



“九五”国家重点电子出版物规划项目·计算机知识普及系列



网站 / 网页编程实作丛书 (1)

JSP/HTML

Programming Instance

编程实作教程

北京希望电子出版社 总策划
张洪波 张黎明 单银根 编写



本光盘内容为：
本版电子书



092



北京希望电子出版社

Beijing Hope Electronic Press
www.bhep.com.cn



“九五”国家重点电子出版物规划项目·计算机知识普及系列



网站 / 网页编程实作丛书 (1)

JSP/HTML

Programming Instance

编程实作教程

北京希望电子出版社 总策划
张洪波 张黎明 单银根 编写



本光盘内容为：
本版电子书

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制



北京希望电子出版社

Beijing Hope Electronic Press
www.bhep.com.cn

内 容 简 介

本书是“网站/网页编程实作丛书”之一。而且本书的作者是专业从事网站建设的专家，他从基础知识入手详细讲述了 JSP 的语言规范、运行环境、基本语法、指令以及建立动态网站的各种技能。

本书共分 8 章，第 1 章：JSP 描述。主要讲述什么是 JSP，JSP 的主要特点、运行模式以及 JSP 运行环境的建立。第 2 章：Java 语言基础，主要讲述 Java 语言的功能、特点、基本语法、程序结构以及编程技巧等；介绍 JSP 动态网站建设的基本语言工具。第 3 章：JSP 基本语法指令及其作用。介绍 JSP 的编写。第 4 章：HTML 表单的处理与交互方法，讲述 HTML 表单的基本概念、表单的标记与处理、表单在客户端的确认方法、表单在服务器端的确认方法以及与客户机交互的过程与方法。第 5 章：JSP 内部对象的使用，讲述了 JSP 内部 Request、Response、Session、Application、Out、Exception 六种对象的使用。第 6 章：JavaBeans 的使用，讲述了 JavaBeans 以及设计目标的服务、基本结构、应用方法，为全面掌握 JSP 开发技术奠定了基础。第 7 章：JSP 访问数据库。全面讲述了 JSP 访问数据库的方法，如怎样连接数据库、怎样连接 Access2000 数据库、怎样连接 SQLServer7.0 数据库、怎样连接 MySQL 数据库等、使建设 JSP 动态网站的技术上升了一个层次。第 8 章：应用实例。本章提供了一个完整的网站建设样例，供读者参考。

本书特点：内容丰富、语言简洁、流畅，重点突出、实用性强，从基础知识入手，教读者学习和掌握

本书可作为从事网站/网页编程的广大科技人员自学指导书；高等院校计算机专业师生教学、自学参考书以及社会相关专业培训班教材。

本光盘内容包括：本版电子书

系 列 书：网站/网页编程实作丛书（2）

书 名：JSP/HTML 编程实作教程

总 策 划：北京希望电子出版社

文 本 著 者：张洪波 张黎明 单银根 编写

C D 制 作 者：希望多媒体开发中心

C D 测 试 者：希望多媒体测试部

责 任 编 辑：周艳

出版、发行者：北京希望电子出版社

地 址：北京中关村大街 26 号，100080

网址：www.bhp.com.cn E-mail：xrl@hope.com.cn

电话：010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

（图书发行）

010-62613322-215（门市） 010-62547735（编辑部）

经 销：各地新华书店、软件连锁店

排 版：希望图书输出中心

C D 生 产 者：北京中新联光盘有限责任公司

文 本 印 刷 者：北京双青印刷厂

开 本 / 规 格：787 毫米×1092 毫米 1/16 开本 20.875 印张 484.6 千字

版 次 / 印 次：2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

本 版 号：ISBN 7-900071-20-2/TP·19

定 价：35.00 元（1CD，含配套书）

说明：凡我社光盘配套图书若有自然破损、缺页、倒页、脱页，本社负责调换。

前 言

本书是以中科院计算所（二部）网络研究开发中心，北京中科天地网络技术有限公司的张洪波为主体，并根据自己建设网站的经验而写成的。本书共分 8 章，他们是：

第 1 章：JSP 描述。主要讲述什么是 JSP，JSP 的主要特点、运行模式以及 JSP 运行环境的建立。向读者介绍 JSP 的整体概况。

第 2 章：Java 语言基础，主要讲述 Java 语言的功能、特点、基本语法、程序结构以及编程技巧等；介绍 JSP 动态网站建设的基本语言工具。

第 3 章：JSP 基本语法指令及其作用。讲述 JSP 语法的基本原理、指令及其用法，JSP 编程中应该注意的问题、介绍 JSP 的编写过程。

第 4 章：HTML 表单的处理与交互方法，讲述 HTML 表单的基本概念、表单的标记与处理、表单在客户端的确认方法、表单在服务器端的确认方法以及与客户机交互的过程与方法；介绍 JSP 中的 HTML 表单处理方法。

第 5 章：JSP 内部对象的使用，讲述了 JSP 内部 Request、Response、Session、Application、Out、Exception 六种对象的使用。

第 6 章：JavaBeans 的使用，讲述了 JavaBeans 以及设计目标的服务、基本结构、应用方法，为全面掌握 JSP 开发技术奠定了基础。

第 7 章：JSP 访问数据库。全面讲述了 JSP 访问数据库的方法，如怎样连接数据库、怎样连接 Access2000 数据库、怎样连接 SQLServer7.0 数据库、怎样连接 MySQL 数据库等、使建设 JSP 动态网站的技术上升了一个层次。

