

JBoss IDE 开发 EJB3.0 Message Driven Bean

罗代均 ldj_work#126.com 2007 年 8 月

一. 准备开发环境

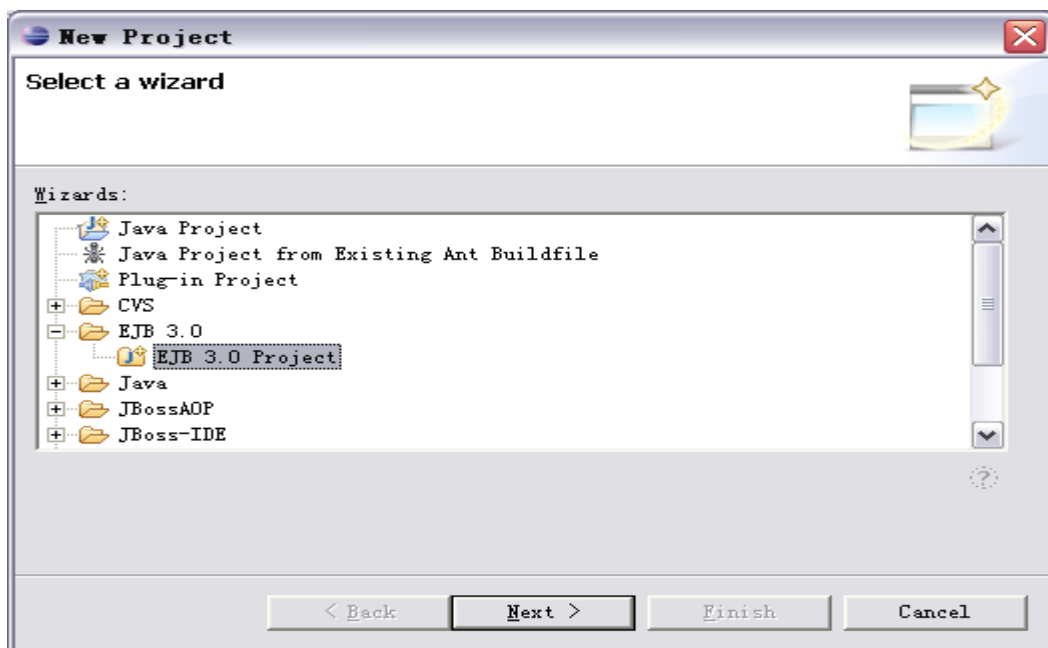
JBoss IDE 1.5GA, JBoss IDE 是 JBoss 公司对 Eclipse 的增强, 包括 EJB3.0, JBPM, JBoss AOP, FreeMaker 等等插件, 对于 JBoss 产品的开发, 是非常方便的。可以去 Jboss.org 网站下载。

Eclipse WTP 插件, 这是开发 Java EE 项目插件, 这里用来开发一个 web 项目, 来测试 Message Driven Bean。

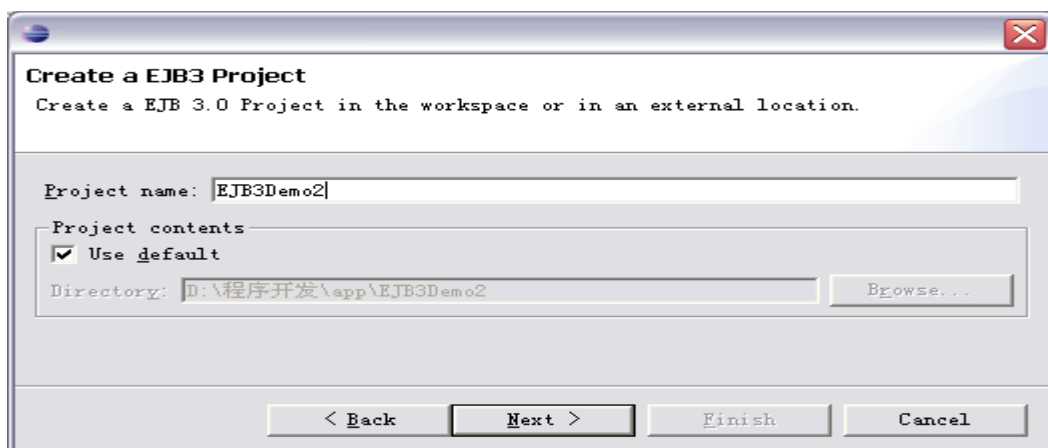
JBoss 4.04GA, 可以从 Jboss.org 网站下载, 安装版, zip 版都可以。

