

从入门到提高丛书

网络跨平台开发的主流技术

JSP 实例教程

阶梯创作室 编著



浦东电子出版社

前 言

随着新世纪的到来, Web 开发技术也在日新月异地发展。以前, 程序员开发一个动态电子商务网站, 除了 CGI、SSI 别无选择。使用 CGI、SSI, 无论是开发还是运行效率都不是很高。虽然 ASP、PHP 等技术的出现, 使开发动态网站变得简单, 但仍然没有一种技术能够使开发和运行效率“鱼和熊掌兼得”, 它能同时具有分布式事件处理的功能和强大的可扩展跨平台能力。随着 JSP 技术的出现, 这一切难题都迎刃而解了。JSP 蕴含了 Java 语言的优势, 是一种真正与平台无关的开发技术, JSP 既有较高的运行效率又有较短的开发周期, 扩展能力和跨平台能力特别强, 任何人都可以按照其公开的规范开发出自己的产品, 因此 JSP 迅速成为了受人青睐的对象。随着众多公司纷纷宣布支持 JSP 技术, 许多网站改头换面, 采用 JSP 作为开发手段, 使 JSP 技术得到了迅猛的发展。

本书主要介绍 JavaServer Pages (JSP) 的编程知识, 突出快学快用的特点, 向广大编程爱好者和网站、数据库开发人员, 详尽地介绍了 JavaServer Pages 开发技术, 结合大量开发实例和技巧, 详解了如何基于 Java 平台构建快速高效的电子商务平台, 充分展示了 JSP 强大的功能和开发网站的优越性。本书叙述由浅入深, 循序渐进, 通过本书的阅读, 广大读者能较快地掌握这一未来网络的开发技术, 成为 JSP 开发能手, 开发出强大的使用 JSP 技术的网站。

本书分为三大部分, 第一部分为入门篇, 包括第一章至第三章。主要讲述了 JSP 的入门知识, 第二部分为实战篇, 包括第四章至第七章, 主要讲述了 JSP 的语法、JSP 与数据库的连接, 并结合一些实例进行说明。第三部分为提高篇, 包括第八章至第十章。介绍了 JSP 与 Java 的结合和高级应用, 并给出一个完整的网络开发实例, 以供广大读者参考。

本书由阶梯创作室谢乐健策划, 马文刚主编, 参与编写的有曹零、管知时、李振、马文刚和谢乐健。参与光盘制作的有周春光。在本书编写中还得到刘斌、陈波、徐振岳、郑祖彬、叶忠华、孙颂华、徐雯燕、陈凌玲、徐英捷、黄涛、奚海峰、曹阳、葛建辉、周成香、陈军、唐新发、吴华雅的帮助和支持, 在此表示感谢。

阶梯创作室的全体成员为本书的出版, 付出了极大的努力, 在时间紧迫的情况下, 全创作室密切合作, 成功地完成了本书的编著, 因为我们创作室愿意作为每一位读者不断进步的阶梯。

感谢浦东电子出版社的领导们, 他们一丝不苟的工作态度深深鼓舞了我们, 他们尽心尽力的工作, 使本书能迅速与广大读者见面。

限于作者水平, 加上编者出版时间仓促, 书中难免有疏漏和不妥之处, 敬请读者批评指正。

阶梯创作室

2001 年 1 月于上海交通大学

内 容 简 介

JSP(JavaServer Pages)是开发 Web 动态网站快速、有效、强大的技术之一。随着 Java 技术的日益成熟和流行, JSP 在网络编程中的地位变得非常重要。JSP 基于强大的 Java 语言, 具有特别强的扩展能力和与平台无关的特点, 在普遍使用 Java 平台构建动态电子商务网站的今天, JSP 拥有无与伦比的优势。本书由经验丰富的 JSP 网站开发人员编写, 面向广大编程爱好者和网站、数据库开发人员, 详尽地介绍了 JavaServer Pages 这一未来强大而先进的动态网站开发的主要技术, 结合大量开发实例, 详尽地讲述了基于 Java 平台来构建快速、高效的电子商务平台的方法, 充分展示了 JSP 强大的功能以及用 JSP 开发网站的优越性。

本书由十个章节和一个附录组成。主要内容包括: JSP 概述、运行环境、JSP 语法简介、JSP 与数据库的连接、JSP 的高级应用、JavaBean、JSP Taglibrary、JSP 的相关资源等。

本书言简意赅, 内容丰富。是计算机编程和开发人员的重要技术参考书。本书面向各个层次的编程、开发和设计人员。本书中的实例具有较好的可移植性, 可供 JSP、Java 编程人员高校相关专业师生借鉴, 开发出更强大的使用 JSP 技术的网站。