第 8 章：应用实例。通过第 7 章的学习掌握了基本技术，而本章提供了一个完整的网站建设样例，供读者参考。网站包括了会员管理、网上购物、聊天室等功能，没有经验的开发者可模仿建设。

本书由张洪波同志主笔编写，参加编写的人员还有刘春阳、张黎明、单银根、黎连业、刘占全、滕华等同志。由于写作时间短促，如有错误请读者批评指正。

作者：张洪波

2001 年 2 月于中科院计算所（二部）网络研究开发中心

第 1 章 JSP 概述

JavaServer Pages(JSP)是基于 Java 的技术,用于创建可支持跨平台及跨 Web 服务器的动态网页。

本章将向读者介绍什么是 JSP? JSP 的优点、JSP 与 ASP 的比较以及它的运行方式。同时还重点讨论了 JSP 运行环境需要的组件以及在 Unix、Windows 和 Linux 操作系统下如何建立 JSP 运行环境。最后,给出了一个简单的 JSP 程序来帮助读者加深对 JSP 的理解。

1.1 什么是 JSP

Java Server Pages(简称 JSP),是由 Sun Microsystems 公司倡导、多家公司参与合作而建立起的一种动态网页技术标准,其网址为 <http://www.javasoft.com/products/jsp>。它正式发布于 1999 年 6 月,是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术,也是 Servlet2.1 API 的扩展。JSP 可以不加修改地在 85%以上的 Web Server 上运行,其中包含了微软的 Windows NT /95 系统。

在传统的网页 HTML 文件(*.htm,*.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(tag),就构成了 JSP 网页(*.jsp)。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片段,然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定义网页以及发送 E-mail 等等,这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行,网络上传送给客户端的仅是得到的结果,对客户浏览器的要求最低,可以实现无 Plugin,无 Activex,无 Java Applet,甚至无 Frame。

目前的 Java2 platform Enterprise Edition 简称 J2EE,是 Sun 公司为企业级应用建立的高水准的结构,它获得了各大开发平台提供商的积极支持,并已成为当今企业应用开发的主流产品。JSP 技术是 J2EE 平台上的一个关键性的组件。由于 JSP 技术被设计成独立于平台和服务器,我们可以利用这一技术建立先进、安全和跨平台的动态网站。它的建立来源于通用的工具、服务器和数据库供应商,因此 JSP 有着非常广泛的发展前景。

在 Sun 正式发布 JSP(JavaServer Pages)之后,这种新的 Web 应用开发技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。按照 Sun 的说法,JSP 能够适应市场上包括 Apache WebServer、IIS4.0 在内的 85%的服务器产品。即使你对 ASP “一往情深”,我们认为,关注 JSP 的发展仍旧很有必要。

JSP 可与微软的 Active Server Pages(ASP)相媲美,但 JSP 使用的是类似于 HTML 的标记和 Java 代码片段而不是用 VBScript。当你使用不提供 ASP 本地支持的 Web 服务器(例如 Apache 或 Netscape 服务器)时,就可以考虑使用 JSP 了。虽然也可以为这些服务器弄一个 ASP 附加软件模块,但是太昂贵了。现在 Sun 公司并不会因为你使用 JSP 而向你收费(虽然将来可能要收),况且用于 Solaris、Linux 以及 Windows 系统的组件都很容易获得。

不要把 JSP 与服务器端的 JavaScript 语言搞混了。Web 服务器自动将通过 JSP 生成的 Java 代码片段转换成 Java 片段(servlets)。JSP 也可自动控制许多功能,如过去用 Perl 脚本

编写功能程序或像 ASP 这样的服务器专用 API(应用编程接口)。

事实证明, JSP 是一种开发 Web 应用的理想技术。JSP 以 Server 技术为基础, 又在许多方面做了改进。利用跨平台运行的 JavaBean 组件, JSP 为分离处理逻辑与显示样式提供了卓越的解决方案。

Sun 公司的 JSP, 有以下几种版本:

JSP 0.91 版本: 它是第一个比较完善的版本。

JSP 0.92 版本: 它是 JSP 整体构架开始完善的版本, 为 JSP 1.0 版提供了很多新概念。

JSP 0.92 版本是目前最受欢迎的 JSP 版本。

JSP 1.0 版本: JSP 1.0 版本中的语法有很大的改动, 不过改变的主要原因是让 JSP 规范化, JSP 1.0 版本是目前 Sun 提供的最新的 JSP 版本。

JSP 1.1 版本: JSP 1.1 版本只有草案, 没有公开的测试版本。

1.2 JSP 的主要特性

JSP 是动态网页开发技术, 它的主要特性如下:

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 标识来设计最终页面。使用 JSP 标识或者小脚本来生成页面上的动态内容(内容是根据请求来变化的, 例如请求帐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识和 JavaBeans 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识和 Beans 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)页面的形式发送回浏览器。这有助于作者保护自己的代码, 而又保证了任何基于 HTML 和 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(JavaBeans 或者 Enterprise JavaBeans 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. 采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的编程人员。而 JavaServer Page 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。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, 以及执行用其他方法更难于编程且耗时的功能。

通过开发定制化标识库, JSP 技术是可以扩展的, 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识库。这使得 Web 页面开发人员能够使用熟悉的工具和执行特定功

