

第 9 章 Web构建工具Eclipse+Lomboz的安装和配置

上一章介绍了 JSP 的页面构成、标准动作和内置对象的使用，让读者学会如何编写一个完整的 JSP 文件。这一章将重点向读者介绍如何使用工具和插件更加简单地架构和开发 Web 应用，以及代码的维护。第 3 章介绍的第一个 JSP 应用，需要手动创建 Web 模板，以及手动使 Web 模板发布到 Web 服务器上，这一切都显得非常繁琐。如果 JSP 页面中使用到适当的 Java 类文件，没有适当的工具，还必须手动将大量的源代码编译成二进制可执行的 class 文件。所以说，适当工具的使用，将是事半功倍。

下面将向读者详细介绍 Eclipse 开发工具以及用于 Web 架构的 Lomboz 插件。除了这一章节所介绍的两种工具之外，在开发过程可能还会使用到其他辅助工具或者第三方插件，届时会逐一向读者介绍。

本章要点包括以下内容：

- ☐ Eclipse 工具的安装
- ☐ Eclipse 工具的配置
- ☐ Lomboz 插件的安装
- ☐ Lomboz 工具的配置
- ☐ Eclipse+Lomboz 架构 J2EE 项目

9.1 Eclipse工具的安裝和配置

Eclipse 是一个不错的 Java 开发工具，并且开发源代码。如果配合第三方插件 Lomboz，可以更快捷、方便地构架和开发 Web 应用。

9.1.1 Eclipse工具的介绍

Eclipse 是当今非常流行的 Java 集成开发环境（IDE）。现在 Java 程序员越来越青睐于 Eclipse。这是因为它是免费的、开放源代码的、质量和界面很好，而且可以通过第三方插件很容易进行定制。没有安装任何插件的 Eclipse（一般集成了 Ant 发布和 CVS 版本控制工具）是核心非常小的程序，开发者可以根据自己的需要安装相应的第三方插件来扩展 Eclipse 的功能。

另外，Eclipse 的安装也是非常简单的。它属于绿色安装，即直接解压到指定目录就可以使用了，不需要向 Windows 注册表里写入信息。卸载时直接删除 Eclipse 目录即可，也不需要运行卸载程序。

除了 Eclipse 可以结合 Lomboz 或者 MyEclipse（和 Lomboz 类似的 J2EE 构架插件）来方便的架构和开发 Web 应用之后，Eclipse 可以结合 SWT/JFace 插件开发出界面优美的 Application 产品。另外还可以使用 Eclipse 开发能够组装到自身中的插件。

由于本书介绍的是 JSP 网站开发，所以对 Eclipse 的介绍也仅仅局限于 Eclipse 和 Lomboz 的 Web 应用开发。

9.1.2 Eclipse的下载

Eclipse 可以去官方网站 <http://www.eclipse.org> 下载软件包。具体的下载地址为 <http://download.eclipse.org/eclipse/downloads/index.php>，本书选择 3.1.0 版本的 Eclipse。下载后的软件包全名为 eclipse-SDK-3.1-win32.zip。

9.1.3 Eclipse的安装

Eclipse 的运行需要有 JDK（提供 Java 运行环境 JRE 和常用库文件）的支持，JDK 的安装在第 2 章已经介绍过。

Eclipse 的安装非常简单，并且属于“绿色”安装。将下载的 eclipse-SDK-3.1-win32.zip 文件解压到相应路径下即可，例如本书将 Eclipse 安装在 D:\zzb\eclipse 目录下。详细目录结构如图 9.1 所示。