本书光盘内容包括: 书中部分实例程序的源文件、服务器安装程序和相关技术资料。

目 录

第一章 JSP 引论.....	1
1.1 JSP 概述.....	1
1.2 JSP 的优势.....	1
1.2.1 开放的技术.....	1
1.2.2 JAVA 的优越性.....	2
1.3 各种动态 Web 页面开发工具的比较.....	3
1.4 JSP 的发展趋势.....	5
第二章 JSP 的常用运行环境.....	6
2.1 JSWDK 服务器.....	6
2.1.1 安装 JSWDK 服务器.....	6
2.1.2 配置 JSWDK 服务器.....	6
2.1.3 JSP 文件的存放路径.....	8
2.2 Tomcat 服务器.....	8
2.2.1 安装 Tomcat 服务器.....	9
2.2.2 配置 Tomcat 服务器.....	11
2.2.3 JSP 文件的存放路径.....	11
第三章 JSP 初步.....	13
3.1 JSP 相关概念.....	13
3.1.1 描述语言 (Scripting Language)	13
3.1.2 Java Servlet	13
3.1.3 动态 HTML(Dynamic HTML).....	13
3.2 JSP 与 Servlet、HTML 的关系.....	14
3.3 JSP 的基本语法规则及语句.....	14
3.3.1 编译器指令.....	14
3.3.2 声明.....	15
3.3.3 表达式.....	15
3.3.4 代码片段/脚本片段 (Scriptlets)	16
3.3.5 注释.....	16
3.3.6 JSP 的内部对象.....	18
3.4 一个 JSP 的小程序.....	18
第四章 JSP 语法精解.....	21
4.1 Scriptlet.....	21
4.2 Include 指令.....	22
4.3 Page 指令.....	23
4.4 Taglib 指令.....	24
4.5 <jsp:forward>.....	25

4.6	<jsp:include>.....	26
4.7	<jsp:plugin>.....	27
4.8	<jsp:setProperty>.....	29
4.9	<jsp:useBean>.....	31
4.10	JSP 示例.....	33
第五章	JSP 与数据库的连接.....	36
5.1	概述.....	36
5.2	JDBC 简介.....	36
5.2.1	java.sql.DriverManager.....	37
5.2.2	java.sql.Connection.....	38
5.2.3	java.sql.Statement.....	39
5.2.4	java.sql.ResultSet.....	40
5.3	与数据库建立连接.....	41
5.3.1	加载驱动程序.....	41
5.3.2	建立连接.....	41
5.3.2	连接到数据库.....	41
5.4	在 JSP 中访问数据库方法及实例.....	42
5.5	一个简单的数据库程序.....	43
第六章	浅谈建立基本的应用程序.....	45
6.1	JSP 与数据库.....	45
6.2	电子商务与应用服务器.....	49
第七章	JSP 应用实例.....	53
7.1	购物程序.....	53
7.2	计数器.....	63
7.3	邮件程序.....	65
7.4	猜测数字.....	69
7.5	网上考试.....	73
7.6	出错处理.....	79
第八章	JSP 的其他应用.....	84
8.1	动态生成图像.....	84
8.1.1	创建一个 BufferedImage 对象.....	84
8.1.2	在 image 对象上绘制图像.....	84
8.1.3	使用 com.sun.image.codec.jpeg 包输出 BufferedImage.....	84
8.2	Cookies.....	86
8.3	Session.....	88
8.4	web 地址重定向.....	89
第九章	JSP 的高级应用.....	91
9.1	JavaBean 的使用.....	91
9.1.1	JavaBean 入门.....	91

■ 9.1.2 JavaBean 与数据库操作	100
■ 9.2 Servlets 介绍	102
■ 9.3 JSP Tag Library 介绍	106
■ 9.4 JSP 安全	112
第十章 一个网站的完整实例	116
■ 10.1 系统简介	116
■ 10.2 数据库结构	121
■ 10.3 模块简介	125
■ 10.4 代码分析	125
附 录	204

JSP 引论

1.1 JSP 概述

什么是 JSP? 引用 SUN 公司关于 JSP 技术的官方语言描述是最恰当不过的:

“JavaServer Pages (JSP)技术使创建动态 Web 网页有了一个简捷而快速的方法。JSP 技术的目的是使构造基于 Web 的应用程序变得更加容易和快捷,而且这些应用程序能够很好地与各种 Web 服务器,浏览器和开发工具共同工作。”如果您访问 <http://java.sun.com/products/jsp/index.html> 这一网址可以在比较醒目的地方看到这段话。

总体来讲,JavaSever Pages (JSP)和微软的 ActiveSever Pages (ASP)在技术方面有许多相似之处。两者都为基于 Web 应用来实现动态交互网页的制作提供了技术环境支持。从同等程度上来讲,两者都能够为程序开发人员提供实现应用程序的编制和自带组件设计网页从逻辑上分离的技术。而且两者都能够替代 CGI,使网站建设与更新变得较为简单和快捷。

