



# JBuilder

*Allen Long*

*Email: [allen@huihoo.com](mailto:allen@huihoo.com)*

<http://www.huihoo.com>

2004-04



- JBuilder介绍
  - 。 JBuilder简介
  - 。 JBuilder功能一览
- JBuilder与Weblogic的集成
  - 。 设置
  - 。 开发EJB

# 内容安排



- JBuilder开发EJB
  - 。开发有状态会话Bean
  - 。开发无状态会话Bean
  - 。开发CMP实体Bean
  - 。开发BMP实体Bean
  - 。开发MDB
- JBuilder开发JSP,Servlet
  - 。开发JSP
  - 。开发Servlet



# JBuilder: 领先的Java解决方案

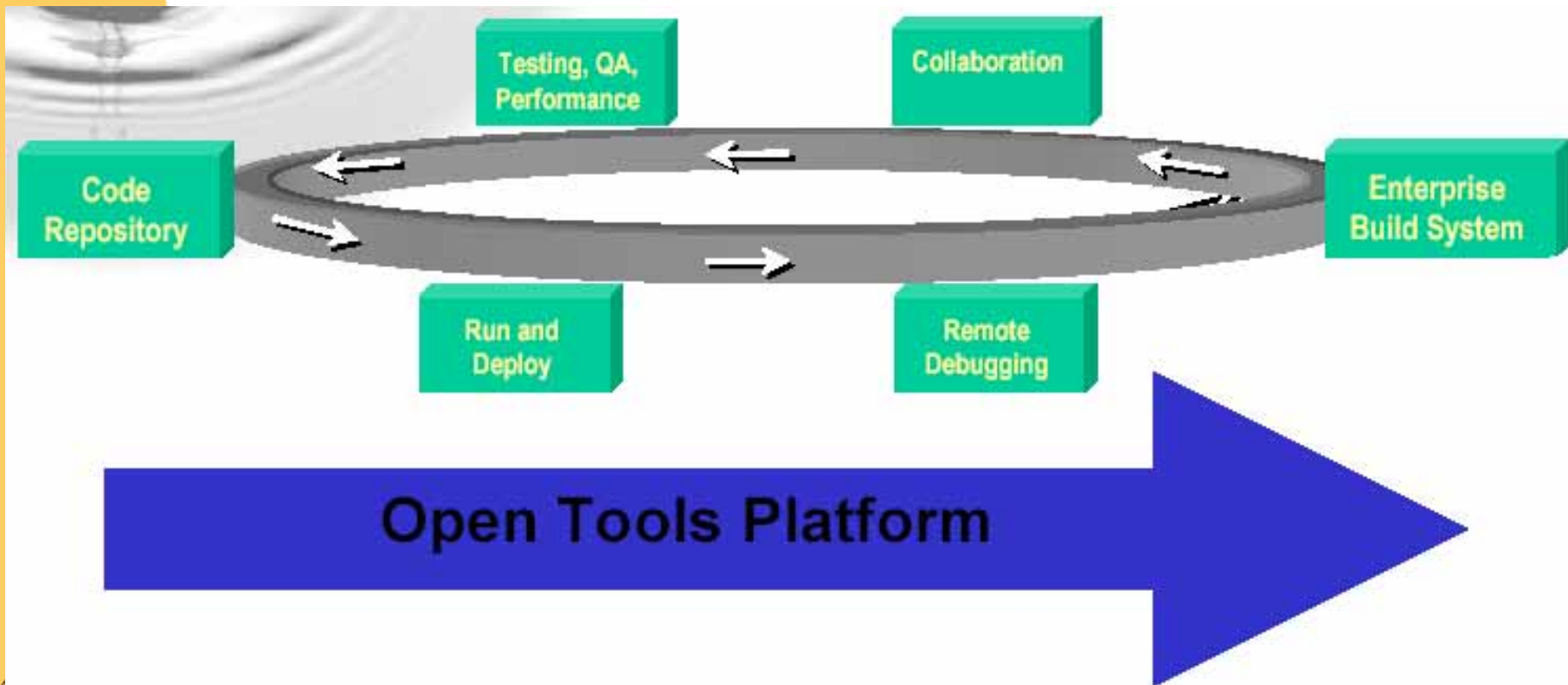


## 主要特性

- 用于J2EE应用程序快速开发的EJB 2.0的可视化设计器
- 面向如下领先应用程序服务器的分发：Borland企业级服务器,Weblogic,WebSphere和SUN ONE
- 简化数据库应用程序开发和分发的向导,工具和组件
- Web应用程序开发和采用JSP和Servlets的分发
- UML代码可视化
- 重构（refactoring）与单元测试
- 集成领先版本的控制系统
- 跨设备公布和集成商务数据的XML工具



