

229

7P3f3.4
u67

JSP 网络编程技术



翁卫兵
李耿等编著
丁卫霞



新 时 代 出 版 社

· 北 京 ·

图书在版编目(CIP)数据

JSP 网络编程技术/翁卫兵等编著. —北京: 新时代出版社, 2002. 1

ISBN 7-5042-0645-8

I. J... II. 翁... III. ①主页制作 - 技术 ②JAVA 语言 - 程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 070024 号

新时代出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 15 3/4 357 千字

2002 年 1 月第 1 版 2002 年 1 月北京第 1 次印刷

印数: 1—3000 册 定价: 22.00 元

(本书如有印装错误, 我社负责调换)

内 容 简 介

JSP 作为 Sun 公司推出的全新网站编程工具, 继承了 Java 的几乎全部优势, 符合“write once, run anywhere”的原则。JSP 在页面中可以嵌入几乎全部的 Java 语言, 并为分离处理逻辑与表示样式提供了最好的解决方案。

本书旨在提高读者的编程实践能力, 因此书中用通俗易懂的语言, 结合大量的实例, 全面系统地介绍了 JSP 及其相关知识, 分别对其安装、语法、表达式、对象及其活动范围、JavaBean 等作了极为细致的描述。

本书适合于初、中级水平的读者, 同时也可作为高级用户参考用书。

前 言

INTERNET 逐渐成为生活中不可缺少的一部分,而制作网页的动态语言也越来越多。目前,最常用的三种动态网页语言有 ASP(Active Server Pages)、JSP(Java Server Pages)和 PHP(Hypertext Preprocessor)。

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。Sun 公司借助自己在 Java 上的不凡造诣,将 Java 从 Java 应用程序和 Java Applet 发展到 JSP——Java Server Page。JSP 可以在 Servlet 和 JavaBean 的支持下,完成功能强大的站点程序。

与 PHP、ASP 相比,JSP 由于具有效率高、不受操作系统的限制等优点而受到了广大网络编程人员的欢迎。由于现在市场上关于 JSP 的书籍比较少,在学习 JSP 时只能根据网上的一些资料,并且其中大部分是英文资料。这使得广大编程人员把宝贵的时间大部分花费在资料的查找和翻译上。鉴于此,我们结合实际工作经验,编写了这本书。

由于时间仓促,限于作者的水平,本书的不足之处在所难免,欢迎广大读者不吝赐教。

编 者
2000 年 10 月

目 录

第一部分 Java Server Pages 基础知识

第一章 JSP 概述	1
1.1 JSP 技术的发展	2
1.2 ASP、PHP 和 JSP 的技术特点	3
1.3 JSP 与 ASP 的比较	5
1.3.1 结构	5
1.3.2 性能和平台无关性	5
1.3.3 Session 的管理	5
1.3.4 Application 管理	6
1.3.5 ServerSideIncludes	6
1.3.6 Java 组件:JavaBeans	6
1.3.7 数据库的连接	7
1.3.8 结论	7
1.4 Java Servlet	10
第二章 Java Servlet 编程	12
2.1 HTTP Servlet API	12
2.1.1 简述	12
2.1.2 常用 HTTP Servlet API 概览	13
2.2 系统信息	14
2.2.1 初始化参数	15
2.2.2 服务器	15
2.2.3 客户端	16
2.2.4 请求	17
2.3 传送 HTML 信息	22
2.3.1 response 的结构	22
2.3.2 发送标准的响应信息	23
2.3.3 使用持续连接	23
2.3.4 生成 HTML	24
2.3.5 状态码	25
2.3.6 HTTP 头	26
2.3.7 错误处理	27