尽管 Java Sever Pages 技术和微软的 Active Sever Pages 在许多方面都有相似之处,但仍然存在很多不同之处,最本质的区别在于:两者是来源于不同的技术规范组织,其实现的基础及 Web 服务器平台要求不同。

1.2 JSP 的优势

1.2.1 开放的技术

JSP 和 ASP 技术明显的不同点在于,开发人员对两者各自软件体系设计的深入了解的方式不同。JSP 技术基于平台和服务器的互相独立,输入支持来自广泛的、专门的各种工具包,服务器的组件和数据库产品由开发商所提供。相比之下,ASP 技术主要依赖于微软的技术支持。

- 平台和服务器的独立性

JSP 技术依附于一次写入之后,可以运行在任何具有符合 Java 语法结构的环境。代替了过去依附于单一平台或开发商的环境,JSP 技术能够运行在任何 Web 服务器上,并且支

持来自多家开发商提供的各种工具包。由于 ASP 是基于 Activex 控件技术来提供客户端和服务端端的开发组件, 因此 ASP 技术基本上局限于微软的操作系统平台之上。ASP 主要工作环境是微软的 IIS 应用程序结构, 而且 Activex 对象具有平台特性, 因此 ASP 技术很难实现在跨平台的 Web 服务器上工作。尽管 ASP 技术通过第三方提供的产品能够得到组件和服务实现跨平台的应用程序, 但是 Activex 对象必须事先放置于所选择的平台中。

- 开放的开发过程和源代码

SUN 应用 JAVA 组合性过程, 开发 JSP 技术。自从 1995 年, SUN 已经用这种开放过程方法同国际 JAVA 组织, 合作开发和修改了 JAVA 技术与规范。针对 JSP 的产品, SUN 授权了工具提供商 (如 Macromedia)、结盟公司 (如 Apache, Netscape)、最终用户、协作商。

最近, SUN 将最新版本的 JSP 和 Java Servlet (JSP 1.1, JAVA SERVLET 2.2) 的源代码发放给 Apache, 以获取 JSP 与 Apache 紧密而相互的发展。Apache、SUN 和许多其它的公司及个人公开成立一个完善的咨询机构以使任何公司和个人都能免费、方便地取得信息。JSP 应用程序界面 (API) 毫无疑问已经取得成功, 并将随 JAVA 组织不断开放、扩大而继续完善。相反, ASP 技术仅依靠微软本身的推动, 其发展是建立在独占的、封闭的开发过程基础之上。

- 从开发人员的角度来看

ASP 和 JSP 技术都能使开发者实现通过点击网页中的组件, 制作交互式、动态的内容和应用程序的 Web 站点。ASP 仅支持组件对象模型 COM, 而 JSP 技术提供的组件都是基于 Javabeans 技术或 JSP 标签库。由此可以看出两者虽有相同之处, 但其区别是很明显的。

- JSP 标签的可扩充性

尽管 ASP 和 JSP 都使用标签与脚本技术来制作动态 Web 网页, JSP 技术能够使开发者扩展 JSP 标签得以应用, JSP 开发者能定制标签库, 所以网页制作者能够充分利用 JSP 与 XML 兼容的强大标签技术, 大大减少对脚本语言的依赖。由于定制标签技术, 使网页制作者降低了制作网页和向多个网页扩充关键功能的工作量。

- JSP 跨平台的可重用性

JSP 的开发人员在开发过程中一直关注其可重用性。JSP 组件 (企业 Javabeans, Javabeans, 或定制的 JSP 标签) 都是跨平台可重用的。企业 Javabeans 组件可以访问传统的数据库, 并能以分布式系统模式, 工作于 UNIX 和 Windows 平台。JSP 技术的标签可扩充功能, 为开发人员提供简便的与 XML 兼容的接口, 即共享网页的打包功能, 可使其完全符合工业标准。

这种基于组件的模式, 能有效地提高应用程序的开发效率, 因为这种模式能够使开发人员利用快捷的子组件, 快速创建模板应用程序, 然后再整合一些附加功能以后, 便可使用。类似这样有效的方法, 在 JSP 中无处不在, 并可将其打包成一个 Javabeans 或一个工业标准化的 Javabeans 组件。

■ 1.2.2 JAVA 的优越性

JSP 技术是用 JAVA 语言作为脚本语言的, 而 ASP 网页, 使用微软的 VBScript 或 Jscript。JAVA 是成熟、强大、易扩充的编程语言, 大大优于基于 BASIC 的脚本语言。如: JAVA

的可执行性优于 VBScript 或 Jscript 语言。因为它们利用 JAVA 技术并且都被编译为 JAVA Servlets, JSP 网页为整个服务器端的 JAVA 库单元提供了一个接口, 来服务于 HTTP 的应用程序。