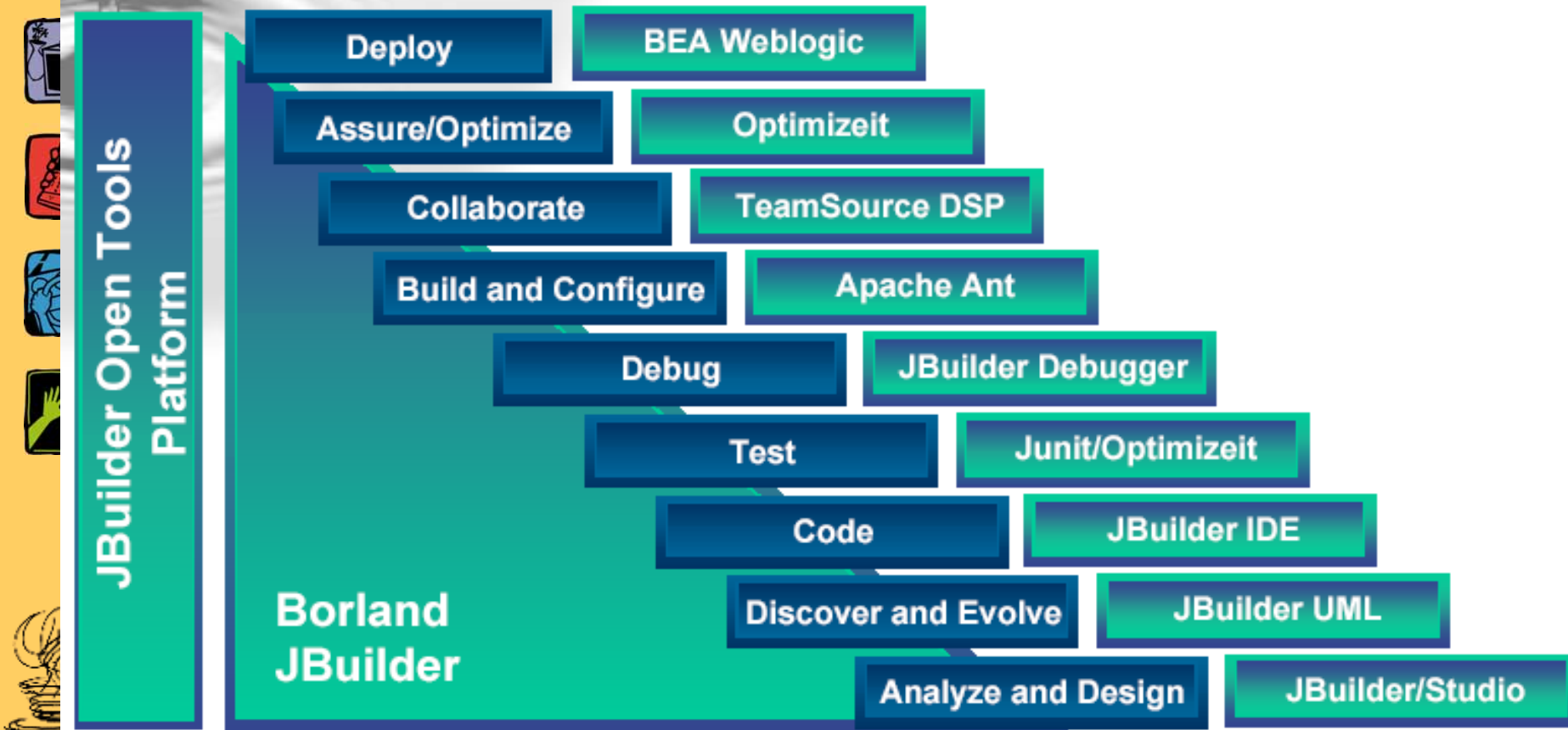
# Borland Java解决方案



# Borland Java解决方案



JBuilder integrates the complete application life cycle:



JAVA

# JBuilder8新特性



1. 支持J2EE1.4规范,在JDK 1.4.1上运行的Jbuilder8大大提高了开发的效率,在客户端可以更高速的执行应用,而且也增加了对鼠标拖拉的支持,再以前版本的Jbuilder中,只有靠拖动滚动条来修改程序,现在可以直接用鼠标的滚动键来浏览代码了。还提供了更多关于集中管理、在Linux上的UI支持。



## 2 . 工程管理



### 工程面板的文件拖拉功能

你可以从外部用拖拉的方式拉一个文件到Jbuilder8的工程面板里面来，不过这个文件必须是Jbuilder认识的文件。你就可以编辑这个文件了，如果你象使它成为工程的一部分，可以按上面的Add Files/Packages按钮。

### 创建一个空的新文件

你可以通过File|New来创建一个新的空文件

### 工程组

你可以在两个相关的工程中创建一个工程组来关联两个工程，两个工程可以在共用相同的代码的基础上，使用自己不同的应用服务器或者JDK。这个功能只在企业版里才支持。

### 目录视图

你可以在工程里面添加一个目录视图，目录视图列出了一个目录下面的所有文件，当外界改变了这个文件的时候，目录视图会相应的更新到工程里去。

### VisualCafe工程的导入

Jbuilder8能够用向导来顺利导入VisualCafe的工程和文件。

### 在Required Library里新增一个工程

如果一个工程依靠另外一个工程的话，你可以在Required Library里加入一个工程。





## 3 . 调试功能的增强



### • Smart Swap

有了Smart Swap，你就可以在debugging的情况下修改代码了。Smart Swap改变、修改并更新你所修改过的文件。有了它，你可以在当前的执行点上在同一个debugging Session下测试代码。（JBuilder8 的Smart Swap技术是基于JDK 1.4的Hot Swap技术）

### • 设定Execution Point

你可以在当前被调试的线程上增加一个Execution Point，这将改变它当前位置的执行断点。

### • Smart Source

Smart Source能够改变你的代码的视图，这样你既可以在debugging的时候浏览Java代码，也可以浏览非Java代码。（JBuilder8 的Smart Swap技术是基于JDK 1.4的JSR45技术）

此外：JBuilder8对在debugging中的编辑、Change Watch command、ExpressionInsight Window、File Modified dialog box等都作了优化。



## 4 . 在单元测试方面的改进



JBuilder8利用Cactus扩展了它在服务器端的单元测试。当然还是和以前一样支持JUnit，以下是JBuilder8再测试方面支持的新的特性。

### 1.Cactus的支持

有了Cactus，JUnit可以支持在服务器端的Java代码的测试。

### 2.Cactus设置向导

您可以从(Wizard——Cactus Setup)菜单来设置Cactus。

### 3.EJB客户端测试向导

EJB客户端测试向导先在可以产生两种类型的Test Client：JUnit Test Client和Cactus Test Client。

### 4.JUnit Text Collector

在RunTime Configuration属性对话框里有一个新的运行期测试类型选项让你更加简单的验证你的测试用例。

### 5.Unit testing Stack Trace Filter

能够让你集中对你有用的栈跟踪信息。

### 6.JUnit3.8

JBuilder8里的JUnit的版本是3.8



## 5 . UML



原来Project Property里的UML页在Jbuilder8里被去掉了，里面的两个选项被移到了其他的页上去了。其中UML diagrams被移到Project Property里的Class Filters页，你可以再Class Filters页里的Name旁边的下拉框中选择UML diagrams来加入或者执行一些类和包。Diagram References From Generated Source被放在General页里的最下方。

## 6 . Web Development



JBUILDER8另外一个很重要的新特性是增加了对Struts的支持。Struts是基于Model2或MVC增加了软件的设计。model包含的数据，view是数据代表，controller控制model和view之间的交互。最典型的view就是一个JSP页面，controller是一个叫ActionServlet的Servlet，model可以是任意的数据访问技术，从JDBC到EJB。整个框架包括了classes、JSP、Servlet、tag library在HTML和Java代码之间建立了一个清晰的分层和业务逻辑的形象表达。Jbuilder8提供了工具和向导来使你快速的建立一个基于Struts的应用。这些工具和向导包括：

- 你可以在Configure Libraries里设置你的Frameworks作为Libraries。
- 基于Struts的WEB应用向导、JSP文件、web.xml文件。
- JSP到Struts的转换向导，把一个已经存在的HTML文件和JSP文件转换成使用Struts标签的文件。
- 可视化的编辑struts-config.xml文件。
- 能够在Action向导里为你的Struts应用创建一个Action Class并注册到struts-config.xml文件中。
- 能够在ActionForm向导里创建一个使用一个特定的JSP字段的ActionForm并注册到struts-config.xml文件中，也可以从ActionForm来创建一个JSP文件。

其它的一些改进包括：

- 在Web Application向导里的WAR generation选项在JBUILDER8里改成了Build WAR，

有四个选项：

- (1) When Building Project Or WebApp：当编译工程或者是Web Application的时候创建WAR文件。
- (2) When Building WebApp Only：只在编译Web Application的时候创建WAR文件。
- (3) When Building Project Only：只在编译Project的时候创建WAR文件。
- (4) Never：不产生WAR文件。

- 你可以在JSP里选择来手工的加入JSP 标签库和框架支持，JBUILDER8自带的Tomcat版本是 4.1,Tomcat4.1不支持JSP的调试。另外在JSTL等方面也由一些改进。



