

第 2 章 搭建JSP运行环境

第一章简单介绍了 Web 应用程序的开发技术，但是在准备创建和开发一个动态网站之前，首先还必须安装和配置 Web 应用程序的运行环境、服务器以及相关开发工具。

首先需要安装 JDK，其提供 Java 程序的运行环境以及 Java 开发工具包。接下来需要安装 Web 容器，本书选择的是 Tomcat。Tomcat 容器负责解析、编译 JSP 页面程序，并对客户端请求作出响应。另外，还需要安装两个 JSP 编辑器，其中 Dreamweaver 为一个可视化的页面编辑器，可以很方便对页面结构进行设置和调整。UltraEdit 编辑器简单好使，是一个轻量级应用程序，类似与 Windows 操作系统自带的写字板，但是 UltraEdit 带有代码闪亮提示。在手写 JSP 代码时，推荐使用 UltraEdit 文本编辑器。

这一章主要介绍这些工具的安装和配置，在之后章节讲解中还会逐渐介绍其他辅助工具或者第三方插件。

本章要点包括以下内容：

- ☐ JDK 的安装和配置
- ☐ Web 容器 Tomcat 的安装和配置
- ☐ Dreamweaver 可视化页面编辑器的安装
- ☐ UltraEdit 文本编辑器的安装

2.1 安装和配置JDK

JDK 工具包是一种用于 Java 平台上发布的应用程序、Applet 以及组件提供的开发环境。它由 SUN 公司提供，其中包括 Java 程序运行所必须的环境以及开发常用库文件。在使用 JSP 开发动态网站之前，首先必须安装的就是 JDK 组件。

2.1.1 JDK的作用

JDK 提供 Java 运行环境 JRE（只有安装了 JRE 了，才能运行 Java 开发的应用程序）以及常用的开发工具包（例如 tools.jar）。其实仅仅只需安装 JRE 就可以运行 Java 程序了。这里之所以安装 JDK，是因为 JDK 已经包含了 JRE，并且还含有 Java 的源代码，可以随时查看这些源代码。这样在调试程序的时候非常有帮助。

2.1.2 安装JDK

JDK 可以去 SUN 公司的网站 <http://Java.sun.com/j2se/1.4.2/download.html> 下载。本书使用 JDK1.4.2_04 版本，下载文件全名为 J2sdk-1_4_2_04-windows-i586-p.exe，是 Windows 操作系统下的安装版本。具体安装过程如下：

（1）双击已经下载下来的 J2sdk-1_4_2_04-windows-i586-p.exe 安装文件，弹出“License Agreement”对话框，如图 2.1 所示。

(2) 选择 “I accept the terms in the license agreement” 选项，单击 “Next” 按钮，弹出 “Custom Setup” 对话框，如图 2.2 所示。

(3) 在该对话框中，选择性安装所需的 JDK 组件（显示共有四个）及其指定 JDK 安装目录。此处选择组件全部安装和默认的安装路径，即 JDK 将被安装在 C:\j2sdk1.4.2_04 目录下。单击 “Next” 按钮，弹出 “Browsers Registration” 对话框，如图 2.3 所示。

