

第二十四章 Tomcat与Eclipse

Eclipse是一个非常优秀的集成开发环境，使用它可以很容易的开发Java程序。在本章中介绍如何使用Eclipse结合Tomcat进行Web开发。

24.1 Eclipse简介与安装配置

Eclipse是替代IBM Visual Age for Java(以下简称IVJ)的下一代集成开发环境(Integrated Development Environment, 以下简称IDE)，但它未来的目标不仅仅是成为专门开发Java程序的IDE环境，根据Eclipse的体系结构，通过开发插件，它能扩展到任何语言的开发，甚至能成为图片绘制的工具。

目前，Eclipse已经开始提供C语言开发的功能插件。而且Eclipse是一个开放源代码的项目，任何人都可以下载Eclipse的源代码，并且在此基础上开发自己的功能插件。也就是说未来只要需要，就会有建立在Eclipse之上的COBOL、Perl、Python等语言的开发插件出现。同时可以通过开发新的插件扩展现有插件的功能，比如在现有的Java开发环境中加入Tomcat服务器插件。可以无限扩展，而且有着统一的外观，操作和系统资源管理，这也正是Eclipse的潜力所在。

24.1.1 安装Eclipse

一般Eclipse提供几个下载版本：Release, Stable Build, Integration Build和Nightly Build, 建议下载Release或Stable版本。

Eclipse可以从Eclipse官方网站(<http://www.eclipse.org>)免费下载，也可以在本书的随书光盘中找到，作者使用的是eclipse-SDK-3.0.1。

Eclipse本身是用Java语言编写，但下载的压缩包中并不包含Java运行环境，需要用户自己另行安装JRE，并且要在操作系统的环境变量中指明JRE中bin的路径。

安装Eclipse的步骤非常简单。只需将下载的压缩包直接解压即可，假定解压的目录是Eclipse，打开这个目录你会发现如下文件或目录：

- /plugins: 存放插件的目录。
- /features: 存放功能部件的目录。
- /links: 其他plugins和features的连接地址的存放目录。
- /readme
- eclipse.exe: 启动Eclipse程序。
- ...

注意：如果有了更新的版本，要先删除老的版本重新安装，不能直接解压到原来的路径覆盖老版本。

在解压缩之后可以到相应的安装路径去找Eclipse.exe运行。如果下载的是Release或Stable版本，并且JRE环境安装正确无误，一般来说不会有什么问题，Eclipse会显示它的欢迎界面，如图24.1。

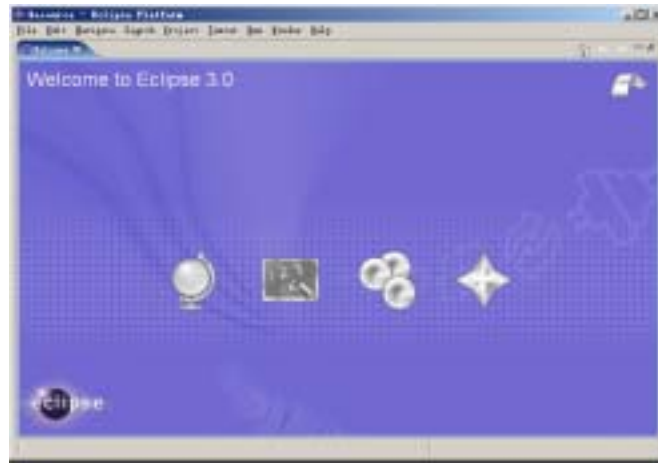


图24.1 Eclipse的欢迎界面

关闭欢迎页面后，可以看到Eclipse的默认界面（资源透视图），如图24.2所示。

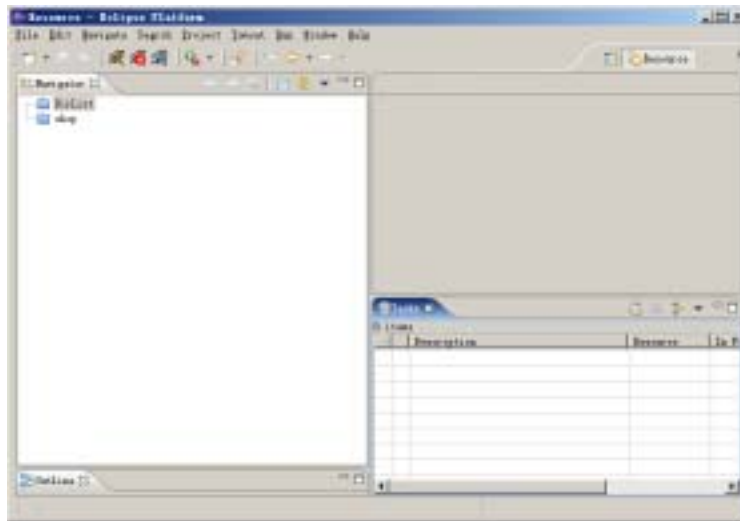


图24.2 Eclipse的默认界面（资源透视图）

24.1.2 Eclipse的多国语言包

Eclipse支持多国语言，只要下载与SDK相应的多国语言包插件就可以实现软件的本地化，多国语言包插件的安装和其他Eclipse插件的安装一样简单，可以按照如下步骤进行。

- (1) 首先把多国语言包插件压缩包解压到本地硬盘。
- (2) 然后把解压目录下features和plugins文件夹中的所有文件夹拷到对应的Eclipse目录中的features和plugins文件夹下。
- (3) 重新启动Eclipse，可以看到Eclipse的界面变成了熟悉的中文环境，如图24.3。