# 7 . Web Services



JBuilder8自帶了Borland Web Services for Java，并提供了向导和工具更快的开发Web Services。Jbuilder8可以和各种各样的Web Services工具包集成工作。只要你选择了Web Services工具包，JBuilder8就可以使你的工程支持倒入服务，输出Web Services的Java Classes。

你只需要把工程设置成支持Web Services的就可以输出Web Services的EJB，JBuilder8默认把所有的无状态Session Bean远程接口里的商务方法显示成Web Services的，你可以不用默认为，手工来设置你想要显示成Web Services的EJB Modules、Beans、方法。

JBuilder8的Web Services功能和你所选得开发包有关，包括：

- 你可以设置你的工程支持Web Services。
- 建立一个WSDL文件来表写你所开发的Web Services。
- 创建调用Web Service的客户端。
- 倒入一个描述Web Services的EAR或者WSDL来创建调用服务的类。
- 创建服务器端的代码来本地化Services类。
- 把Java类转换成Web Services。
- 自动的转换无状态Session Bean远程接口里的商务方法。
- 再调用服务的时候监听客户端和服务端端的SOAP信息。
- 利用Web Services Explorer来查找或者发布一个Web Services作为一个UDDI ( Universal Description, Discovery, and Integration ) 注册。





- **SAX Handler wizard**

JBuilder8的SAX Handler wizard使用了JDK1.4里的JAXP。

- **Cocoon**

在JBuilder8里Cocoon的版本升级到了2.0.3,在Web Application向导里可以选择Cocoon作为framework，如果你使用Cocoon Web Application向导的话，Cocoon前的框是自动选择上的。

- **在IDE里的XML选项**

在JBuilder8的(Tools|IDE Options)里有个新的选项——Ignore DTD。这个选项是默认选择上的，这样JBuilder不解析DTD，也不报告结构面板里的错误，如果不选上的话，JBuilder每次都解析DTD并报告结构面板里的错误。



## 9 . 运行设置



JBuilder8重新设计了运行期设置属性对话框，在Run这一页里面包含了一个下拉框，你可在里面选择你需要设置的应用程序的类型。当你选择Edit或者Copy功能时，这个框是不能被改动的。

JBuilder8包含了可以让你在JBuilder中象其他工程一样运行、调试、最优化你的开放工具工程的开放工具运行其设置类型，该类型你载下拉框中可以看得到。这样你就不用退出JBuilder，创建一个JAR文件再COPY到相应的目录中去，然后再重新启动JBuilder了。



# 10 . 编译系统



- Apache Ant的版本更新至1.5.1，以下是几个新增的功能：

--Build.xml以相对路径显示在工程面板里，你可以右击build.xml文件后选择Properties，在Ant页里修改这个选项。

--使用Ant向导来向工程里新增编译文件，如果你使用向导，系统会自动识别编译文件为Ant编译文件。

--如果你想要Ant使用JDK工程，在Project Properties里的Build页的Ant页里选择User Project JDK When Running Ant。

- Project Properties里的Build页

在Project Properties里的Build页里有了一些变化：

--Java页——你能选择编译的编译器。把General页里的Enable Assert Keyword选项放到这里来了。

--General页——新增了Autosave All Files Before Compiling、Refresh Project Before Building、Generate Source To Output Path等选项。

--ANT页——新增了User Project JDK When Running Ant选项。

--Menu Item页——你能在这里新增一些指定的对象。（例如你设置的ANT命令）

--Web Service页——新增了Regenerate Deployment选项，覆盖Web Services的配置文件。

- 编译工程组

你能够在Project Group Properties改变一个工程组里个各子工程的编译顺序和定制菜单项目。

- 资源管理

可以让你用JBuiler打开你不认识的文件，并在编译的时候跟随工程一起复制到相应的目录





## • 排除档案文件的依赖

你现在可以排除档案文件的依赖当你用Archive Builder和Native Executable Builder创建档案文件和创建WAR文件的时候。Archive Builder和Native Executable Builder在档案文件的属性里的Content页，Web App里的Class页里指定了用不同方式把类和资源整合在一起的几种方式：

Classes: (1)Specified Only (2)Specified And Dependent (3)All Resource: (1)Specified Only (2)All

例如：如果你想在档案文件中包括所有的类和资源的话，你可以在Classes面板和Resources面板里都选择All。如果你不想加入这些依赖的资源，而只是想加入自己指定的一些东西的话，你可以在两个面板里都选择Specified Only选项，然后可以用Add Classes按钮假如你想要的东西。如果你加类的话必须加工程输出路径的类，如果是加文件的话必须加入工程的Source路径里的文件。

## • Archive Builder

Archive Builder现在有了一种新的档案类型，可执行的JAR文件。如果你要把一个已经存在的JAR文件变成可执行的话，你可以选择这种类型。你也可以为执行选择一种运行期的设置。

重新设计了Archive Builder和Native Executable Builder里的Content页，可以排除档案文件的以来，见前面。

如果你在Configuration页里为一个应用程序选择主类的时候选择了<Auto Select>，工程在运行的时候就找到默认的运行设置包括默认的参数来运行你的应用程序。如果没有默认的设置或者默认的不是一个应用程序的设置，系统就自动使用第一个应用程序运行期设置。



# 11 . 配置



## • 创建本地执行

如果你选择了Native Executable或Executable JAR档案类型的话，你就可以使用Archive Builder来创建可执行的文件。你也可以使用Native Executable Builder来创建。

## • 设置本地执行文件

当你用Archive Builder或Native Executable Builder创建了可执行的文件后，你也可以选择属性面板里的Runtime来设置创建或者重写配置文件。有以下的设置类型：

- (1) Create executable configuration.
- (2) Create executable configuration and save a copy in the specified file.
- (3) Override the executable configuration with the specified file.

如果你选择2或者3的时候，设置文件将被加到工程里去。



# 12 . 生产力的增加



- 消息框

你可以使消息框在屏幕的任何位置变成一个自由移动的窗口。

- 状态栏消息

你可以使用Tools|IDE Options|Browser|Status Message Timeout选项设置状态栏信息保持时间。

- 使用Regular expressions选项查找

在JBuilder8里的查找选项里新增了使用Regular expressions查找。

- 编辑器

大多数效率的改进是针对编辑器的

- 拖放

你可以拖放文本, 选择上了一段文本后, 可以用鼠标把被选择的部分拖放到你指定的地方。这样的功能以前大家很熟悉是在微软的工具里。

- 行数

在代码的左边默认显示了各行的行数, 你可以在(Tools|Editor Options)里去掉这个选项。也可以在Line Number页边直接按右键来快速取消这个选项。

- 选择文本

你可以利用左边的行数来选择一整行或者一行的一部分。

- 代码格式

你可以在Project|Project Properties里的Formatting页里选择相关的选项来制定你的代码格式的偏好和自动对齐你的代码。在旁边的一个窗子里会显示出你的设置的改变。