第三章 JSP 的安装	32
3.1 JSP 的运行平台的软硬件要求	32
3.1.1 硬件要求	32
3.1.2 软件要求	32
3.2 JSP 运行平台的建立	32
3.2.1 在 Linux 上安装 JSP	32
3.2.2 在 NT 上建立 JSP 环境	36
第四章 JSP 名词定义	39
第五章 JSP 指令和脚本元素	41
5.1 HTML 注释	42
5.2 隐藏注释	43
5.3 声明	43
5.4 表达式	44
5.5 脚本片段 (Scriptlet)	45
5.6 JSP 指令列表	46
5.6.1 Page 指令	46
5.6.2 Include 指令	52
5.6.3 Taglib 指令	53
5.6.4 <jsp:forward>	53
5.6.5 <jsp:getProperty>	54
5.6.6 <jsp:include>	55
5.6.7 <jsp:plugin>	56
5.6.8 <jsp:setProperty>	58
5.6.9 <jsp:useBean>	59

第二部分 Java Server Pages 的对象

第六章 JSP 和 HTTP 协议	62
6.1 Response 对象的使用	64
6.2 Request 对象的使用	68
6.2.1 getQueryString() 方法	68
6.2.2 request.querystring() 方法	68
6.2.3 getParameter() 方法	69
6.2.4 getAuthType() 方法	69
6.2.5 在一个 JSP 的请求处理中如何停止 JSP 的执行	69
第七章 会话的操作	73
7.1 会话的简介	73
7.2 会话的状态行	73
7.3 Session 对象的应用的例子	74

7.4	控制会话的结束时间	77
7.5	Cookie 对象的概述	78
7.6	与 Cookie 有关的例子	79
第八章	HTML 表单的处理	82
8.1	获取表单数据	82
8.2	表单数据的确认	86
第九章	JSP 其他隐含对象	87
9.1	Application 对象	87
9.2	Out 对象	89
9.3	pageContext 对象	90
9.4	Config 对象	93
9.5	Page 对象	93
9.6	Exception 对象	94
第十章	JSP 非隐含对象	95
10.1	Servlet 对象	95
10.2	ServletRequest 对象	96
10.3	ServletResponse 对象	97
10.4	JspFactory 对象	98
10.5	String 对象	99
10.6	Enumeration 对象	102
10.7	HttpSession 对象	103
10.8	HttpSessionContext 对象	104
10.9	HttpServlet 对象	104
10.10	Write 对象	104
10.11	Principal 对象	105
10.12	URL 对象	106
10.13	InputStream 对象	107
10.14	RequestDispatcher 对象	108
10.15	GenericServlet 对象	108
10.16	异常类	109

第三部分 Java Server Pages 高级主题

第十一章	JDBC 技术	110
11.1	概述	110
11.2	目标与哲学	111
11.3	接口概貌	112
11.3.1	JDBC API	113
11.3.2	JDBC Driver API	113

11.4 JDBC 的使用场合	113
11.4.1 Applet	113
11.4.2 Application	114
11.4.3 其他场合	114
11.5 安全性考虑	115
11.5.1 JDBC 和未验证的 applet	115
11.5.2 JDBC 和 Java 应用程序	115
11.5.3 Driver 的安全责任	115
第十二章 JSP 与数据库	117
12.1 数据库连接	117
12.2 对数据库进行操作	118
12.3 对数据库进行操作的示例	118
第十三章 文本文件操作	128
13.1 文件的读取	128
13.2 文件的写入	132
13.3 将数据追加到一个文件中	134
第十四章 电子邮件的处理	138
第十五章 中间件与 JSP 技术	143
15.1 中间件技术	143
15.1.1 什么是中间件	144
15.1.2 中间件的作用和分类	144
15.1.3 中间件模型和形态	145
15.2 分布式应用开发技术概述	145
15.2.1 CORBA、DCOM、RMI	145
15.2.2 发展趋势	146
15.2.3 三种主流技术比较	148
15.3 CORBA 技术及实例	149
15.3.1 CORBA 简介	150
15.3.2 用 Java 做 CORBA 开发实例	151
15.4 JSP 与 CORBA 技术的结合使用	157

第四部分 Java Server Pages 样本应用程序

第十六章 JSP 应用程序示例	158
16.1 猜数字游戏	158
16.2 获取与日期有关的程序示例	161
16.3 获取有关系统信息的程序	163
16.4 错误处理示例程序	164
16.5 日程表	168

