

JSP 技术概述

JSP 全称 Java Server Pages,它是由 Sun Microsystems 公司于 1999 年 6 月推出的动态网页制作技术,是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术。

1.1 JSP 技术特点

在早期,开发网络数据库应用程序主要采用 CGI 技术(Common Gateway Interface,公共网关接口)。编写 CGI 程序可以使用不同的程序语言,如 Perl、Delphi 或 C/C++。虽然 CGI 技术已经发展成熟而且功能强大,但由于其具有编程困难、效率低下、修改复杂等缺陷,已不能满足 Web 开发的需要,于是新的技术纷纷面世,如 Microsoft 的 ASP(Active Server Pages),Tcx 的 PHP(PHP: Hypertext Preprocessor),Sun 的 Java/JSP/Servlet 等。其中 Sun 的 Java/JSP/Servlet 技术被许多人认为是未来最有发展前途的动态网站技术。

JSP 技术使用 Java 编程语言编写类 XML 的标记和脚本代码,来封装产生动态网页的处理逻辑。网页还能通过标记和脚本代码访问存在于服务端的资源的应用逻辑。JSP 将网页逻辑与网页设计和显示分离,支持可重用的基于组件的设计,使基于 Web 的应用程序的开发变得迅速和容易。

Web 服务器在遇到访问 JSP 网页的请求时,首先执行其中的程序段,然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户。插入的 Java 程序段可以操作数据库、重新定向网页等,以实现建立动态网页所需要的功能。

JSP 与 Java Servlet 一样,是在服务器端执行的,通常返回该客户端的就是一个 HTML 文本,因此客户端只要有浏览器就能浏览。

JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端请求以后对这些 Java 代码进行处理,然后将生成的 HTML 页面返回给客户端的浏览器。Java Servlet 是 JSP 的技术基础,而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成。JSP 具备了 Java 技术的简单易用,完全的面向对象,具有平台无关性且安全可靠,主要面向因特网的所有特点。

1.2 为什么使用 JSP

1. 将内容的生成和显示进行分离

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或者 XML 标记来设计和格式化最终页面。使用 JSP 标记或者小脚本来生成页面上的动态内容(内容是根据请求来变化的,例如请求账户信息或者特定的一瓶酒的价格),生成内容的逻辑被封装在标记和 JavaBean 组件中,并且捆绑在小脚本中,所有的脚本在服务器端运行。如果核心逻辑被封装在标记和 Bean 中,那么其他人,如 Web 管理人员和页面设计者,能够编辑和使用 JSP 页面,而不影响内容的生成。

在服务器端,JSP 引擎解释 JSP 标记和小脚本,生成所请求的内容(例如,通过访问 JavaBean 组件,使用 JDBC 技术访问数据库,或者包含文件),并且将结果以 HTML(或者 XML)页面的形式发送回浏览器。这有助于作者保护自己的代码,而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 生成可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件(JavaBean 或者 Enterprise JavaBean 组件)来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件,或者使得这些组件为更多的使用者或者客户团体所使用。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

3. 采用标记简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能,这些功能是在易用的、与 JSP 相关的 XML 标记中进行动态内容生成所需要的。标准的 JSP 标记能够访问和实例化 JavaBean 组件,设置或者检索组件属性,下载 Applet,以及执行用其他方法更难于编码和耗时的功能。

通过开发定制化标识库,JSP 技术是可以扩展的。今后,第三方开发人员和其他人员可以为常用功能创建自己的标记库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标记一样的执行特定功能的构件来工作。

4. Servlet 功能

与 Servlet 相比,JSP 能提供所有 Servlet 功能,但它比用 `println` 书写和修改 HTML 更方便。此外,还可以更明确地进行分工,Web 页面设计人员编写 HTML,只须留出地方让 Servlet 程序员插入动态部分即可。

5. 健壮的存储管理和安全性

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的,而且所有的 JSP 页面都被编译成为 Java Servlet,JSP 页面就具有 Java 技术的所有优点,包括健壮的存储管理和安全性。

6. 一次编写,随处运行

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,随处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中,用户可以使用自己所选择的服务器和工具,更改工具或服务器并不影响当前的应用。

1.3 所需软件

一般说来,JSP 应用开发需要运行环境和开发工具两个部分。运行 JSP 应用的软件一般称为应用服务器,也称为 JSP 容器。与 Java 其他技术一样,JSP 技术同样是跨平台的技术。也就是说,在不同硬件平台下,不同操作系统下,都可以支持 JSP 的开发和应用。本书选择 Windows 操作系统平台下的免费的 Java 开发工具箱 JDK 和 JSP 引擎 Apache Jakarta Tomcat 来搭建 JSP 开发平台,而 JSP 源代码的编辑在基础部分使用通用的文本编辑器,在 JSP 高级技术部分将会介绍 Eclipse+MyEclipse 的使用,这将会大大加快 JSP 的开发速度。

(1) JDK

JDK(Java Development Kit,Java 开发工具箱)是 Sun Microsystems 针对 Java 开发人员的产品。自从 Java 推出以来,JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。不同的 Java 应用开发对应应有 3 个不同的版本:

- ① JavaSE(Java Platform Standard Edition,Java 标准版);
- ② JavaME(Java Platform Micro Edition,Java 微型版);
- ③ JavaEE(Java Platform Enterprise Edition,Java 企业版)。

其中 Java SE 是标准的开发版本,可开发 Java 相关的几乎所有软件。Java ME 是针对移动设备、嵌入式设备的应用。Java EE 则是基于 Java SE 的企业版,增加了对大型商业应用的支持。JSP 技术就是 Java EE 技术中的一个组成部分。

(2) Tomcat

Tomcat 是 Sun 的 JSWDK(Java Server Web Development Kit)中 JSP 和 Servlet 的运行环境,同时它也具有 Web 服务器的功能,作为 JSP 引擎,它负责接受浏览器客户端的 Web 请求,将请求传送给 JSP Web 应用进行处理,并将处理结果(响应)返回浏览器客户端。

1.4 安装与配置 JDK

JSP 引擎需要 JRE(Java Runtime Environment,Java 运行时环境)的支持才能运行 JSP 应用程序。JRE 包含 Java 虚拟机、类库及其他文件,可支持以 Java 编写的程序。JDK 是 Java 的软件开发工具,是 Java 应用程序的基础。JSP 是基于 Java 技术的,所以配置 JSP 环境之前必须要安装 JDK。