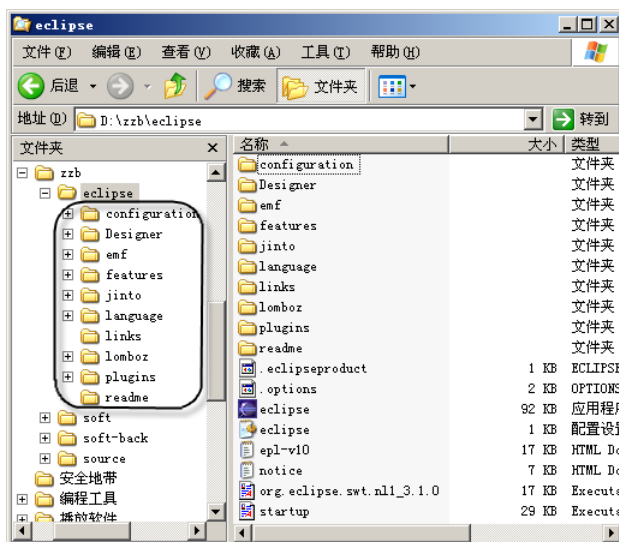


图 9.1 Eclipse 安装目录

这样就已经成功安装了 Eclipse。直接双击 D:\zzb\eclipse\eclipse.exe 执行文件就可以启动 Eclipse。

9.1.4 Eclipse的配置

Eclipse 第一次启动时，会弹出“Workspace Launcher”对话框，用来指定项目文件所存放的路径，如图 9.2 所示。此处把工作空间的路径设置为 D:\zzb\source。选中“Use this as the default and do not ask again”选项，单击“OK”按钮，这是为了下次再启动时不再出现类似提示。如果 Eclipse 启动后出现“Welcome to Eclipse 3.0”的欢迎界面，表示 Eclipse 已经成功启动。

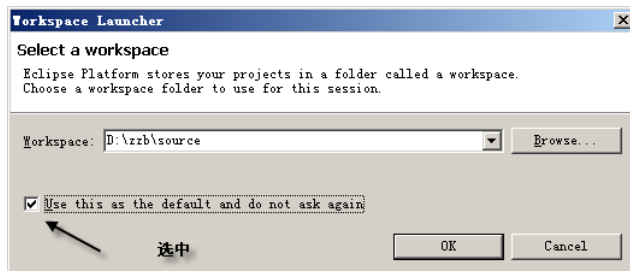


图 9.2 Workspace Launcher 对话框

另外需要查看 Eclipse 的 Java 虚拟环境是否设置正确，一般情况 Eclipse 会自动识别到 JRE 目录位置，无须手动设置。选择菜单中的“Window”|“Preferences”命令，弹出如图 9.3 所示的窗口。