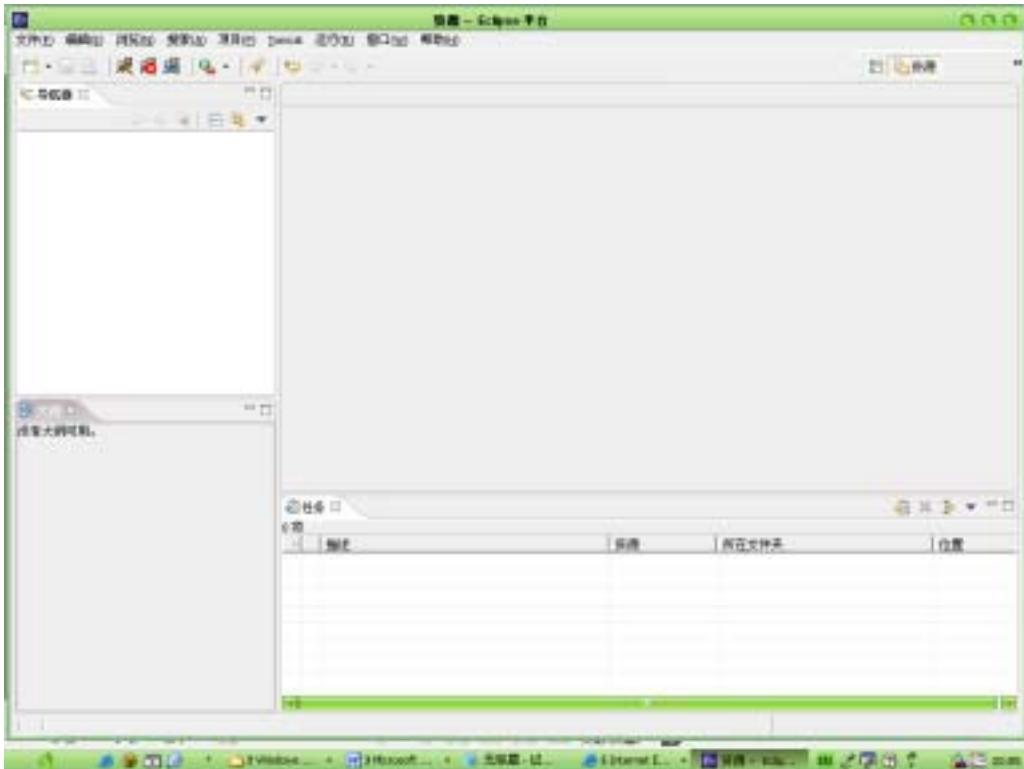


图24.3 Eclipse的中文默认界面（资源透视图）

24.2 使用Eclipse开发Web应用程序

通过安装Eclipse支持Struts的插件可以使Struts的开发变成一件非常简单的事情。

在Eclipse 3.0之前可以使用Easy Struts插件，由于Eclipse 3.0不再支持 Easy Struts插件，所以，如果要使用3.0以上版本的Eclipse就要换用其他插件了，笔者推荐使用MyEclipse插件。下面讲述如何下载、安装MyEclipse并使用MyEclipse开发struts Web应用程序。

24.2.1 下载安装MyEclipse插件

MyEclipse是一款功能强大的J2EE开发插件，支持代码编写、配置、测试以及除错，还包括HTML智能编辑器，Struts、JSF、CSS、JavaScript、SQL和Hibernate，带有自动完成与语法高亮显示功能的J2EE编辑器等。根据Eclipse的不同版本提供不同版本的MyEclipse下载，不过MyEclipse不是免费软件，不过可以有30天的试用期，下载地址是：

<http://myeclipseide.com/index.php>

安装MyEclipse之前要先安装JDK和Eclipse，它的安装相对要麻烦一点，具体可以按照如下步骤进行。

- (1) 双击下载的安装文件，出现MyEclipse的Introduction。
- (2) 单击【Next】按钮，出现最终用户许可协议。
- (3) 选中I Accept后，单击【Next】按钮。
- (4) 设置为Eclipse的安装目录，单击【Choose】按钮可以改变默认值，此时是E:\Eclipse，单击【Next】按钮。

(5) 设置MyEclipse的安装目录，单击【Choose】按钮可以改变默认值，单击【Next】按钮。

(6) 选择创建快捷方式的位置或不创建任何快捷方式，单击【Next】按钮。

(7) 单击【Install】按钮，开始安装。

(8) 安装完成后退出安装程序。

(9) 启动Eclipse，在工具栏中单击【Window】，在弹出的下拉菜单中选择【Preferences】，可以看到窗口左边的框中多出一个选项MyEclipse，如图24.4。

(10) 如果能够看到多出的MyEclipse选项，表明已经成功安装了MyEclipse，需要使用MyEclipse时可以在其中做相应的设置。

24.2.2 使用Eclipse+MyEclipse开发Struts Web应用程序

在成功安装MyEclipse后，就可以使用MyEclipse进行J2EE的开发了，下面介绍使用Eclipse+MyEclipse开发Struts Web应用程序，这样可以简化开发Struts Web应用程序的过程。

(1) 在工具栏中单击【File】菜单，在弹出的下拉列表中将鼠标指向【new】选项，在列表中单击【Project】选项，选择新建一个Web工程，Web工程项的选择见图24.5。