能的构件来工作。

4. 具有“一次编写、各处运行”的特点

JSP 在设计时,充分考虑到应用平台的无关性。因此 JSP 完全与平台无关,可以在 Apache、Netscape、IIS 等服务器上执行。支持的操作系统平台有 Solaris、Windows、Mac OS、Linux 等。

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,各处运行”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,你可以使用自己所选择的服务器和工具,更改工具或服务器且不影响当前的应用。

5. 程序执行的效率高

在 JSP 中,代码被编译成 Servlet 并由 Java 虚拟机执行,编译过程只在程序第一次执行时进行,不需要每次执行程序时都编译,因此,代码的执行效率很高。

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,所以 JSP 页面就具有 Java 技术的所有好处,包括健壮的存储管理功能和安全性。与 SERVLETS 相比,JSP 能提供所有的 SERVLETS 功能,但它比用 `println` 书写和修改 HTML 更方便。

6. 强大的数据库支持

JSP 可以和任何与 JDBC 兼容的数据库建立连接,操纵数据库数据。Sun 公司还开发了 JDBC-ODBC bridge,用此技术,JSP 程序就可以访问带有 ODBC 驱动程序数据库。目前大多数数据库系统都带有 ODBC 驱动程序,所以 JSP 程序能访问诸如 Oracle、Sybase、MS SQL Server 和 MS Access 等主流数据库。

7. JSP 的效率和安全性更高

ASP 以源码形式存放,以解释方式运行,每次 ASP 网页调用都需要对源码进行解释,运行效率不高。另外,IIS 的漏洞曾使得许多网站的源程序大曝光,包括笔者以前用 ASP 开发的网站,ASP 程序就全部被人下载了去。

JSP 在执行以前先被编译成字节码(byte code),字节码由 Java 虚拟机(Java Virtual Machine)解释执行,比源码解释的效率高;服务器上还有字节码的 Cache 机制,能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢,因为它被编译成 Cache,以后就快得多了。同时,JSP 源程序不大可能被下载,特别是 JavaBean 程序完全可以放到不对外的目录中。

8. JSP 的组件(Component)方式更方便

ASP 通过 COM 来扩充复杂的功能,如文件上载、发送 E-mail 以及将业务处理或者复杂分离出来成为独立可重复利用的模块。JSP 通过 JavaBean 实现了同样的功能扩充。

在开发方面,COM 的开发远比 JavaBean 复杂和繁琐,学会 ASP 不难,但学会开发 COM 可不简单。而开发 JavaBean 就简单多了。

在维护方面,COM 必须在服务器上注册,如果修改了 COM 程序,就必须重新注册,甚至必须关机和重新启动。JavaBean 则不需要注册,放在 CLASSPATH 包含的目录中就行了。如果 JavaBean 进行修改,则 JSWDK 和 Tomcat 现在还需要关闭和重新运行(但不是关机),但开发者已经许诺将在以后的版本中做到不需要关闭服务器。

另外 JavaBean 是完全的 OOP,可以针对不同的业务处理功能方便地建立一整套可重复利用的对象库,例如用户权限控制、E-mail 自动回复等等。

9. JSP 的适应平台更广

ASP 目前仅适用于 NT 和 IIS。虽然 Unix 下有 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，而且 Unix 下的 COM 实现起来非常困难。

JSP 则不同，几乎所有平台都支持 Java，所以 JSP+JavaBean 可以在所有平台下通行无阻。NT 下 IIS 通过一插件，例如 JRUN(<http://www3.allaire.com/products/jrun>) 或者 ServletExec(<http://www.newatlanta.com/>)，就能支持 JSP。著名的 Web 服务器 Apache 已经能够支持 JSP，由于 Apache 广泛应用在 NT、Unix 和 Linux 上，因此 JSP 有更广泛的运行平台。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 字节码都是标准的与平台无关的。笔者将在 NT 下实验的 JSP 程序原封不动地拿到 Linux 下就运行起来了，感觉非常满意。

1.3 JSP 与 ASP 的比较

JSP 与 Microsoft 的 ASP 技术非常相似。两者都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。在 ASP 或 JSP 环境下，HTML 代码主要负责描述信息的显示样式，而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器，而 ASP 和 JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中，然后一起发送给浏览器。ASP 和 JSP 都是面向 Web 服务器的技术，客户端浏览器不需要任何附加的软件支持。

总的来讲，JSP 与 ASP 在技术方面有许多相似之处，两者都是为基于 Web 应用实现动态交互网页制作的提供的技术环境。两者都能够为程序开发人员提供实现应用程序的编制与自带组件设计网页的从逻辑上分离的技术。

尽管 Sun 的 JSP 和 Microsoft 的 ASP 在许多方面都相似，但仍然存在很多不同之处，其中最本质的区别在于：两者来源于不同的技术规范组织，其实现的基础：Web 服务器平台要求不同。

JSP 与 ASP 的比较应从以下几个方面来考虑：

1. JSP 与 ASP 技术的比较

JSP 与 ASP 技术明显的区别如下：

- JSP 技术基于平台和服务器的互相独立，技术支持来自广泛的、专业化的各种工具包，由服务器的组件和数据库产品开发商所提供。
- ASP 技术主要依赖于 Microsoft 的技术支持。

