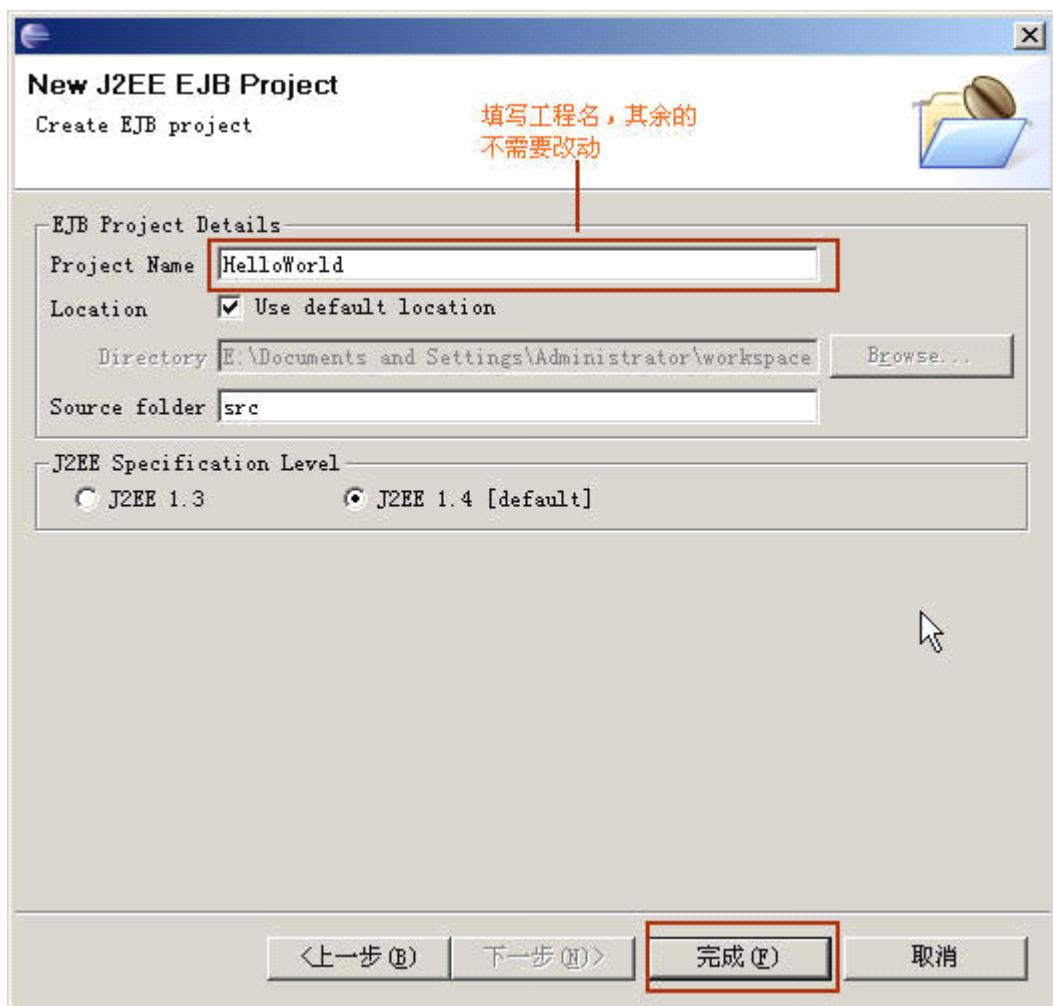


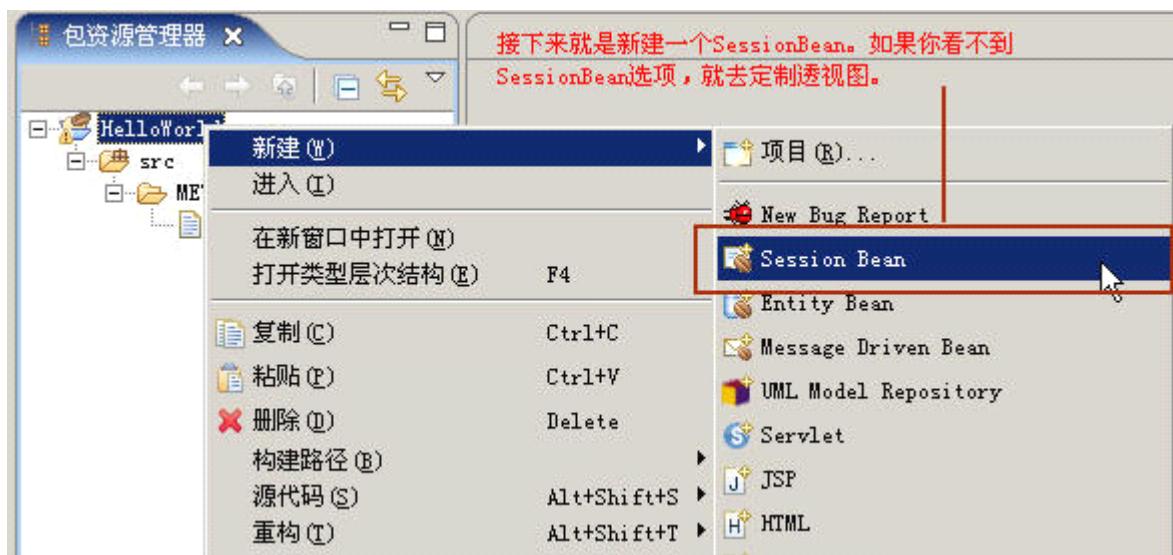
注意 weblogic8 的 JDK 版本必须是 JDK1.4。

4. 开始 Ejb 服务项目

接下来就开始我们的开发了。









下面就是 SessionBean 的代码

```

1 package com.ejb;
2
3 import java.rmi.RemoteException;
4
5 import javax.ejb.EJBException;
6 import javax.ejb.SessionBean;
7 import javax.ejb.SessionContext;
```

```
8
9 /**
10 * XDoclet-based session bean. The class must be declared
11 * public according to the EJB specification.
12 *
13 * To generate the EJB related files to this EJB:
14 *      - Add Standard EJB module to XDoclet project properties
15 *      - Customize XDoclet configuration for your appserver
16 *      - Run XDoclet
17 *
18 * Below are the xdoclet-related tags needed for this EJB.
19 *
20 * @ejb.bean name="HelloWorld"
21 *      display-name="Name for HelloWorld"
22 *      description="Description for HelloWorld"
23 *      jndi-name="ejb>HelloWorld"
24 *      type="Stateless"
25 *      view-type="remote"
26 */
27 public class HelloWorld implements SessionBean {
28
29     /** The session context */
30     private SessionContext context;
31
32     public HelloWorld() {
33         super();
34         // TODO 自动生成构造函数存根
35     }
```

```
36
37  /**
38   * Set the associated session context. The container calls this method
39   * after the instance creation.
40   *
41   * The enterprise bean instance should store the reference to the context
42   * object in an instance variable.
43   *
44   * This method is called with no transaction context.