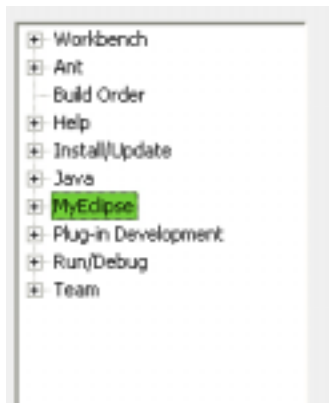


图24.4 MyEclipse安装成功后效果

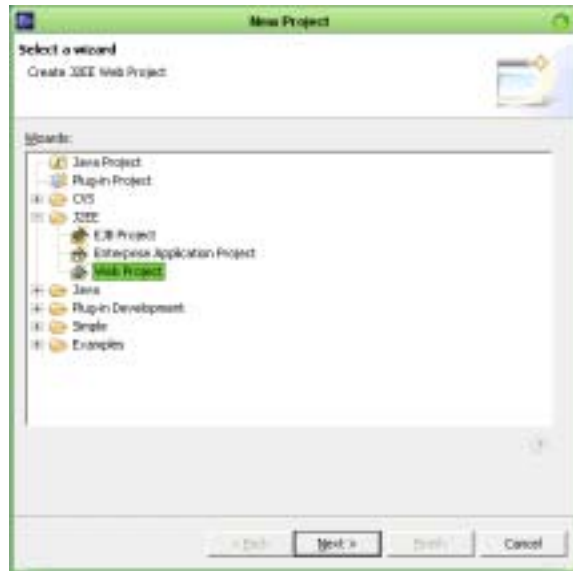


图24.5 新建Web工程

在弹出的对话框中设置project name 为HelloWorld，单击【Finish】按钮。

(2) 为工程添加Struts框架支持，使用1.1版：在【Package Explorer】视图中右击HelloWorld（新建的工程名），在弹出的下拉菜单中将鼠标指向【MyEclipse】，然后在出现的菜单中单击【Add Struts Capability...】，在弹出的对话框中可以做相应的设置，这里使用默认值就可以了。

(3) 添加工程可用Server。在工具栏中单击【Window】，在弹出的下拉菜单中选择【Preferences】，单击窗口左边框中【MyEclipse】选项左边的加号，在下拉目录中单击【Application Server】左边的加号，然后在下拉目录中单击【Tomcat 5】设置其状态为Enable，并设置下面的各个目录（主要设置Tomcat Home Directory），单击【Tomcat 5】左边的加号，在下拉目录中选择Paths，单击Append to classpath项右边的【Add JAR/ZIP】按钮，添加JDK安装目录中lib目录下的tools.jar文件。

(4) 为工程添加布局: 在【Package Explorer】视图中右击HelloWorld(新建的工程名), 在弹出的下拉菜单中将鼠标指向【MyEclipse】, 在出现的菜单中单击【Add or Remove project Deployments...】, 在弹出的对话框中选择要操作的工程HelloWorld, 单击【Add】按钮, 在弹出的对话框中选择Server为Tomcat 5(第(3)步设置其为可用状态), 单击【Finish】按钮, 可以看到Project Deployments对话框中表格中多了一个Tomcat 5项, 如果配置成功可以看到在最左边有一个钩。

(5) 使用向导添加模型文件。新建Struts 1.1 Form Action&jsp(选择路径见图24.6), 单击【Next】按钮, 在出现的对话框中填写Use case为helloWorld, 可以看到name变为helloWorldForm, 选择supperclass为org.apache.struts.action.ActionForm, 在Form Property 选项卡中单击【Add】按钮, 在name域中输入userName, 单击【Add】按钮, 在新弹出的对话框中单击【Close】按钮, 在JSP选项卡中的Create JSP Form? 打上钩, 用来创建jsp页面, 此时关于Form模型文件的设置就完成了, 单击【Next】, 进入Action模型文件的设置, 此时使用默认值就可以了, 单击【Finish】完成添加。



图 24.6 新建Struts 1.1 Form Action&jsp

(6) 修改资源文件: 打开ApplicationResources.properties文件, 在文件最末尾添加如下一行:

```
jsp.hello.title=Hello World!!!  
jsp.hello.error=This is Error!!!
```

(7) 修改Form模型文件: 打开HelloWorldForm.java文件, 把reset方法的内容清空后添加

```
this.userName=null;
```

把validate方法的内容清空后添加:

```
ActionErrors errors= new ActionErrors();  
if((userName==null)|| (userName.length()<1)){  
errors.add("userName",new ActionError("jsp.hello.error"));  
}  
return errors;
```

ActionError处会提示错误, 原因是没有正确的引入包, 找不到类路径, 选中ActionError字符串后, 使用Ctrl + Shift + o可以自动添加。

(8) 修改Action模型文件：新建Struts 1.1 Forward配置，名称设置为Hello，路径设置为/form/helloWorld.jsp，完成后可以在struts-config.xml文件中看到多了一句

```
<forward name="Hello" path="/form/helloWorld.jsp" />
```

打开HelloWorldAction.java文件，把方法execute中throw new UnsupportedOperationException部分去掉，并添加语句

```
return mapping.findForward("Hello");
```

(9) 启动Tomcat，并在浏览器地址栏中输入http://192.168.1.19:8080/HelloWorld/测试所作的工作是否成功。

24.3 Tomcat与Eclipse集成开发Servlet

下载一个Tomcat插件，就能很容易的调用外部独立安装的Apache Tomcat进行JSP、Servlet的开发。

24.3.1 Tomcat插件的安装和初始化设置

Tomcat插件的安装并没有什么特别之处，但是对其进行初始化设置就要费一番心思了，而且也是在学习使用Eclipse过程中容易碰到问题的地方，下面介绍如何安装和初始化设置Tomcat插件。