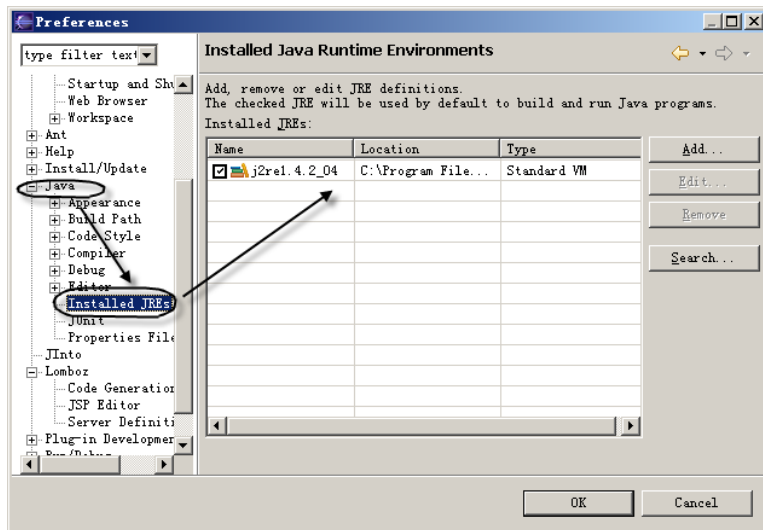


图 9.3 Preferences 窗口

选择图 9.3 左侧的“Java”|“Installed JREs”命令，弹出图中右侧所示的界面。查看 JRE 路径是否正确，如果有误，需要手动进行修改。

除此，在建立 Web 项目之后，还需要进行其他配置（例如，源代码编译后的路径以及 Tomcat 服务器的配置等等）。有关项目属性的配置将在以后实例讲解中再介绍。

9.2 Lomboz插件的安装和配置

Lomboz 插件类似于 MyEclipse，都属于 J2EE 开发插件。它和 Eclipse 工具的组合可以加大 Web 开发的方便和快捷性。

9.2.1 Lomboz插件的介绍

前面介绍的 Eclipse 工具是由一个很小的核心和安装在核心之上的大量插件组成。插件的安装可以由开发者根据需要自定。Eclipse 默认安装并不支持 Web 应用的架构，所有需要安装第三方插件，但是 Eclipse 默认继承了 Ant 和 CVS 工具。

Lomboz 是一个很好的 J2EE 开发工具，它也完全是开源和免费的。使用 Lomboz 可以方便的架构 Web 项目，自动生成一些配置文件（web.xml 文件等）和目录（WEB-INF/classes）。另外，还可以使用 Lomboz 来将开发的 Web 应用发布到 Web 服务器上，并且可以通过 Lomboz 来启动和停止 Web 服务器。

注意：J2EE 是一个非常广的概念，它不仅仅涉及到 Web 模块，还包括 EJB 模块以及整体构架和安全方面。本书主要介绍 Web 模块的开发，不涉及 EJB 中间件部分。

9.2.2 Lomboz插件的下载

Lomboz 的具体下载地址为 http://forge.objectweb.org/project/showfiles.php?group_id=97。同时还要下载 emf 包，用户可以在下载页面中查看到这个文件。下载后的这两个插件文件名为：

- ❑ org.objectweb.lomboz_3.1.0.N20050106.zip。
- ❑ emf-sdo-runtime-I200412160800.zip。

注意：下载 Lomboz 时，一定要根据 Eclipse 的版本进行选择。如果版本不对应，将导致 Lomboz 工作不正常。

9.2.3 Lomboz插件的安装

Eclipse 插件的安装也非常简单，主要有如下两种方法：

(1) 将以上下载的 org.objectweb.lomboz_3.1.0.N20050106.zip 和 emf-sdo-runtime-I200412160800.zip 文件解压，并分别给目录取名为 lomboz 和 emf，然后将解压后的这两个目录剪切到 Eclipse 安装的根目录（本书为 D:\zzb\eclipse），具体目录结构如图 9.4 所示。

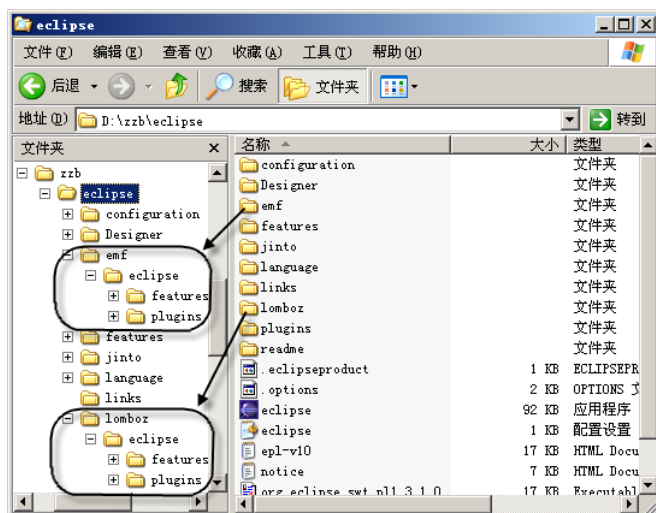


图 9.4 Lomboz 和 emf 插件的安装目录

从图中可以看出，Lomboz 和 emf 目录下都有一个 eclipse 子目录，然后还有 feature 和 plugins 二级子目录。请读者注意，如果这两个插件解压之后不存在这样的目录，请读者自行创建。