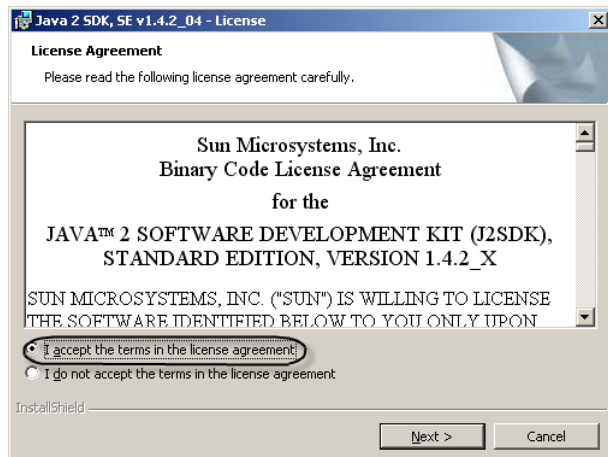


图 2.1 License Agreement 对话框

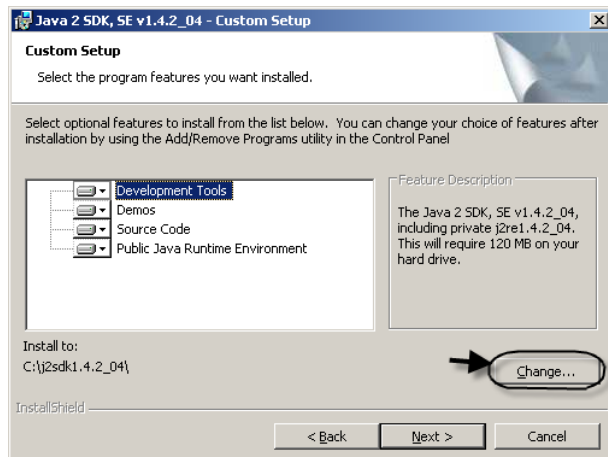


图 2.2 JDK 组件和安装路径的选择

(4) 选择需要注册到 JDK 的浏览器，此处选择默认的选项，然后单击 “Install” 按钮开始安装。

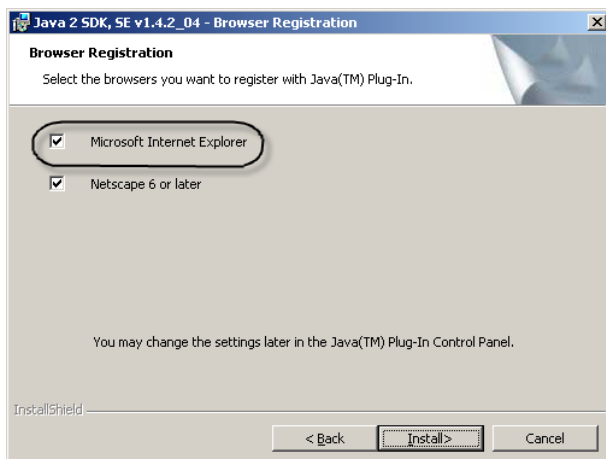


图 2.3 浏览器插件的选择

注意：在 JDK 安装完成之后系统一般需要进行重启才能有效。


至此，就已经成功地安装了 JDK。在 Windows XP 或者 2000 下安装该版本的 JDK 不需要再手动设置 JDK 的环境变量。但是有可能在 Linux 操作系统下或者安装其他版本的 JDK 时，需要手动设置 JDK 的 path、classpath 等环境变量

下一小节将具体介绍 JDK 环境变量的设置，但是这一步不是必须的，如果读者在安装了 JDK 之后发现执行 Java 程序时查找不到对应的 JDK 时，才按照下面步骤进行相应 JDK 设置。

2.1.3 配置JDK环境变量

现有版本的 JDK 一般不再需要手动进行环境变量的设置。JDK 安装后，就可以直接使用。但如果

安装错误或者其他原因,运行程序可能出现“找不到JDK”的提示错误。这时需要检查环境变量对JDK的设置是否有误,必要时需要手动配置JDK环境变量。配置步骤如下:

- (1) 右击“我的电脑”图标,在下拉菜单中选择“属性”命令,弹出“系统属性”对话框。
- (2) 选择“高级”选项卡,单击“环境变量”按钮,弹出“环境变量”对话框,如图2.4所示。

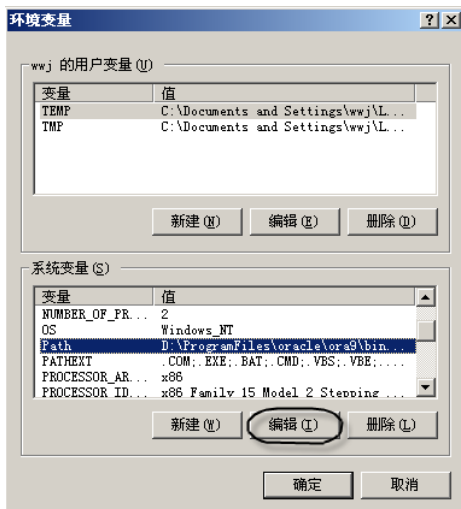


图 2.4 编辑 path 环境变量

(3) 选中“系统变量”列表框中的 path 变量,单击“编辑”按钮,弹出“编辑系统变量”对话框,如图2.5所示。

(4) 在“变量值”文本框中添加 C:\j2sdk1.4.2_04\bin (C:\j2sdk1.4.2_04 为 JDK 的安装目录),并以分号与前面变量值隔开。单击“确定”按钮,完成编辑。