1. Tomcat插件的安装

从<http://www.sysdeo.com/>上免费下载Tomcat插件（目前最高版本为3.0），下载后解压缩文件到临时目录并把临时目录中com.sysdeo.eclipse.tomcat_X.X.X（X.X.X为版本号）文件夹拷贝到Eclipse主目录下的plugins文件夹就可以了。

2. Tomcat插件初始化设置

安装成功后，为了能在Eclipse中使用Tomcat插件，还需要进行一定的初始化设置。

(1) 单击工具栏的【Window】菜单，在弹出的下拉菜单中选择【Customize Perspective】

(2) 在弹出的对话框中选择【Commands】选项卡，确认【Available command groups】组中Tomcat选项前面有一个钩（选中状态），如图24.7。

(3) 单击工具栏的【Window】菜单，在弹出的下拉菜单中选择【Preferences】。

(4) 在弹出窗口左边的框中双击【Tomcat】项，选择机器中安装的Tomcat版本，并设置Tomcat安装的主目录，设置【JVM settings】面板中，单击classpath（Before generated classpath）项右边的【Add JAR/ZIP】按钮，添加JDK安装目录中lib目录下的tools.jar文件。

(5) 单击面板右下方【OK】按钮，退出首选项设置面板。

(6) 这时在工具栏中应该有一个Tomcat菜单，点击其下拉列表中【Start Tomcat】，启动Tomcat服务，如果启动正常，说明安装和设置已经成功了。

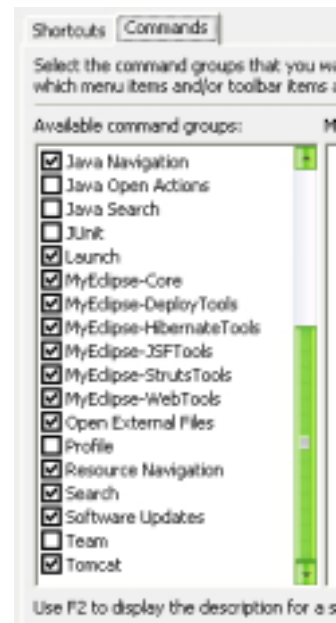


图24.7 选中Tomcat

24.3.2 Tomcat与Eclipse集成开发Servlet实例

在把Tomcat与Eclipse整合成功后，下面就以一个实际的例子来讲述如何在Tomcat与Eclipse环境下开发Servlet。

- (1) 新建一个Tomcat工程，其选择路径如图24.8。
- (2) 在弹出的对话框中输入Project的名字ServletDevelop，单击【Next】按钮。
- (3) 单击【can update server.xml file】选项前面的方框，使其为选中状态，单击【Finish】按钮，一个Tomcat Project就建好了，建好的Project结构如下图24.9。



图24.8 新建Tomcat工程

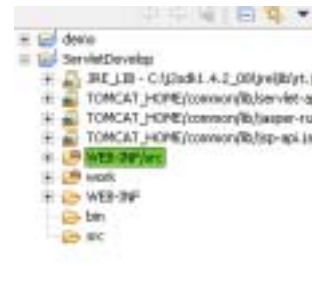


图24.9 Project结构

(4) 打开【new class】向导，在Name中填入HelloWorld，在Superclass中填入javax.servlet.http.HttpServlet，单击【Finish】按钮，一个servlet就创建好了。

(5) 修改生成的文件，为了让Servlet能正确的找到类，需要在文件开头加入如下语句。

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
```

(6) 添加doGet方法：

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
throws IOException, ServletException {
    response.setContentType("text/html");

    PrintWriter writer = response.getWriter();

    writer.println("Hello World!");
```

```
}
```

(7) 为工程添加配置文件，新建ServletDevelop.xml文件，并在其中添加如下内容：

```
<?xml version='1.0' encoding='utf-8'?>
<Context docBase="E:/eclipse/workspace/ServletDevelop"
path="/ServletDevelop"
reloadable="true" workDir="E:\eclipse\workspace\ServletDevelop\work">
</Context>
```

保存在TOMCAT_HOME\conf\Catalina\localhost目录下。

(8) 为Servlet建立映射关系，新建web.xml文件，并在其中添加如下内容：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<servlet>
<servlet-name>HelloWorld</servlet-name>
<servlet-class>HelloWorld</servlet-class>
</servlet>

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

保存文件在{Project主目录}（这里是ServletDevelop）\WEB-INF目录下。

(9) 在浏览器中输入<http://127.0.0.1:8080/ServletDevelop/HelloWorld>，可以看到如下的效果图24.10。



图24.10 程序运行结果

24.4 Tomcat与Eclipse插件开发 - Lombok

24.4.1 Lombok介绍

Lombok是Eclipse的一个主要的开源插件(open-source plug-in)，Lombok插件能够使Java开发者更好的使用Eclipse去创建，调试和部署一个100%基于J2EE的Java应用服务器。

Lombok插件的使用，使得Eclipse将多种J2EE的元素、Web应用的开发和最流行的应用服务器结合为一体。

1. Lombok的主要功能

Lombok的主要功能有如下几项。

- 使用HTML、servlets、JavaServer™ Page(JSP)等方式建立Web应用程序。

- JSP的编辑带有高亮显示和编码助手。
- JSP语法检查。
- 利用Wizard创建Web应用和EJB应用。
- 利用Wizard创建EJB客户端测试程序。
- 支持部署J2EE Web应用档案(EAR), Web模块文件(WAR)和EJB档案文件(JAR)。
- 利用Xdoclet开发符合EJB1.1和2.0的应用。
- 能够实现端口对端口的本地和远程的测试应用服务。
- 能够支持所有的有可扩展定义的Java应用服务。
- 能够利用强大的Java调试器调试正在运行的服务器端代码(JSP&EJB)。
- 通过使用Wizard和代码生成器提高开发效率。
- 创建Web服务客户端的WSDL形式的文件。