- 缩放工具

在编辑器的底部有一个放大镜的按钮, 用这个按钮你可以缩放你的代码。

- 快捷键的改进

你可以在编辑器的底部改变快捷键的设置, 可以设置不同的方式也可以在Editor Option里使用定制来改变某个方式里具体的某个键。



# 13 . 团队开发的改进



## • Merge Conflicts

再History面板里有一个Merge Conflicts页，你可以在这里察看或者解决在工作台和版本控制库里相应的文件的不一致情况。（Merge Conflicts只支持CVS和VSS）

## • CVS的改进

CVS方面有两个关键的改进：

- （1）如果你在JBuilder8里用pserver方式连接的时候，你可以指定端口号。
- （2）你能在Pull Project From CVS向导里检库里面的一系列的模块和分支。

## • Clear case的改进

Clear case在版本控制设置管理方面也有很多改进，Status Browser主要是一个浏览工具：可以浏览激活的工程，显示每个文件的版本控制状态，显示可用的版本和不同版本之间的差别。Commit Browser提供了Status Browser浏览功能，提供改变文件版本控制操作的一般的访问，有了Commit Browser，你能够为每一个文件设定你想要设定的版本控制命令。输入一些单个文件或者整个组的注释，然后点一下就可以执行所有的命令了。另外还提供动态和快照支持、控制台输出、对于一个文件多个人的修改合并功能、用命令来访问ClearCase工具和提供UCM的支持。



# 14 . 支持J2EE服务器



以下是JBuilder8支持的J2EE服务器：

支持BEA WebLogic Server

支持IBM WebSphere

支持Borland Enterprise Server

支持Oracle9i Application Server

支持Sybase EAServer



# 15 . JavaDoc的改进



JBUILDER8包含了一个包编辑器来让用户创建、编辑、删除工程中那些单独的包的package.html文件



# 16 . Refactoring的改进



在结构面板上有以下的Refactoring命令：

Find References 、 Optimize Imports 、 Rename Package 、  
Rename Class 、 Rename Method 、 Rename Field 、  
Change Method Parameters 。

# 小结



JBuilder8主要在以下方面提供了大量的功能：

- (1) 使用一个新的标准的Web开发框架来加速Web应用程序。
- (2) 通过工程组、代码格式化、比以前更紧密的和代码管理控制系统联系在一起。
- (3) 比以前更快的在支持J2EE、Web Applications、EJB的单元测试工具的支持下，使用灵活的开发方法开发软件。
- (4) 使用改进的编辑器增加了生产力和代码可靠性，还有HotSwap debugging和改善的性能。





# Jbuilder集成Weblogic



点击Tools>Configure Servers

。Jbuilder已经预设置了多个应用服务器的基本配置.你现在所要做得就是Enable Server,并将缺省设置改为应用服务器实际安装的位置.

。这里的设置会直接影响到File>New中的选项目,如你选择了Tomcat,那么New中有关EJB的部分就变灰而不可用.而选择WebLogic后,这些选项变为Enable. CORBA的选项也是这样影响的




# 创建一个Weblogic域



BEA Configuration Wizard -

## Choose Domain Type and Name

Select a template:

 WLS Examples	 WLS Petstore	 WLS Domain
--	--	--

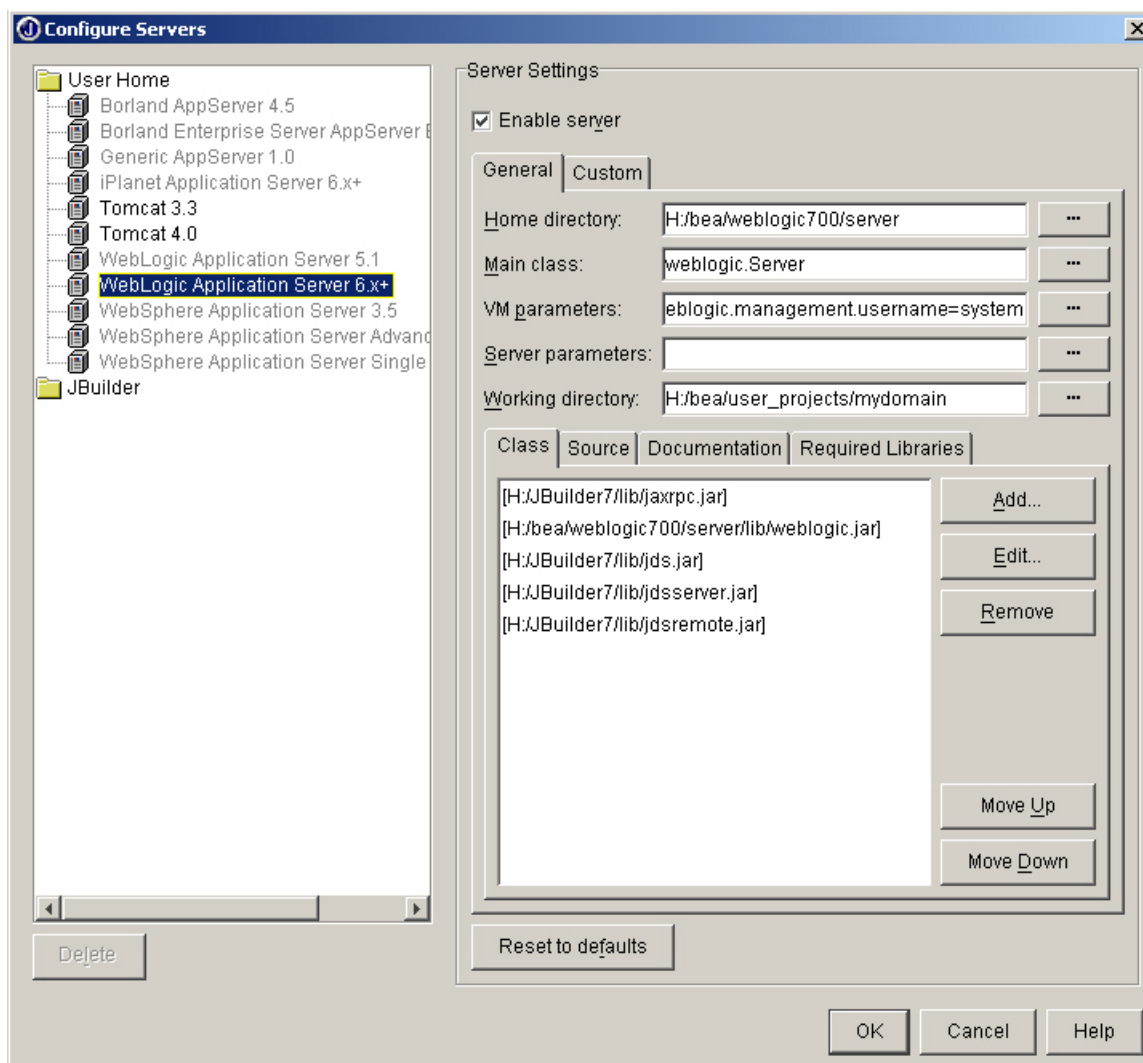
Description:  
Typical WebLogic Domain with no custom applications

Author:  
BEA Systems, Inc.