(5) 在“环境变量”对话框中,单击“创建”按钮,弹出“新建系统变量”对话框,如图2.6所示。

(6) 在“变量名”文本框中输入 classpath,在“变量值”文本框中输入 C:\j2sdk1.4.2_04\lib\dt.jar;C:\j2sdk1.4.2_04\lib\tools.jar。具体设置要以 JDK 安装路径为准,此处的安装目录为 C:\j2sdk1.4.2_04。单击“确定”按钮,完成了JDK环境变量的配置。



图 2.5 编辑 path 变量

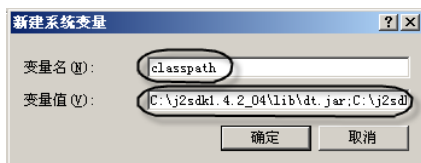


图 2.6 创建 classpath 变量

成功安装和配置JDK后,就可以编写Java程序并编译和执行。但是要运行JSP文件(Web容器会使其编译成Java Servlet文件再执行),还需要安装Web容器(包含JSP引擎),下一步将具体介绍其安装和配置步骤。

2.2 安装和配置Web容器Tomcat

安装了Web容器的主机一般作为Web服务器,对客户端请求进行相应处理,并返回响应结果。这其中包括对JSP页面程序的解析、编译以及执行等操作。商业性的Web服务器可以包括更多辅助功能(例如安全和远程调用等功能)。本书选择的是Tomcat容器,它是一个开源软件、操作方便,Web开

发人员一般选择它构建开发环境。下面详细讲解 Tomcat 容器的安装和配置过程。

2.2.1 Web容器Tomcat的作用

Web 容器 Tomcat 是 Servlet 的容器并提供 JSP 引擎，第一章已经对 Web 容器有所介绍。安装了容器的主机一般作为 Web 服务器，负责 JSP 文件执行和客户响应。JSP 首先会翻译成 Java Servlet 源文件（后缀.java），然后再编译成 Java Servlet 的二进制文件（后缀.class），Web 容器并根据客户请求来调用和执行相应的 JSP 文件。

只有把开发的 Web 应用放置在这个容器中，才能通过浏览器来访问。没有它，Web 应用程序无法和真正的网络服务连接起来。

2.2.2 安装Tomcat容器

下载 Tomcat 安装文件的网址为 [Http://jakarta.apache.org](http://jakarta.apache.org)。本书中使用的 Tomcat 版本为 5.0.30，下载后的全名为 jakarta-tomcat-5.0.30.exe。详细的安装过程如下：

（1）双击安装 Tomcat 的可执行文件 jakarta-tomcat-5.0.30.exe，弹出“Welcome to the Apache Tomcat Setup”对话框。