二. 新建 EJB3.0 工程

选择菜单 New | Project..., 在弹出对话框中, 选择 EJB 3.0 Project



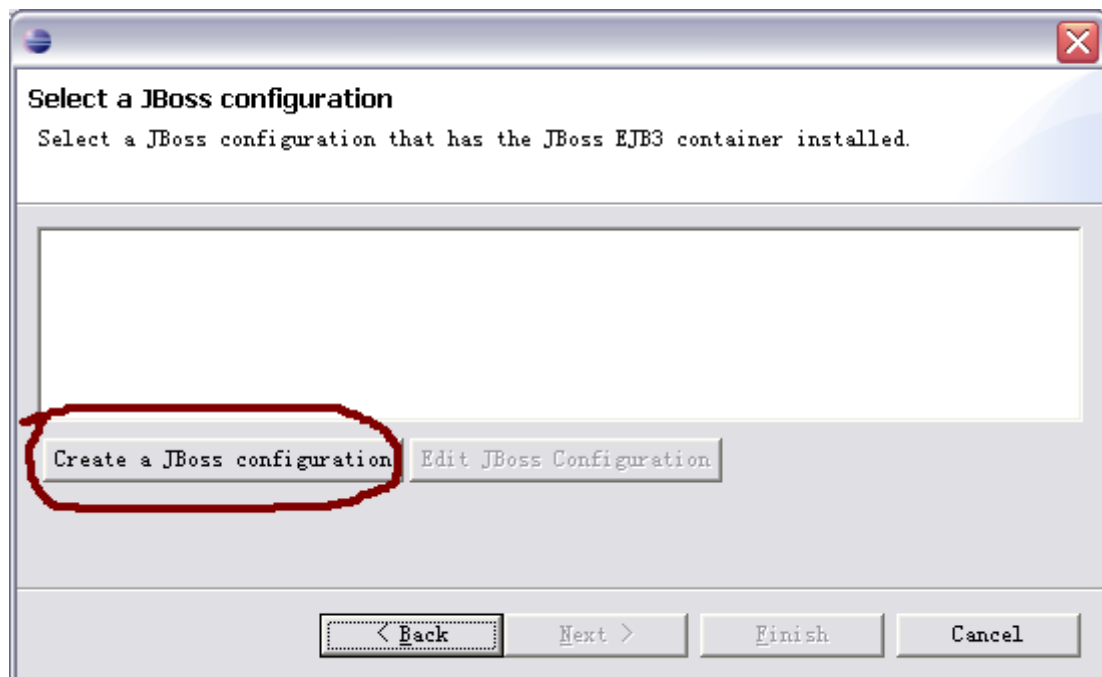
单击[Next]按钮, 填写工程名称 EJB3Demo2



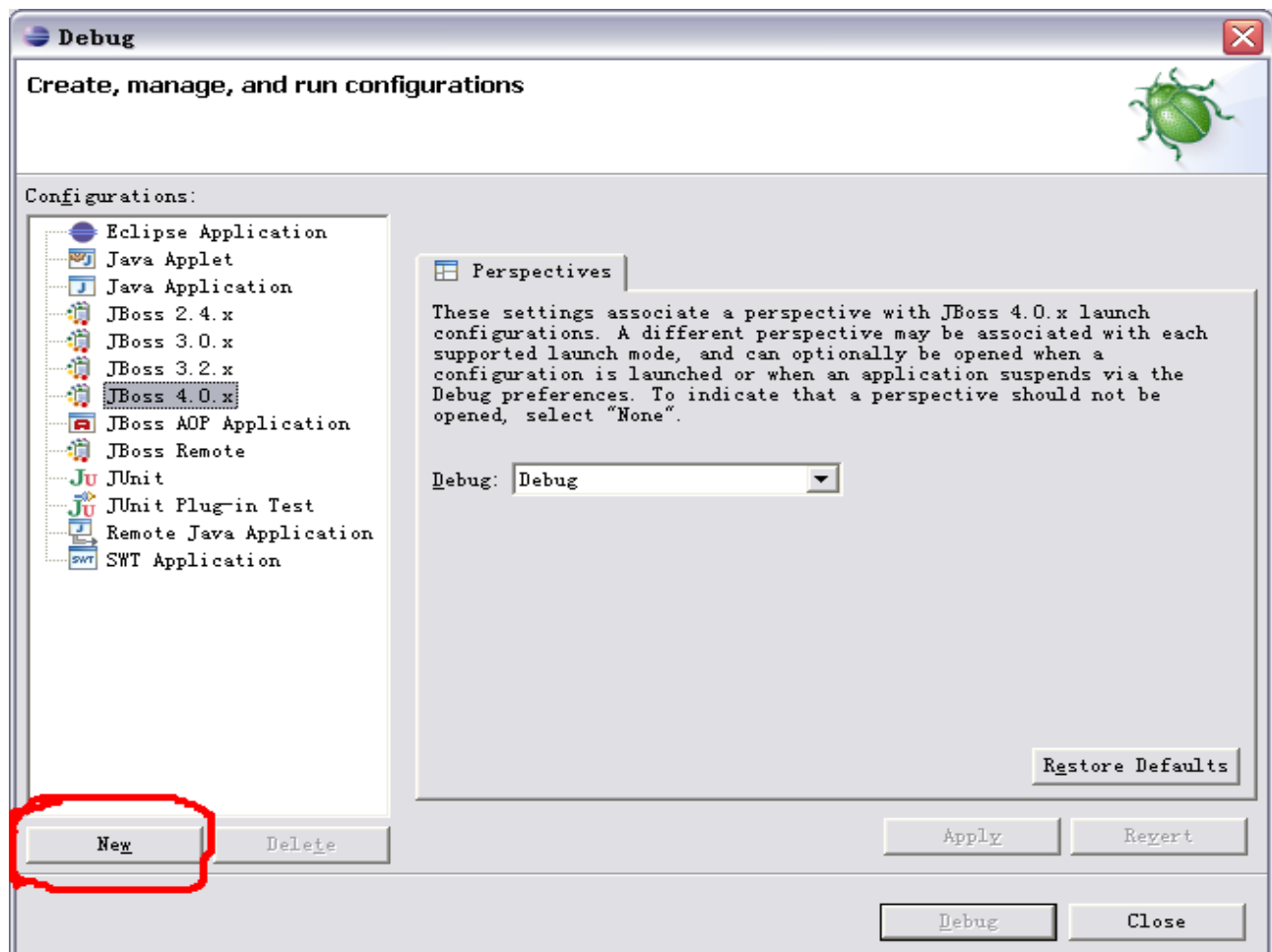
EJB3.0 入门图文教程---开发 Message Driven Bean

单击[Next]按钮, 选择 JBoss 配置

我们这里还没有配置, 因此选择[create a jboss configuration]按钮, 新建 JBoss 配置.