Name:  
mydomain

Exit Previous Next

# 在Jbuilder中设置Weblogic



# 在Jbuilder设置CORBA



**Edit Configuration**

CORBA configurations consist of information about an ORB vendor's tools and classes. JBuilder uses this to construct command lines for building CORBA source files, for project classpaths and to locate tools. The IDL compiler command and output directory option are pre-defined for VisiBroker and OrbixWeb.

Name for this configuration:

Path for ORB tools:  
 ...

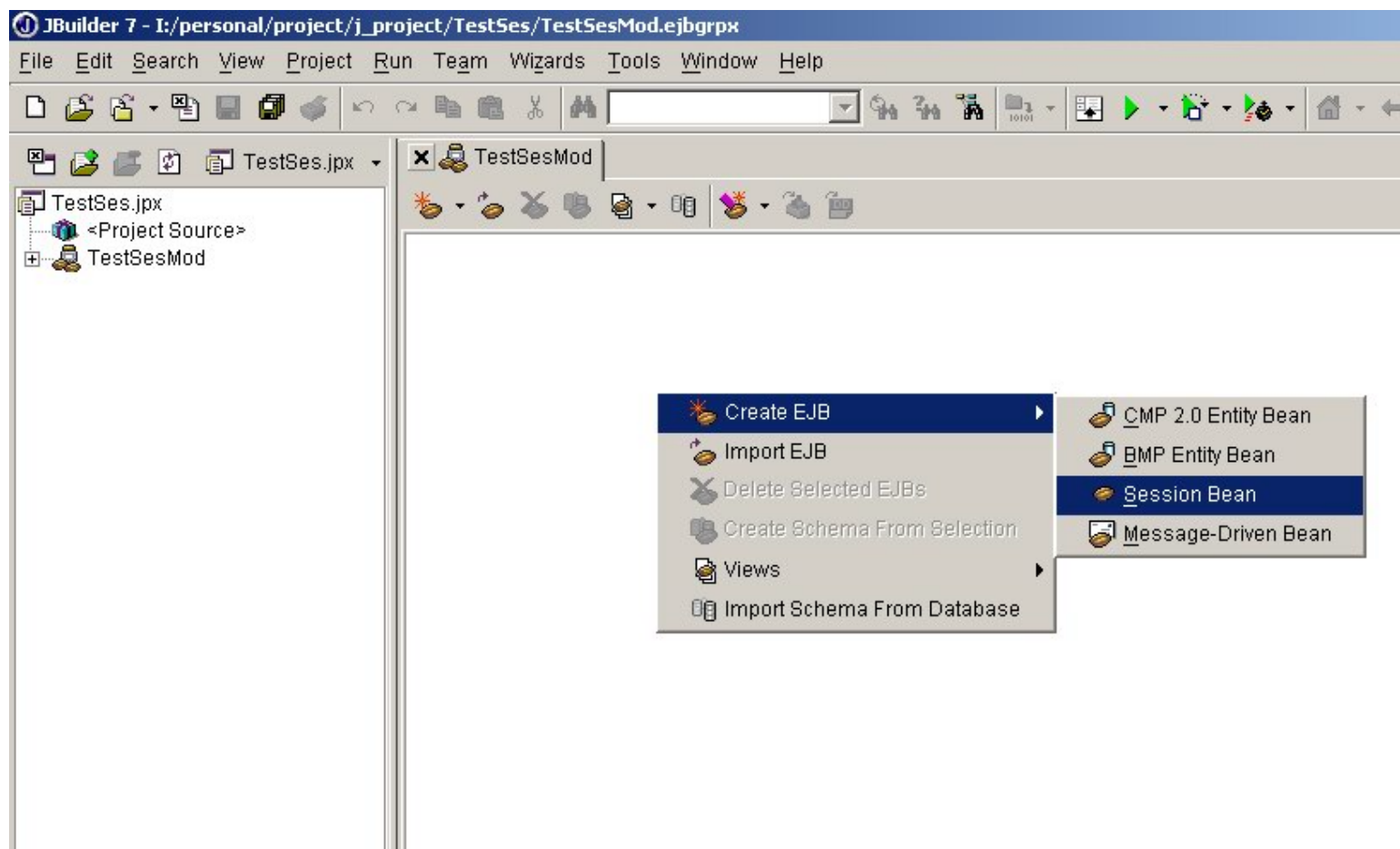
Library for projects:  
 ...

IDL compiler command:

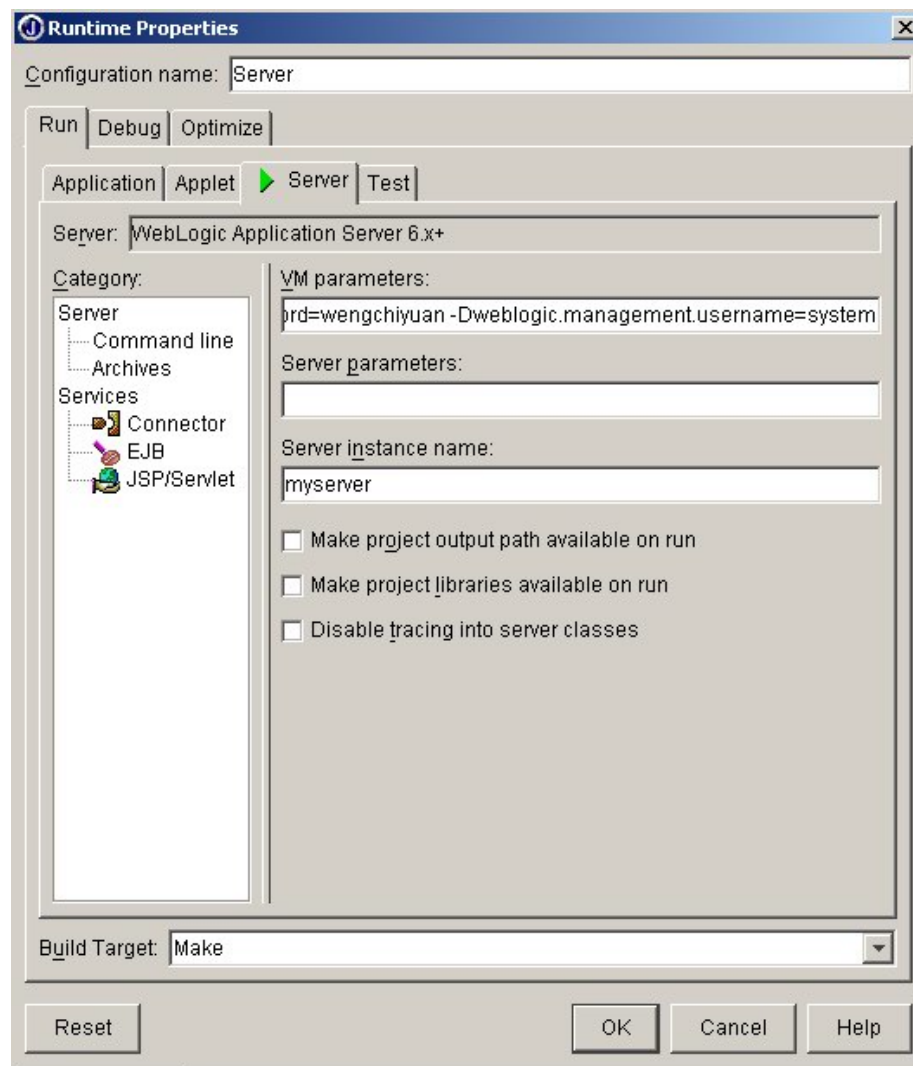
Command option for output directory:

OK Cancel Help

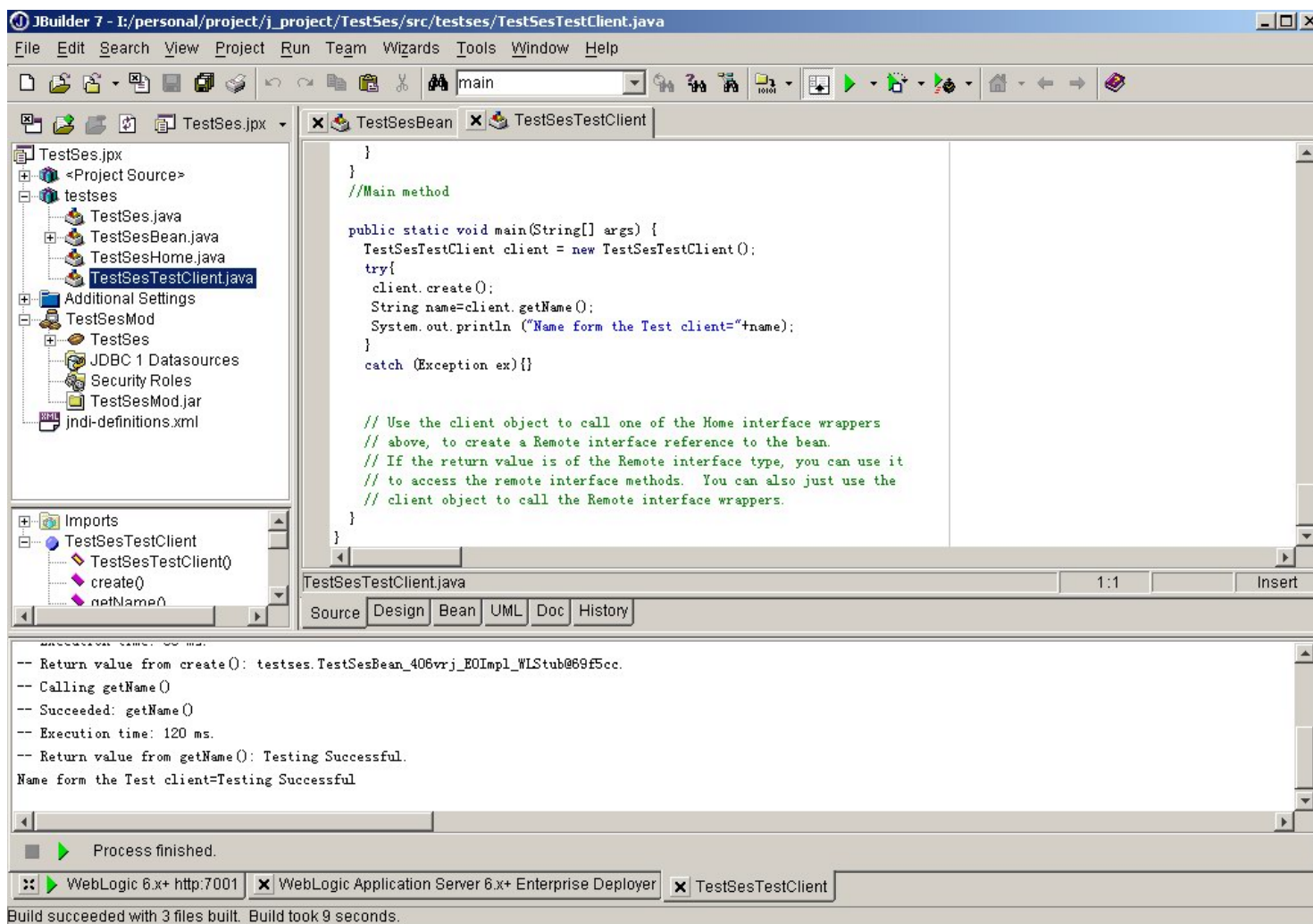
# 创建一个Session Bean



# 创建运行时的设置



# 编写客户端





# 使用JDataStore数据库



JDataStore是Jbuilder提供的一个高性能、小规模、纯java实现的数据库。JdataStore的数据库文件扩展名为jds，在D:\JBuilder9\samples\JDataStore\datastores（我的Jbuilder9安装在C盘）目录中可以找到Jbuilder自带的JdataStore数据库文件，如employee.jds。

要将其在Weblogic中使用，首先请在Weblogic Server（我的Weblogic安装在C盘）的启动文件

D:\bea\weblogic700\samples\server\config\examples\setExample sEnv.cmd文件中添加JDataStore的驱动程序

D:\JBuilder9\lib\jdsremote.jar，所以在CLASSPATH语句中添加：  
set

```
CLASSPATH=%WEBSERVICES_JAR%;%CLIENT_CLASSES%;  
%SERVER_CLASSES%;%SAMPLES_HOME%\server\eval\point  
base\lib\pbserver42ECF183.jar;%SAMPLES_HOME%\server\eva  
l\pointbase\lib\pbclient42ECF183.jar;%WL_HOME%\server\lib\cla  
sses12.zip;D:\JBuilder9\lib\jdsremote.jar;%COMMON_CLASSES  
%
```



# Jbuilder集成Weblogic



在WebLogic的控制面版中设置Connection Pools

Name: jDataStorePool

URL:

jdbc:borland:dslocal:D:/JBui lder9/samples/JDataStore/datas  
tores/employee.jds

Driver Classname:

com.borland.datastore.jdbc.DataStoreDriver

Properties: user=Borland

Password=

Initial Capacity: 5

Maximum Capacity: 10

Test Table name: employee



# Jbuilder集成Weblogic



## 配置数据源DataSource

Name: myDataSource

JNDI Name: myDataSource

Pool Name: jDataStorePool

## 检查是否配置成功

重新启动Weblogic Server，如果没有报错，证明以上所做配置没有问题。此时，启动JdataStore Server，通过监控界面可以看到有5个到employee.jds的连接，这5个连接就是由Weblogic Server连接池建立的（初始容量设置为5）。



# 开发会话EJB



From the JBuilder menu bar, click **File -> New -> Enterprise** tab of Object Gallery -> EJB Module. Hit **OK**.

选择the module name is **TestSesMod** and

the version is **EJB 2.0 compliant**.

Right-click on the **Editor Pane**. Click **Create EJB**. Click **Session Bean**.

the name of the session bean is **TestSes**.