JSP 与 ASP 技术比较如表 1-1 所示。

表 1-1 JSP 与 ASP 技术比较

比较项目	ASP 技术	JSP 技术
Web 服务器	微软的 IIS 或 Personal Web Server	任何 Web 服务器, Apache、Netscape、IIS 等
操作系统平台	微软的 Windows 系统	微软的 Windows 系列平台、Sun Solaris 平台、Mac OS、Linux 平台等
跨平台访问	需要第三方 ASP 的引入产品	支持 Web 系统中不同系列的计算机群, 即保护用户在当前软硬件及人力资源上的投资完全兼容。JSP 技术提供灵活、开放的选择; 可以使用各种各样的工具及提供商提供的工具, 高度体现工业化标准输入与配置
自定义扩充标签	不能	可以
可将资料与程序逻辑分开	可以, 使用 Com 组件	不能, 使用 JavaBean 技术
建立大型 Web 应用程序	可以	可以
与数据库连接	ODBC 或 Com 组件	JDBC API 或 Java Bean
执行模式	编成 P Code	以 Java byte code 存在
自定义函数	可以	可以
页面直接包含代码	可以	可以
扩展名	.ASP	.JSP

2. JSP 与 ASP 结构的比较

JSP 和 ASP 在结构上类似, 两者都是面向对象的程序设计语言, 都是采用 HTML 文件内直接嵌入脚本语言的方法, 使静态的 Web 页面升至为动态 Web 页面。它们的脚本语言标签都是以“<%”和“%>”作为标签符, 但不同的是 JSP 标签内的内容是 Java 代码段, 而 ASP 标签内的内容是 VBScript 代码段。

由于 ASP 的编程语言是 VBScript 之类的脚本语言, JSP 使用的是 Java, 这是两者最明显的区别。此外, ASP 与 JSP 还有一个更为本质的区别: 两种语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下, VBScript 代码被 ASP 引擎解释执行; 在 JSP 下, 代码被编译成 Servlet 并由 Java 虚拟机执行, 这种编译操作仅在对 JSP 页面的第一次请求时发生。

3. JSP 与 ASP 平台和服务器独立性的比较

JSP 与 ASP 平台和服务器独立性的比较如下:

- JSP 技术依附于一次写入之后便可以运行在任何具有符合 Java 语法结构的环境。取而代之过去的依附于单一平台或开发商; 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各样的工具包。

Java 与 C++、VB 等语言比较, 被看作是个效率不太高的语言, 但它是牺牲效率换来了平台无关性, 使 Java 可以在各式各样不同种类的机器和大多数操作系统上运行而不需要重新编译, 许多人愿意为此而付出代价。JAVA 的编译代码是一种字节代码, 在运行时由操作系统上的一个 Java VirtualMachine(JVM)虚拟机解释。这个字节代码可以在所有平台上迁移, 而不需要任何改动。交叉平台编程倾向于采用所有操作系统的“最小公分母”, 尤

其是使用图形化界面时(servlets 不需要任何图形界面)。Servlets 比传统的 java 程序(applets, javaapp)要快, 因为它们在服务器端运行, 不需要加载沉重的 GUI (HTML 的 GUI 是非常少的)。另外, servlets 的字节代码只有在客户请求时才执行, 所以尽管在首次调用 servlets 时会有几秒钟的加载时间, 但后续的请求非常迅速, 因为服务器已经缓存了运行的 servlets。当前的 jsp 服务器, 都带有 java 即时编译器(JIT), 因此, JSP 的执行比每次都要解释执行的 ASP 代码要快, 尤其是在代码中存在循环操作时, JSP 的执行速度要比 ASP 快 1 到 2 个数量级。

- ASP 是基于 ActiveX 控件技术的客户端和服务器的开发组件, 因此 ASP 技术基本上是局限于微软的操作系统平台之上的。ASP 主要工作环境是微软的 IIS 应用程序结构, 又因 ActiveX 对象有平台局限性, ASP 技术不能很容易地实现在跨平台的 Web 服务器上的工作。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。

4. JSP 与 ASP 在开发过程的源代码上的比较

JSP 与 ASP 在开发过程的源代码上的比较如下:

- JSP 是 Sun 公司采用开放过程方法同国际 Java 组织合作开发和修改了 Java 技术与规范而开发的产品。针对 JSP 产品, Sun 授权了工具提供商(例如: Macromedia)、结盟公司(例如: Apache、Netscape)、最终用户以及协作商等。Sun 将 JSP 和 Java Servlets 的源代码发放给 Apache, 以求 JSP 与 Apache 紧密地相互发展, 使得 JSP 应用程序接口(API)取得了成功。
- ASP 技术只能依靠 Microsoft 本身的推动, 其发展建立在独立的、封闭的开发过程基础之上。

5. JSP 与 ASP Session 管理的比较

为了跟踪用户的操作状态, ASP 使用了 Session 对象, 而 JSP 使用了一个叫 HttpSession 的对象来实现同样的功能。Session 的信息保存在服务器端, Session 的 ID 保存在客户机的 cookie 中。如果客户机禁止 cookie, Session ID 就必须跟在 URL 后面。Session 一般在服务器上设置了一个 30 分钟的过期时间, 当客户停止活动后会自动失效。Session 中保存和检索的信息不能是基本数据类型, 如: int, double, 而必须是 java 的相应对象, 如: Integer, Double。

Session 语句

HttpSession session = request.getSession(); //获取一个 Session 对象或者创建一个 Session 对象。

session.getId(); //获取 Session ID 数

Integer id = (Integer)session.getValue("id"); 检索出 Session 的值并转化为整型

session.putValue("ItemValue", itemValue); //ItemValue 必须不是简单类型

6. JSP 与 ASP 中 Application 管理的比较

- ASP 的 Application 是服务器管理面向整个应用的参数, 它使得每个客户都能获得同样的参数值。
- JSP 与之对应的是 ServletContext 对象, 操作的方法和 Session 一样。