2. Lomboz适用的服务器

Lomboz对Apache Tomcat、JBOSS、JOnAS、Resin、Orion、JRun、Oracle IAS、BEA WebLogic Server和IBM WebSphere等服务器大多数版本都支持的很好。

24.4.2 建立Lomboz环境

要在Eclipse中建立Lomboz环境可以按照如下步骤进行。

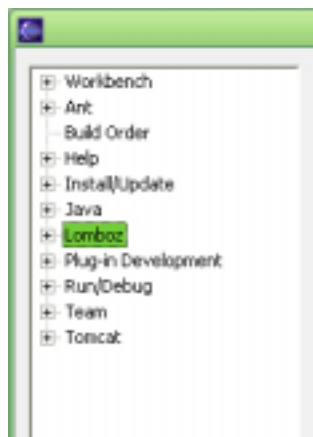
1. 安装Lomboz插件

Lomboz插件的安装与其他Eclipse插件的安装基本相同。

(1) 把Lomboz插件压缩包(笔者使用的是Lomboz 3.01)解压出的plugins目录拷贝到Eclipse安装目录下。

(2) Lomboz的运行还需要emf运行环境的支持, 还要把emf-sdo-runtime-2.0.0解压出的eclipse目录下的两个目录拷贝到Eclipse安装目录下。

(3) 为了验证安装的正确性, 运行Eclipse程序后, 单击工具栏的【Window】, 在弹出的下拉菜单中选择【Preferences】, 可以看到在左边的框中多了Lomboz一项, 如图24.11。



24.11 Lomboz安装成功效果

2. Lomboz配置

按照如下步骤对Lomboz进行配置。

- (1) 运行Eclipse程，单击工具栏的【Window】。
- (2) 在弹出的下拉菜单中选择【Preferences】，在设置对话框的左边找到Lomboz（如果没有找到，说明Lomboz没有安装正确，要重新检查Lomboz的安装）。
- (3) 此时右边的设置项只有一个要修改：JDK tools.jar，把它指向JAVA_HOME/lib/tools.jar，改好后单击【Apply】按钮。
- (4) 打开左边的Lomboz设置选项，找到【Server definitions】。
- (5) 在右边的Server types里选择【Apache Tomcat v5.0.x】。这里有两个地方要设置：Application Server Dictionary和Classpath Variable，其中Application Server Dictionary和Classpath Variable都指向Tomcat的安装目录，Classpath Variable是用于设置ClassPath用的，单击【Apply】按钮。
- (6) 打开最后一个选项卡【Project Classpath】，在里面加一个Library，按【Add library】按钮，在Tomcat的安装目录中找到common/lib/jsper-runtime.jar，效果如图24.12。

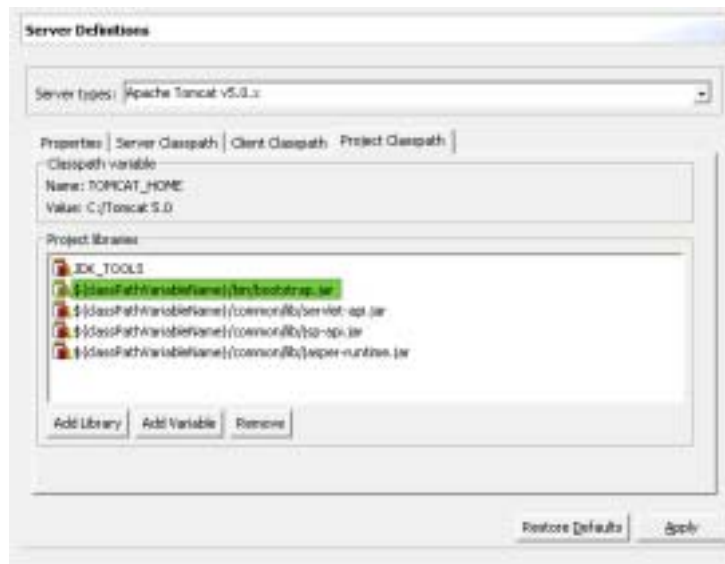


图24.12 Lomboz配置效果

- (7) 在设置对话框中左边找到【Java】项，双击其下的【BuildPath】，在右边的【Source and output folder】里选择【Folders】；As JRE library use项选择【JRE_LIB variable】，单击【Apply】按钮。

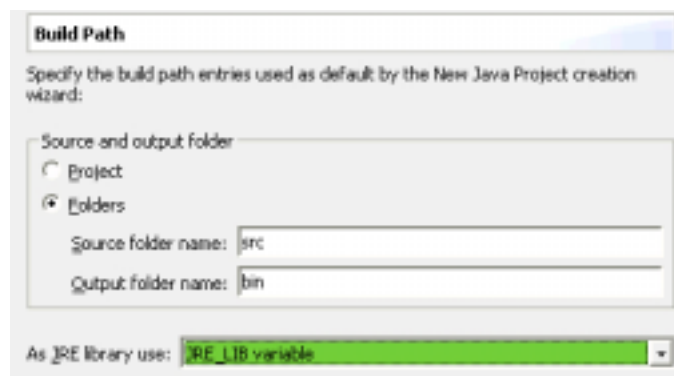


图24.13 BuildPath设置效果

按照上述步骤配置后，Lomboz的设置就结束了，可以使用下一节的例子验证是否设置成功。

24.4.4 应用实例

在本节中介绍一个使用Lomboz插件进行开发的一个实际例子，可以按照如下步骤进行。

1. 新建Lomboz工程

- (1) 新建一个Lomboz J2EE Project，在出现的对话框里添上Project的名字（LombozTest），单击【Next】按钮。
- (2) 在弹出的Java Setting对话框中均使用默认值，然后单击【Next】按钮。
- (3) 在【Create J2EE Module】对话框中，添加Web Module（单击【Add】按钮，填入Web Module的名字，例如HelloWorld），然后还需要添加一个Server，选择Apache Tomcat v5.0.x后单击【Add】按钮。
- (4) 单击【Finish】按钮，完成工程的创建。