（2）单击“Next”按钮，将弹出“License Agreement”对话框，如图 2.7 所示。

（3）单击“I Agree”按钮，弹出“Choose Components”对话框，如图 2.8 所示。

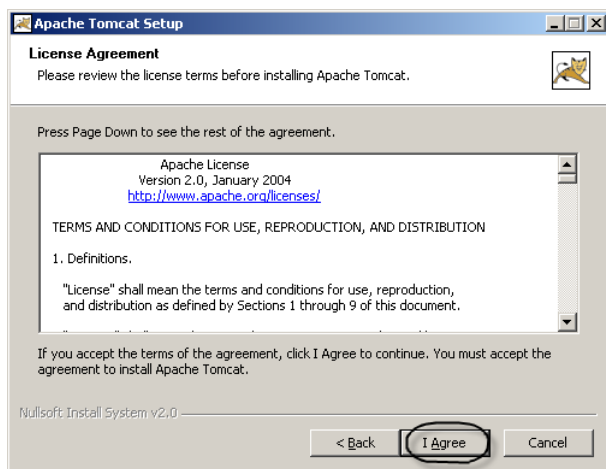


图 2.7 “License Agreement”对话框

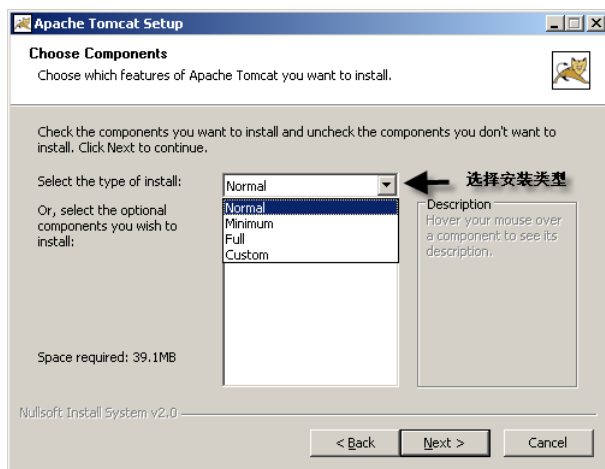


图 2.8 “Choose Components”对话框

（4）选择 Tomcat 安装类型，这里就选择默认的“Normal”选项。选择后，单击“Next”按钮，弹出“Choose Install Location”对话框，如图 2.9 所示。

（5）选择 Tomcat 容器的安装路径，此处将 Tomcat 安装在 D:\zzb\soft\Tomcat 5.0 目录下，读者可以单击“Browse”选项来更改 Tomcat 的安装路径。路径设定后，单击“Next”按钮，将会弹出“Configuration”对话框，如图 2.10 所示。

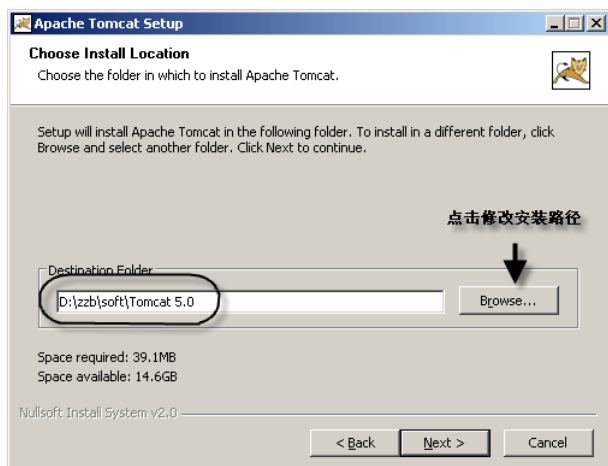


图 2.9 “Choose Install Location”对话框

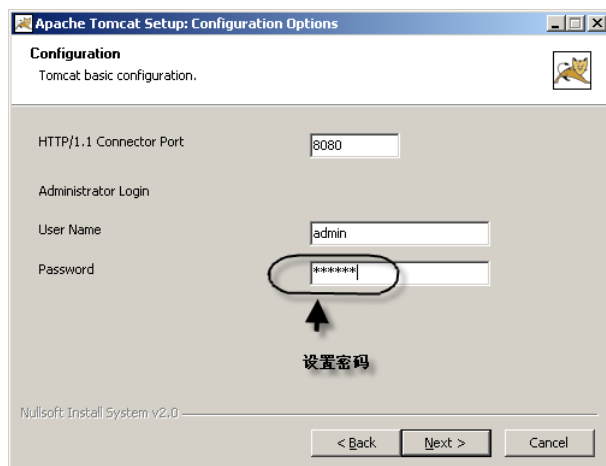


图 2.10 “Configuration”对话框

(6) 设置 Tomcat 服务器的端口号（访问 Web 服务器时，都是通过 IP 地址+端口号，IP 就相当于一户人家的详细地址，而端口则表示房子具体的门。Tomcat 的端口号默认为 8080）及其管理员密码。这里保持端口号 8080 以及用户名 admin 不变，仅在“Password”文本框中输入密码 123456。设置后，单击“Next”按钮，将弹出“Java Virtual Machine”对话框中，如图 2.11 所示。