①设置一个应用变量

getServletContext().setAttribute("Item", ItemValue);

②获取 item

```
Integer i=(Integer)getServletContext().getAttribute("ItemName");
```

7. JSP 与 ASP 重用组件的比较

JSP 和 ASP 都使用封装组件, 都可以让开发者通过访问网页中的组件, 将生成的内容与布局分离。ASP 支持 COM, 而 JSP 提供了基于 JavaBeans 技术的组件或 JSP 标签。重用组件使得页面简单快速。COM 和 ASP 的结合等同于 JavaBeans 和 JSP 的结合, 就它们之间的差异比较如下:

①从编程语言上来看

- JSP 的 JavaBeans 只能用 java 语言开发。
- ASP 的 COM 可以由符合标准对象模型的任何语言开发(如: Visual J++), 因为 COM 组件由编译语言(C++,VB)构成(而不是解释型的 scripting 语言 VBScript, JavaScript)。

②从掌握的难易程度来看

- JSP 的 JavaBeans 更容易开发, 因为一旦掌握了 Java, 了解 JavaBeans 的结构是一件非常容易的事情。
- ASP 的 COM 比较复杂, 需要大量的学习和培训才能掌握, 即使你具有 C++和 VB 的基础, 也需要付出许多额外的努力。

③从访问权限来看

- JSP 的 JavaBeans 不需要重新注册, 如果开发者不需要得到对服务器的完全访问权限, 则无须重新注册, 这是它的一个巨大的优点。
- ASP 的 COM 组件需要在服务器上注册, 如果你修改了一个现有的组件, 服务器需要重新启动才能使用它。

8. JSP 与 ASP 连接数据库的比较

- JSP 通过 JDBC 技术联接数据库

目标数据库需要一个 JDBC 驱动程序, 一个数据库与 Java 的接口, 使得 Java 可以用标准的方式访问数据库。JDBC 不使用服务器端的数据源, 只要有 JDBC 驱动程序, Java 就可以访问数据库了。

如果一个特定的数据库没有 JDBC 驱动程序, 而只有 ODBC 驱动程序, Java 提供一个 JDBC-ODBC 桥来将 JDBC 调用转化为 ODBC 调用。所有的 Java 编译器都带有一个免费的 JDBC-ODBC 桥, 理论上 JDBC-ODBC 桥可以访问任何常见的数据库产品。

- ASP 使用 ODBC 通过 ADO 连接数据库。

9. JSP 与 ASP 自定义扩充标记的比较

虽然 ASP 和 JSP 都用标记和脚本来建立动态 Web 页面。但 JSP 技术可以让开发者自己扩展 JSP 标签, 而 ASP 不能。

JSP 开发者能建立自定义标签库, JSP 技术的这一标签扩展能力, 给开发者在访问网页时共享函数包提供了一个简单类 XML 的接口。因为开发者可用类 XML 标签访问更多的函数, 且只依靠很少的脚本。所以运用自定义标签功能, 开发者可以建立更易读的网页, 使得逻辑设计及扩展函数的繁琐内容在程序中不予显现。

10. JSP 与 ASP 的 Server Side Includes 语句

JSP 和 ASP 都支持在页面直接包含代码的功能, 即在服务器(如: IIS,Apache)上实现服

务器端包含虚文件。但 JSP 不仅仅包含程序代码本身，即静态文件，也能将动态页面的结果包含进来，当你包含的文件在另一个服务器上时，不包含任何代码和对象是一个非常有效的功能。

ASP 的包含语句：`<!--#include file="subfile.asp"-->`

JSP 的包含语句之一：`<%@ include file="subfile.jsp" %>`

JSP 的包含语句之二：`<jsp: include page="subfile.jsp"></jsp:include>`

归纳以上 JSP 与 ASP 的比较，可以得到以下的结论：

- JSP 使用了灵活而强大的 Java 语言，取代了 ASP 效率不高的 Scripting 语言；在平台技术上，JSP 率先实现了跨平台操作，摆脱了 ASP 使开发者只能使用基于 Windows 平台的束缚；在编译技术方面，JSP 的一次编译一直运行的高效率，使得 ASP 每次请求每次解释的执行效率受到了怀疑。
- JSP 模型是在 ASP 之后定义的，它借用了 ASP 的许多优点。更为重要的一点是，JSP 技术是通过 Java 开放模式开发的，它可以广泛地被其他开发工具、Web 服务器和应用服务开发商所支持。这使得它比其它技术的发展更具多元化、融入了更多的智慧。

1.4 JSP 的运行模式

1. JSP 运行模式具有的特点

①JSP 的所有程序都在服务器端执行，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代码，然后将执行结果以 HTML 格式返回给客户。网络上传送给客户端的仅是得到的结果。

②它对客户浏览器的要求最低，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame 等。它的程序代码可以操作数据库、重新定向网页以及发送 email 等等。

③它实现了动态页面与静态页面的分离，完全摆脱了硬件平台的束缚，以及编译后运行等方式，大大提高了程序执行效率，成为当今因特网上的主流开发工具。

④它使得开发者的工作比用其它方式更容易。Java 语言拥有内存区域管理，可以防止内存漏洞和提供对难于发现的降低应用效率的指针 Bugs 的保护。此外，JSP 还为现实世界的应用提供了强有力的例外处理能力。