然后在 eclipse 安装的根目录下（本书为 D:\zzb\eclipse）创建一个 links 文件夹（links 名为固定，不能随意变动，不然 eclipse 不能识别）。然后在 links 文件夹中创建两个名为 lomboz.link 和 emf.link 的文件（这里的文件名，以及后缀都可以任意取名）。在 lomboz.link 文件中添加如下一行代码：

```
path=lomboz
```

代码说明：这里使用了相对路径，当然用户也可以写成绝对路径，例如 path=D:\zzb\eclipse\lomboz 或者 path=D:\\zzb\\eclipse\\lomboz（是两个反斜杠）。建议使用相对路径，这样方便移植。

类似地，在 emf.link 文件中添加如下一行代码：

```
path=emf
```

(2) 第二种方法就直接将 Lomboz 和 emf 解压，然后将各自的 feature 和 plugins 目录下的所有文件复制到安装 eclipse 根目录下的 feature 和 plugins 目录下。

经过以上设置之后，还需要重新启动一个 Eclipse 才能有效。Eclipse 启动过程中会自动查找 links 目录下的所有文本文件，然后根据文件中 path 设置来加载相应的插件。

使用前，还需要检查插件是否成功安装。打开 Eclipse，在菜单中选择“File”|“New”|“Project”命令，弹出“Select a wizard”对话框，如图 9.5 所示。如果在窗口的列表框中出现“Lomboz J2EE Wizard”选项，即说明插件已经安装成功。

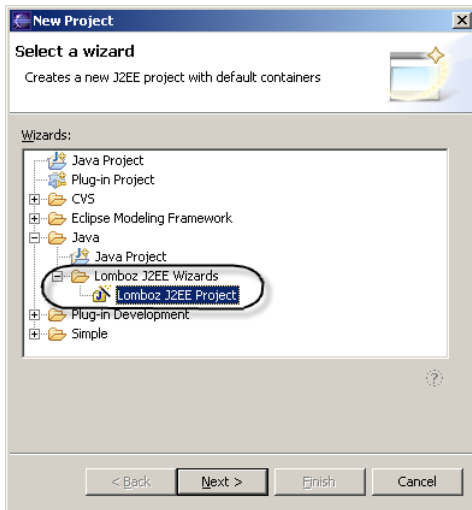


图 9.5 向导对话框

在安装和配置 Lomboz 过程中，还需要注意如下多个事项：

(1) 如果需要在 Eclipse 中安装多个插件，这时候既可以将所有设置插件的 path 放在同一个 link 文件中，也可以一个插件对应一个 link 文件（如上设置方法，这些 link 文件都创建在根目录下的 links 文件夹中）。

(2) 如果 Lomboz 安装之后，path 路径也设置正确，但图 9.5 所示的“Select a wizard”窗口始终没有显示出“Lomboz J2EE Wizard”选项，这时把 D:\eclipse\eclipse\configuration 目录下的除 config.ini 文件之外的所有文件都删除掉，再重新启动 Eclipse。

第二种安装方法虽然比较简单，但是不利于插件的管理和卸载，所以本书建议使用第一种 link 安装方法。

9.3 Eclipse+Lomboz架构J2EE项目

在使用 Eclipse+Lomboz 进行 Web 项目构架之前，为了更好的 Web 开发，还需要配置一些环境。详细介绍如下。

9.3.1 设置Java的构建路径

在使用 Eclipse+Lomboz 构建 Web 应用之前，首先需要设置 Eclipse 的构建路径，即指定 Java 源码和编译文件的路径（可以设置使两者分离）。这一设置步骤非常重要，以便于以后对项目代码进行管理和维护，从而避免出现乱码现象，具体设置步骤如下：

(1) 打开 Eclipse，单击菜单中的“Window”|“Preferences”命令，会弹出“Preferences”对话框。

(2) 选择左侧列表框中的“Java”|“Build Path”选项。在“Source and output folder”选项框中，选中“Folders”选项。然后在“Source folder name”文本框中输入存放 Java 源文件的文件夹名 src，在“Output folder name”文本框中输入存放输出文件的文件夹名 bin，如图 9.6 所示。