45   *
46   * @throws EJBException Thrown if method fails due to system-level error.
47   */
48  public void setSessionContext(SessionContext newContext)
49    throws EJBException {
50    context = newContext;
51  }
52
53  public void ejbRemove() throws EJBException, RemoteException {
54    // TODO 自动生成方法存根
55
56  }
57
58  public void ejbActivate() throws EJBException, RemoteException {
59    // TODO 自动生成方法存根
60
61  }
62
63  public void ejbPassivate() throws EJBException, RemoteException {
```

```
64    // TODO 自动生成方法存根
65
66 }
67
68 /**
69 * An example business method
70 *
71 * @ejb.interface-method view-type = "remote"
72 *
73 * @throws EJBException Thrown if method fails due to system-level error.
74 */
75 public String hello() throws EJBException {
76     // rename and start putting your business logic here
77     return new String("HelloEJBWorld!");
78 }
79 //add by ganwg note: 在 weblogic 发布时, hello 方法被检查到没有 RemoteException 异常捕获, 所以改成
/*
public String hello() throws EJBException,RemoteException {
    return new String("HelloEJBWorld!");
}
如果生成了接口类, 那么也需要加上 RemoteException 异常捕获。
/*
80 }
81
```

其实就修改了其中的一个方法:

```
1 /**
2  * An example business method
3  *
4  * @ejb.interface-method view-type = "remote"
5  *
6  * @throws EJBException Thrown if method fails due to system-level error.
7  */
8 public String hello() throws EJBException {
9     // rename and start putting your business logic here
10    return new String("HelloEJBWorld!");
11 }
```