2. JSP 的运行模式

Web 服务器向浏览器传送一个页面的过程如下：

- 浏览器发出对 Web 页面的请求；
- 浏览器利用 URL 辨别 Web 服务器的地址，找到自己需要的主页，并给出 Web 服务器需要的其它信息。这些信息包括名称(Netscape)、版本(4.5)、操作系统(Windows 98)以及由用户来填写的表。
- 当请求的是 HTML 文件时，Web 服务器就简单地找到该文件，然后传送该文件内容到浏览器，浏览器得到内容后就开始将基于 HTML 的代码译成页面。
- 当请求的是 JSP 页面时，并不立即发送文件的内容。首先执行脚本。由脚本产生一些 HTML 代码，然后将这些代码传送给 JSP 执行引擎，再由它将 HTML 文件传送给 Web 浏览器。JSP 的运行模式如图 1-1 所示。

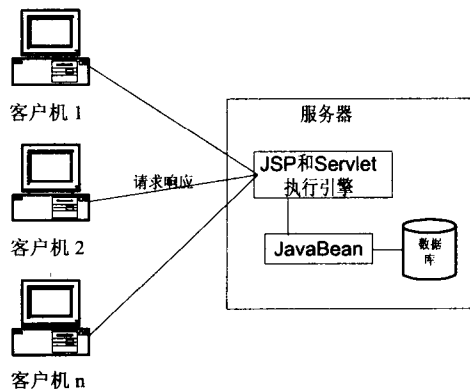


图 1-1 运行模式

3. JSP 的运行过程

当 WEB 服务器接收到一个后缀为".jsp"页的请求时，触发 JSP 引擎。JSP 引擎首先检查 JSP 文件是新的还是未修改过的文件，对这两种情况会有不同的处理。

JSP 引擎把 JSP 文件转换成 Java Servlet，然后使用标准的 Java 编译器编译 Servlet，使用标准的 API 执行 Java Servlet。这个翻译步骤也是 JSP 为什么慢的一个原因。

对于旧文件则直接进行编译，省略了前面翻译的工作，接下来的步骤是一样的。

JSP 是将网页的表现形式和服务端端的代码逻辑分开。作为服务器进程的 JSP 页面，首先被转换成 Servlet(一种服务器端运行的 Java Bytecode)。Servlet 支持 HTTP 协议的请求和响应。多个用户同时请求一个 JSP 页面时，应用实例化线程来响应请求。这些线程由 Web 服务器进程来管理。

JSP 在执行以前先被编写成字节码，然后字节码由 Java 虚拟机解释执行，比源码解释的效率高——服务器上还有字节码的 Cache 机制，能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢，因为它被编译成 Cache，以后就快得多了。另外，JSP 源程序不大可能被下载，特别是 JavaBeans 程序完全可以放到不对外的目录中。JSP 的运行过程如图 1-2 所示。

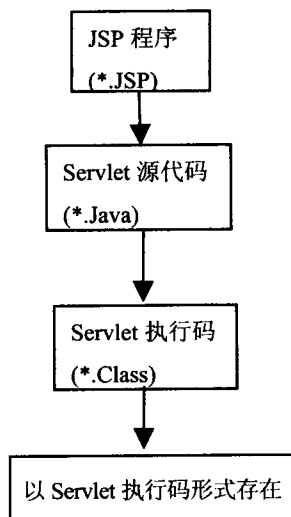


图 1-2 JSP 的运行过程

1.5 如何建立 JSP 运行环境

要使用 JSP 技术,就必须先配置好 JSP 的运行环境,选择好 JSP 的开发工具。如果要发布一个网站,除了要有服务器硬件外,还要有相应的服务器软件。

1.5.1 JSP 运行环境需要的组件

需要安装 Java2 软件开发工具箱(J2SDK),过去称之为 Java 开发工具箱(JDK)、JavaServer 网络开发工具箱(JSWDK)、Tomcat,或者其它能支持 JSP 的 Web 服务器。Sun 公司为 Windows、Solaris 以及 Linux 平台提供免费的 J2SDK 和 JSWDK。

如果你想在现有的 Web 服务器上使用 JSP,而此服务器本身不支持 JSP 和 Java 代码片段,可以试用 Allaire 公司的 Jrun。它可以为 Netscape 的 Enterprise Server 和 Fasttrack Server、微软的 Internet Information Server(IIS)和 Personal Web Server(PWS)、Apache 以及其他服务器充当 Web 服务器的附件。也可以使用 Apache 服务器的 Java 版,其中包含最新的 JSWDK。

- J2SDK 可在<http://java.sun.Com/jdk>处下载;
- JSWDK 可在<http://java.sun.Com/Products/jsp/>处下载;
- Tomcat 可在<http://jakarta.apache.org/>处下载。

支持 JSP 的 Web 服务器必须要有相应的配置。当今流行的 Web 服务器有: JSWDK、TOMCAT、RESIN 等。分别介绍如下:

1. JSWDK

Java Server 网络开发工具箱(JSWDK)是 JSP 最容易使用的 Web 服务器。目前, J2SDK 的最新版本为 1.2.2-001。你可在<http://Java.Sun.Com/Products/jsp/>处下载 JSWDK (Java Server Web Development Kit 1.0.1), JSWDK 以压缩文档的方式在 Web 上提供。JSWDK 的安装只需将 jsjdk1-0-1-Win.Zip 文件释放到硬盘目录下(c:\等),然后就可以在硬盘上找到\jsjdk-1.0.1\目录。如果不想保留 JSWDK 时,删除这个目录就可以了,没有任何系统文件和注册表遗留问题。