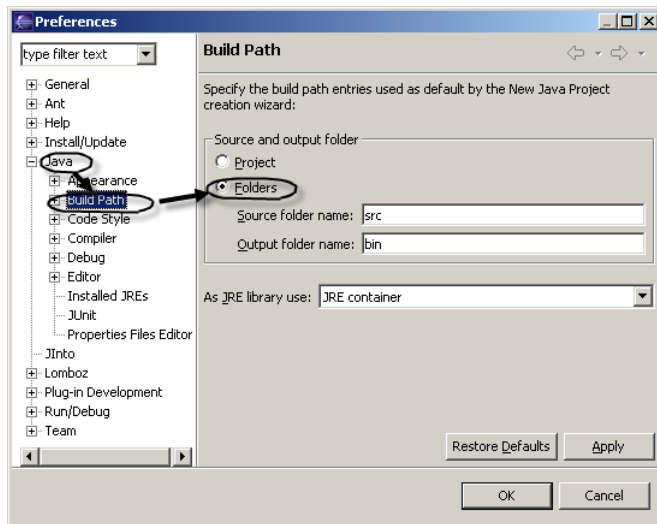


图 9.6 设置 Java 项目的构建路径

说明：“Source folder name”文本框中输入的目录为 Java 源文件所在的位置，这样设置之后就可以把源文件和编译之后的 class 文件分离开来。其中编译文件（以.class 为后缀）会生成在 Web 模块下的 WEB-INF/CLASS 文件夹中；“Output folder name”文本框中输入的目录为 JSP 运行之后生成的 servlet 等文件的所在位置。如果不进行这样的设置，以后会出现很多种类的文件混在一起（例如 Java 源文件和编译后的 class 文件），给后期开发会带来很大的不便。

9.3.2 Lombok环境设置

在使用 Lombok 之前也需要进行适当地设置，其中包括 JDK 中 tool.jar 包路径设置和 Lombok 所使用 Web 服务器的选择。

9.3.2.1. tool.jar 包路径设置

在 Eclipse 主菜单中选择“Window”|“Preferences”命令，弹出“Preferences”对话框。单击左侧列表框中的“Lombok”选项，在右边窗口的“JDK Tools jar”文本框中设置 JDK 的 tools.jar 包位置，一般会检测到。由于本书安装的 JDK 在 C 根目录下，所以这里设置的路径为 C:\jdk1.4.2_04\lib\tools.jar，如图 9.7 所示。