1.4.1 下载 JDK 1.5

最新版本的 JDK 可以通过访问 <http://java.sun.com> 来下载,是由 Sun Microsystems

公司免费提供的 Java 开发平台下载。在本书使用的实例中,采用的是 JDK 1.5 版本。下面详细说明 JDK 1.5 安装配置过程。

1.4.2 安装 JDK 1.5

安装 JDK 1.5 的具体操作步骤如下。

(1) 下载完 JDK 后,双击该可执行文件。这时将出现如图 1-1 所示的界面,提示正在释放安装需要的文件。

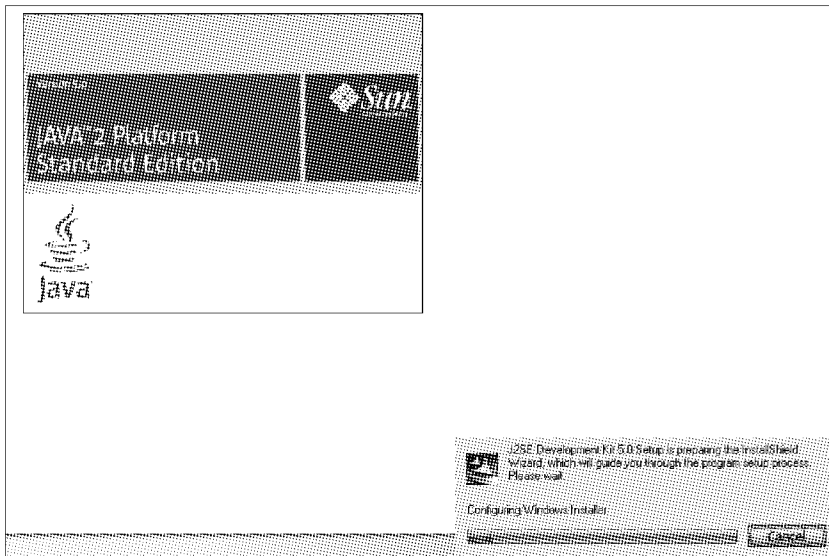


图 1-1 释放 JDK 安装文件

(2) 释放完安装文件后,将打开安装许可证书界面,如图 1-2 所示。

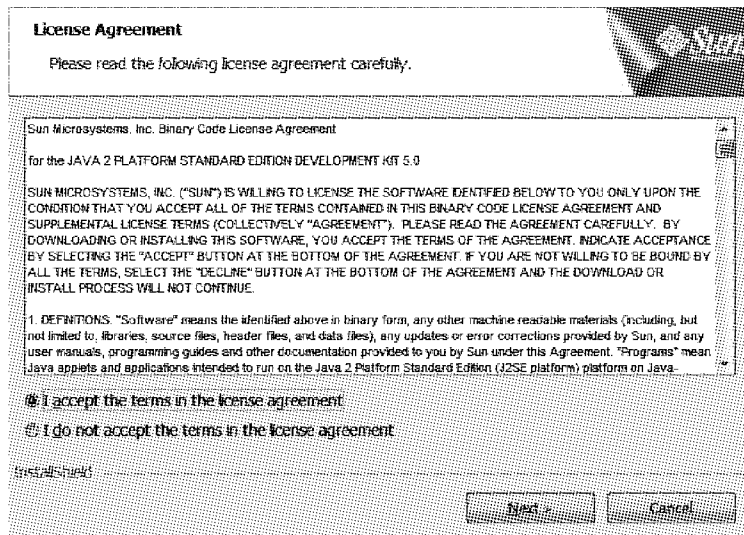


图 1-2 安装许可证书界面

(3) 选择 I accept the terms in the license agreement 单选框,单击 Next 按钮,进入自定义安装界面,如图 1-3 所示。

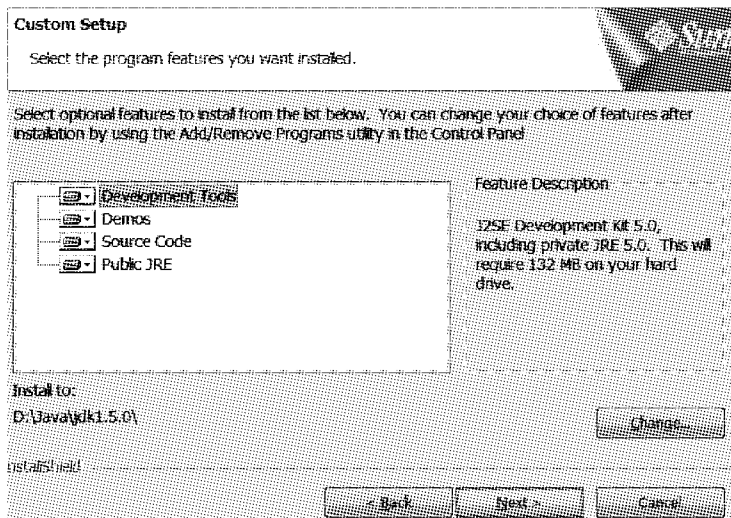


图 1-3 自定义安装

(4) 用户可以根据需要选择安装的内容和安装的路径。在这里,选择 D:\Java\jdk1.5 作为 JDK 1.5 的安装路径。请读者一定要注意,在一般情况下安装路径最好不要留有空格,因为这将会给以后的编译调试带来不必要的错误和麻烦,例如类库路径定位的问题。