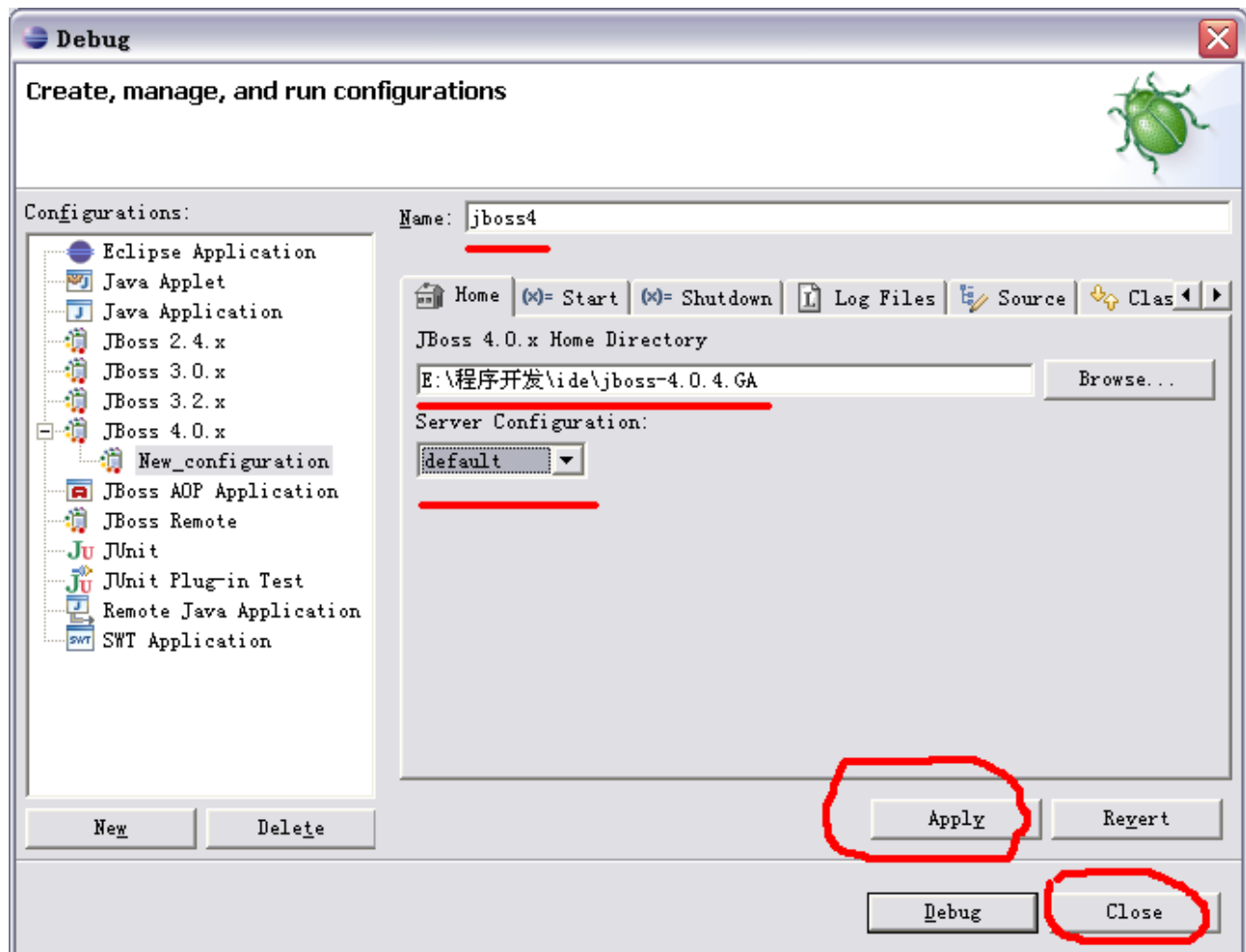


选择 JBoss4.0.x, 并单击[New]按钮

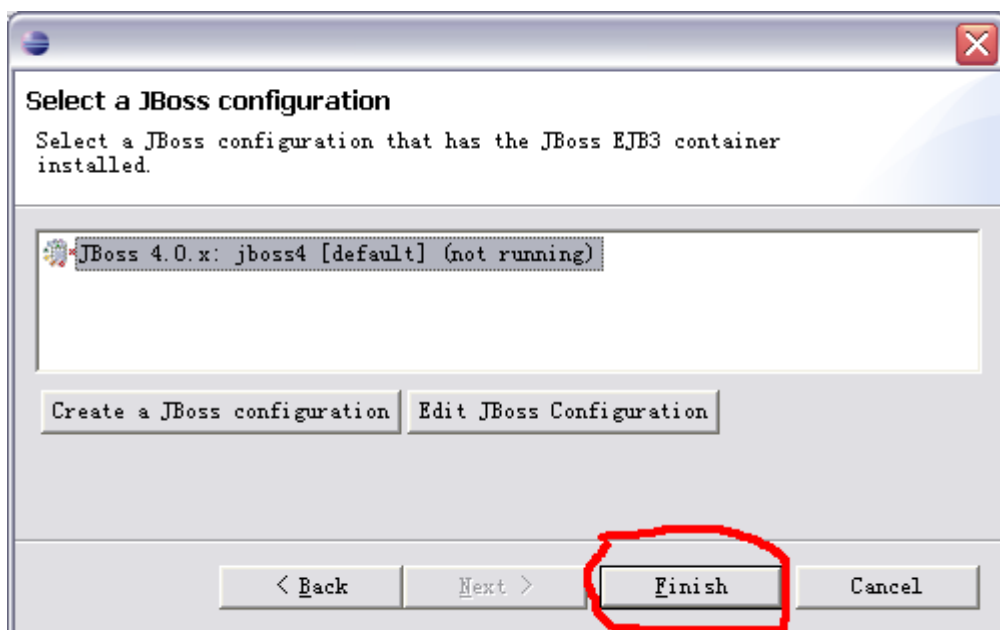


EJB3.0 入门图文教程---开发 Message Driven Bean

按下图 输入配置名称, JBoss 安装目录, Server Configuration 选择 default



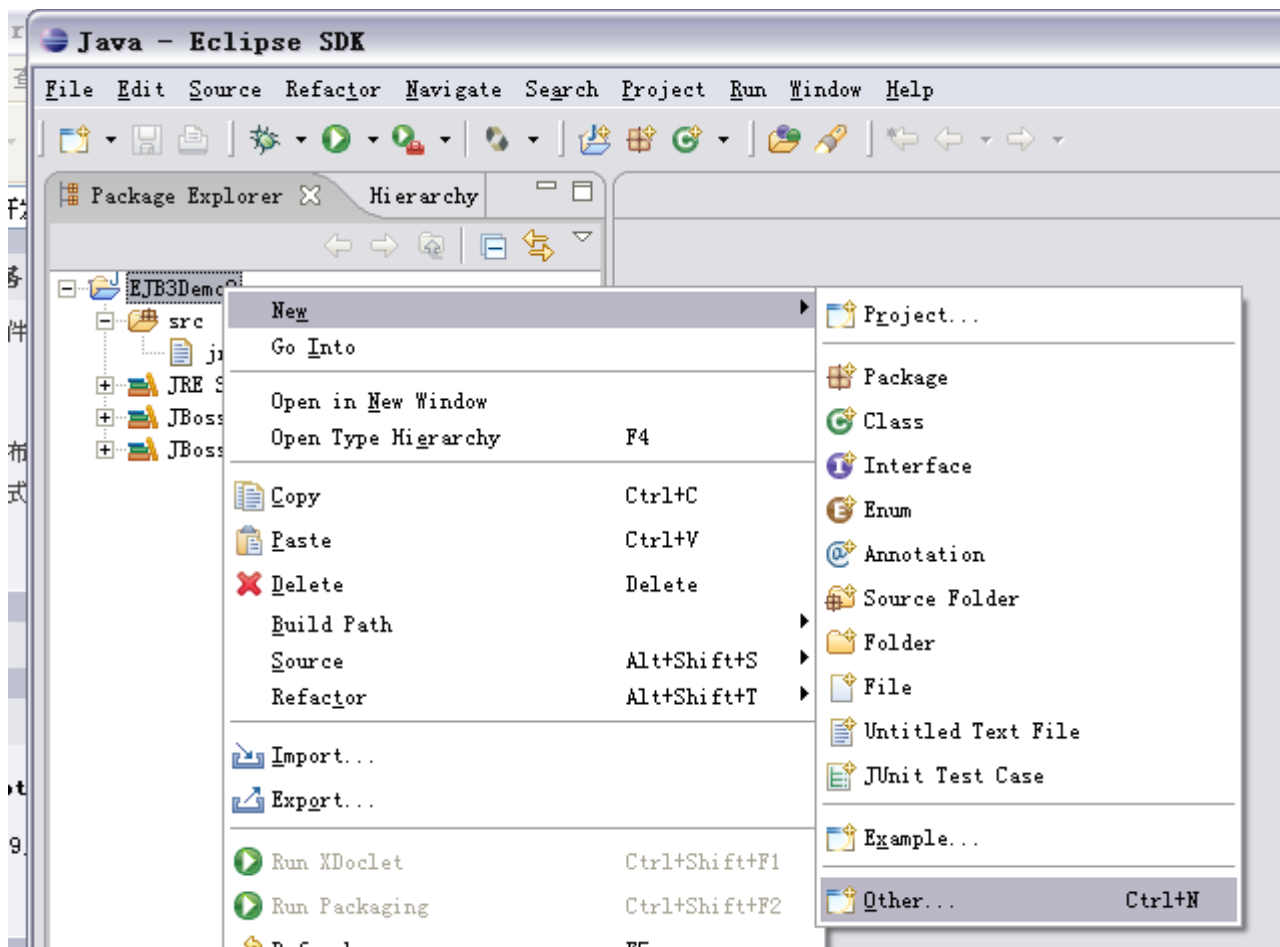
单击[Apply]按钮, 然后单击[Close]按钮



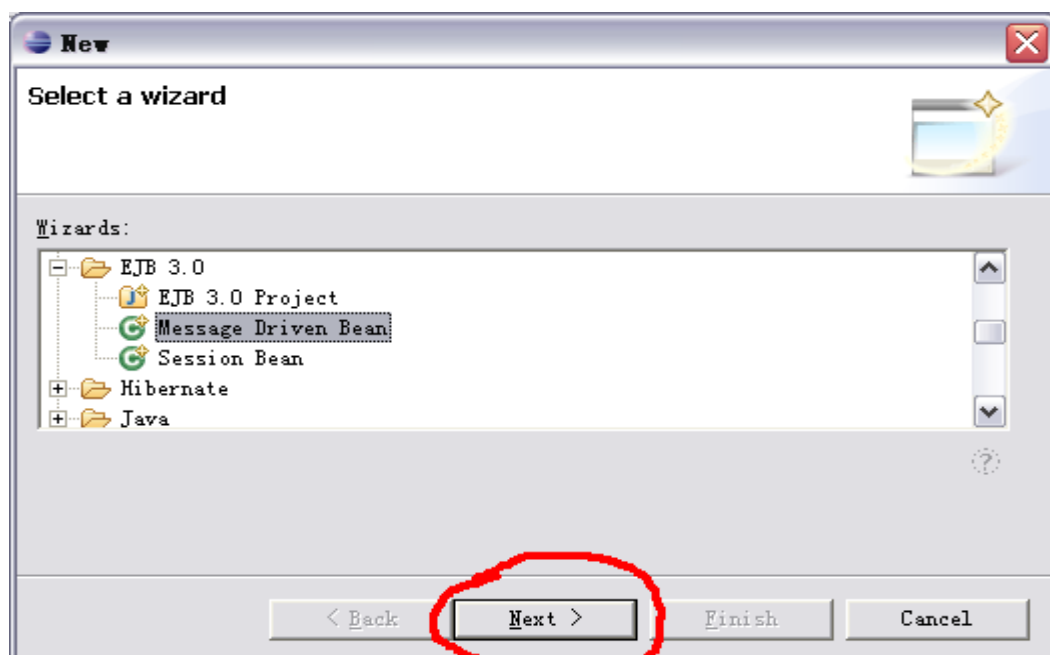
单击[Finish]按钮, 完成.

三. 新建 Message Driven Bean

单击右键，选择 New | Other...



在弹出对话框中，选择 EJB3.0 | Message Driven Bean



单击[Next]按钮，输入包名，类名等

The screenshot shows a 'New' dialog box with the following fields and options:

- Source folder: EJB3Demo2/src
- Enclosing type: (empty)
- Package: demo
- Name: HelloMDB
- Modifiers: ☒ public, ☐ default, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass: java.lang.Object
- Interfaces: javax.jms.MessageListener
- Buttons: < Back, Next >, Finish, Cancel

单击[Finish]按钮完成。

四. 完成 Message Driven Bean 的代码

4.1 设置 *ConnectionFactory* 和 *Queue*

ConnectionFactory 使用 JBoss 默认的 ConnectionFactory, 因此不用设置。

设置 Queue 方式如下:

找到 jbossmq-destinations-service.xml 文件, 该文件在 jboss 安装目录 \server\default\deploy\jms 下. 在文件最后, 加入下面代码, 新增一个 Queue.