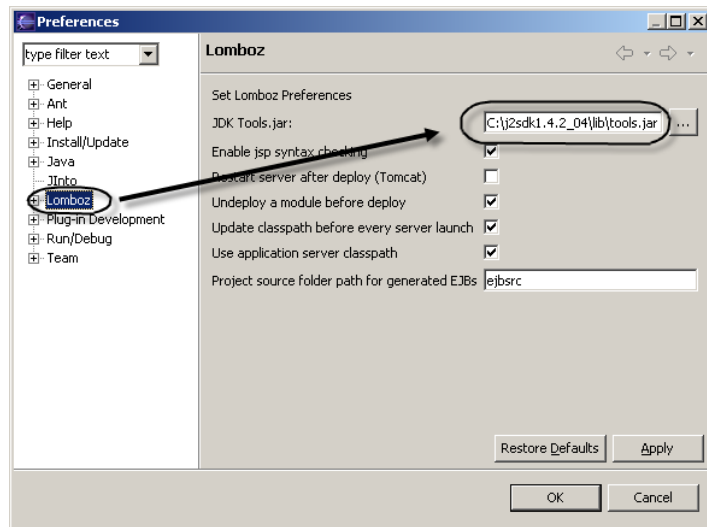


图 9.7 JDK 的 tools.jar 包设置

9.3.2.2. Web 服务器选择

同样在弹出的“Preferences”对话框中，单击左侧的“Lomboz”|“Server Definitions”选项，如图 9.8 所示。

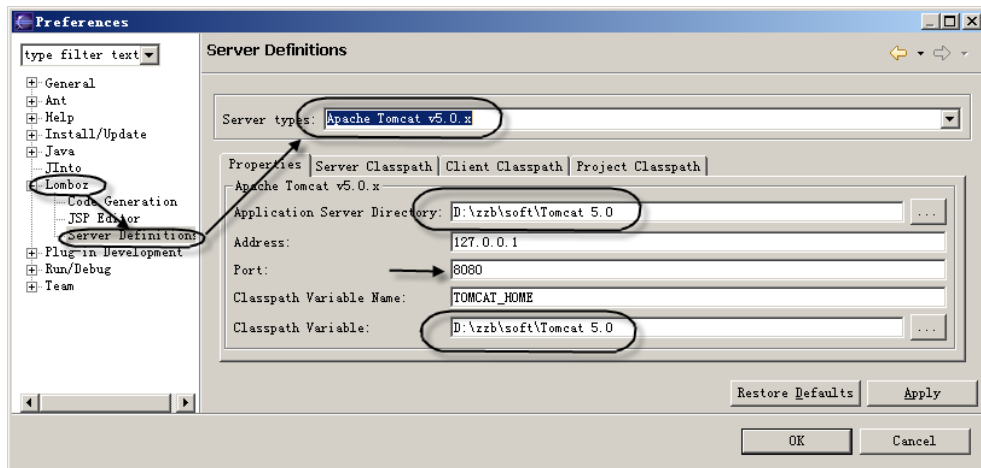


图 9.8 Lomboz 中设置 Tomcat 服务器

首先需要在右边窗口的“Server types”下拉框中选择相应 Web 服务器。然后设置 Web 服务器路径，由于本书的 Tomcat 安装目录为 D:\zzb\soft\Tomcat 5.0，所以在“Apache Server Directory”和“Classpath Variable”文本框中如图 9.8 所示进行设置。每步设置完之后都要单击“Apply”按钮，最后单击“Ok”按钮完成所有设置。

Lomboz 和 Tomcat 之间有时会存在一点冲突，需要作一点修改。

(1) 在 Lomboz 插件的安装目录 eclipse\lomboz\eclipse\plugins\com.objectlearn.jdt.j2ee_3.0.1\servers 中可以查看到各种 Web 服务器的配置文件，它们会一一显示在图 9.8 的“Server types”下拉框中。把除了 Tomcat50x.server 配置文件之后的所有其他配置文件都删除，如果不删除，在 Web 发布和使用 Lomboz 启动 Tomcat 时有时会出错。

(2) 使用记事本打开 Tomcat50x.server 配置文件，将大约在 35 行和 39 行的所有 `${serverRootDirectory}/bin;${serverRootDirectory}/common/endorsed` 程序代码改成

`${serverRootDirectory}/common/endorsed`，即把`${serverRootDirectory}/bin` 部分删除。

9.3.3 创建一个J2EE项目实例

完成了以上所有设置之后，就可以使用 Eclipse+Lomboz 来创建一个 J2EE 项目（注意，这里并不创建 EJB 模块，仅包括 Web 模块）。具体创建方法和步骤如下：

（1）启动 Eclipse，然后单击菜单中的“File”|“New”|“Project”命令，弹出“Select a Wizard”对话框，如图 9.9 所示。

（2）选择“Java”|“Lomboz J2EE Wizard”|“Lomboz J2EE Project”选项，单击“Next”按钮继续下一步操作，弹出“Project Settings”对话框，如图 9.10 所示。