JSWDK 的 Web 服务器参数保存在\jsjdk-1.0.1\Web Server.Xml 中,用 Windows 写字板打开并编辑这个文件就可以修改 JSWDK 的默认设置值。

JSWDK 默认的文档目录是\jsjdk-1.0.1\Webpages,在此目录下可以建立子目录,例如:\jsjdk-1.0.1\Webpages\test,就能在浏览器中用 <http://localhost:8080/test> 访问这个目录,并且执行其中的 JSP 程序。JSWDK 主文档在默认状态下为 index.html 和 index.jsp。也就是说访问 <http://localhsot: 8080> 等于访问\jsjdk-1.0.1\Webpages\index.html。

JSWDK 软件包中的示例页面分为两类,它们或者是 JSP 文件,或者是包含一个表单的 HTML 文件,这些表单均由 JSP 代码处理。与 ASP 一样, JSP 中的 Java 代码均在服务器端执行。因此,在浏览器中使用“查看源文件”菜单命令是无法看到 JSP 源代码的,而只能看到结果 HTML 代码。所有示例的源代码均通过一个单独的“examples”页面提供。JSWDK 的初始页面如图 1-3 所示。

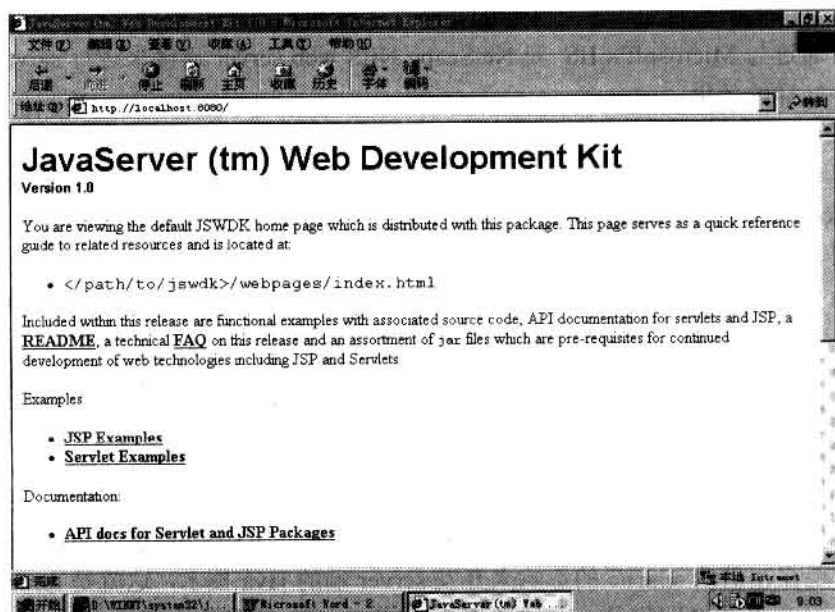


图 1-3 JSWDK 的初始页面

2. Tomcat

Tomcat 是一个 servlet 容器和 JavaServer Pages(tm)执行器。它可以单独使用或和以下几个流行的网络服务器关联在一起使用：

- Apache, version 1.3 或更新版本
- Microsoft Internet Information Server, version 4.0 或更新版本
- Microsoft Personal Web Server, version 4.0 或更新版本
- Netscape Enterprise Server, version 3.0 或更新版本

Apache Tomcat 3.1/3.2 是 Apache Group Jakarta 小组开发的一个免费服务器软件，适合于嵌入 Apache 中使用。而且，它的源代码可以免费获得，你可以自由地对它进行扩充。不足之处是它的配置十分麻烦，而且有一些安全性的问题没有解决，例如，低版本的 Tomcat 不支持大写的 JSP 扩展名，如果用户请求大写扩展名的 JSP 程序，那么 Tomcat 可能会将 JSP 程序的源代码全显示在客户端的浏览器上。

(1) 安装和运行 Tomcat

Tomcat 需要一个 Java 运行环境(包含在任何 Java2 平台系统里的 JRE1.1 或更新版本)。假如你想开发应用，你需要一个 Java 编译器，比如包括在 Java Development Kit1.1 或更新版本里的编译环境。

(如果想知道更多细节，请查阅 tomcat 目录下 doc/uguide/tomcat_ug.html 文件)

(2) 使用 Tomcat 进行应用开发

如果你在使用 Tomcat 进行 Servlet 或基于 JSP 的应用开发，可参照应用开发者指南中这方面的信息。具体可参照 doc/appdev/index.html。

(3) 版本中的新特征

在 Tomcat3.0 版本后加入了一些新的特征，主要包括以下几方面：

——可通过与 Microsoft's IIS 和 NetScape NES web server 连接来运行 servlet，能获得与单独运行一样的效果。

——可自动产生 Apache 的配置文件。

——可自动调度一个应用并描述出一个 Web 文档(WAR)文件。配置一个应用，复制这个文件到 Tomcat 目录\webapps 目录下并且重新启动 Tomcat，WAR 文件将会自动扩展。移除一个应用，只要删除那个应用的目录以及相应的 WAR 文件即可。