(5) 然后单击 Next 按钮,开始安装 JDK,如图 1-4 所示。

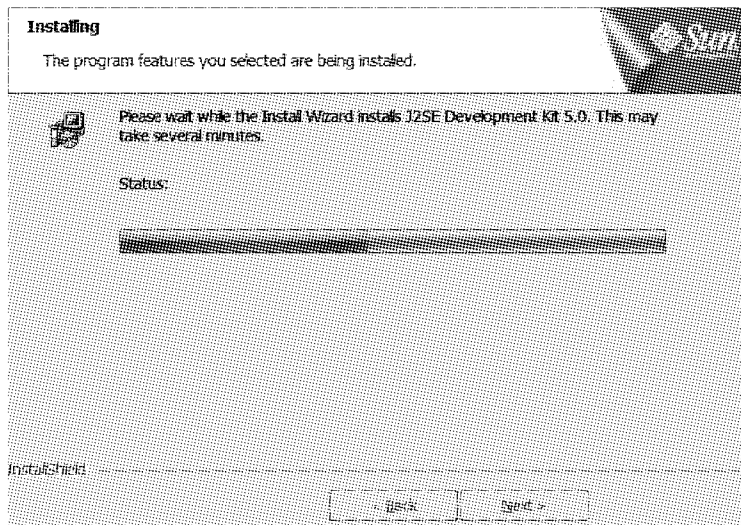


图 1-4 开始安装 JDK

(6) 当安装快结束时,会出现安装 JRE 的对话框,如图 1-5 所示。

(7) 选择需要安装的内容和安装的路径(最好是和 JDK 的安装路径在同一目录下,

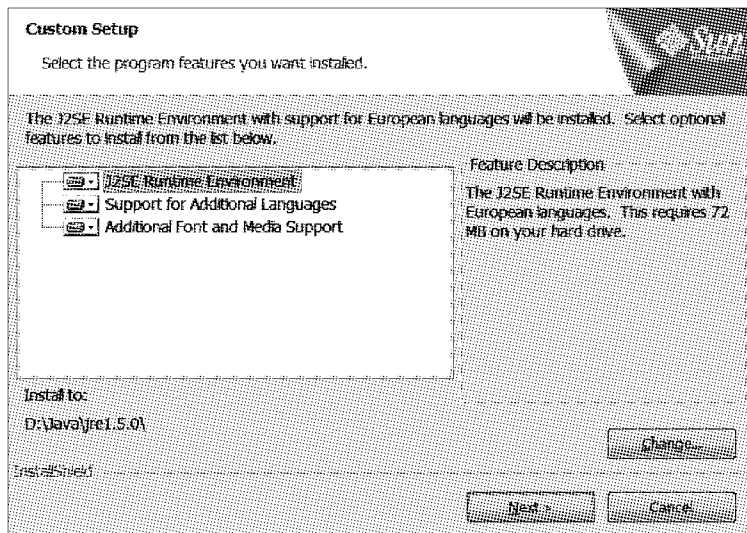


图 1-5 JRE 安装对话框

以方便以后查找)。单击 Next 按钮,会出现浏览器注册对话框,让用户选择是否让 Internet Explorer 默认支持 Java,如图 1-6 所示。

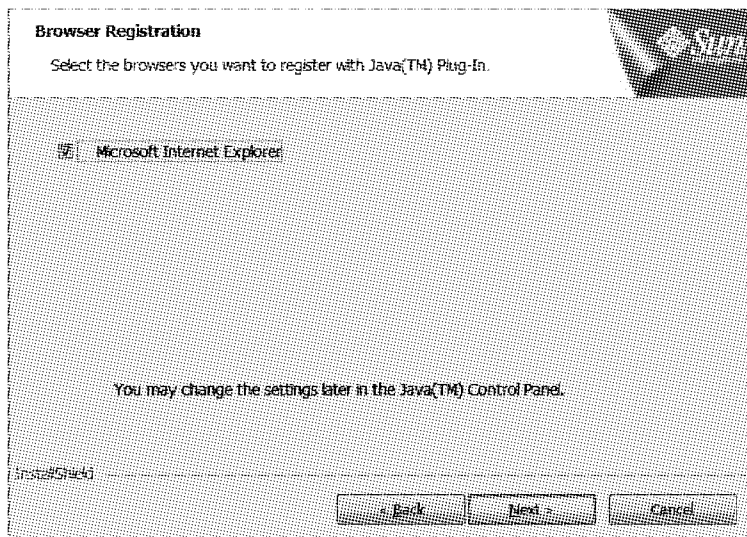


图 1-6 浏览器注册

(8) 选中 Microsoft Internet Explorer 复选框,单击 Next 按钮,进入 JRE 的具体安装,如图 1-7 所示。

(9) 安装完 JRE 后,出现 Installation Completed 对话框,如图 1-8 所示。

单击 Finish 按钮,JDK 安装结束。但如果要使用 JDK 来开发 Java 程序,还得配置一下环境变量才能使用,下面介绍如何配置 JDK。

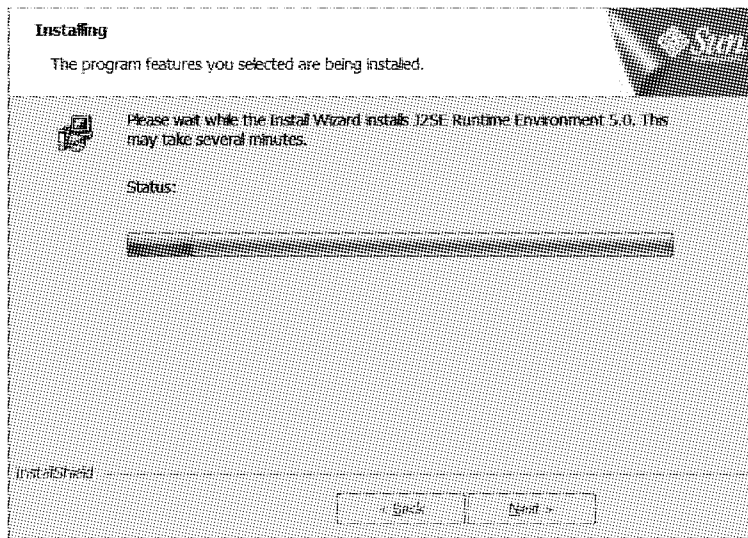


图 1-7 安装 JRE

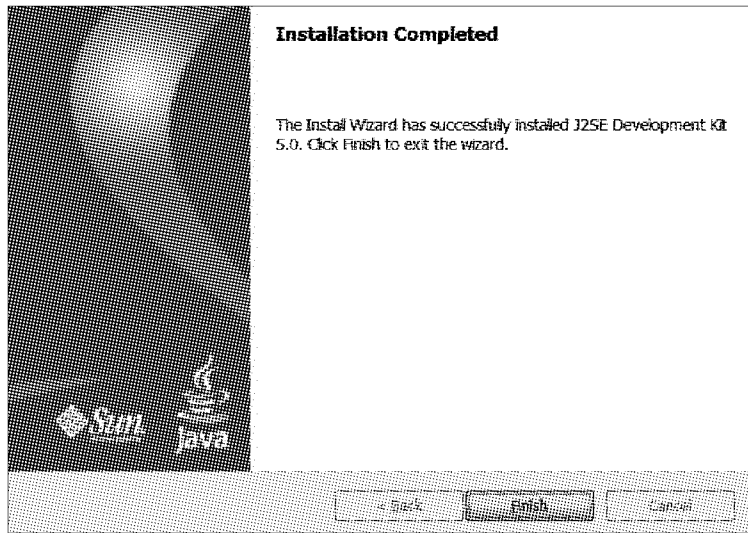


图 1-8 JDK 安装结束

1.4.3 JDK 环境变量设置

JDK 环境变量设置的具体步骤如下。

(1) 右击“我的电脑”，在快捷菜单中选择“属性”命令，打开“属性”对话框，选择“高级”选项卡，如图 1-9 所示。

(2) 单击“环境变量”按钮，打开如图 1-10 所示环境变量列表框。

在“Administrator 的用户变量”列表框里新建 3 个属性：

- ① JAVA_HOME: D:\Java\jdk1.5.0(刚才 JDK 的安装路径)；
- ② PATH: D:\Java\jdk1.5.0\bin(JDK 可执行文件的路径)；

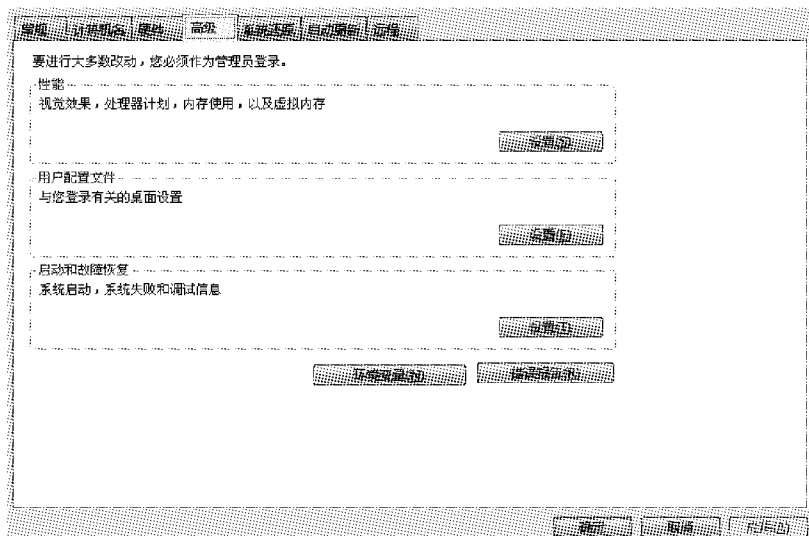


图 1-9 “高级”选项卡

③ CLASSPATH: . ; D:\Java\jdk1.5.0\lib\tools.jar; D:\Java\jdk1.5.0\lib\dt.jar。

(3) 单击“新建”按钮,弹出“新用户变量”对话框,输入如图 1-11 所示的变量名和变量值,完成用户变量的添加。

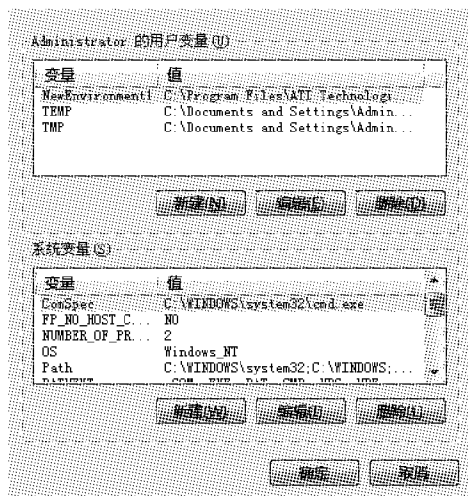


图 1-10 环境变量设置

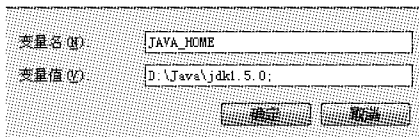


图 1-11 设置参数

设置好 JDK 的环境变量后,做一个简单的 Java 例子来测试一下 JDK 是否正常工作。最经典的莫过于 HelloWorld 了。打开记事本程序,输入如下代码:

```
import java.io.*;
public class HelloWorld{
    public static void main(String[] args){
```

```
        System.out.println("Hello World");  
    }  
}
```