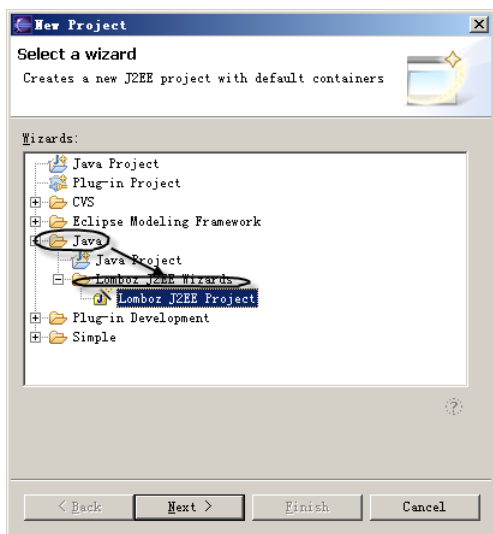


图 9.9 选择向导对话框

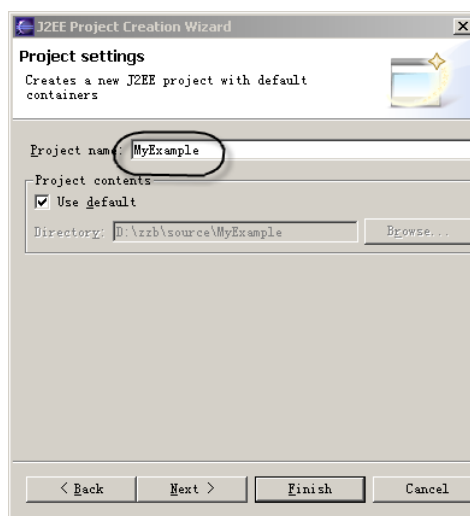


图 9.10 创建项目名对话框

（3）在“Project Name”输入框中输入该创建的项目名称，本书取名为 MyExample（用户可任意取名，只要不和之前项目名相同即可）。设置完之后，单击“Next”按钮进行下一步操作，弹出“Java Settings”对话框，如图 9.11 所示。

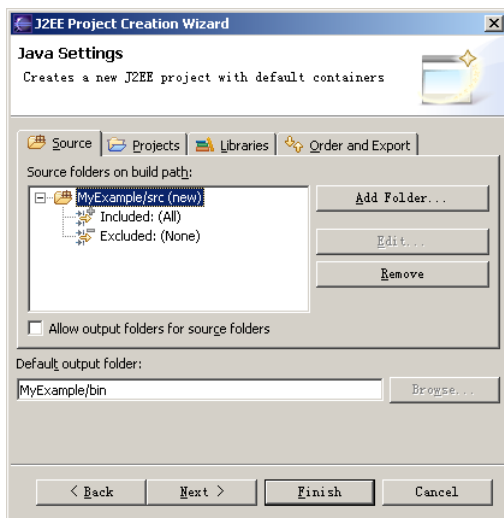


图 9.11 定义 Java 构建设置对话框

(4) 保持如图 9.11 所示的默认选项，然后单击“Next”按钮，弹出“Create J2EE Module”对话框，如图 9.12 所示。

(5) 在图 9.12 所示的“Web Modules”选项卡中，单击右边的“Add”按钮添加一个 Web 模块，取名为 Example（和项目名称有区别，也可以同名，这不冲突）。

(6) 在“Create J2EE Module”对话框的“Targeted Server”选择卡中，选择 Apache Tomcat v5.0.x 选项（由于本书使用 Tomcat5.0 作为 Web 服务器），然后单击“Add”按钮，如图 9.13 所示。

(7) 另外，由于本例用不到 EJB 模块，所有在“EJB Modules”选择卡中不需要添加任何东西。以上设置完成之后单击“Finish”按钮，就生成了一个名为 MyExample 的 J2EE 项目。