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=myQueue">
  <depends optional-attribute-name="DestinationManager">
    jboss.mq:service=DestinationManager
  </depends>
</mbean>
```

4.2 完成代码

设置好 Queue 后，编辑 HelloMDB.java，加入 Connection 和 Queue 说明。

```
package demo;

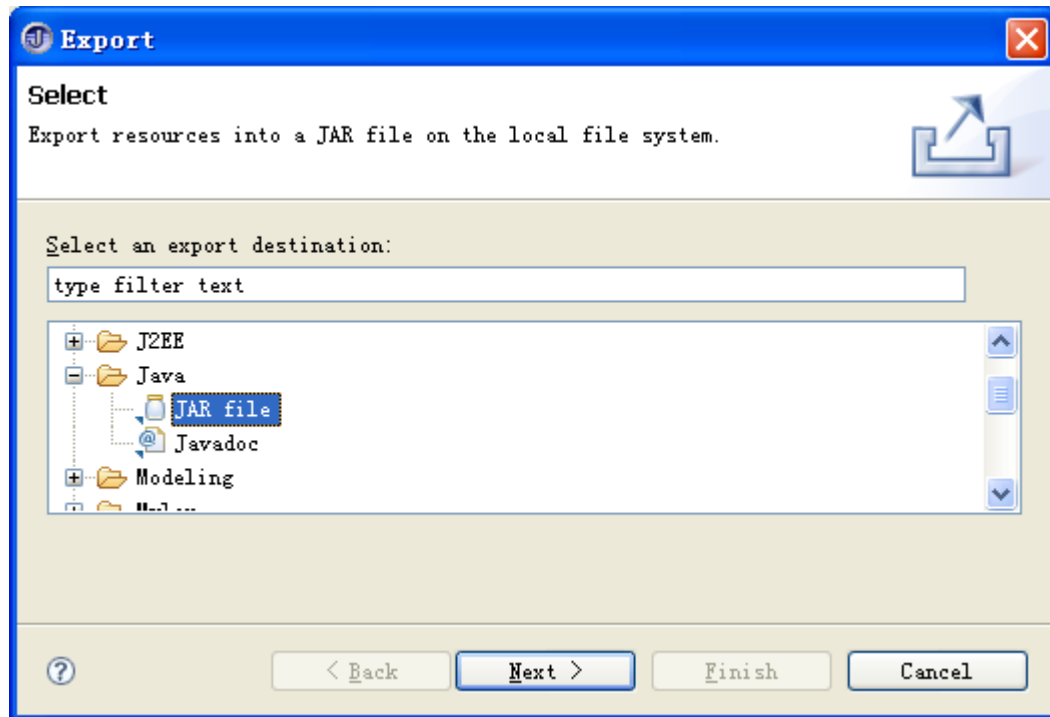
import javax.jms.JMSEException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;
import javax.ejb.ActivationConfigProperty;
import javax.ejb.MessageDriven;