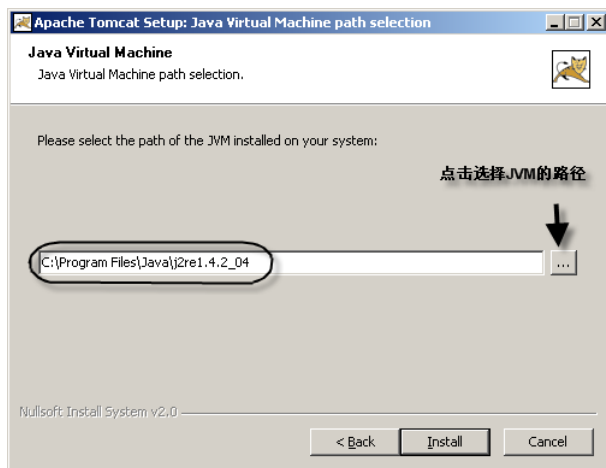


图 2.11 “Java Virtual Machine”对话框

(7) 单击右边按钮设置 Tomcat 需要使用到的 JVM 路径，系统一般会自动查找到 JVM 的位置，很少需要手动去设置。确认路径设置正确无误，单击“Install”按钮，进行 Tomcat 安装。

至此，就成功完成了 Tomcat 容器的安装。下一步将介绍如何启动和停止 Tomcat 服务器以及如何根据特定的需求对 Tomcat 进行相应的配置。当然用户可以使用 Tomcat 的默认设置，例如 Tomcat 默认的 Web 应用程序的调用路径为 Tomcat 5.0\webapps，如果用户想调用别的目录下的 Web 应用，则需要进行相应的配置。下面首先介绍 Tomcat 服务器的启动和停止方法。

2.2.3 启动和停止Tomcat服务器

Tomcat 服务器提供了两种方法来启动和停止服务器。方法如下：


2.2.3.1 通过服务启动和停止 Tomcat 服务器

Tomcat 服务器成功安装后，就可以到“控制面板”|“管理工具”|“服务”（如果控制面板不是经典视图，则为“控制面板”|“性能和维护”|“管理工具”|“服务”）中查看 Tomcat 的进程状态（Tomcat 安装完之后，就会作为 Windows 操作系统的一个服务存在），并可以通过右击“Apache Tomcat”列表选项手动地对 Tomcat 进行“启动”和“停止”操作，如图 2.12 所示。



图 2.12 通过服务启动和停止 Tomcat 服务器

2.2.3.2 通过图标命令启动和停止 Tomcat 服务器

除上面介绍的启动和停止方法，还可以单击 Windows 的“开始”|“所有程序”|“Apache Tomcat5.0”|“Monitor Tomcat”命令。这时，在 Windows 界面的右下角会出现 Apache Tomcat 的图标。右击 Tomcat 图标，在下拉菜单中有“Start Service”、“Stop Service”命令，如图 2.12 所示。使用这两个命令，同样可以进行 Tomcat 的“启动”和“停止”操作。

在下拉菜单中，还有一个“Configure Tomcat”命令。单击该命令，弹出如图 2.13 所示的 Tomcat 配置界面。此对话框包含了 Tomcat 的很多参数设置，但一般都不需要去修改。在 Tomcat 配置界面中，也可以单击“Start”或者“Stop”按钮来启动或者停止 Tomcat 服务器。