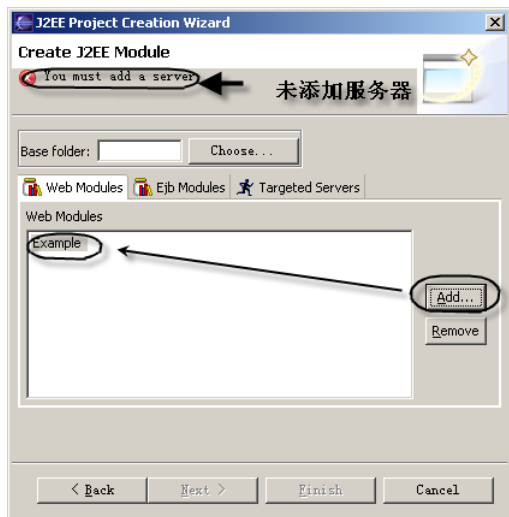


图 9.12 添加 Web 模块

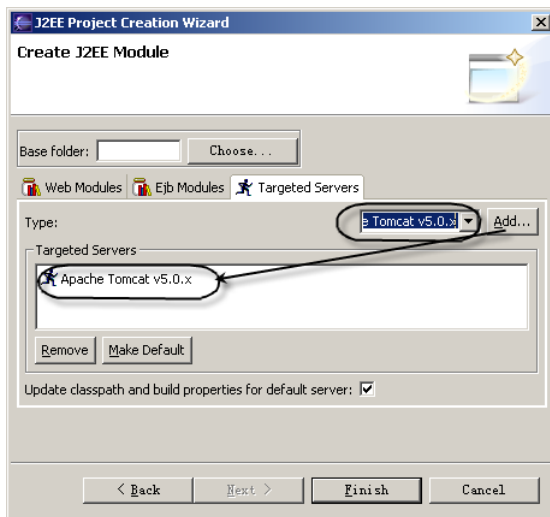


图 9.13 添加服务器

(8) 生成 MyExample 项目之后，可以很清晰地在 Eclipse 的导航栏中看到该项目的目录结构，如图 9.14 所示。

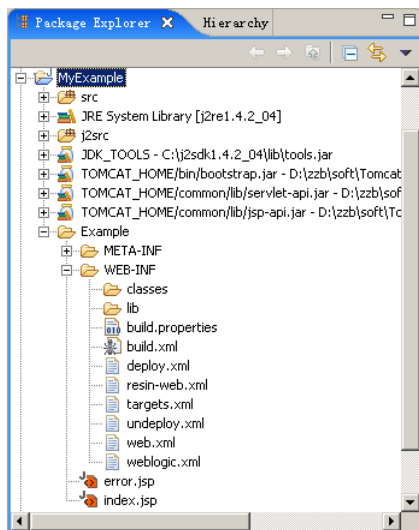


图 9.14 项目结构图

至此，就已经成功地使用 Eclipse+Lomboz 工具创建了该实例项目。从图 9.14 可以看出，Lomboz 已经自动生成了很多必备的配置文件和目录。

□ 在 Example 模块目录下有个 META-INF 目录。这目录下的 MANIFEST.MF 文件存储着各文件配

置信息，即记录着各个文件在什么位置可以查找到。

- ❑ 在 Web 模块 Example 根目录下有两个文件是创建 Web 模块时自动生成的两个 JSP 文件。
- ❑ 在 Example 目录下还有一个 WEB-INF 目录，这目录下的一些文件都是相关的配置文件。目录下的 class 目录是存放编译之后的 class 文件，lib 目录是存放 jar 包。另外 Register 根目录下还存在很多的配置文件，例如在 Web.xml 文件需要配置相关 Web 信息。
- ❑ MyExample 项目根目录下的 src 目录是存放 java 源文件的，bin 是 JSP 运行之后生成的 Servlet 等文件的输出目录。

9.4 本章小结

本章的内容虽然不是教会读者如何创建 JSP，但是熟练这些工具的使用，会大大方便整个 Web 的开发以及代码的维护。Eclipse 工具可以实现 Java 源码和编译后的二进制文件的分离（后面实例介绍时还会重点提及），Lomboz 可以很快架构起需要的 Web 站点，并且方便 Web 项目的发布。另外，使用 Lomboz 可以更加便捷地启动和停止 Web 服务器。至于这些工具的更多好处，在实际使用中再慢慢体会。