@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType",
                                propertyValue = "javax.jms.Queue"),
    @ActivationConfigProperty(propertyName = "destination",
                                propertyValue = "queue/myQueue") })
public class HelloMDB implements MessageListener {
    public void onMessage(Message msg) {
        if (msg instanceof TextMessage) {
            TextMessage txtMsg = (TextMessage) msg;
            try {
                System.out.println("Hello " + txtMsg.getText() + ", MDB 欢迎您!");
            } catch (JMSEException e) {
                e.printStackTrace();
            }
        }
    }
}
```

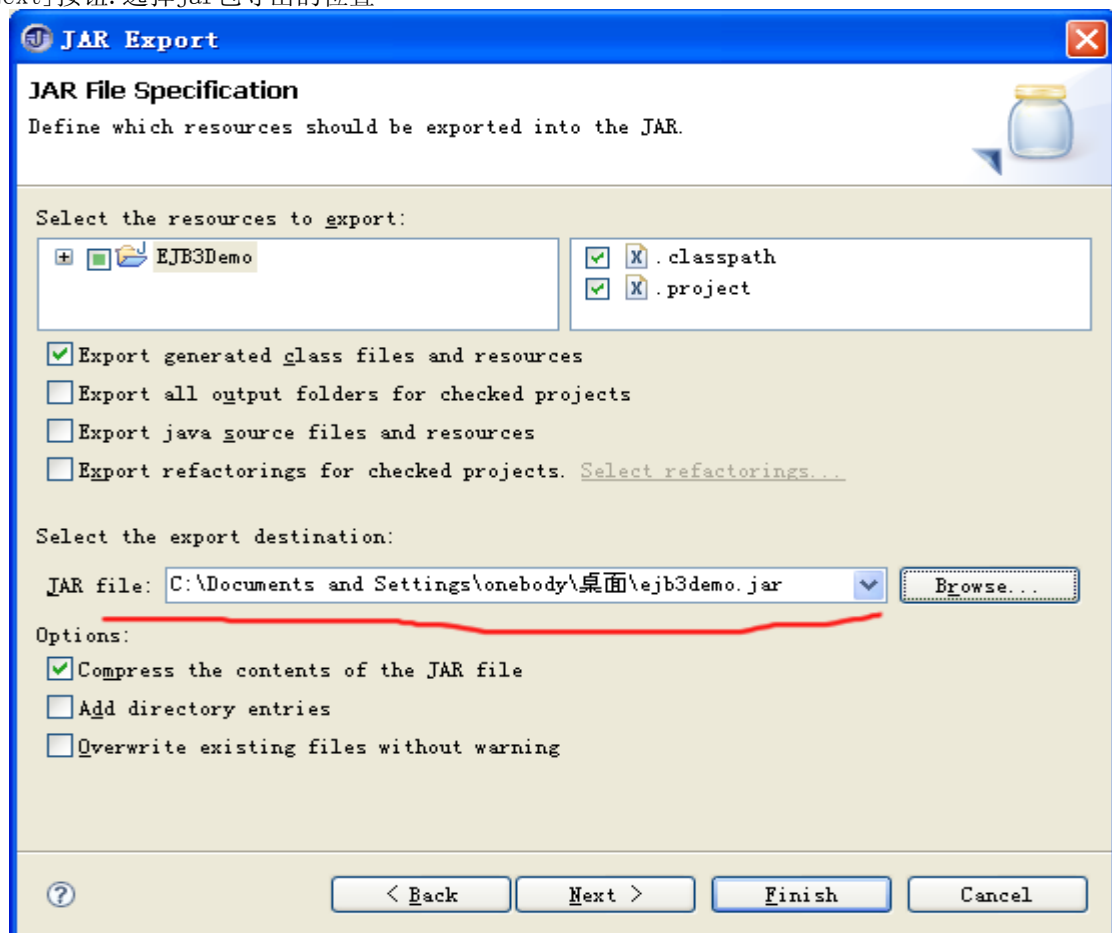
4.3 导出 jar 包