JAVA 使开发人员的工作, 在其它方面也变得一样容易、简单。例如, 当 ASP 应用程序在 Windows NT 系统被怀疑可能会崩溃时, JAVA 能有效地防止系统的崩溃。JAVA 语言通过提供防止内存泄漏的方法, 在内存管理方面也能大显身手。而且, JSP 为应用程序提供了健壮的意外事件处理机制。

- 易于维护性

基于 JSP 技术的应用程序比基于 ASP 的应用程序更易于维护和管理。脚本语言能很好服务于小的应用程序, 但不能适应大型、复杂的应用程序。因为, JAVA 是结构化的, 它比较容易创建和维护庞大、组件化的应用程序。JSP 突出的组件技术使修改内容而不影响逻辑或修改逻辑而不影响内容变得很容易实现。企业级的 Javabeans 结构整合了企业逻辑, 例如数据库的访问, 安全, 事务完整性和独立性。因为 JSP 技术是一种开放的, 跨平台的结构, 因此, Web 服务器, 平台, 其它的组件能很容易升级或切换, 且不会影响 JSP 基本的应用程序。这一特点使得 JSP 能够适用于当今各种 Web 应用程序的不断变化和发展。

- 企业产品的多样性

JAVA2 企业平台(J2EE)是适用于多企业应用程序的 JAVA 结构, 作为 J2EE 的部分, JSP 网页可访问所有 J2EE 的组件, 包括 Javabeans, 企业级 Javabeans 及 JAVA Servlets。JSP 网页都能完全编译成为 Servlets, 所以它们都享有灵活性的特点。J2EE 平台内容不仅包括管理复杂的企业应用程序, 而且包括事务管理技术和 Pooling 资源管理技术。

总之, 因为 JSP 技术是基于 JAVA 的开放性过程的产品, 因此它能够广泛支持不同提供商提供的工具、Web 服务器和应用程序的服务, 这样能够使用户选择最佳的开发方法, 选择最适应他们的应用程序开发的工具包, 同时有效地保护用户在代码和人员培训上的投资。

图 1.3 各种动态 Web 页面开发工具的比较

目前, 最常用的动态网页语言有 ASP、JSP 和 PHP。在此, 对这三种语言作一个简要介绍和比较, 从中可以发现 JSP 具有的优势。

- ASP ActiveServer Pages (动态服务器主页), 内含于 Internet Information Server(IIS)当中, 提供一个服务器端的 Scripting 环境, 站点服务器会自动将您设计的 Active Server Pages 的程序码解释为标准 HTML 格式的主页内容, 在用户端的浏览器上显示出来。用户端只要使用常规可执行 HTML 代码的浏览器, 即可浏览。

ASP 无须编译或链接即可执行, 集成于 HTML 中, 使用常规文本编辑器, 与浏览器无关, 用户端只要使用常规的可执行 HTML 码的浏览器即可浏览。Script 语言(VBscript, Java script)在站点服务器执行, 用户不需要执行这些 Script 语言。它还可通过 ActiveX 服务器组件来扩充功能, ActiveX Server Component 可使用 VB、Java、C++、COBOL 等语言来实现。此外, Active Server Pages 与任何 ActiveX scripting 语言兼容。除了可使用

VBScript 或 Java Script 语言来设计, 还可通过插件程序 (Plug-in) 的方式, 使用由第三方所提供的其它 Scripting 语言。它的源程序代码不会传到用户的浏览器中, 可以保护源程序。传到用户浏览器的是由 Active ServerPages 解释得出的常规 HTML 码。

● PHP 是一种跨平台的服务器端嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法, 并结合 PHP 自身的特性, 使 Web 开发者能够快速编写出动态 Web 页面。它支持目前绝大多数数据库。PHP 是完全免费的, 您可以自由下载, 甚至可以不受限制地获得源代码。

PHP 可以编译成与许多数据库相连接的函数。PHP 与 MySQL 是绝佳的组合。您还可以自己编写外围的函数来间接存取数据库, 而当您更换使用的数据库时, 可以轻松地更改编码, 以适应这样的变化。PHPLIB 就是最常用的可以提供一般事务需要的一系列基库。但 PHP 提供的数据库接口支持不统一, 比如对 Oracle、MySQL、Sybase 的接口, 彼此都不一样。这也是 PHP 的一个弱点。

● JSP 是 Sun 公司推出的新一代站点开发语言, 它完全解决了目前 ASP、PHP 的一个通病——脚本级执行 (据说 PHP4 也已经在 Zend 的支持下, 实现编译运行)。JSP 可以在 Servlet 和 JavaBean 的支持下, 完成功能强大的站点程序。