将程序保存为 HelloWorld.java, 打开命令提示符, 进入到 HelloWorld.java 所在目录下, 输入如下命令:

```
Javac HelloWorld.java  
Java HelloWorld
```

如果能出现“Hello World”的字样, 就说明 JDK 配置成功。下面就可以进行 Web 服务器的安装配置了。

1.5 安装与配置 Tomcat

Tomcat 是一个免费的开源的 Servlet 容器和 JSP 引擎, 它是 Apache 基金会的 Jakarta 项目中的一个核心项目, 由 Apache、Sun 和其他一些公司及个人共同开发而成。

1.5.1 下载 Tomcat 5.5

不同版本的 Tomcat 可以到其官方网站 <http://tomcat.apache.org> 下载。本书所用的 Tomcat 版本为 5.5.1.7。下面将详细介绍 Tomcat 的安装和配置过程。

1.5.2 安装 Tomcat 5.5

安装 Tomcat 的具体步骤如下。

(1) 下载完 Tomcat 后, 文件名为 apache-tomcat-5.5.17.exe, 双击运行该文件启动 Tomcat 的安装过程。开始界面如图 1-12 所示。



图 1-12 启动 Tomcat 安装

(2) 单击 Next 按钮, 出现 Tomcat 安装的许可协议, 如图 1-13 所示。

(3) 单击 I Agree 按钮进入 Tomcat 的安装类型选择界面, 如图 1-14 所示。

(4) 在 Select the type of install 下拉列表框中选择 Full, 单击 Next 按钮, 进入

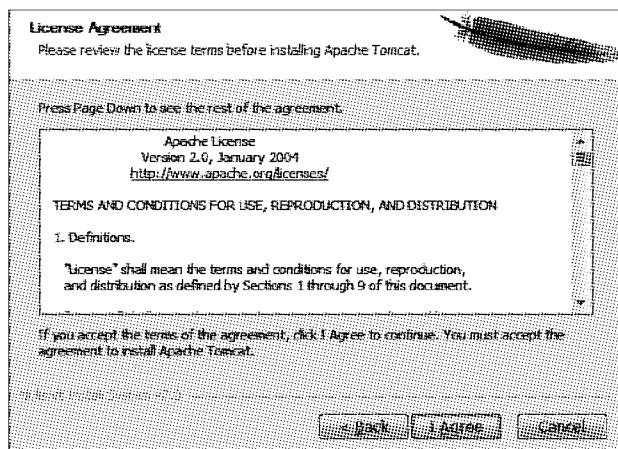


图 1-13 Tomcat 许可协议

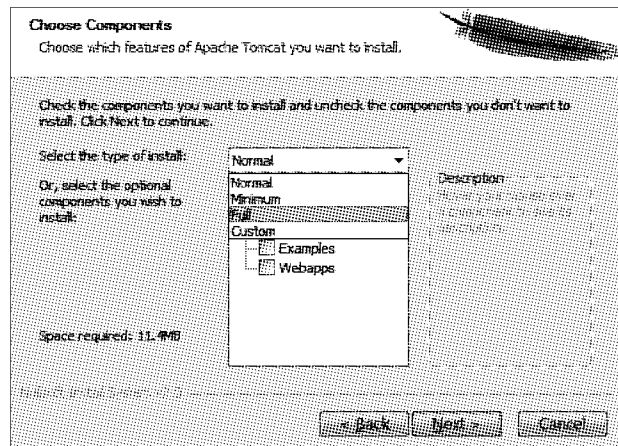


图 1-14 Tomcat 安装类型选择

Tomcat 安装位置的选择对话框,这里选择 D:\Tomcat5.5 作为安装目录,如图 1-15 所示。

(5) 单击 Next 按钮,进入 Tomcat 的基本配置对话框。其中 HTTP/1.1 连接端口是 Tomcat 提供 HTTP 连接的网络端口,默认为 8080。管理员登录设置是管理员配置和运行 Tomcat 需要进行的安全验证,这里你可以自己随便设置用户名和密码。为了方便,本例将用户名和密码都设为 admin,如图 1-16 所示。