选择菜单 File | Export, 弹出对话框选择 JAR File

EJB3.0 入门图文教程---开发 Message Driven Bean



单击[Next]按钮, 选择jar包导出的位置



单击[Finish]按钮, 完成导出

五. MDB 的部署与测试

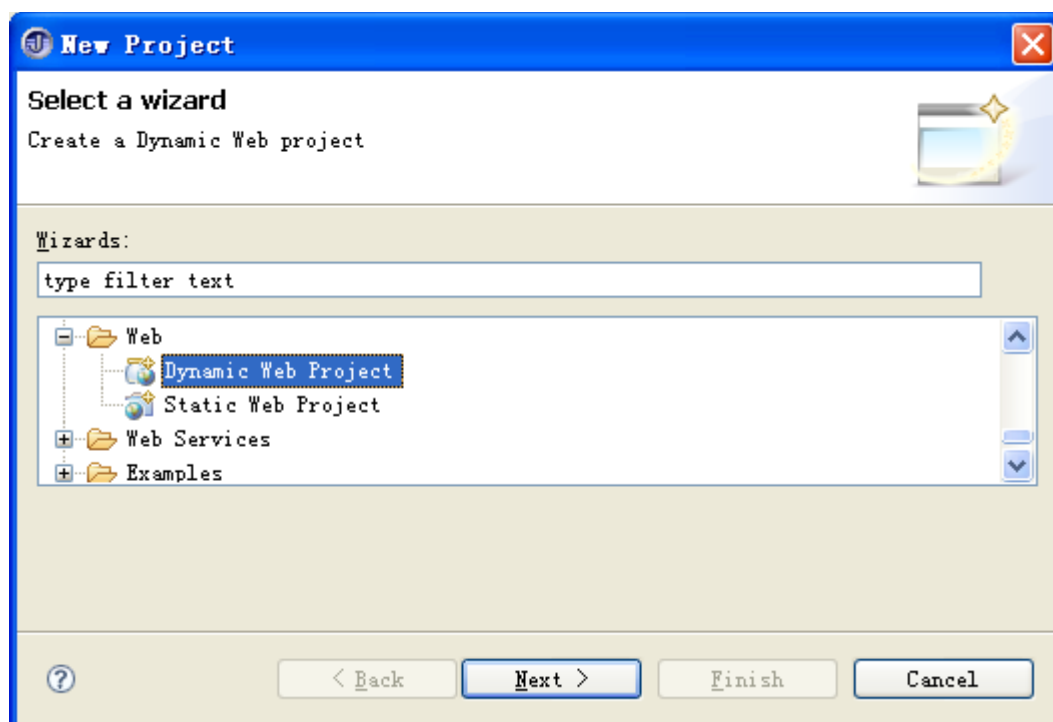
5.1 部署 EJB

把我们刚才导出的 jar 文件. 放到 JBoss 安装目录下的 server\default\deploy 下
ok, ejb 部署好了. 下面我们写一个 servlet 测试一下

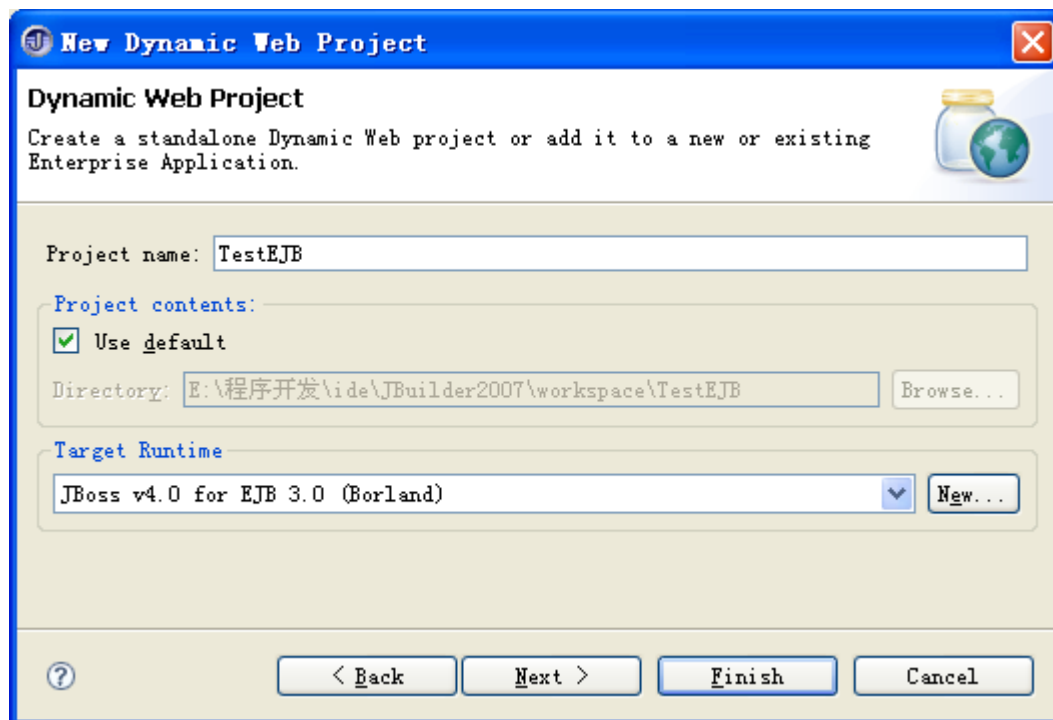
5.2 测试 EJB

5.2.1 新建动态 web 项目

选择菜单 File | new | Project , 在弹出对话框中选择 web | Dynamic Web Project



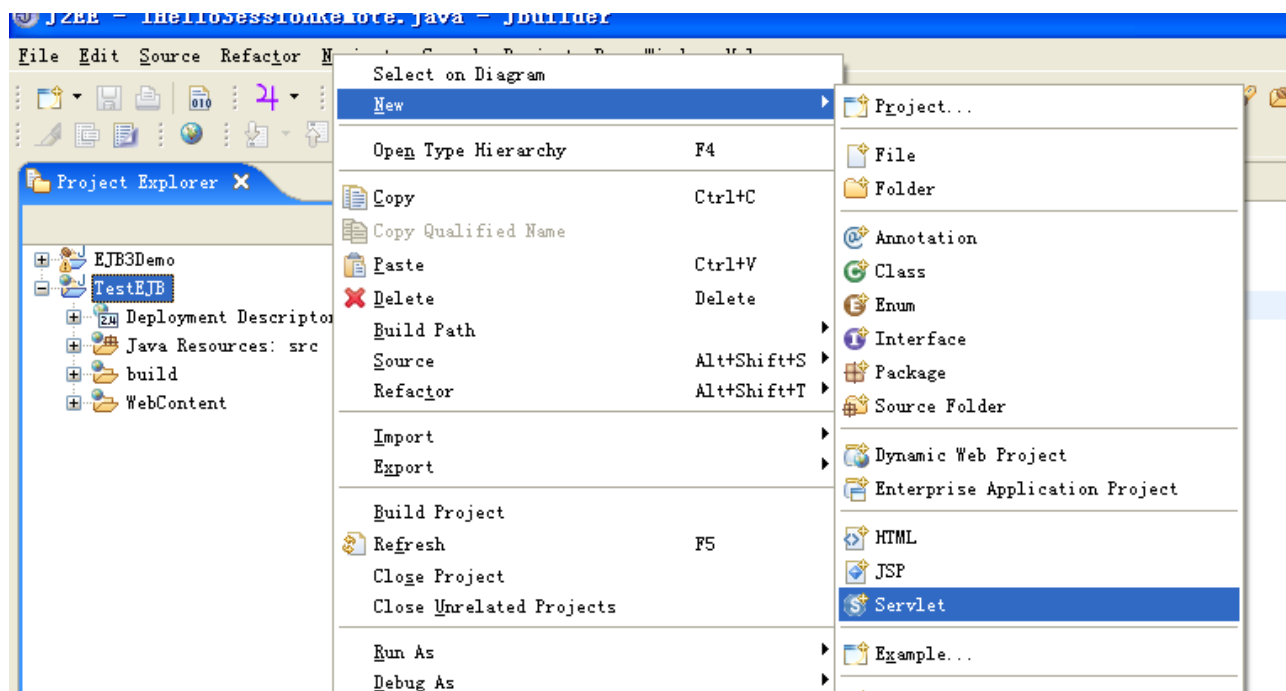
单击[Next]按钮, 填写工程名称



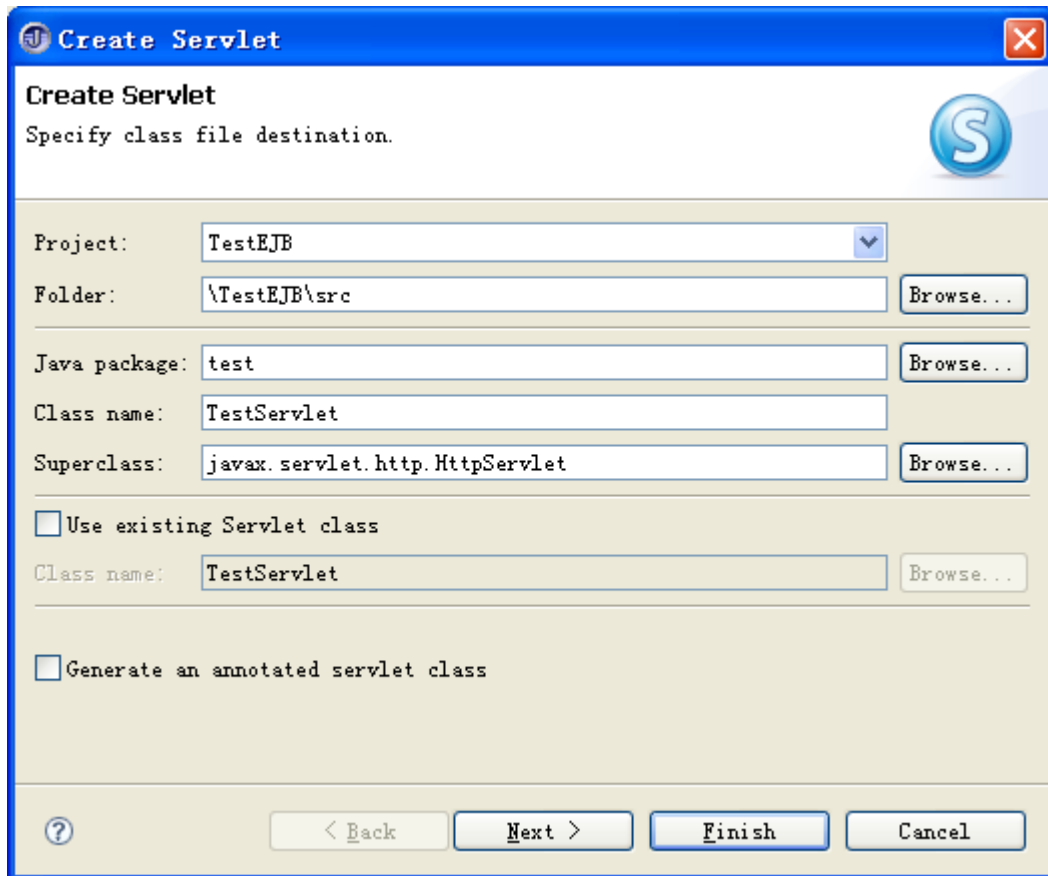
单击 [Finish] 按钮, 完成.

5.2.2 新建 Servlet

单击右键, 选择 New | Servlet



弹出对话框中, 按下图填写



单击[Finish]按钮完成

5.2.3 在 TestServlet 的 doGet 方法中, 编写测试 EJB 的代码

TestServlet.java 内容如下