JSP 的最大特点是将内容的生成和显示进行分离。使用 JSP 技术的 Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。使用 JSP 标识或者小脚本来生成页面上的动态内容。生成的逻辑内容被封装在标识和 JavaBeans 组件中, 并且捆绑在小脚本中, 所有的脚本都在服务器端运行。如果核心逻辑被封装在标识和 Beans 中, 那么 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 标识和小脚本, 生成所请求的内容 (例如, 通过访问 JavaBeans 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。

Web 页面开发人员不会都是熟悉脚本语言的编程人员。Java Server Pages 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标识中生成动态内容所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件, 设置或者检索组件属性, 下载 Applet, 以及实现用其它方法难于编码的功能。

通过开发定制化标识库, JSP 技术可以扩展。技术人员可以为常用功能创建自己的标识库, 这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。

JSP 技术很容易整合到多种应用体系结构中, 以利用现存的工具和技巧, 并且扩展到能够支持企业级的分布式应用。

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的, 拥有 Java 编程语言“一次编写, 各处运行”的特点, 而且所有的 JSP 页面都被编译成为 Java Servlet, JSP 页面就具有 Java 技术的所有好处, 包括完全的存储管理和安全性。

在各自的应用范围中, ASP 只能运行于微软的服务器产品上。Unix 下也有 ChiliSoft 的插件来支持 ASP, 但是 ASP 本身的功能有限, 必须通过 ASP+COM 的组合来扩充, Unix 下的 COM 实现起来非常困难。而 PHP 可在 Windows、Unix、Linux 的 Web 服务器上正常

运行,还支持 IIS, Apache 等通用 Web 服务器,用户更换平台时,无需变换 PHP 代码,即可拿即用。

JSP 同 PHP 类似,几乎可以运行于所有平台。如 Win NT、Linux、Unix。NT 下 IIS 通过一个插件,例如 JRUN 或者 ServletExec,就能支持 JSP。著名的 Web 服务器 Apache 已经能够支持 JSP。虽然现在 NT 操作系统占了很大的市场份额,但是在服务器方面 Unix 的优势仍然很大,而新崛起的 Linux 更是来势不小。从一个平台移植到另外一个平台,JSP 和 JavaBean 甚至不用重新编译,因为 Java 字节码都是标准的和与平台无关的。

在性能比较上,曾有人做过试验,对这三种语言分别做循环性能测试及存取 Oracle 数据库测试。无论是在循环性能测试中,还是在数据库测试中,JSP 都是最快的。

目前在国内 PHP 与 ASP 应用最为广泛。但在国外,JSP 已经是比较流行的一种技术,尤其是电子商务类的网站,多采用 JSP。

采用 PHP 的网站如新浪网(sina)、中国人(Chinaren)等,但由于 PHP 本身存在的缺乏规模支持和缺乏多层结构支持的缺点,使得它更适合一些小型的商业站点。此外,由于 PHP 提供的数据库接口支持不统一,这就使得它不适合运用在电子商务中。

ASP 和 JSP 则没有以上缺陷,ASP 可以通过 Microsoft Windows 的 COM/DCOM 获得 ActiveX 规模支持,通过 DCOM 和 Transcation Server 获得结构支持;JSP 可以通过 SUN Java 的 Java Class 和 EJB 获得规模支持,通过 EJB/CORBA 以及众多厂商的 Application Server 获得结构支持。

1.4 JSP 的发展趋势

总体来说,JSP 应该是电子商务未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet。比较出名的如 IBM 的 E-business。

JSP 技术被设计为一个开放的、可扩展的建立动态 Web 页面的标准。开发人员可以使用 JSP 页面来创建可移植的 Web 应用,在不同的 Web 和应用服务器上为不同的场合所运行。

通过与业界领袖的合作,Sun 保证 JSP 规范是开放的和可移植的。可以使用任何客户机和服务器平台,在任何地方编写和部署它们。将来,工具供应商和其它厂商将通过为专门的功能,提供客户化的标识库,而扩展平台的功能。

JSP 规范的 1.0 版本,是通向动态 Web 页面生成的一个开放的工业标准方法的第一步。1.0 版本通过一个核心标识集、隐含对象以及开始创建动态 Web 页面所需的基本功能,构成了该方法的基础。已经有许多 Web 服务器,应用服务器和开发工具供应商正在为它们的产品添加 JSP1.0 的支持,这样在业界已经拥有了最初的、立即的支持。在 1999 年晚些时候已完成的 1.1 版本,通过更多的 XML 支持、客户化标识、以及与 J2EE 的集成而扩展这个版本。并且供应商们可能会有选择地扩展和扩充在 JSP 规范中基本的、必需的功能。JSP 引擎能够强有力地支持多种脚本语言和对象模型。在业界扩充和使用 JSP 技术能力的同时,也向 Sun 承诺将保证 JSP 技术保持平台和服务器间固有的可移植性。