2. 更改设置

新建Lomboz工程成功后，在Package Explorer中右击新建的工程LombozTest，选择【Properties】，双击属性对话框左边的Java Build Path，单击【Source】选项卡，选中选项卡下部的【Allow output folders for source folders】，把【Default output folder】改为刚才建的Project下的WEB-INF/classes。它是Tomcat默认放Class的地方。

经过这样的更改后，就可以调试Bean和JSP文件了，此时Lomboz生成的文件结构如图24.14。



24.14 Lomboz生成的文件结构

3. 新建JSP文件并、建立Tomcat配置文件并进行语法检查

经过上面的设置，就可以调试JSP程序了，当然了，首先要建立工程需要的JSP文件，这里，使用Lomboz自动生成的index.jsp，仅稍做修改，修改后程序如下。

```
<html>
  <head>
    <title>Welcome</title>
  </head>
  <body>
    <center><%= "Hello World!!!" %></center>
```

```
</body>
</html>
```

建立Tomcat配置文件HelloWorld.xml，将其保存到TOMCAT_HOME\conf\Catalina\localhost目录下，文件内容如下：

```
<?xml version='1.0' encoding='utf-8'?>
<Context path="/HelloWorld"
docBase="E:\eclipse\workspace\LombozTest\HelloWorld"
workDir="E:\eclipse\workspace\LombozTest\j2src">
```

其中，path表示Web应用的URL名字；docBase表示web应用的物理位置；workDir表示jsp所对应的java文件产生的位置，workDir只能指向j2src，否则在Eclipse中无法正确找到jsp对应的java文件，从而无法进行调试（在Eclipse2中在j2src下必须创建org\apache\jsp这样的目录结构和名称，Eclipse3以上不需要）。

在调试之前，JSP应该是没有语法错误的。Lomboz提供了JSP语法检查的功能，具体操作步骤如下。

- （1）选择要进行语法检查的JSP文件，单击鼠标右键。
- （2）在弹出的下拉菜单中选择【Lomboz j2ee...】，然后单击【Check Jsp Syntax】。
- （3）这样就可以进行语法检查了，可以根据检查的结果进行修改或转入下一步。

到现在为止，调试JSP程序所需要做的准备工作都结束了，下面就可以对jsp程序进行调试了。

4. 调试JSP程序

解决完语法错误之后，接下来就是解决逻辑错误的时候了。对于这种错误就只能通过调试来解决了。使用Lomboz调试JSP的过程与普通Java应用程序的调试非常类似，因为它的原理就是先利用服务器生成JSP对应的Java文件，然后对它们进行调试。下面是进行调试的步骤。

（1）单击【Window】菜单项，在弹出的菜单中选择【Show View】，在出现的View列表中选择【Lomboz J2EE View】，如果列表中没有这一项，可以单击【Other...】后在弹出的对话框中选择。

（2）在【Lomboz J2EE View】中展开要调试的工程，找到工程的服务器，然后右击服务器名，然后单击【Debug Server】项，如果这个工程的Tomcat配置文件被正确建立，则Tomcat会启动。如果Tomcat无法启动，可以从Lomboz初始化配置和Tomcat配置文件两方面来排除错误。

（3）启动Tomcat服务器后，在浏览器中输入http://127.0.0.1:8080/HelloWorld/index.jsp，如果JSP页面的内容被正确显示，会在j2src\org\apache\jsp目录下生成这个JSP文件对应的Java文件，在【Package Explorer】的空白中右击，在弹出的菜单中选择【Refresh】（或按F5快捷键）刷新工程，可以在j2src目录下看到生成的Java文件，如图24.16。

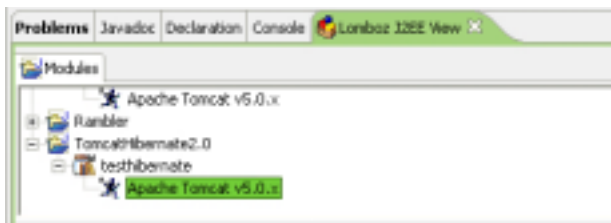


图24.15 Lomboz J2EE View

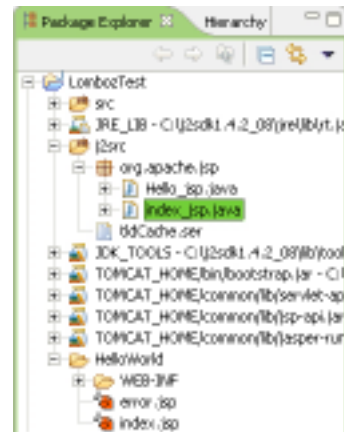


图24.16 JSP文件对应的Java文件

这一步主要是触发服务器根据jsp产生可供调试的java文件，不必太在意结果是否正确。因为一旦java文件生成，那么这些错误都可通过调试排除。

(4) 打开要调试的Java文件，在其中设置断点，然后在IE的地址栏再次输入这个URL，此时Eclipse的调试功能即被激活。至此就可有针对性地对JSP进行调试了，为了更好的进行调试并查看调试的结果，可以更改为Debug试图（单击【Window】菜单项，在弹出菜单中选择【Open Perspective】，然后选择Debug），更改试图后，效果如图24.17。