16.6	留言板	173
16.7	网上商店	183
16.8	聊天室	193

第五部分 附录

附录 A	关于 JSP 的常见问题	214
附录 B	Java 程序基础	221
附录 C	SQL 参考	233
附录 D	JSP 参考资料相关网址	241

第一部分 Java Server Pages 基础知识

第一章 JSP 概述

本章主要内容:

- (1) JSP 的发展;
- (2) JSP 的介绍;
- (3) JSP 的版本;
- (4) JSP、ASP、PHP 的技术特点以及 JSP 与 ASP 的比较。

INTERNET 越来越成为生活中不可或缺的一部分,制作网页的动态语言也越来越多。目前,最常用的三种动态网页语言是 ASP(Active Server Pages)、JSP(Java Server Pages)和 HPP(Hypertext Preprocessor Pages)。

ASP 是一个 Web 服务器端的开发环境,利用它可以产生和运行动态的、交互的、高性能的 Web 服务应用程序,同时 ASP 采用了脚本语言 VB Script(Java script)作为自己的开发语言。

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,并耦合 PHP 自己的特性,使 Web 开发者能够快速地生成动态页面。它支持目前绝大多数数据库。另外,PHP 是完全免费的,不用花钱,完全可以从 PHP 官方站点(<http://www.php.net>)自由下载。而且可以不受限制地获得源码,甚至加进自己的特色。

JSP 是 Java Server Page 的缩写,是 Sun 公司于 1999 年 6 月推出的新技术,该技术由 Sun 公司主导,并采纳计算机软硬件、通信、数据库领域多家厂商的意见而共同制定的一种基于 Java 的 Web 动态页面技术。

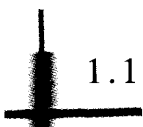
在 Sun 正式发布 JSP 之后,这种新的 Web 应用开发技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。按照 Sun 的说法,JSP 能够适应市场上包括 Apache WebServer、IIS4.0 在内的 85% 的服务器产品。

JSP 可以在 Servlet 和 JavaBean 的支持下,完成功能强大的站点程序。它提供在 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。JSP 代码被编译成 Servlet,并由 JVM(Java 虚拟机)解释执行,这种编译操作仅在对 JSP 页面的第一次请求时发生。在 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 JSP 页面需要附加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。JSP 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。



现在的企业应用已从 2-tier 朝 3-tier 结构转移,开发者有许多不同的方法分发应用数据到用户。JSP API 是 J2EE 的一部分,给 Web 和 Java 开发者创建各种应用提供了一个简单而有力的工具。

JSP 给开发者提供了两个重要的能力:(1) 它提供了访问远程数据的机制,如 EJB、RMI、JDBC;(2) 它让开发者从表现中封装和分离程序逻辑,从而使其达到最大程度的代码重用和灵活性。逻辑与表现的分离是 JSP 对比其他 Web 应用结构的一个主要优势。



1.1 JSP 技术的发展

Java 技术是由美国 Sun 公司研制和开发的,它包括 Java 语言和 Java Media APIs、Security APIs、Management APIs、Java Applet、Java RMI、JavaBeans、JavaOS、Java Servlet 以及 JDBC 等,现把 Java 技术的发展简述如下:

- 1990 年,Sun 公司的 James Gosling 领导小组设计了一种平台独立语言 Oak,主要用于为各种家用电器编写程序。

- 1995 年 1 月,Oak 被改名为 Java,并与 1995 年 5 月 23 日,Sun 公司在 Sun World'95 上正式发布 Java 和 HotJava 浏览器。

- 1995 年 8 月至 12 月,Netscape 公司、Oracle 公司、Borland 公司、SGI 公司、Adobe 公司、IBM 公司、AT&T 公司、Intel 公司获得 Java 许可证。

- 1996 年 1 月,Sun 宣布成立新的业务部门——JavaSoft 部,以开发、销售基于 Java 技术的产品。同时推出 Java 开发工具包 JDK(Java Development Kit) 1.0,为开发人员提供用来编制 Java 应用软件的工具。

- 1996 年 2 月,Sun 发布 Java 芯片系列,包括 picoJava、MicroJava 和 UltraJava,并推出 Java 数据库链接 JDBC (Java Database Connectivity)。