JSP 的常用运行环境

2.1 JSWDK 服务器

JSWDK 是由 Sun 提供的 JSP 运行平台，只包含必需的内容，是一种比较简洁的运行环境，初学 JSP 的朋友可用此平台连接本机的数据库（比如使用 Microsoft Access 做数据库）进行 JSP 开发的初步体验。JSWDK 只是一个简化的服务器平台，性能和稳定程度都比较有限。

2.1.1 安装 JSWDK 服务器

JSWDK 的安装。仅需将 jsjdk1_0_1-win.zip 带目录释放到硬盘根目录下(c:\, d:\ 等)，然后就可以在硬盘上找到 \jsjdk-1.0.1\ 目录，将来如果不想保留 JSWDK，删除这个目录就可以了，没有任何系统文件和注册表遗漏问题，是一个“绿色软件”。

然后，修改系统环境参数，JDK 可以是 Sun JDK1.2.2 或 1.3。可在本书配套光盘中安装：

如果您在 Windows98 上安装，只需将 set JAVA_HOME=[x:]jdk1.2.2（其中[x:]是安装 JDK 的硬盘 c:, d: 等。）加到你的 autoexec.bat；如果是 Windows NT，可在控制面板/系统/环境页中选择 path 参数并加入[x:]jdk1.2.2 in；同样可增加新的环境参数 CLASSPATH=[x:]jdk1.2.2\lib*.jar，然后重新启动即可。

更详细安装的过程以及 Solaris/Unix 和 Linux 下 JDK 的安装，可以参考下载得到文件包中的安装说明。

2.1.2 配置 JSWDK 服务器

每次开机以后都要执行一下 startserver.bat 才可启动 JSP 服务器，这是在 DOS 窗口中运行的程序，如果要退出，可执行 stopserver.bat。如图 2-1、2-2、2-3 所示。

（如果是 unix 系统，则分别可执行 %startserver 和 stopserver。）

如果发生问题，最大的可能是 JDK 的环境参数没有正确设置，您可以再次加以检查。如果是 8080 端口已经被使用，可以打开 JSWDK 目录下的 webserver.xml 对此端口重新进行设置。

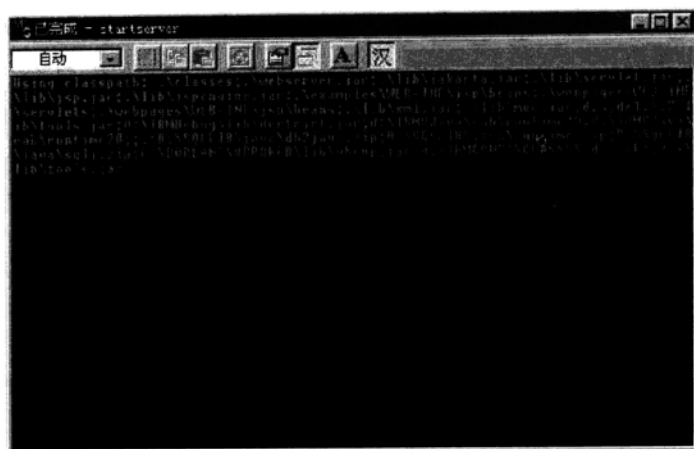


图 2-1



图 2-2



图 2-3

具体如下:

```
id ID #REQUIRED
port NMTOKEN \"8080\"
hostName NMTOKEN \"\"
inet NMTOKEN \"\"
docBase CDATA \"webpages\"
workDir CDATA \"work\"
workDirIsPersistent (false | true) \"false\">
```

将 port 口设为其它端口即可。

■ 2.1.3 JSP 文件的存放路径

打开 JSP 页面的方式为: <http://localhost:8080/examples/xxx.jsp>

如果您要建立自己的执行目录,可以在 `webserver.xml` 文件中增加这样的设置:

<http://localhost:8080/myweb/xxx.jsp>

■ 2.2 Tomcat 服务器

Tomcat 3.1 是由 Apache Group Jakarta 小组开发的一个免费服务器软件,而且源代码可以免费获得,有特殊要求的程序员可以自由对它进行扩充。Tomcat 3.1 实现了最新的 Servlet2.2 和 JSP 1.1 的标准, SUN 公司也非常推荐 Tomcat,甚至将 Tomcat 的图标,直接放置在 SUN 站点中 JSP 的主页上。

Tomcat 是一个带有 JSP 环境的 Servlet 容器。Servlet 容器是一个根据用户的行为可以管理和激活 Servlet 的运行的 shell。Tomcat 有以下三种运行方式。

- 作为一种独立服务器

由于 Tomcat 内置有 Web 服务器的一部分,虽然其 Web 功能要弱于主流的 Web 服务器。当使用基于 Java 的 Web 服务器的情形,例如 Servlet 容器是 JavaWebServer 的一个部分。独立的 Servlet 容器是 Tomcat 的默认模式。即 Tomcat 本身既可以实现 JSP 和 Servlet 的服务,也可以完成普通的 Web 服务。这就为使用者学习和测试 JSP 提供了很大的方便。