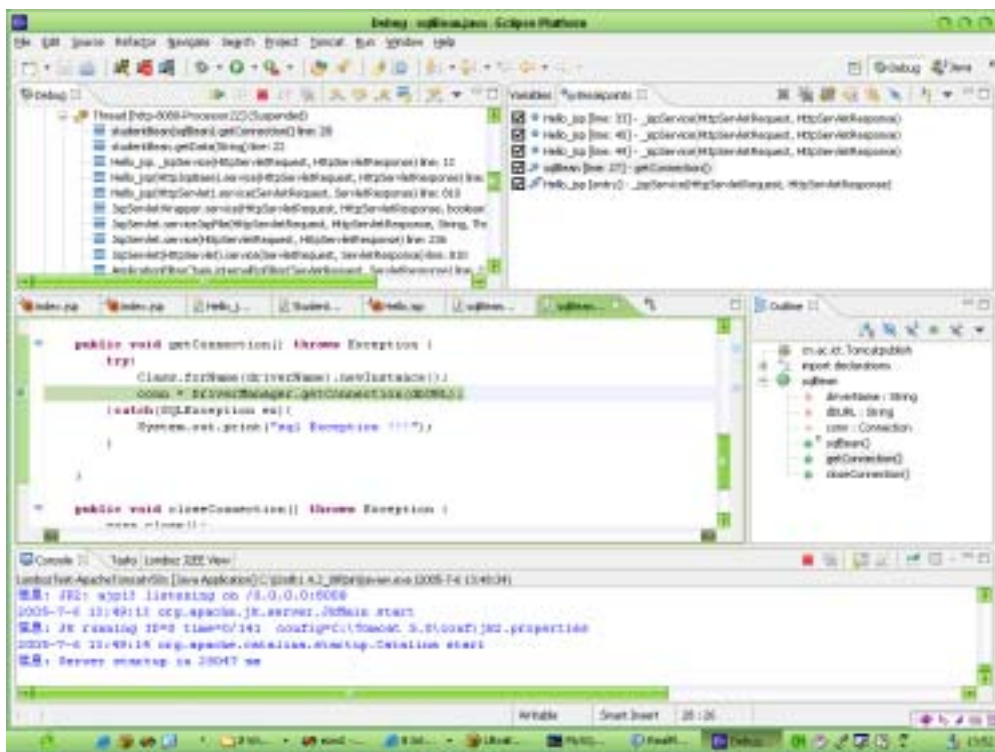


图24.17 Debug试图

可以看到在Debug试图中可以很容易的添加、删除断点，查看变量的值以及程序的运行情况的信息。

虽然目前JSP的调试还没有如同普通的Java文件调试般方便，但是Lomboz工具的出现至少结束了手动调试JSP的阶段。从此Java的web应用开发者不必因为一些错误而时不时地停下程序来手工添加打印语句来观察变量的值，相反完全可以如调试普通Java应用一样在程序运行的同时来观察这些值的变化。这大大方便了基于WEB应用的开发，减少了以前为调试所浪费的时间。

24.5 用Eclipse、Spindle和Tomcat搭建Tapsetry开发环境

24.5.1 Tapestry简介

Jakarta Tapestry是一个开源的Java Web应用框架。它是一个基于控件的框架以致于用它开发Web应用类似开发传统的GUI应用。用Tapestry开发Web应用时无需关注以操作为中心的(operation-centric) Servlet API，引用Tapestry网站上的一句话：“Tapestry用对象(objects)，方法(methods)，属性(properties)替代以往的URLs和查询参数，重新诠释Web应用开发。Tapestry的特性主要有如下几项。

1. 简单性

Tapestry应用与传统的Servlet应用相比代码量更少。大多数传统的Servlet应用包含如下厌烦和重复的任务：解析查询参数，处理HttpSession对象，构建URLs等。Tapestry消除了传统Servlet应用中许多无趣的“衔接”代码("plumbing" code)使开发者把精力集中到应用逻辑上来。

2. 一致性

Tapestry为开发Web应用的页面提供了一致的方式。这样有助于消除传统Servlet应用开发中臆测。由于所有Tapestry应用中的页面都是用相同的可复用的控件组织而成，所以工作方式是相似的。

3. 高效率

Tapestry应用拥有高度的可升级性，它利用缓存和对象池使每个请求的处理时间最小化。Tapestry应用拥有跟传统Servlet应用相仿的性能。

4. 错误反馈

开发过Servlet/JSP应用的人都有类似的经历：为了找出Web.xml文件有什么错误，不得不花费大量时间察看浏览器中的堆栈信息。Tapestry拥有优秀的错误报告方式，最值得一提的是它会指出哪个文件以及那一行导致了错误。

24.5.2 安装Spindle插件

Spindle是用Eclipse开发Tapestry项目的一个插件（Spindle主页 <http://spindle.sourceforge.net/>），它在Eclipse上的安装与其他插件的安装有很大差别，需要用升级的方式获得，可以按照如下步骤安装Spindle插件。

(1) 单击工具栏的【Help】菜单项，在弹出的下拉菜单中选择【Software Updates】，然后选择【Find and Install】。

(2) 在弹出的对话框中选择【Search for new features to install】，单击【next】按钮。

(3) 在【Update site to visit】对话框中单击【New Remote Site...】，Remote site的Name命名为Spindle Update Site，URL设为<http://spindle.sourceforge.net/updates/>，效果如图24.18，