(6) 单击 Next 按钮,进入 Java 虚拟机的路径选择对话框。安装程序会自动找到前面所安装 JDK 时安装的 JRE 目录,如图 1-17 所示。

(7) 单击 Install 按钮,便开始 Tomcat 组件安装,如图 1-18 所示。

(8) 安装结束后,选中 Run Apache Tomcat 复选框,单击 Finish 按钮,如图 1-19 所示。

(9) 这时系统将打开如图 1-20 所示的界面,提示正在启动 Tomcat 服务。

(10) Tomcat 服务启动后,系统托盘中会出现一个知识图标,双击该图标,会出现如

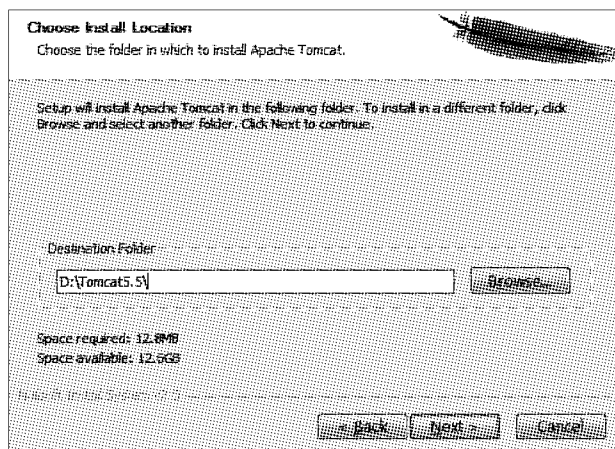


图 1-15 选择 Tomcat 安装位置

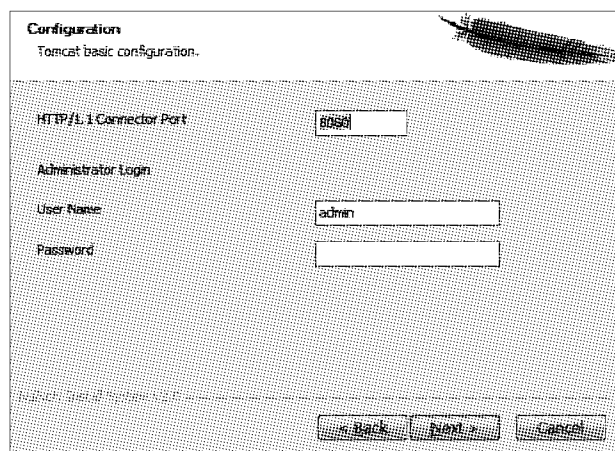


图 1-16 Tomcat 基本配置

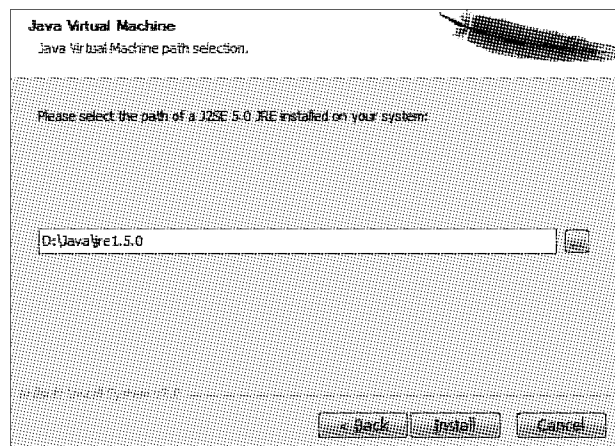


图 1-17 安装 Java 虚拟机

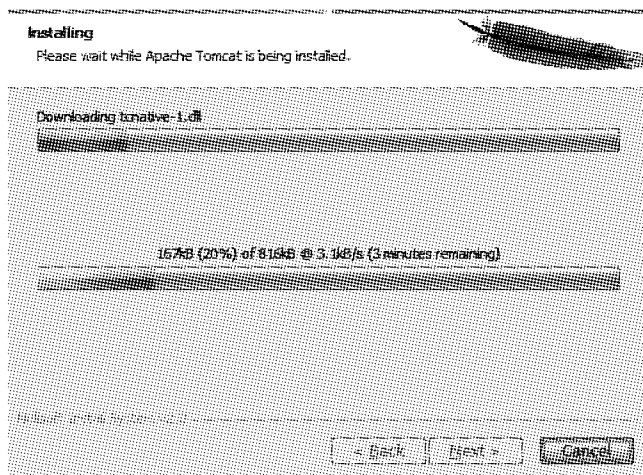


图 1-18 Tomcat 安装进度

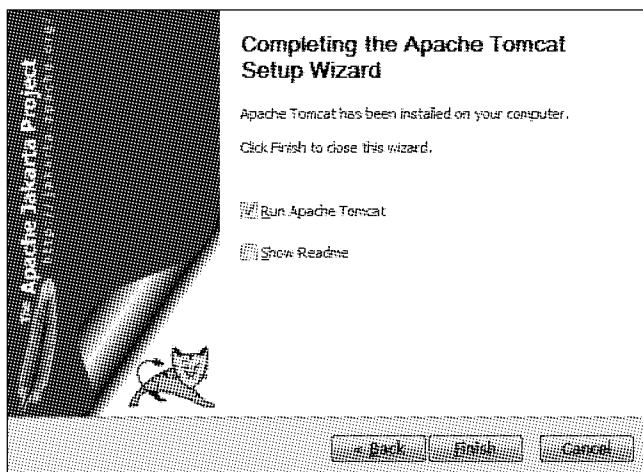


图 1-19 安装结束后选择运行 Tomcat

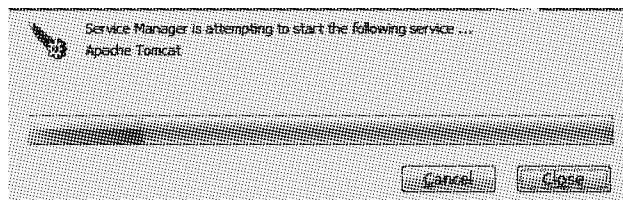


图 1-20 Tomcat 服务启动

图 1-21 所示的管理对话框。

(11) 在该对话框中,用户可以选择启动、停止、暂停、重启 Tomcat 服务,此外,用户还可以进行其他的配置。这里就不一一详述了。

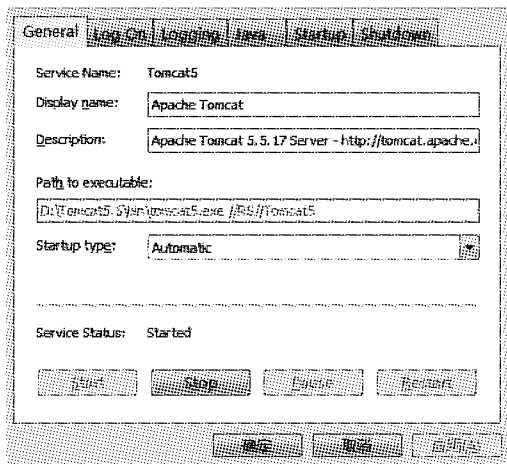


图 1-21 Tomcat 管理

1.5.3 测试 Tomcat

安装完 Tomcat 后,就可以测试一下 Tomcat 是否可以正常工作。

前一节用户已经知道如何启动 Tomcat 服务了,所以在 Tomcat 服务启动后,打开浏览器,在地址栏输入 `http://localhost:8080`,可以打开如图 1-22 所示的页面。