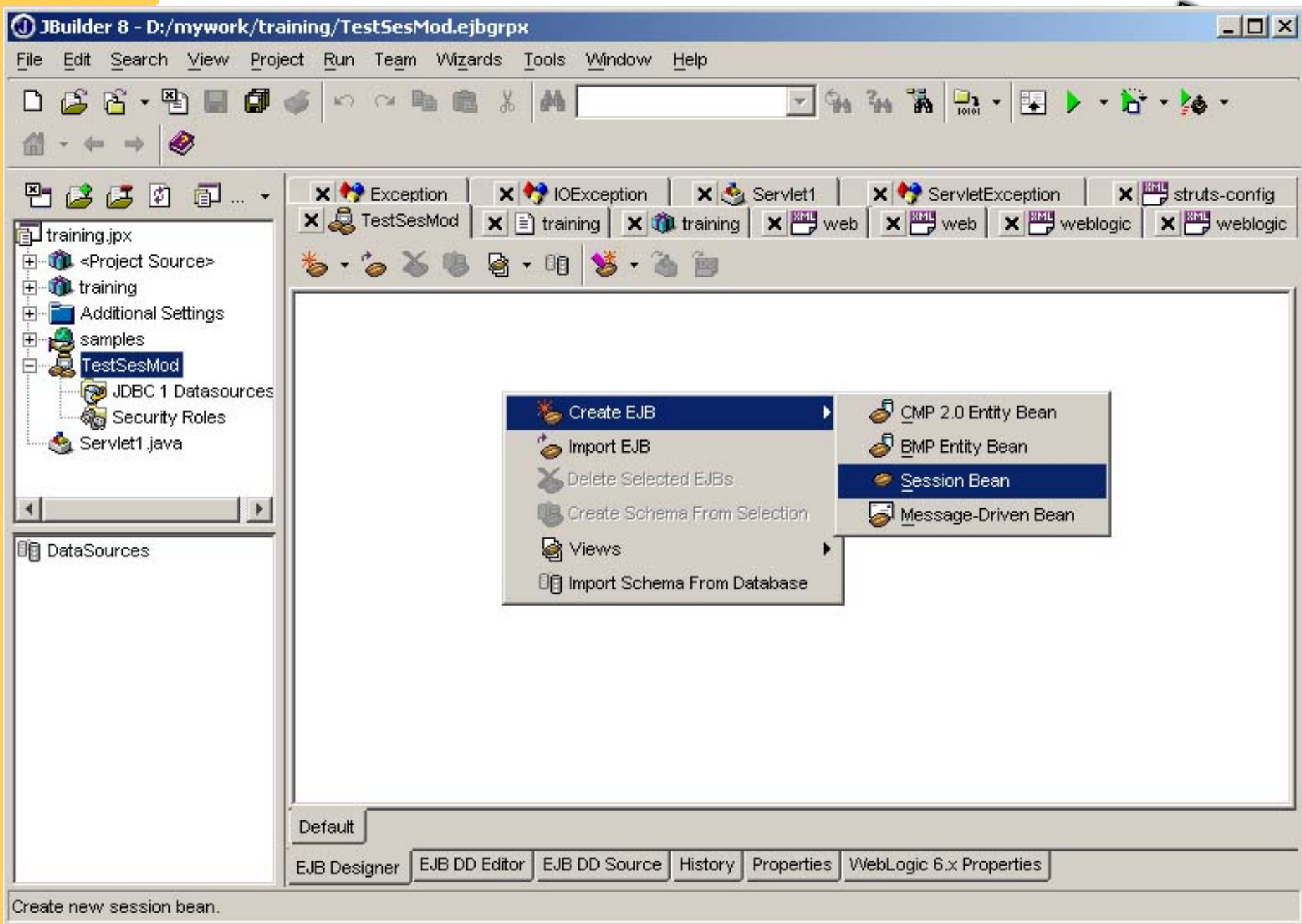
加入方法:

the method name is **getName()**, return type as **string** and interfaces as **remote**.

**TestSesBean.java**

```
public String getName() {  
    return "Testing Successful"; //change null to Testing Successful  
}
```

编译项目,生成TestSesMod.jar



# Create a test client



click **File -> New -> Enterprise tab of Object Gallery -> EJB Test Client**.  
Hit **OK**.

**名字:TestSesTestClient 添加以下内容**

```
public static void main(String[] args) {  
    TestSesTestClient client = new TestSesTestClient();  
    try {  
        client.create();  
        String name = client.getName();  
        System.out.println("Name from the Test Client = " +  
            name);  
    }  
    catch (Exception ex) { }  
}
```



# 开发实体EJB



From the JBuilder menu bar, click **File -> New -> Enterprise** tab of Object Gallery -> EJB Entity Module. Hit **OK**.

选择the module name is **TestEntMod** and

the version is **EJB 1.X compliant**.

Right-click on the **Editor Pane**. Click **Create EJB**. Click **Session Bean**.

the name of the session bean is **TestSes**.

加入方法:

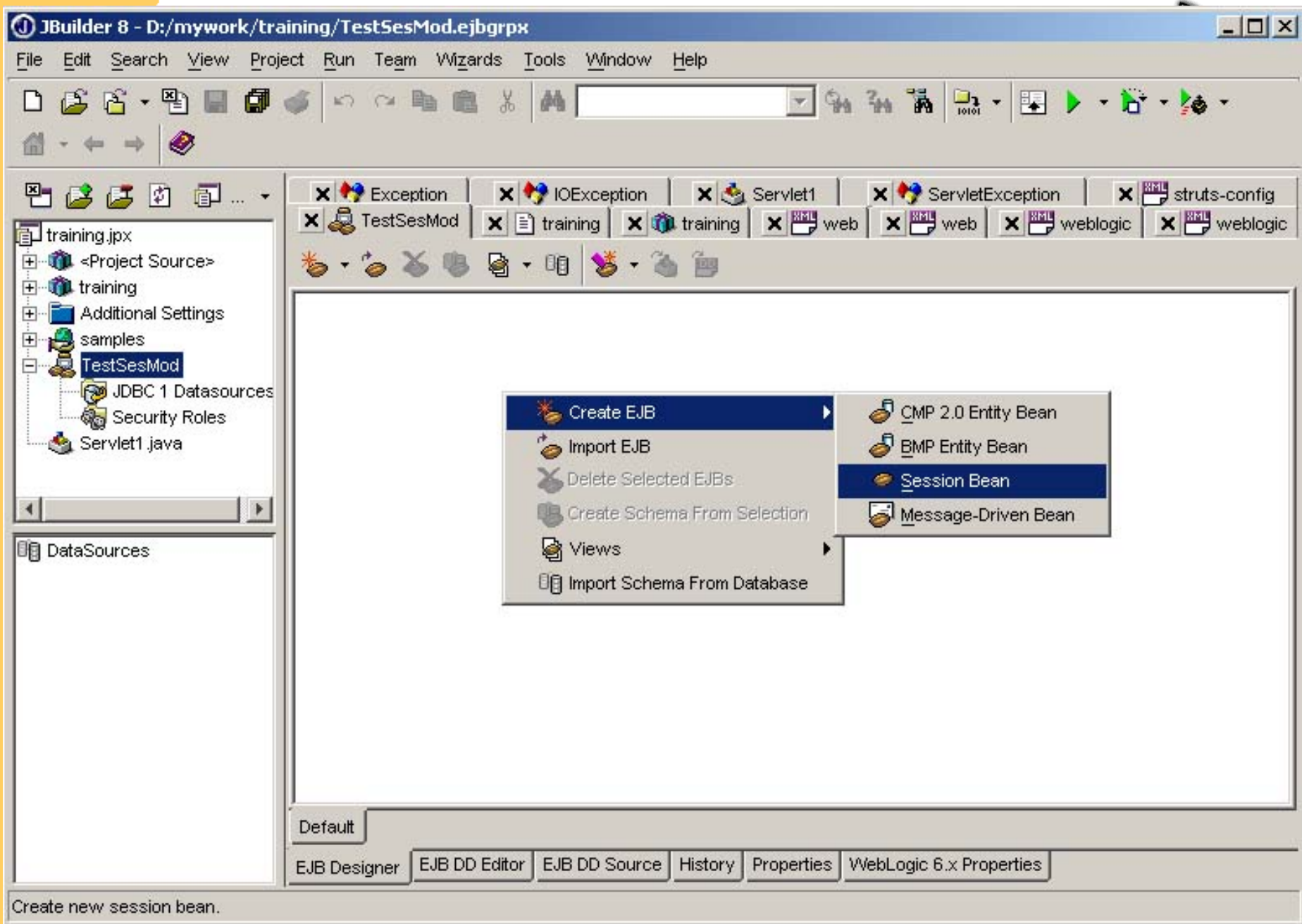
the method name is **getName()**, return type as **string** and interfaces as **remote**.