- 作为其他 Web 服务器的进程内插件。

由于主流的 Web 服务器,大多不提供 JSP 的服务。为了使其提供 JSP 的支持,可将 Tomcat 作为 Web 服务器的插件,运行在服务器的进程内,可以提供良好的运行速度,但伸缩性有所不足。

- 作为 Web 服务器的进程外插件。

Tomcat 运行在服务器进程之外。其反映速度不如进程内的运行方式,但进程外插件有更佳的可伸缩性和稳定性等优点。

Tomcat 既可作为独立的容器(主要是用于开发与调试),又可作为对现有服务器的附加(当前支持 Apache、IIS 和 Netscape 服务器)。即任何时候配置 Tomcat,您都必须决定如何应用,如选择第二或第三种模式,您还需要安装一个 web 服务器接口。

2.2.1 安装 Tomcat 服务器

由于 Windows 依然占据操作系统市场的大部分份额, 因此在此主要介绍 Windows 平台上 Tomcat 的安装。

● JAVA 运行环境的建立

由于 JSP 是 JAVA 技术的一个分支, 因此首先需要安装 JAVA 的运行环境。首先需要获得 JDK 软件包。JDK 软件包有多个版本, 大量被使用的是 JDK1.22 版本, 而随着 JDK1.3 的发布, 笔者推荐采用 JDK1.3, 其中一个主要的原因 JDK1.3 中对中文的支持大为加强!

JDK1.3 可以在 <http://java.sun.com> 获得 j2sdk1_3_0-win.exe, 在此采用 JDK1.3 Windows 版本。

运行并安装该软件包后, 设置 JAVA 运行所需要的环境变量 path 和 classpath 两项如图 2-4 所示。

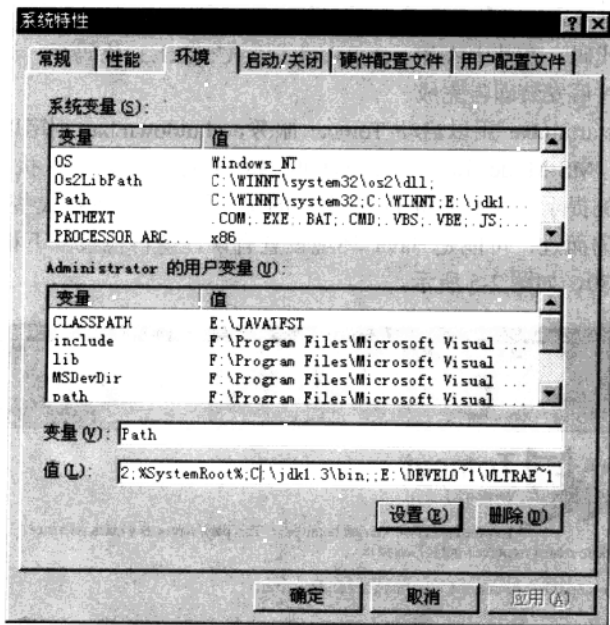


图 2-4

在 Windows NT 中: 右键单击“我的电脑”->选择属性->环境

在 path 变量中添加 JDK 中 bin 文件夹的路径。

如果 Jdk 安装在 C:\jdk1.3, 则需要在 path 中添加 C:\jdk1.3\bin; 。

如果系统中没有 classpath 变量, 则添加此环境变量, 该变量用于存放运行具体应用程序所需要的 JAR 文件和 *.class 程序包的路径。如用命令行形式执行 JAVA 程序, 则需要将需要运行的 *.java 文件路径设置到 classpath 中。如果使用 Windows98/95 系统, 就需要修改 autoexec.bat 文件完成相应的设置。

● Tomcat 安装

Tomcat 软件包可以在 <http://jakarta.apache.org/download/binindex.html> 处下载。

可将其解压缩到任意目录下, 比如将其解压缩到 C 盘根目录下。在 C 盘下就会出现

一个 jakatar-tomcat 的目录, 为方便起见, 将其改名为 Tomcat。在打开 c:\tomcat\bin 目录找到 tomcat.bat 文件, 进行编辑。

```
rem Set up the CLASSPATH that we need
set cp=%CLASSPATH%
set CLASSPATH=
set java_home=C:\jdk1.3
set CLASSPATH=%TOMCAT_HOME%\classes
set CLASSPATH=%CLASSPATH%; %TOMCAT_HOME%\lib\webserver.jar
set CLASSPATH=%CLASSPATH%; %TOMCAT_HOME%\lib\jasper.jar
set CLASSPATH=%CLASSPATH%; %TOMCAT_HOME%\lib\xml.jar
set CLASSPATH=%CLASSPATH%; %TOMCAT_HOME%\lib\servlet.jar
set CLASSPATH=%CLASSPATH%; %JAVA_HOME%\lib\tools.jar
```