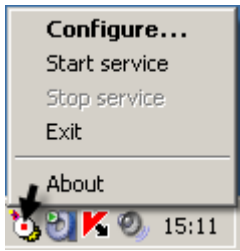


图 2.13 Tomcat 右键菜单

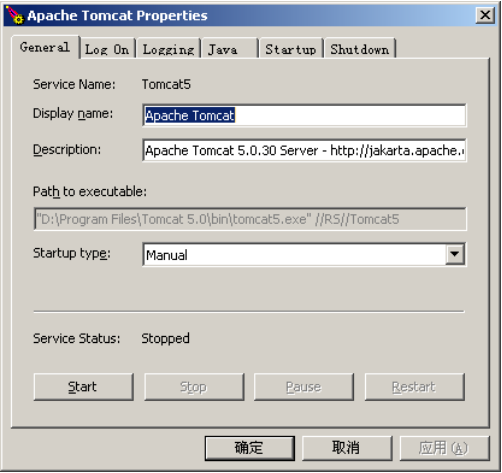



图 2.14 Tomcat 配置界面

2.2.4 检测Tomcat服务器启动的方法

在成功安装完 Tomcat 容器，就可以将开发的 Web 应用程序放在指定的位置，然后通过 IE 进行请求访问。但有时需要检测 Tomcat 是否已经成功启动，可以有如下多个方法：

2.2.4.1 Tomcat 图标显示

通过以上介绍的 Tomcat 启动方法启动服务器，如果服务器启动成功，在 Window 界面的右下角，

Tomcat 图标显示  图案。

2.2.4.2 系统服务中查看 Tomcat 服务器状态

如果 Tomcat 服务器启动成功，在 Windows 的“服务”窗口中，Tomcat 进程显示“启动”的状态，如图 2.15 所示。

名称	描述	状态	启动类型	登录为
Alerter	通知所...		手动	本地服务
Apache Tomcat	Apache ...	已启动	手动	本地系统

图 2.15 Tomcat 的服务状态

2.2.4.3 通过 IE 浏览器查看 Tomcat 服务器是否启动成功

打开 IE 浏览器，在地址栏中输入 `http://localhost:8080` 或者 `http://127.0.0.1:8080`（这里 localhost 和 127.0.0.1 都是指定本机）。此处 Tomcat 服务器的端口号为 8080（类似于指定房子的某扇门）。如果 Tomcat 已经成功启动，则会出现图 2.16 所示界面。如果没有出现该界面，则说明安装不成功或者 Tomcat 没有启动成功，读者需要检查 Tomcat 的安装过程是否正确以及启动状态。

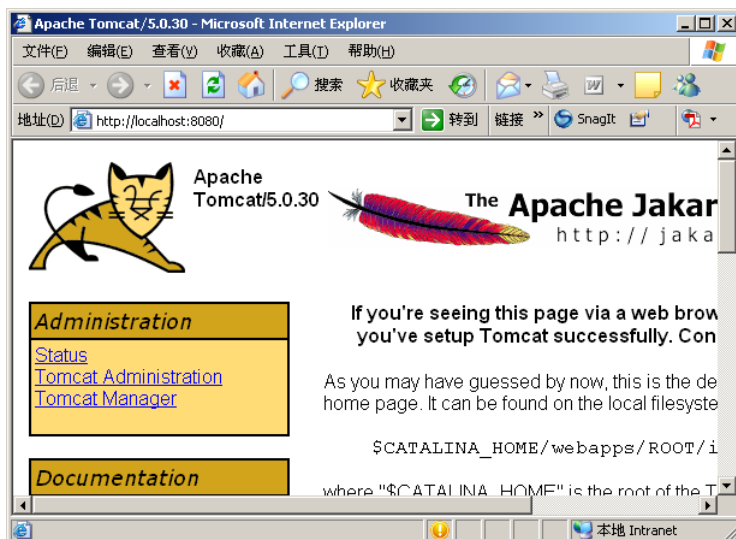


图 2.16 Tomcat 启动成功界面

通过以上三种可以查看 Tomcat 服务器的启动状态。一般 Tomcat 服务器安装完成后，会使用第三种方法，即使用 IE 浏览器来查看 Tomcat 的安装和启动是否正确。另外还可以通过如图 2.16 所示的界面来对 Tomcat 服务器进行相应的配置和管理，具体方法下面会逐一介绍。

2.2.5 Tomcat 服务器端口号的配置

用于 Tomcat 服务器配置的文件主要有 Tomcat 5.0\conf 目录下的 `server.xml` 和 `web.xml` 配置文件。下面将介绍使用 `server.xml` 配置文件更改 Tomcat 服务器的端口号。在介绍之前，先让读者了解什么是端口号，以及它存在的作用。