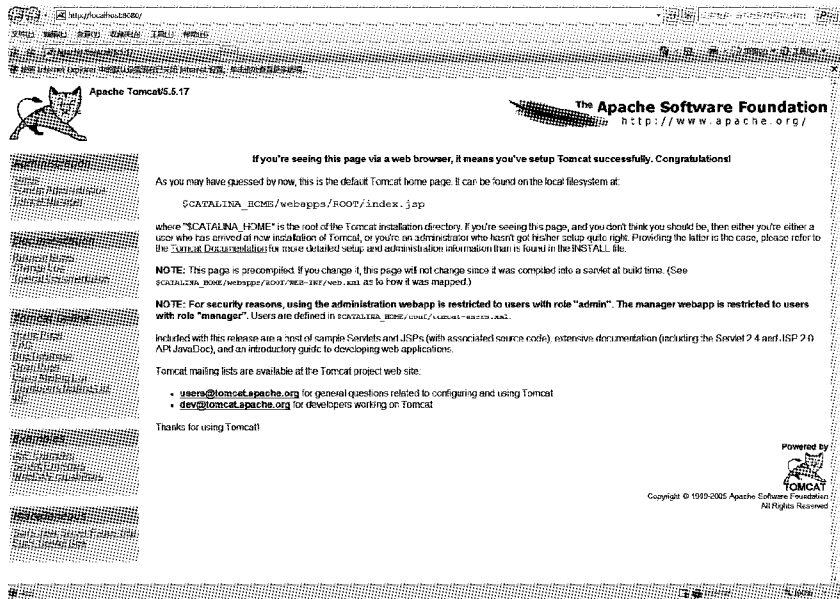


图 1-22 Apache Tomcat 首页

单击页面左边的 JSP Examples 链接,会出现 JSP Samples 页面,检查实例程序是否能够运行,如图 1-23 所示。

执行 Date 实例,将会出现如图 1-24 所示的页面,表明 Tomcat 已经成功安装并运行正常。

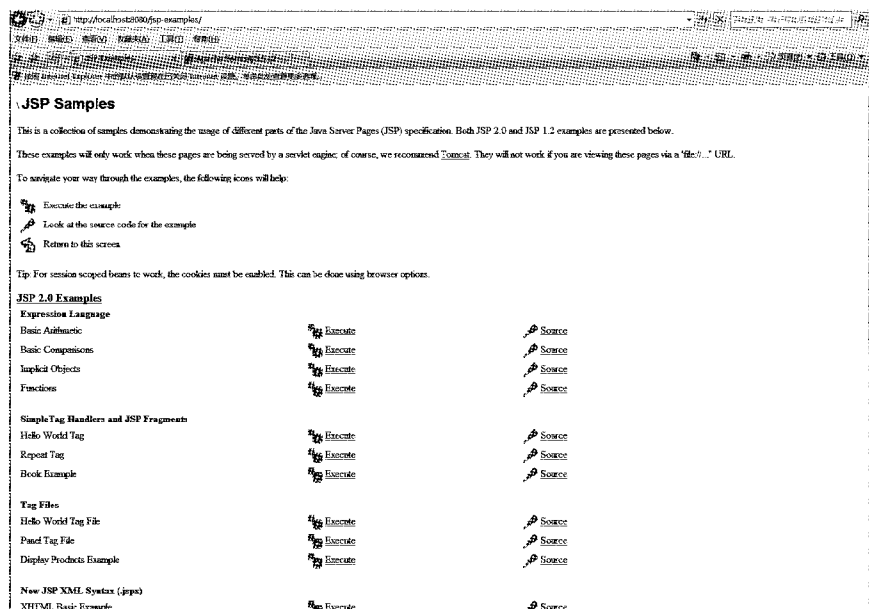


图 1-23 JSP Examples 页面

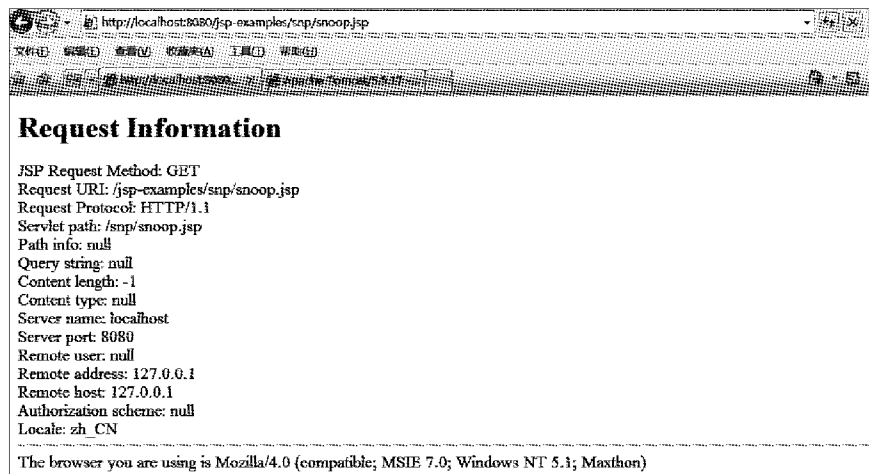


图 1-24 Date 实例

1.6 JSP 开发工具——MyEclipse

1.6.1 MyEclipse 安装与配置

在使用 MyEclipse 之前,先来介绍如何安装和配置 MyEclipse 插件。

安装 MyEclipse 插件之前,首先到其官方网站(<http://www.myeclipseide.com>)上下载安装文件,如图 1-25 所示。

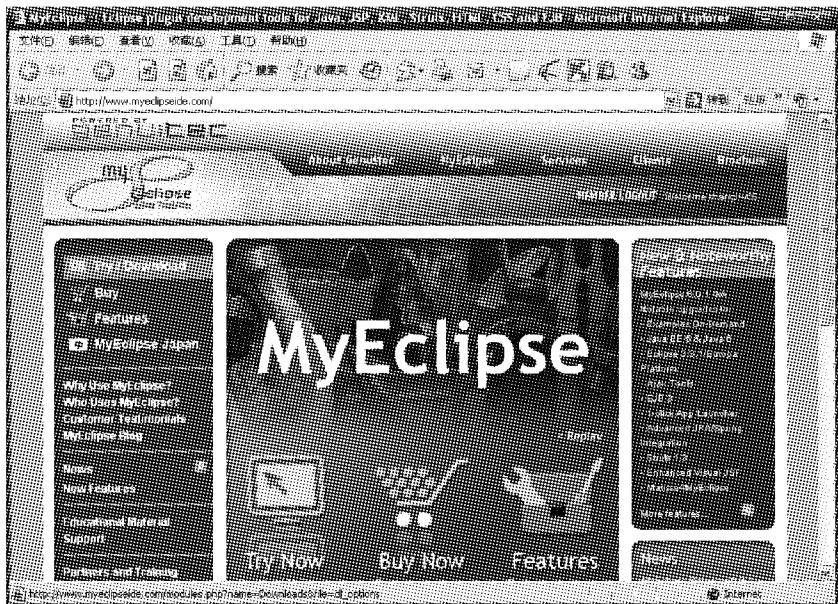


图 1-25 MyEclipse 官方网站

单击 Try/Download 链接,打开如图 1-26 所示的页面。

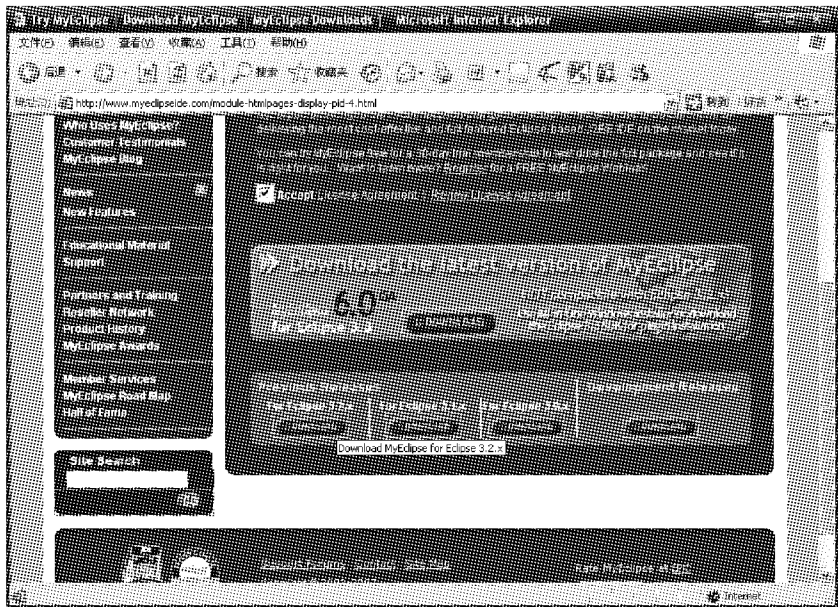


图 1-26 选择版本页面

在该页面中,首先选中 Accept License Agreement 复选框,然后单击 For Eclipse 3.2x 下方的 DOWNLOAD 按钮,进入下载页面,如图 1-27 所示。

本书中所使用的操作系统是 Windows XP,所以选择下载第一项 For Windows 版本

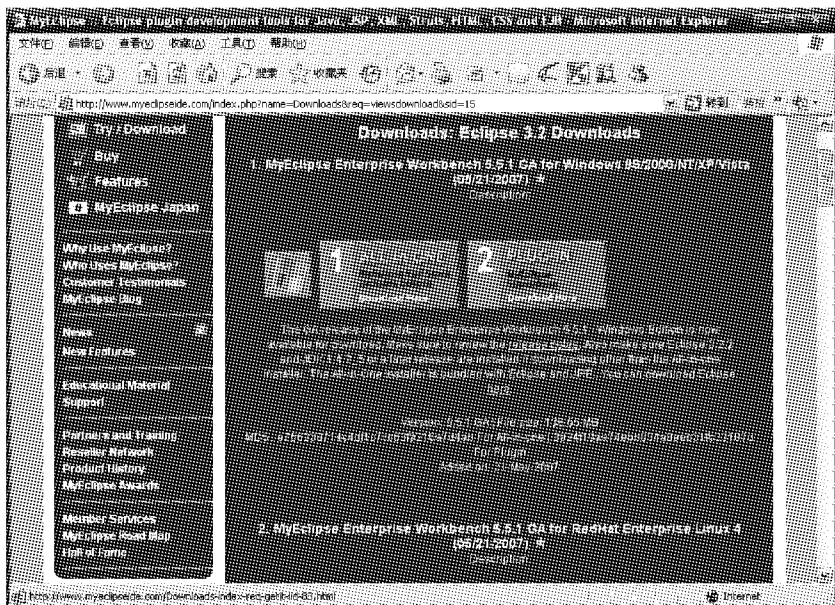


图 1-27 MyEclipse 下载页面

的 MyEclipse, 其中还有两个不同版本, 读者可以根据自己的喜好选择不同的安装方式: ALL-in-ONE——使用独立的安装包进行安装; PLUG-IN——以 Eclipse 插件的形式安装。这里选择前者, 下载得到安装文件 MyEclipse_5.5.1GA_E3.2.2_FullStackInstaller.exe。