**TestSesBean.java**

```
public String getName() {  
    return "Testing Successful"; //change null to "Testing  
    Successful"  
}
```

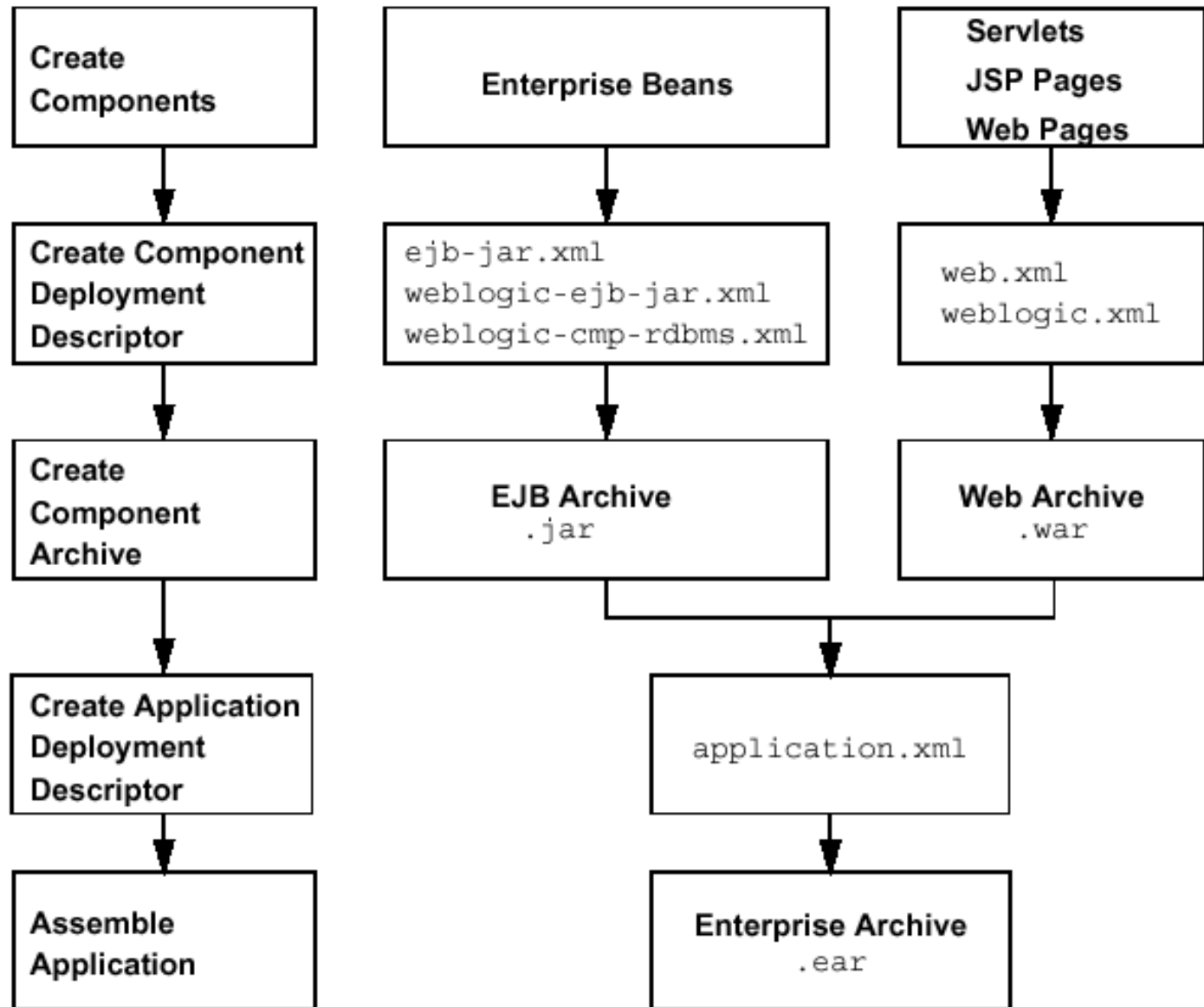
编译项目,生成TestSesMod.jar







# 创建企业应用 EAR





# Ejb-jar文件



- 。 Enterprise Bean
- 。 Remote interface
- 。 Home interface
- 。 Deployment descriptor

生成ejb-jar文件: `jar cmf HelloWorld.jar *`



# EJB部署

将ejb-jar文件部署到应用服务器中，使它能被客户端访问



- 。 核对Ejb-jar中文件
- 。 Ejb容器提供的工具生成EJB对象和Home对象
- 。 EJB容器提供的工具生成必需的RMI存根(stub)和骨架(skeletons)





- JBuilder介绍
  - 。 JBuilder简介
  - 。 JBuilder8功能一览
- JBuilder与Weblogic的集成
  - 。 设置
  - 。 开发EJB

# 总结



- JBuilder是领先的Java解决方案
- Jbuilder提供的软件开发整个生命周期管理

# 参考资料



- <http://www.borland.com>  
borland公司站点
- <http://www.huihoo.com>  
国内一个关于中间件的专业站点



# 结束



## 谢谢大家！

[Allen@huihoo.com](mailto:Allen@huihoo.com)

<http://www.huihoo.com>