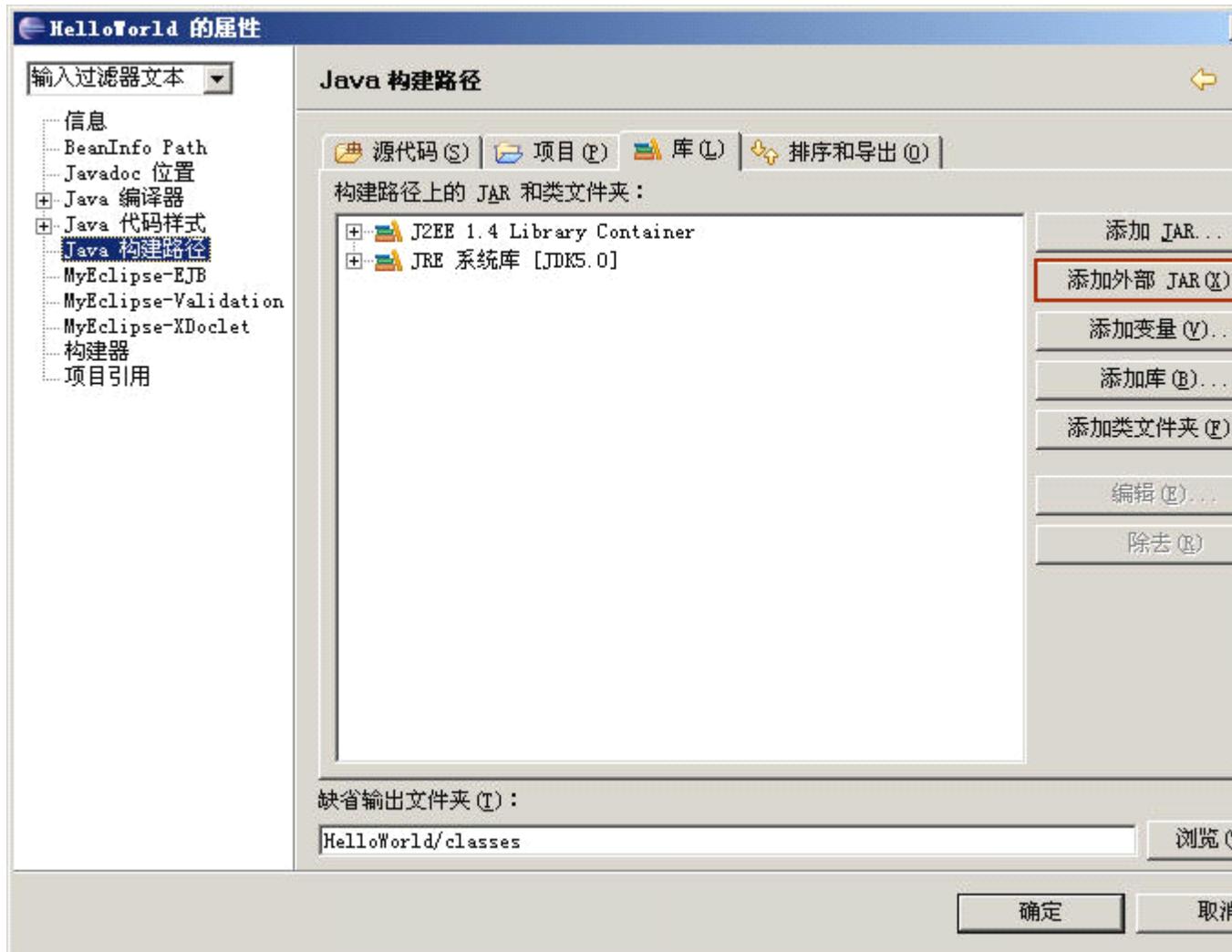
注意：代码中的解释文字不要删除，因为 XDoclet 需要。

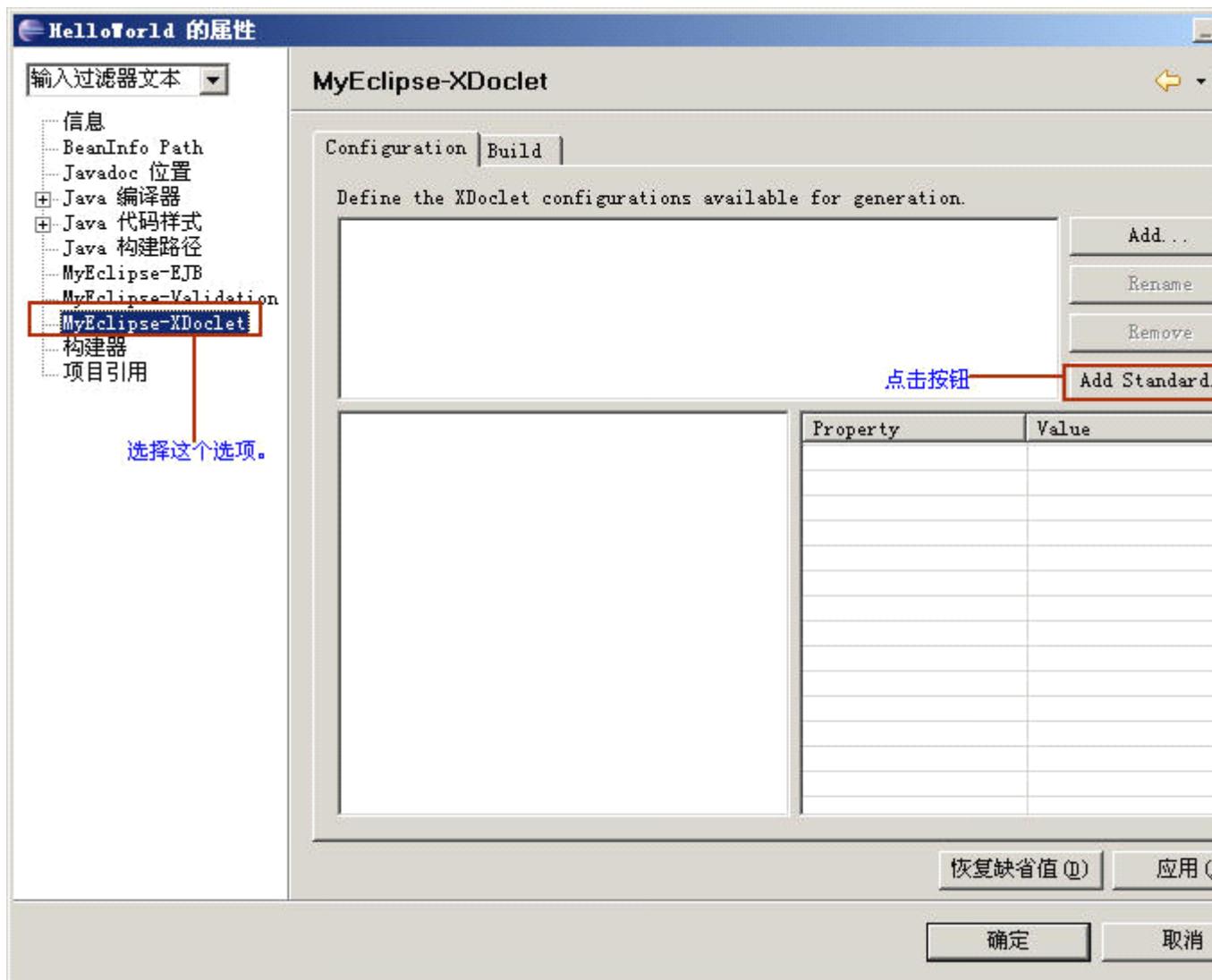
5. Ejb 服务其他代码自动生成

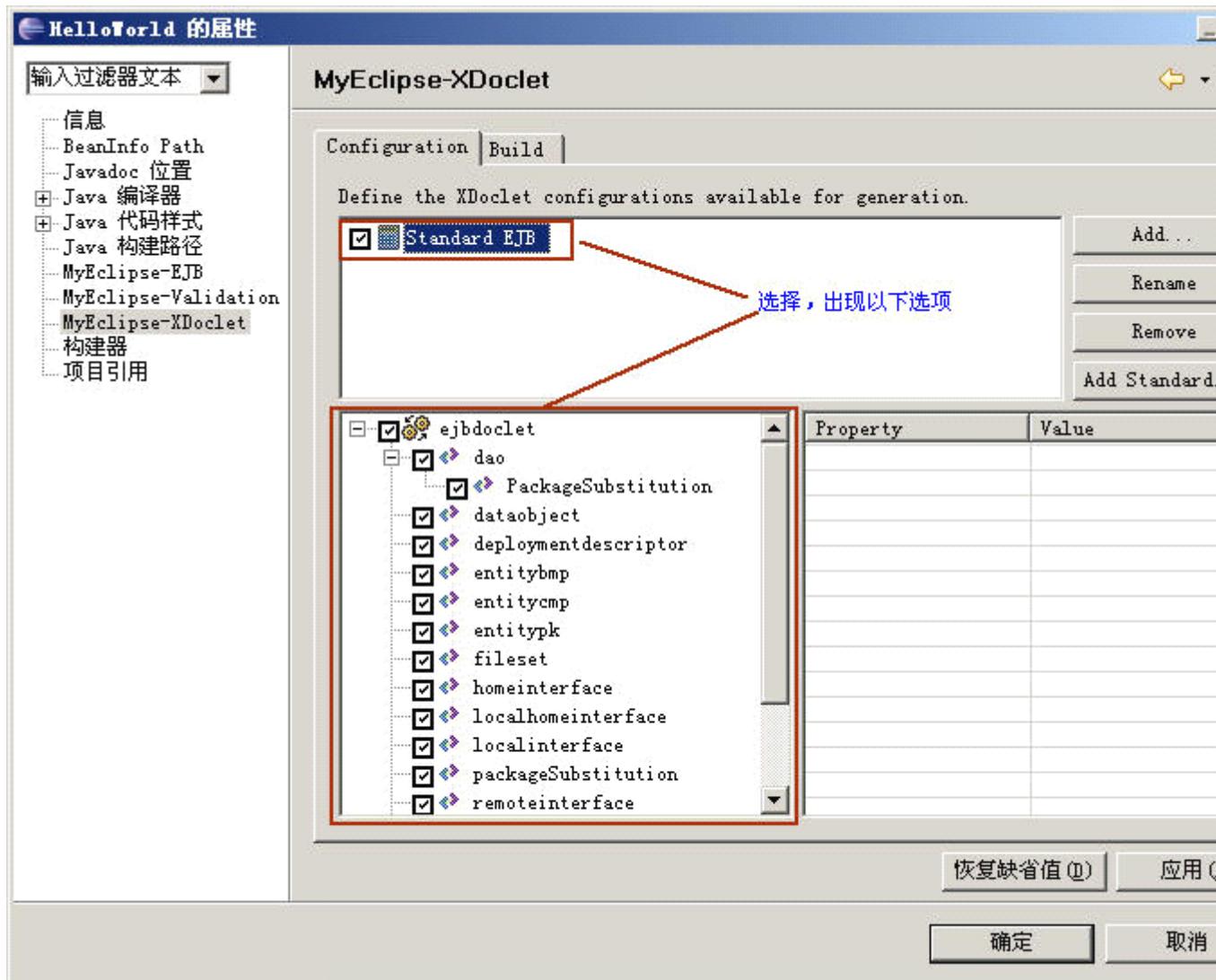
配置属性

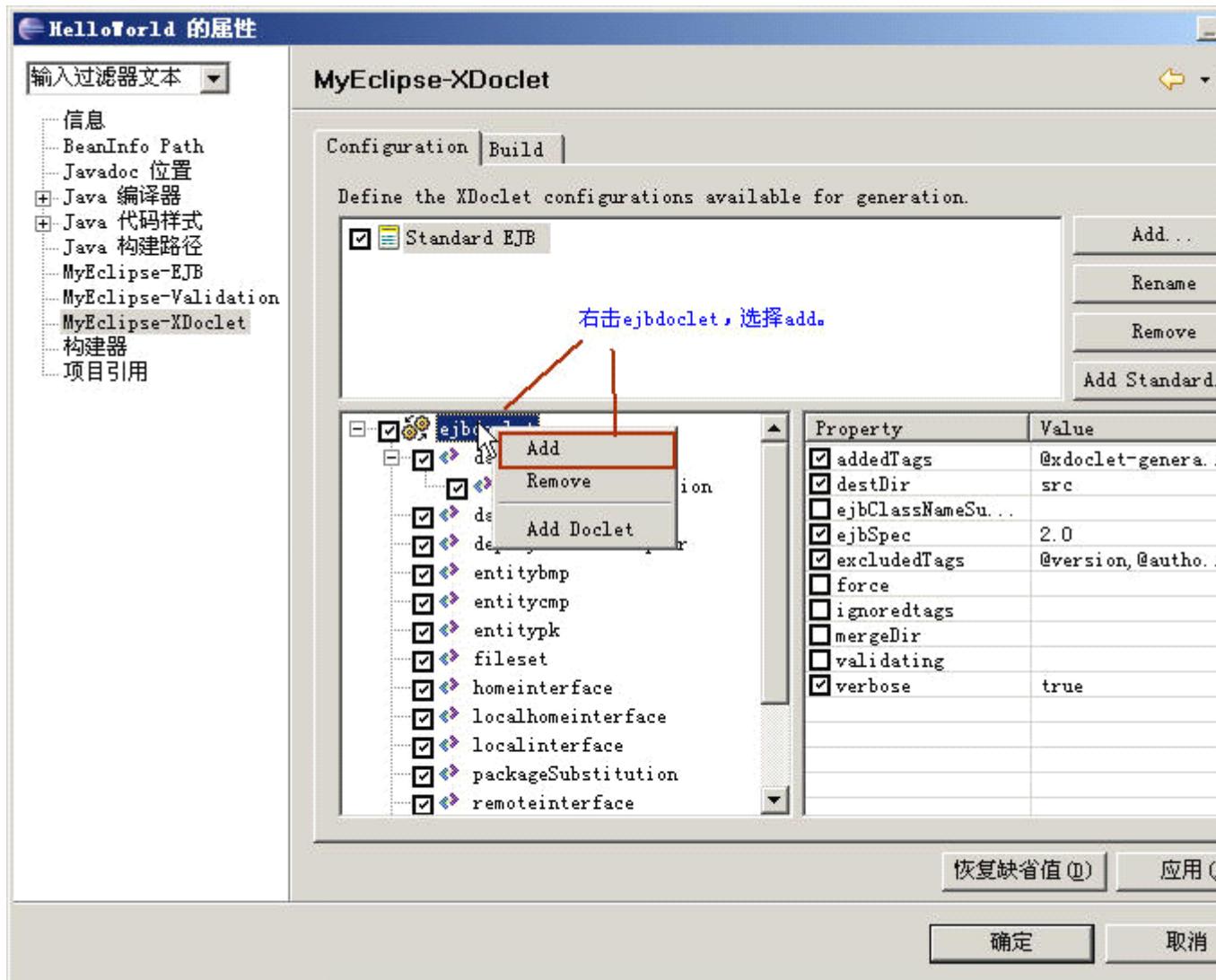


添加 weblogic.jar。我的路径是: bea\weblogic90\server\lib\weblogic.jar

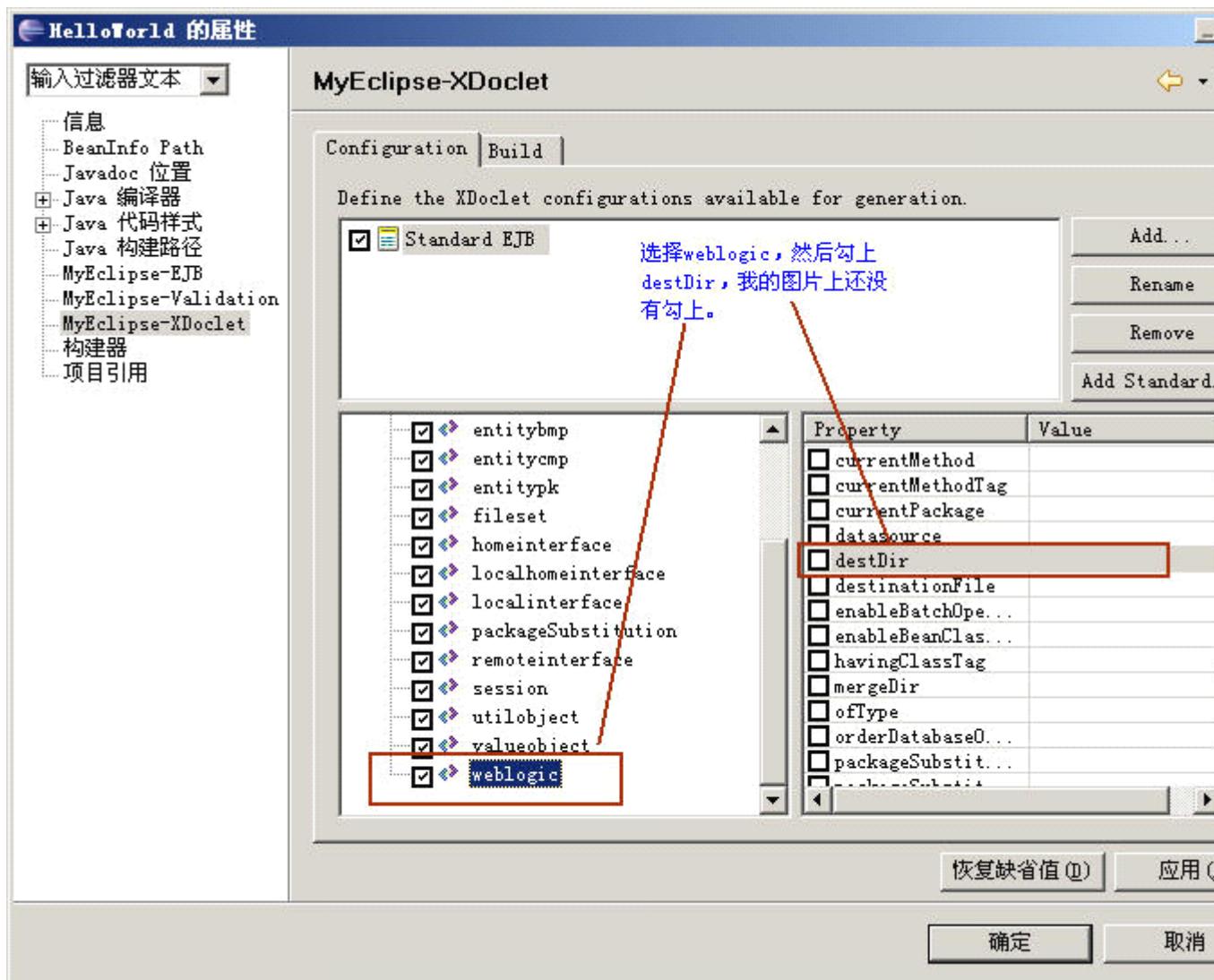


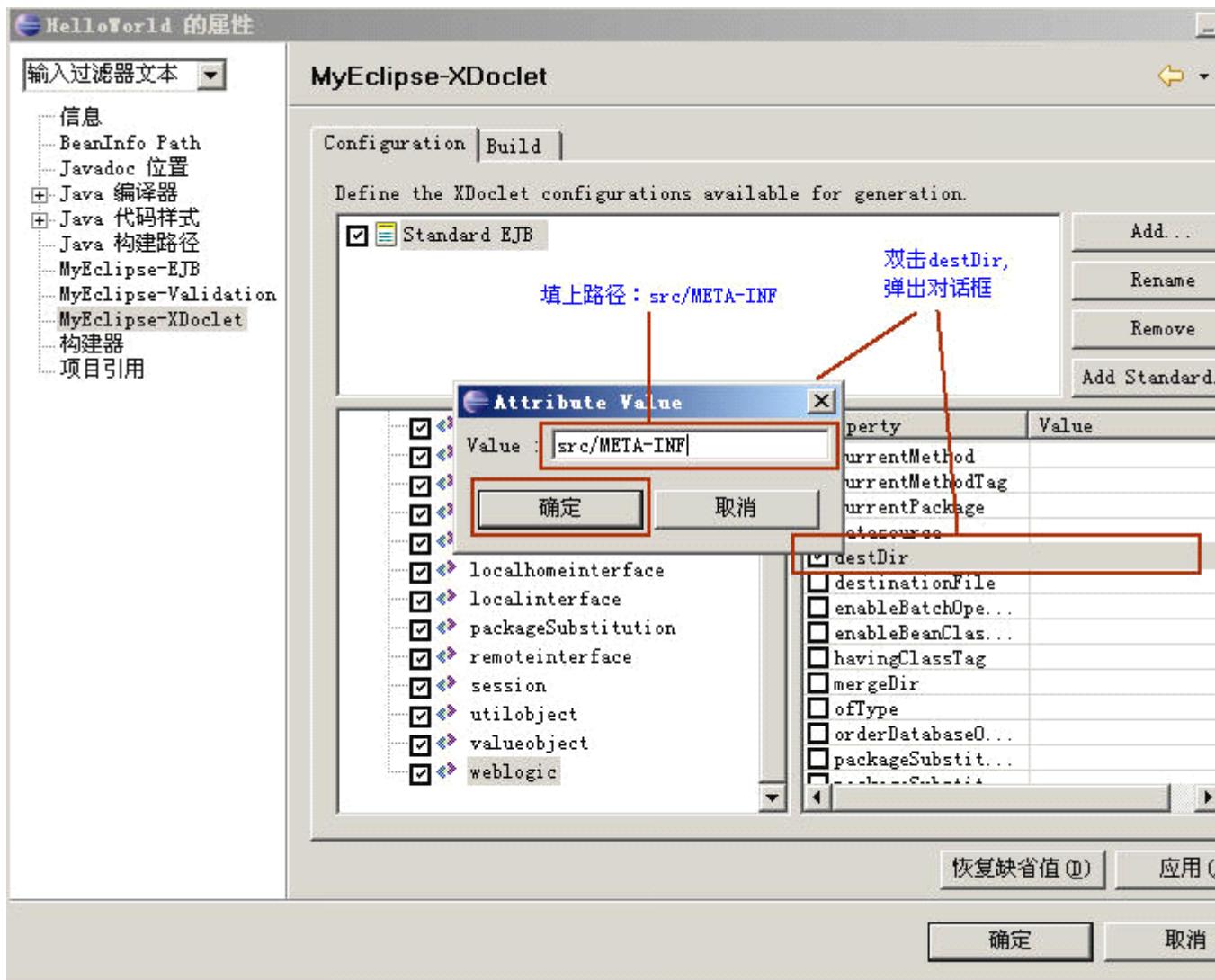