下载安装文件后, 就可以开始安装 MyEclipse, 具体步骤如下。

双击 MyEclipse_5.5.1GA_E3.2.2_FullStackInstaller.exe 图标, 运行安装程序。首先是解压的过程, 如图 1-28 所示。

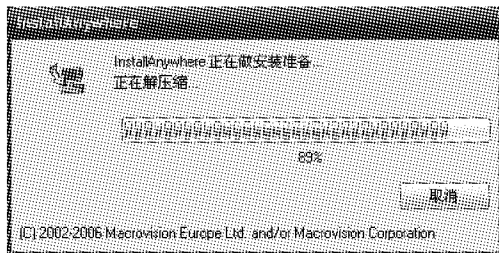


图 1-28 正式安装前的解压过程

解压完成之后进入正式安装, 安装过程不是很复杂, 只要一路单击 Next 按钮即可, 不过有一个地方要注意, 就是在选择安装路径时, 尽量不要选包含中文字符的路径, 如图 1-29 所示。

在其他一些版本中, 在选择安装目录之前, 可能还会让用户选择已经安装的 Eclipse 的根目录, 只要找到原来安装 Eclipse 的根目录即可。

在让用户选择建立快捷方式的位置时, 建议选择 On the Desktop 项, 如图 1-30 所示。接下来一路单击 Next 按钮即可, 这里就不详述了。

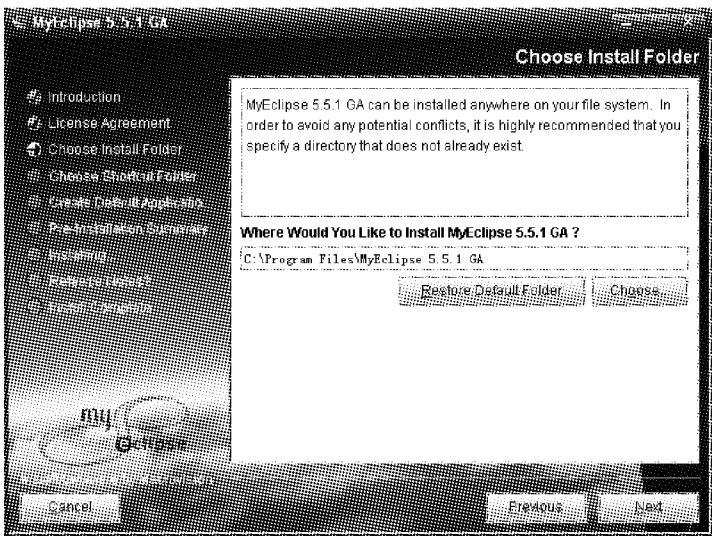


图 1-29 选择安装路径

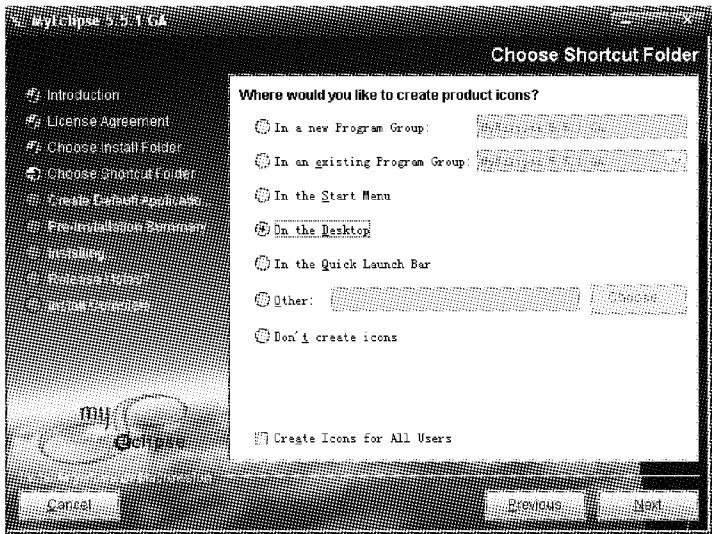


图 1-30 选择建立快捷方式的位置

启动 MyEclipse, 可以看到, 与 Eclipse 相比窗口菜单上多了一个 MyEclipse 菜单项。为了更好地使用 MyEclipse, 可以对其进行配置。由于配置选项参数很多, 本节只对主要的配置选项进行介绍。

(1) 选择 Window|Preferences 命令, 弹出 Preferences 窗口。在左边列表框中单击 MyEclipse 选项, 可以看到 MyEclipse 子项中有非常多的配置选项, 如图 1-31 所示。

(2) 单击 Application Servers|Tomcat 5. x 选项, 弹出如图 1-32 所示的窗口。在该窗口中可以对 Tomcat 服务器进行配置。MyEclipse 可以支持多种 J2EE/Web 应用服务器, 例如 JBoss、Resin、WebLogic、Tomcat 等服务器版本。