然后单击【OK】按钮。

(3) 选中新建的站点【Spindle Update Site】，单击【Next】按钮，出现【Search Results】对话框，从中选择Spindle的最高兼容版本，单击【Next】按钮，进入安装过程。

(4) 安装完成就可以了，安装成功后，可以单击【Help】菜单，然后选择【About Eclipse Platform】，在对话框中单击【Plug-in Details】查看安装结果，可以看到Eclipse的已安装插件列表中有Spindle的项了，如图24.19。



图24.18 新远程站点



图24.19 Spindle安装成功

24.5.3 Tapestry实例

下面给出一个简单的例子，讲述在Eclipse和Tomcat环境下如何开发Tapestry。

(1) 新建【Tapestry Web Project】，命名为【Tapestry】，并把context设置为tapestry，如图24.20。

(2) 在以后的设置中均采用默认值，完成工程的建立。

(3) 从<http://jakarta.apache.org/tapestry>下载Tapestry-3.0.3-bin，并解压缩；从<http://www.ognl.org/>下载ognl-2.6.3.jar；从<http://sourceforge.net/>下载javassist-3.0beta2.zip，解压后，找到javassist.jar包。

(4) 把Tapestry-3.0.3-bin解压缩后文件夹下的lib目录中的jar包、ext文件夹下的jar包以及ognl-2.6.3.jar包和javassist.jar拷贝到新建工程的context下WEB-INF下的lib目录下，这里是E:\eclipse\workspace\Tapestry\tapestry\WEB-INF\lib，此时此目录下应该有如下包文件。

- > bsf-2.3.0.jar
- > commons-beanutils-1.6.1.jar
- > commons-codec-1.2.jar
- > commons-collections-2.1.jar
- > commons-digester-1.5.jar
- > commons-fileupload-1.0.jar
- > commons-lang-1.0.jar
- > commons-logging-1.0.2.jar
- > jakarta-oro-2.0.6.jar
- > javassist.jar
- > ognl-2.6.3.jar
- > tapestry-3.0.3.jar

➤ `tapestry-contrib-3.0.3.jar`

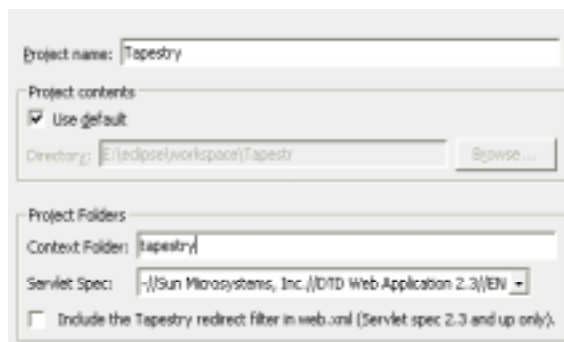


图24.20 新建Tapestry Web Project

(5) 打开Home.html文件，在其中`<body>`和`</body>`之间添加语句` Hello,Tapestry! `。

(6) 新建Tomcat配置文件`tapestry.xml`，保存在`<Context path="/tapestry" reloadable="true" docBase="E:\eclipse\workspace\Tapestry\tapestry" workDir="work\Catalina\localhost\tapestry"/>`。

(7) 打开浏览器，在地址栏中输入`http://localhost:8080/tapestry/app`，出现以下画面，如图24.21。



图24.21 `tapestry`实例效果图

24.6 相关问题

在Tomcat与Eclipse的整合使用过程中，总是会碰到很多问题，有些问题与各个软件的版本有关，有的则是开发者在设置的过程中由于各种原因造成的，下面是初学者容易犯的几个错误以及相应的解决办法。

1. 安装lomboz后在preferences中看不到

(1) 删除整个目录`/eclipse/configuration/org.eclipse.update/`，重启Eclipse

(2) 在启动Eclipse时带上`-clean`参数

如：`d:\eclipse\eclipse.exe -clean`

(3) 如果Eclipse启动找不到插件了，解决办法：在`/configuration/config.ini`文件中加入一行

```
osgi.checkConfiguration=true
```

这样它会寻找并安装插件，找到后可以把那行再注释掉，这样以后每次启动就不会因寻找插件而显得慢了。

2. Eclipse + Tomcat输出日志无法显示中文

在初始化设置Tomcat时，把【JVM Settings】中【Boot Classpath】加上了虚拟机的rt.jar包后，会发现日志文件输出的中文都变成了???，要解决这个问题，只要再把和rt.jar包在同一个目录的charsets.jar包也加到【Boot Classpath】中就可以了。

3. Tomcat+Eclipse开发，启动Tomcat时出问题，提示java.lang.ClassNotFoundException: org.apache.catalina.loader.DevLoader错误。

这是由Tomcat中缺少Devloader类包引起。

解决方法：

在Eclipse插件中找到\com.sysdeo.eclipse.tomcat_3.0.0目录，copy DevLoader.zip文件到Tomcat的安装路径下的server目录下的lib文件夹中。解压，并把解压后的目录拷贝到server目录下的classes文件夹中。

24.7 小结

Eclipse是一个集成的开发工具，它对于Web应用的开发可以提供很好的支持。在本章中介绍了如何在Eclipse中开发Struts、集成Tomcat、使用Lomboz插件和开发Tapestry程序等，这些都是进行Web开发时可能会用到的技术，读者并不一定会用到所有的方面，学习时也可以有选择的学习。其中最常用的就是把Eclipse和Tomcat进行集成的部分了，读者可以重点掌握。