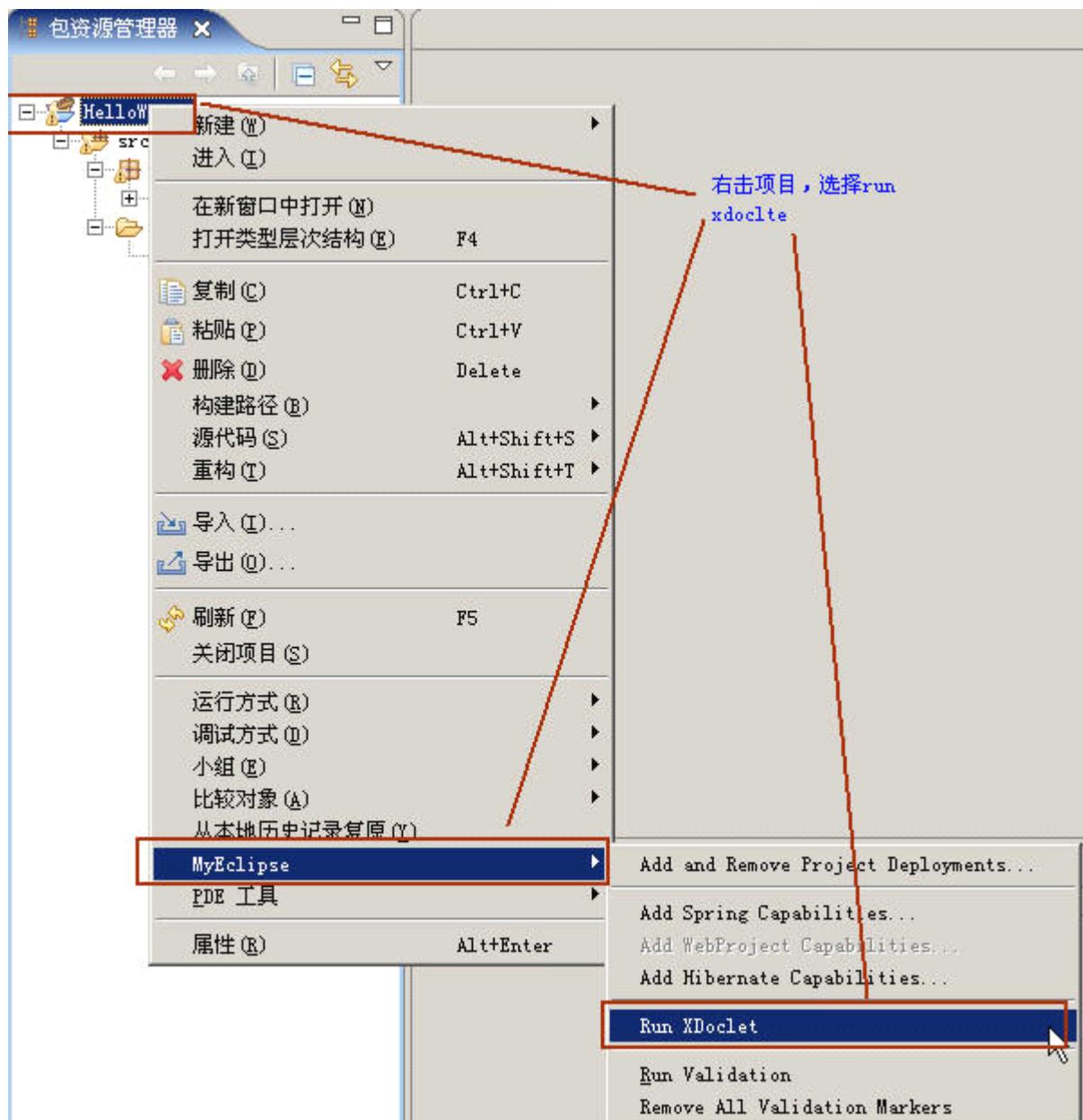


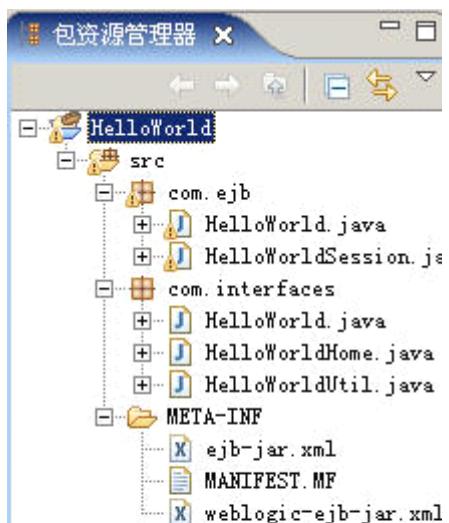












现在的项目结构,是不
是多了不少东东

6. ejb 测试程序

就下来写 EJBTest 类:

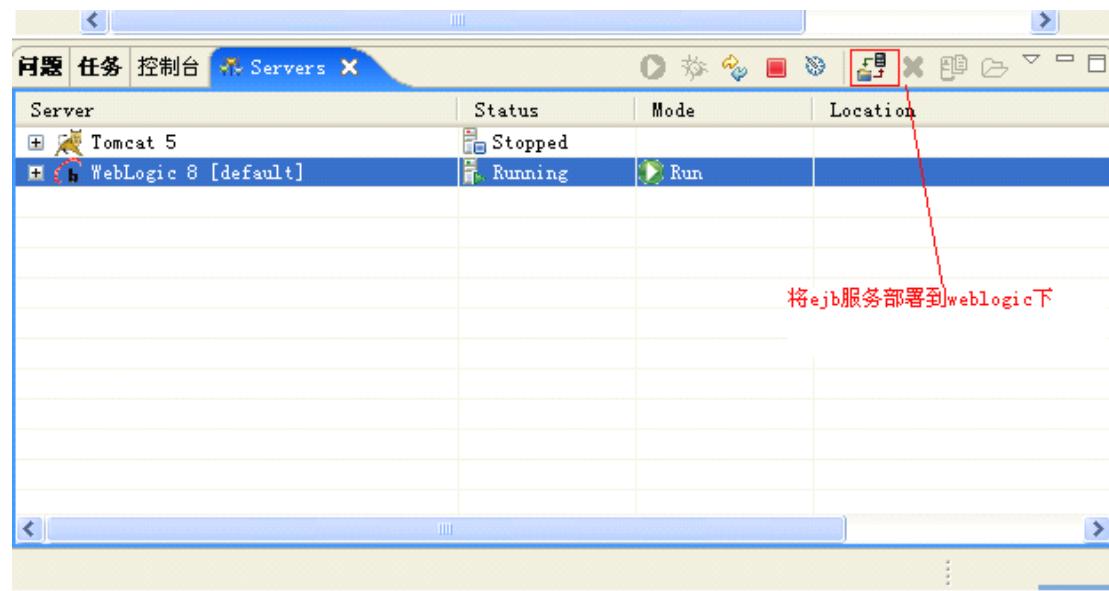
```
1 package com;
2
3 import java.rmi.RemoteException;
4 import java.util.Properties;
5
6 import javax.ejb.CreateException;
7 import javax.naming.Context;
8 import javax.naming.InitialContext;
9 import javax.naming.NamingException;
10
11 import com.interfaces.HelloWorld;
12 import com.interfaces.HelloWorldHome;
13
14 public class EJBTest {
15
16     /**
17      * @param args
18     */
19     public static void main(String[] args) {
20         // TODO 自动生成方法存根
21         Properties properties=new Properties();
22         properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,"weblogic.jn
di.WLInitialContextFactory");
```

```
23     properties.setProperty(Context.PROVIDER_URL,"t3://localhost:7001");  
24  
25     Context context;  
26     try {  
27         context = new InitialContext(properties);  
28         HelloWorldHome hwh=(HelloWorldHome)context.lookup("ejb/HelloWo  
rld");  
29         HelloWorld hw=hwh.create();  
30         String s=hw.hello();  
31         System.out.println(s);  
32     } catch (NamingException e) {  
33         // TODO 自动生成 catch 块  
34         e.printStackTrace();  
35     } catch (RemoteException e) {  
36         // TODO 自动生成 catch 块  
37         e.printStackTrace();  
38     } catch (CreateException e) {  
39         // TODO 自动生成 catch 块  
40         e.printStackTrace();  
41     }  
42  
43 }  
44  
45  
46 }  
47
```