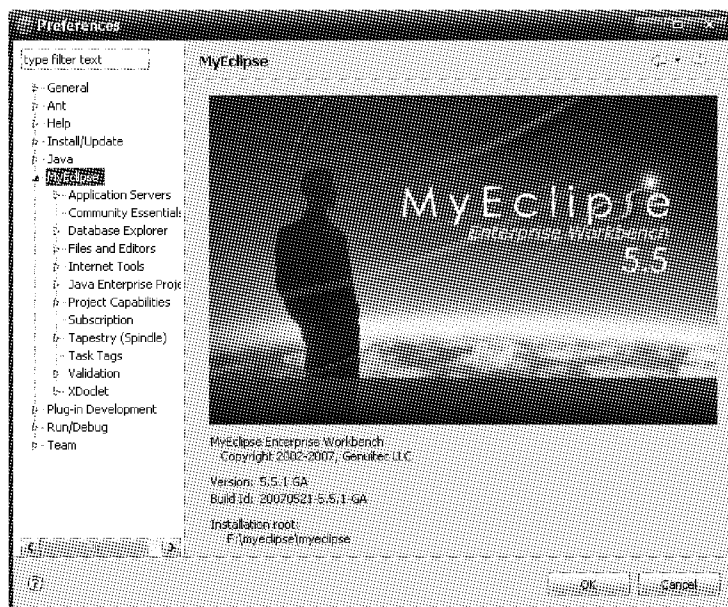


图 1-31 MyEclipse 配置项

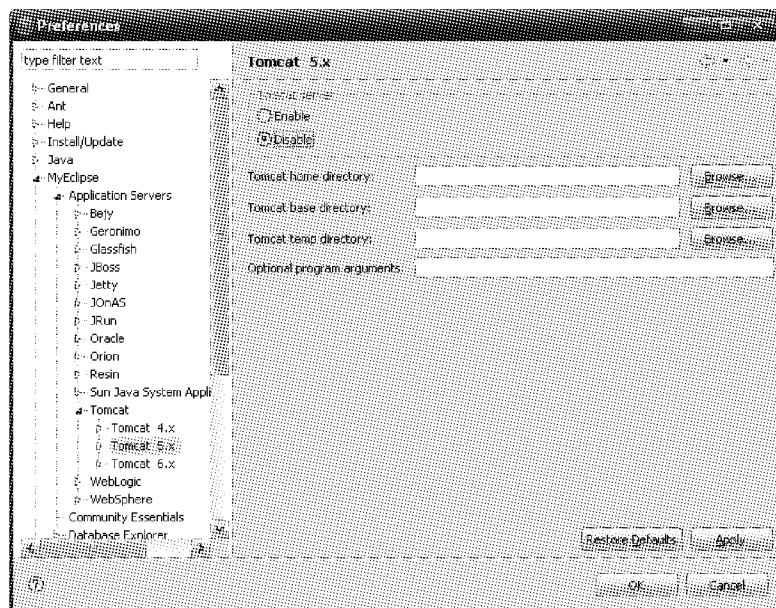


图 1-32 Tomcat 5. x 配置窗口

1.6.2 MyEclipse 下开发 JSP 应用

在 Eclipse 中一般都是用手写 JSP 的方式来开发应用,而 MyEclipse 支持以可视化(图形化)方式进行 JSP 开发,其中还支持 Struts、JSF 等标准控件,大大简化了界面开发,提高了工作效率。本节简要介绍如何利用 MyEclipse 进行可视化开发 JSP 应用。

新建 Web 项目的操作步骤如下所示。

(1) 选择 File|New|Project 命令,弹出如图 1-33 所示的对话框,可以看到其中有 MyEclipse 项目及其子项目 Java Enterprise Projects 等。

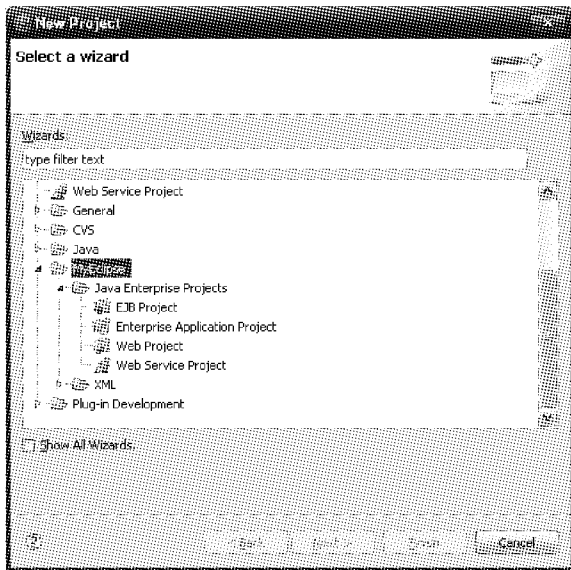


图 1-33 新建 Web 项目

(2) 选择 Web Project,单击 Next 按钮,弹出如图 1-34 所示的对话框。设定其中的 Project Name 为 mytest,Source folder 为 src,Web root folder 为 WebRoot,Context root URL 为 /mytest。如果项目中还需要 JSTL 支持,应选择 Add JSTL libraries to WEB-

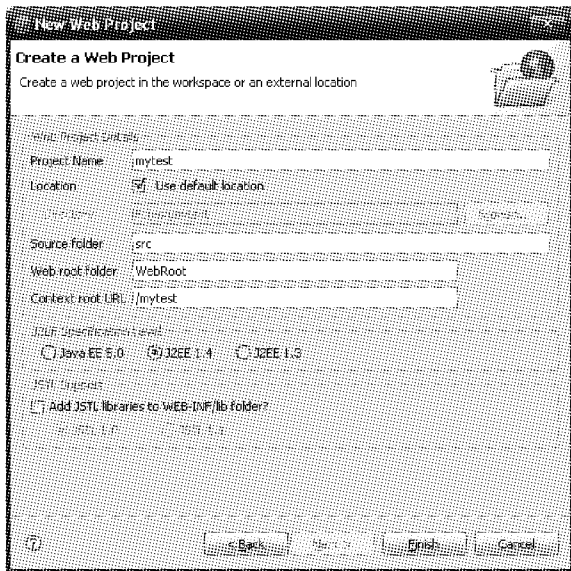


图 1-34 新建 Web 项目的相关信息

INF/lib folder 复选框,并设定 JSTL 版本为 JSTL1.1。

(3) 单击 Finish 按钮,即可结束新建 Web Project 的操作。新建的项目结构如图 1-35 所示。

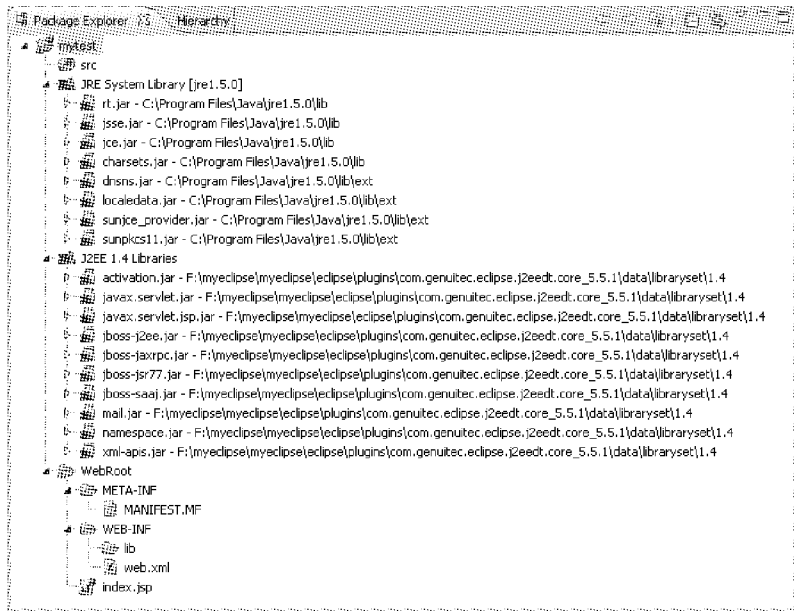


图 1-35 新建 Web 项目结构图

下面为 Web 项目 mytest 创建一个 JSP 页面文件,其操作步骤如下。

(1) 选择 File|New|JSP 命令,弹出 JSP Wizard 对话框,设置子目录为 WebRoot,文件名为 MyFirstJsp.jsp,如图 1-36 所示。

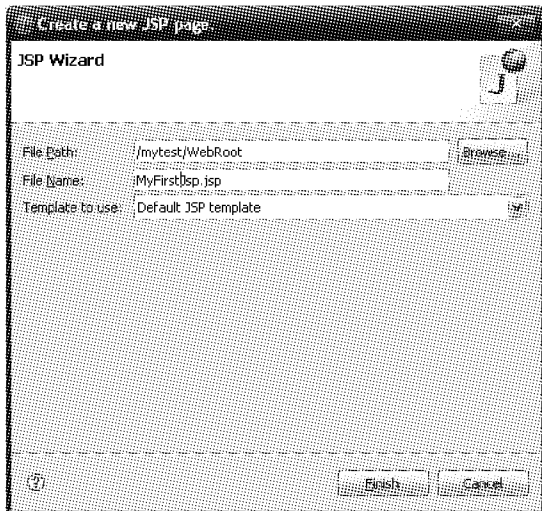


图 1-36 新建 MyFirstJsp.jsp 文件