```
package test;

import java.io.IOException;

import javax.jms.JMSException;
import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class TestServlet extends javax.servlet.http.HttpServlet implements
    javax.servlet.Servlet {
    private static final long serialVersionUID = 1L;
```

```

protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    try {
        //得到上下文
        Context ctx=new InitialContext();
        //得到ConnectionFactory
        QueueConnectionFactory connFacotry=
            (QueueConnectionFactory)ctx.lookup("ConnectionFactory");
        //得到消息队列
        Queue queue=(Queue)ctx.lookup("queue/myQueue");
        QueueConnection queueConn=connFacotry.createQueueConnection();
        QueueSession queueSession =
            queueConn.createQueueSession(false, Session.AUTO_ACKNOWLEDGE);
        QueueSender queueSender =queueSession.createSender(queue);
        //组装消息
        TextMessage txtMsg=queueSession.createTextMessage();
        txtMsg.setText("Luo Daijun");
        queueSender.send(txtMsg);
        //关闭资源
        queueSender.close();
        queueSession.close();
        queueConn.close();

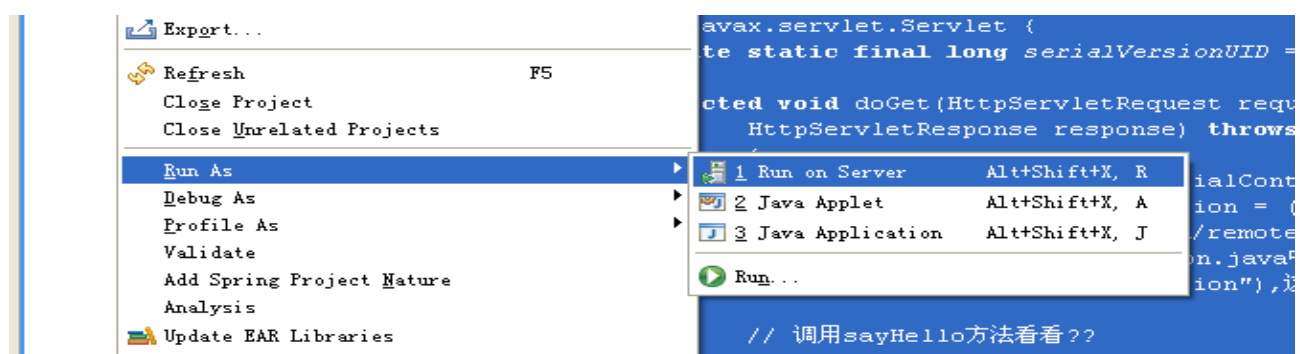
    } catch (NamingException e) {
        e.printStackTrace();
    } catch (JMSException e) {
        e.printStackTrace();
    }
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
}
}

```

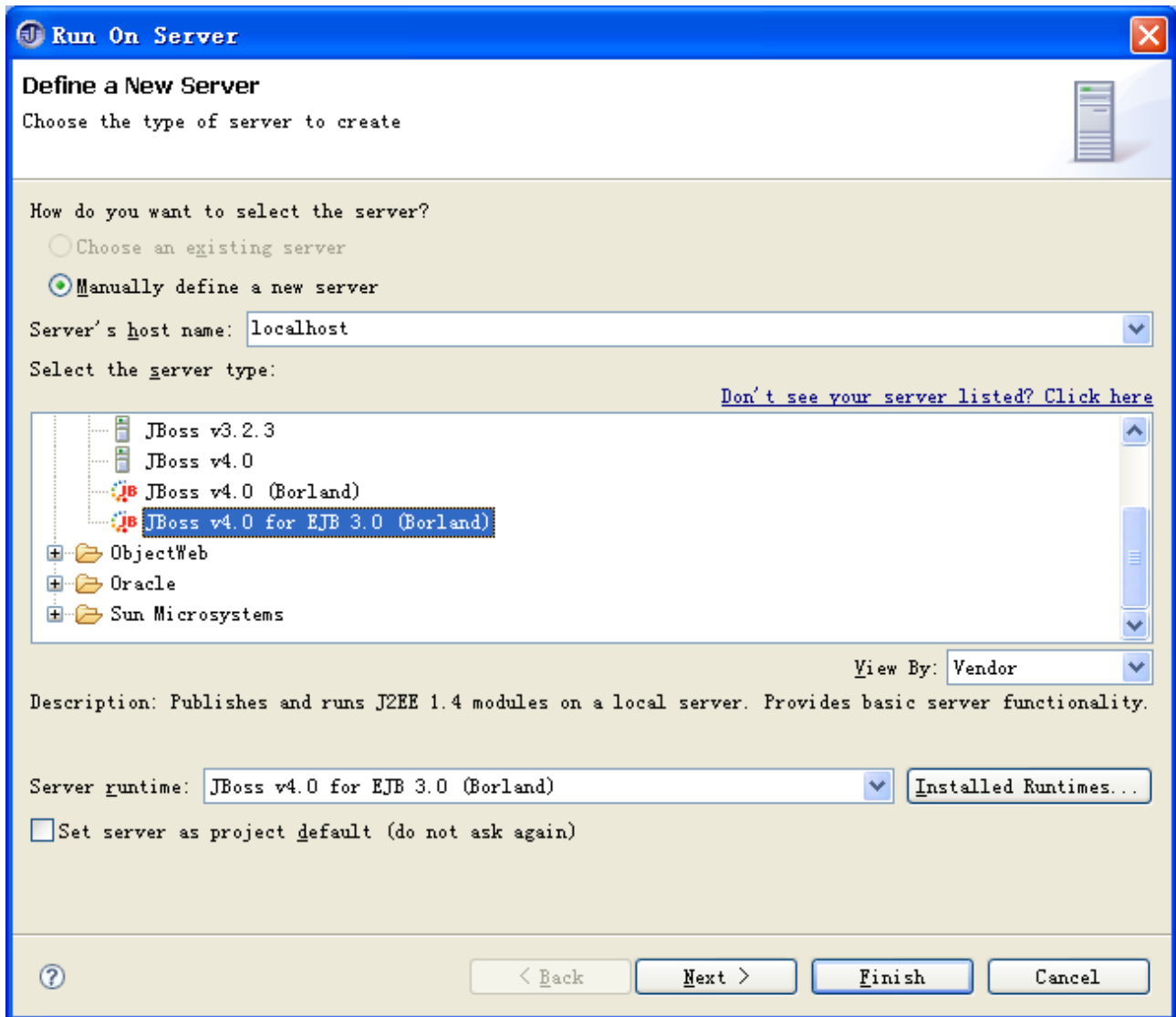
5.2.4 运行 TestServlet, 测试效果 ^_^

右键, 选择 Run As | Run on server



选择 JBoss 4.0 for EJB3.0

EJB3.0 入门图文教程---开发 Message Driven Bean



单击[Finish]按钮, 完成

打开 IE 浏览器, 输入 <http://localhost:8080/TestEJB/TestServlet>

嗯..看看 JBuilder 的控制台吧

```
01:23:51,750 INFO [TomcatDeployer] deploy, ctxPath=/testEJB,
01:23:51,937 INFO [TomcatDeployer] deploy, ctxPath=/jmx-conso
01:23:52,156 INFO [Http11BaseProtocol] Starting Coyote HTTP/1
01:23:52,750 INFO [ChannelSocket] JK: ajp13 listening on /0.0
01:23:52,781 INFO [JkMain] Jk running ID=0 time=0/94 config=
01:23:52,796 INFO [Server] JBoss (MX MicroKernel) [4.0.5.GA (
01:24:04,484 INFO [STDOUT] Hello Luo Daijun, MDB欢迎您!
```

嗯. 结束, 打完收工!!