2.2.5.1 端口号的概念和作用

在网络应用中会经常听到端口这个概念。例如，如果一台服务器上安装了多个 Web 服务器（例如 Tomcat、Resin 或者 WebLogic），为了避免多个 Web 服务器中的模块名称相同而起冲突，这里需要用

端口号来区分这多个 Web 服务器的使用。Tomcat 默认使用的端口号是 8080，Resin 的默认就是 80。可以很形象地将 IP 地址比喻成某户人家的家庭地址，而端口号再表示该房子中某房间的门。

2.2.5.2 Tomcat 端口号的修改

可以在 Tomcat 安装过程中指定服务器的端口号（Tomcat 服务器默认为 8080），如图 2.10 所示。除此之外，还可以在 server.xml 配置文件（在 Tomcat 5.0\conf 目录下）中修改 Tomcat 的端口号。配置端口号的一段代码如下：

```
.....
<Connector
<port="8080"                maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
                             enableLookups="false" redirectPort="8443" acceptCount="100"
                             debug="0" connectionTimeout="20000"
                             disableUploadTimeout="true" />
.....
```

代码说明：Tomcat 默认的端口号为 8080，读者也可以根据需要进行修改。把上面代码有下划线的部分修改为：port="8090"，则调用 Hello 模块的地址变为：http://localhost:8090/hello 或者 http://127.0.0.1:8090/hello。代码中有关其他属性的配置，读者先不需要了解。

2.3 安装Dreamweaver编辑工具

Dreamweaver 是在多媒体方面颇有建树的 Macromedia 公司开发出来的可视化页面制作工具。使用 Dreamweaver 开发页面文件，效率比较高，因为它对页面格式进行调整和修改都非常方便。

本书将要安装的是 Dreamweaver MX 2004 版本，它现在不仅仅是简单的页面编辑软件，已经具备了完整的数据库脚本的可视化开发，并且能够支持的动态脚本已经包括了当前最为流行的所有格式：ASP/JSP/ASP.NET/PHP/COLDFUSION。有了 Dreamweaver MX，开发人员可以在一个环境中快速地创建和开发 Web 页面。

Dreamweaver 安装比较容易，只要选择默认安装即可，所以这里不再具体介绍。Dreamweaver MX 2004 或者其他更高版本的安装文件可以到 Macromedia 公司门户网站下载。

2.4 安装UltraEdit文本编辑器

UltraEdit 文本编辑器相当于 Windows 操作系统自带的写字板，但是它的功能要强大得多，提供代码闪亮功能，更容易对代码、文件的管理。使用 UltraEdit 文本编辑器更加适合手写代码的工作人员。UltraEdit 文本编辑器在后面实例开发过程会经常使用，它的安装和使用也都比较简单，这里就不详细介绍。

2.5 本章小结

本章介绍了开发 JSP 网站之前所要做的准备工作，其中包括 JSP/Java 的运行环境以及开发工具。主要介绍了 JDK、Web 容器 Tomcat、Dreamweaver 以及 UltraEdit 编辑器的安装和配置。

JDK、Web 容器 Tomcat 是开发 JSP 网站所必需的。JDK 提供了 Java 运行环境 JRE 以及常用的库文件。Web 容器是 HTML 以及 Servlet 的大容器，使得 Web 应用程序和真正的网络服务连接起来。当然市

场除了 Tomcat 容器，还有很多产品，如 Resin、JBoss 以及 Weblogic 和 Websphere，Resin 和 JBoss。这些产品都是开放源代码的。JBoss 除了嵌入了 Web 容器 Tomcat，并且还提供了 EJB 容器。Weblogic 和 Websphere 都属于商用服务器，使用在大型或者超大型应用系统中。鉴于本书主要讲解的是 J2EE 的 Web 模块部分，不牵涉到 EJB 模块开发，所以 Tomcat 是比较理想的选择。

最后介绍两种用于开发页面程序的编辑器，一种为可视化编辑器 Dreamweaver MX，方便对页面格式进行调整和修改；另一种编辑器为 UltraEdit，适用于手写页面程序。工欲善其事，必先利其器。了解这一章节介绍的内容后，下一章节将开始创建第一 JSP 应用程序。