找到以上该段代码, 在其中添加 set java_home=C:\jdk1.3 代码行。该行指明 jdk 的安装目录。保存该文件后安装即告完成。

Bin 目录下的 startup.bat 用以启动 Tomcat 服务, shutdown.bat 则用以停止服务。运行 startup.bat, 将出现 MS-Mode 窗口。在 WIN95/98 平台上, 如果在执行时出现 Out of environment space 的提示, 就需要将 DOS 窗口的缓冲区尽量设置得大些, 然后再次运行。如果该启动窗口一闪而过, 可能是 Java 环境设置有误, 请再检查一下相关设置。Tomcat 默认服务端点为 8080, 如图 2-5 所示。

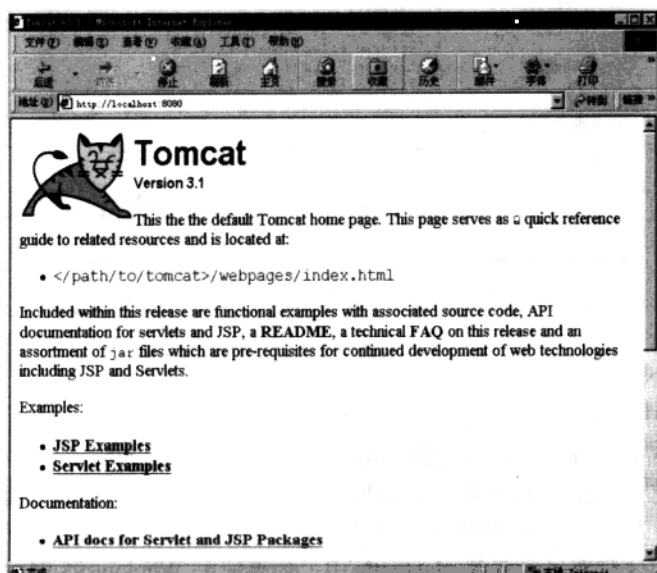


图 2-5

打开浏览器在地址栏中键入: http://localhost:8080/ 回车。点击 JSP Examples 链接, 运行其中范例, 如通过则说明安装成功。若出现错误提示, 则请重点检查 Tomcat.bat 中的

java_home 设置。值得一提的 Tomcat 默认页面中 API docs 链接，将提供开发必不可少的 JSP 和 Servlet 资料，特别对高级开发很有帮助。

■ 2.2.2 配置 Tomcat 服务器

Tomcat 安装成功后，可进行一些必要的设置。

Tomcat 的配置基于两个配置文件：

- conf/server.xml - Tomcat 的全局配置文件，完成两个目标：
- 提供 Tomcat 组件的初始配置。
- 说明 Tomcat 的结构、含义，使得 Tomcat 通过实例化组件完成启动及构建自身。

比如

```
<Connector className="org.apache.tomcat.service.SimpleTcpConnector">
<Parameter name="handler" value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
<Parameter name="port" value="8080"/>
</Connector>
```

该段代码指定 8080 为服务器端口，现可以将其修改为其它端口。

- web.xml - 在 Tomcat 中配置不同的关系环境

Tomcat 可以让用户通过将缺省的 web.xml 放入 conf 目录中，来定义所有关系环境的 web.xml 的缺省值。建立一个新的关系环境时，Tomcat 使用缺省的 web.xml 文件，作为基本设置和应用项目特定的 web.xml(放在应用项目的 Web-INF/web.xml 文件)来覆盖这些缺省值。然而对于初学者和普通的应用，只需要使用默认配置即可，无须改动这些配置文件。

■ 2.2.3 JSP 文件的存放路径

Web 开发中有个必不可少的工作，就是设置 Web 路径和实际路径的映射。首先在 C:\tomcat\webapps 目录下建立 study 文件夹。然后在浏览器地址栏中键入：
http://localhost:8080/admin/contextAdmin/contextAdmin.html 回车。然后点击 View all Contexts 按钮。就可以察看目前 tomcat 管理的路径，并在 Path 文本框中填入 Web 路径/study，在 Document Base 文本框中填入实际的文件路径 c:\tomcat\webapps\study，然后按 Add Context 完成路径映射。这样就可以通过浏览器，键入 http://localhost:8080/study 访问该目录了。事实上如果新建立的文件夹位于 webapps 路径下，重新启动 Tomcat，系统也能自动完成相同的路径映射工作。而利用 Web 控制台，则可以不必重新启动服务，且可以自定义 Web 路径的名称。而要想取消一个路径映射，只需在 Context Name 文本框中，输入该 Web 路径，按 Remove Context 即可完成，如图 2-6 所示。