——当 WEB-INF/classes 目录下的 class 文件或 WEB-INF/lib/*.jar 文件有改动时自动重新装载。

——最小限度的管理工具，允许一个新的 web 应用被扩展到正在运行的 Tomcat 环境或从一个已经存正的应用中移除。

——充分改进的安装和开发文档。

Tomcat 很可能成为下一代 Java Web Server 的主流。因为 Tomcat 受到 Sun 公司的全力支持，并由非常强大的开发组织 apache 来进行发展，这一工程被称为 Jakarta 计划。从战略上看，Sun 现在正借助 apache 的影响来开发 server 端的 java 技术，这就是 tomcat。因此可以相信 tomcat 已经或者即将是一个较理想的 jsp & servlet 开发和支撑平台。相对地，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，而且实际上 SUN 并不许可将它作为 internet 上的商业平台。

tomcat 的功能比 jws 或 jsjdk 强大得多，可以访问 tomcat 的站点 <http://jakarta.apache.org> 查看详细介绍，或者订一个 tomcat 的 maillist，还可以加入到他们的开发组织中去。

tomcat 有不同的工具来共同实现强大的功能。在 jakarta 工程有一些相当不错的相关子项目：

- Ant: 基于 java 的跨平台开发工具，支持 XML。
- Regexp: 100%纯 Java 表达与调试工具包，可是 debug 和测试 java 兼容性。
- Slide: 是一个包含了 Servlet API 内容管理，用以推进 WebDAV 协议(web-based Distributed Authoring and Versioning)，从而使 servlet 可以在任何支持 api 2.2 或以上版本的容器内运行。
- Struts: 用于制作 jsp/servlet 的 web 通用应用开发框架，即 MVC(Model-View-Controller)。
- Taglibs: 是一个与 JSP1.1 标准兼容的自定义标记的库。作为一个强有力的 jsp 结构特性，这将为 jsp 开发注入更多功能，开发也变得更友好。
- Watchdog: 包含在 tomcat3.1 版中，用于检测 servlet 和 jsp 的兼容性。

3. Resin

Resin 提供了最快的 jsp/servlets 运行平台。在 java 和 javascript 的支持下，Resin 可以为任务灵活地选用合适的开发语言。Resin 的一种先进的语言 XSL(XML stylesheet language)可以使得形式和内容相分离。

如果选用 jsp 平台作为 internet 商业站点的支持，那么速度、价格和稳定性都是要考虑到的，resin 十分出色，表现更成熟，很具备商业软件的要求。而且，它是全免费的，从站点下载的就是完整版本，所以值得向您推荐！

相对于 tomcat3.1,后者更象是一个正在研究的项目。目前 resin 可以支持 sun 的 j2ee, 而 tomcat 不能直接支持, 而 j2ee 是基于 java 服务器端大系统的基础。但 tomcat 结构非常合理, 而且是 apache 组织的产品, 因此有着很好的远景。

Resin1.1 的特性包括:

- (1) 支持 jsp1.1 和在服务器端编译的 javascript。
- (2) 比 mod_perl,mod_php 更快, 比 Jakarta Tomcat 快 3 倍。您可以在 Caücho 的站点看到很全面的各种测试报告。
- (3) 使用 servlet2.2 引擎。
- (4) 自动的 servlet/bean 编译!。
- (5) Servlet,XSL Filtering。
- (6) 支持 IIS,Apache,Netscape 和其它内置了 HTTP/1.1 的 Web 服务器。
- (7) XSLT 和 XPath1.0 引擎。
- (8) 企业级的共享软件(基于一个开放源码的协议)。

1.5.2 J2SDK 及 JSDK 环境安装与配置

1. 安装

J2SDK 的最新版本为 1.2.2-001, 它以压缩文档方式提供, 文件大小大约为 20MB。这个软件提供一个彻底的 Java 开发环境, 让用户创造任何基于 Java 并且利用了其核心 API 的解决方案。实际上, Web 服务器使用 JSP 的唯一条件是要有 Java 编译器。要让 Web 服务器知道编译器的位置, 需要将 Java-HOME 环境变量设置为 J2SDK 安装目录。

修改系统环境参数, JDK 可以是 Sun JDK1.2.2 或 1.3, IBM JDK1.1.7 或 Blueprint's JDK。如果你是在 Windows98 上安装, 将这行 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\lib\tools.jar, 然后重新开机即可。

本人使用的是 JDK1.2.2, 在 Win2000 中右击“我的电脑”, 点击“属性”, 找到系统中的环境变量设置 path=\jak1.2.2\bin; CLASSPATH=[x:]\jdk1.2.2\lib\tools.jar。

在 Windows2000 下, 系统环境参数的配置按以下步骤进行:

1. 在桌面上用右键单击“我的电脑”, 然后选择“属性”命令。
2. 在弹出的“系统特性”对话框中选择“高级”标签。
3. 单击“环境变量”按钮, 出现“环境变量”对话框, 如图 1-4 所示。
4. 在“系统变量”列表中选择变量“path”, 点击“编辑”按钮。
5. 在弹出的“编辑系统变量”对话框中的“变量值”编辑栏中加入[x:]\path\bin, 其中[x:]\path 为 J2SDK 的安装目录。如图 1-5 所示。
6. 在“环境变量”对话框中单击“新建”按钮, 在弹出的“新建系统变量”对话框中加入变量名为“CLASSPATH”, 值为“[x:]\path\lib\tools.jar”的环境变量。如图 1-6 所示。