7. 服务部署

//add by ganwg

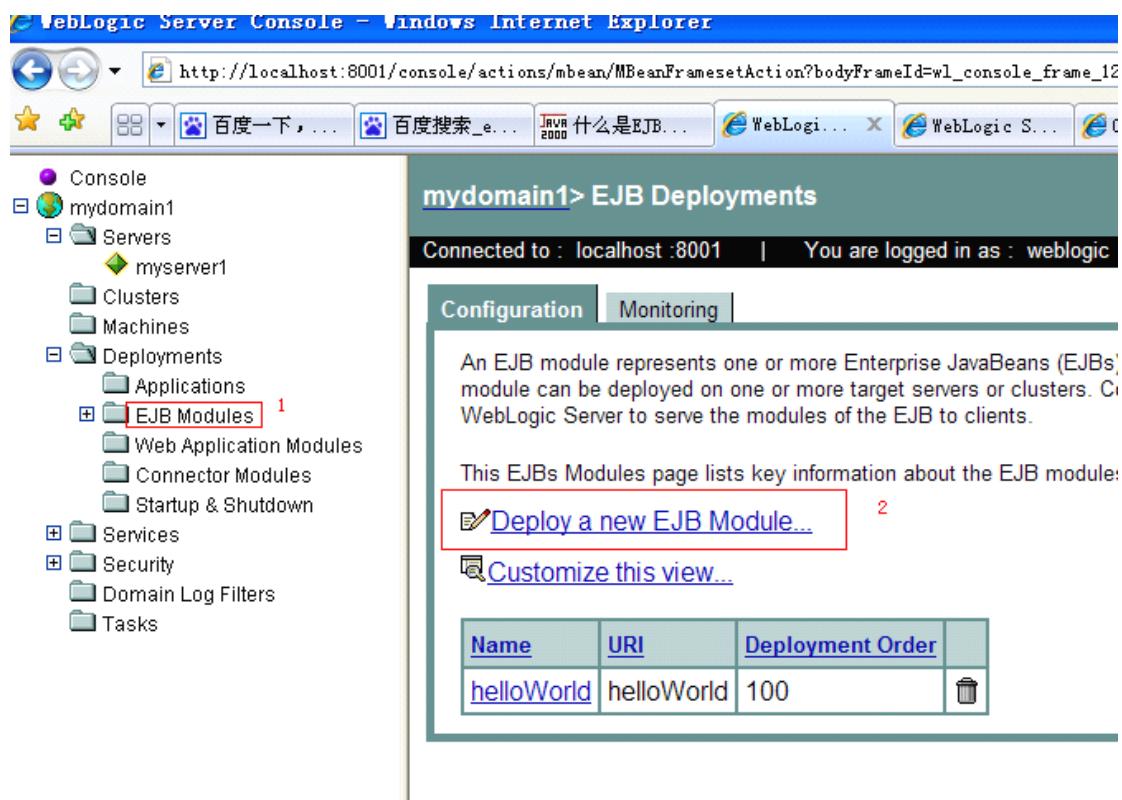
这里需要部署服务



8. weblogic 上发布 ejb 服务

部署成功后，需要到 weblogic 发布

<http://localhost:7001/console>



Connected to : localhost :8001 | You are logged in as : weblogic

EJB Deployments

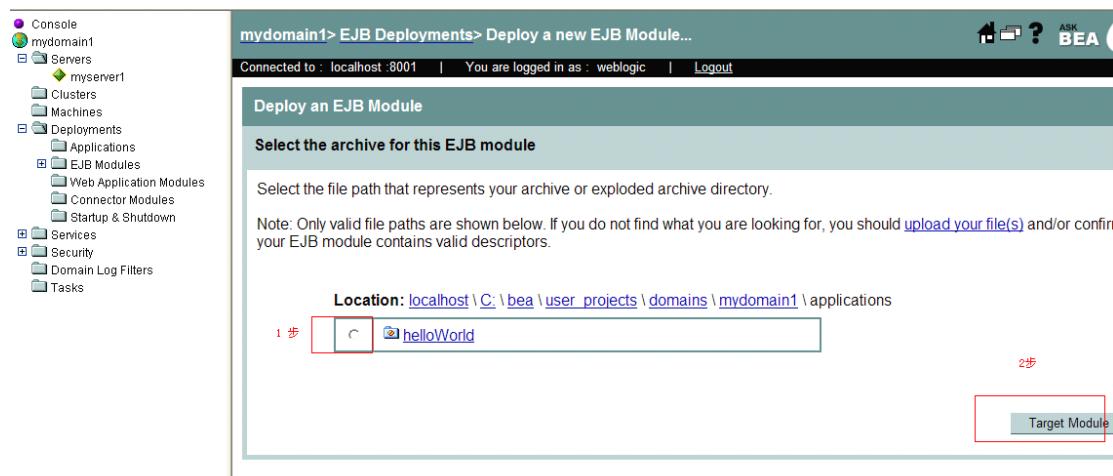
An EJB module represents one or more Enterprise JavaBeans (EJBs); module can be deployed on one or more target servers or clusters. Click WebLogic Server to serve the modules of the EJB to clients.

This EJBs Modules page lists key information about the EJB module:

[Deploy a new EJB Module...](#) 2

[Customize this view...](#)

| Name | URI | Deployment Order | |
|------------|------------|------------------|--|
| helloWorld | helloWorld | 100 | |



Connected to : localhost :8001 | You are logged in as : weblogic | Logout

Deploy an EJB Module

Select the archive for this EJB module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should [upload your file\(s\)](#) and/or confirm your EJB module contains valid descriptors.

Location: `localhost \ C:\bea\user_projects\domains\mydomain1\applications`

1步 2步

Connected to : localhost:8001 | You are logged in as : weblogic | Logout

Deploy an EJB Module

Review your choices and deploy

Deployment Targets

Your EJB module will be deployed to the following locations:

helloWorld will be deployed to
Servers - myserver1

Source Accessibility

Since this is a single server environment, no further stage configuration is required. The server will access this EJB module's files from the location specified.

Identity

Enter a name to be used to identify this EJB module.

Name:

The name of this EJB deployment.

发布时报异常，是 hello 方法没有 RemoteException 异常，加上就 ok 了，（注：helloWorld.java 和 com.interfaces/helloWorld.java 都需要加）

//add end

9. 运行测试程序

最后就是看结果了，先启动 weblogic，然后运行 EJBTest 程序。



The screenshot shows a Windows-style application window titled "控制台" (Console). The window has tabs at the top: 问题 (Problems), 任务 (Tasks), Web Browser, 控制台 (Console) (which is selected and highlighted in blue), 属性 (Properties), and 调试 (Debug). The main pane displays the following text:
<已终止> EJBTest [Java 应用程序] E:\Program Files\Java\jdk1.5.0_04\bin\javaw.exe (2005-12-26 19:23:47)
HelloEJBWorld!