- 1996 年 3 月,Sun 推出 Java WorkShop。

- 1996 年 4 月,Microsoft 公司、SCO 公司、(Apple)公司、NEC 公司等获得 Java 许可证。Sun 宣布苹果电脑、HP、日立、IBM、微软、Novell、SGI、SCO、Tandem 等公司可以把 Java 平台嵌入到其操作系统中。

- 1996 年 5 月,HP 公司、Sybase 公司获得 Java 许可证。北方电讯公司宣布将把 Java 技术和 Java 微处理器应用到其下一代电话机中的计划。5 月 29 日,Sun 在旧金山举行第一届 JavaOne 世界 Java 开发者大会,在大会上推出一系列 Java 平台新技术。

- 1996 年 8 月,Sun 通过互联网提供其第一个产品 Java WorkShop。

- 1996 年 9 月,Addison-Wesley 和 Sun 推出 Java 虚拟机规范和 Java 类库。

- 1996 年 10 月,“德州仪器”等公司获得 Java 许可证。Sun 提前完成 JavaBeans 规范的发布。发布第一个 Java JIT(Just-In-Time)编译器,并计划在 Java WorkShop 和 Solaris 操作系统中加入 JIT。10 月 29 日,Sun 发布 Java 企业计算技术,其中包括:JavaStation 网络计算机,65 家公司发布的 85 个 Java 应用产品,7 个新的 Java 培训课程和 Java 咨询服务,基于



Java 的 Solstice 互联网邮件软件,新的 Java 开发者服务支持,演示 HotJava Views、Java Tutor,完成 Java Card API 等。Sun 宣布完成 Java Card API 规范,这是智能卡使用的第一个开放 API。

- 1996 年 11 月,IBM 公司获得 JavaOS 和 HotJava 许可证。Novell 公司获得了 Java WorkShop 许可证。Sun 和 IBM 宣布双方就提供 Java 化的商业解决方案达成一项广泛协议,IBM 同意建立第一个 Java 检验中心。

- 1996 年 12 月,Xerox 等公司获得 Java 及 JavaOS 许可证。Sun 发布 JDK 1.1、Java 商贸工具包、JavaBeans 开发包及一系列 Java APIs。并推出新的 Java Server 产品系列,其中包括 Java Web Server、Java NC Server 和 Java Server Toolkit。Sun 发布 100% 纯 Java 计划。

- 1997 年 1 月,Sun 公司交付完善的 JavaBeans 开发包,这是在确定其规范后不到 8 个月内完成的。

- 1997 年 2 月,Sun 和 ARM 公司宣布同意使 JavaOS 能运行在 ARM 公司的 RISC 处理器架构上。Informix 公司宣布在其 Universal Server 和其他数据库产品上支持 JDK 1.1。Netscape 公司宣布其 Netscape Communicator 支持所有 Java 应用软件和核心 APIs。

- 1997 年 3 月,HP 公司获得 Java WorkShop 许可证并用于 HP-UX 操作系统;西门子公司、AG 公司获得 Java 许可证;日立半导体公司、Informix 公司等获得 JavaOS 许可证;Novell 公司获得 Java Studio 许可证。Sun 公司还发售 HotJava Browser 1.0,这是一种 Java 浏览环境,可以方便地按剪裁方式来编制专用的信息应用软件,如客户自助台和打上公司牌号的网络应用软件。

- 1999 年 6 月,Sun 发布 JDK 1.3 和 Java Web Server 2.0。

1.2 ASP、PHP 和 JSP 的技术特点

ASP 的技术特点

(1) ASP 使用 VBScript、JScript 等简单易懂的脚本语言,结合 HTML 代码,可快速地完成网站的应用程序。

(2) ASP 无须 Compile 编译,容易编写,可在服务器端直接运行。

(3) ASP 可以使用的编辑器较多,如 INTERDEV、普通的文本编辑器等。利用 Windows 的记事本,即可进行编辑设计。

(4) 与浏览器无关(Browser Independence)。用户端只要使用可执行 HTML 代码的浏览器,即可浏览 Active Server Pages 所设计的网页。Active Server Pages 所使用的脚本语言(VBScript、Jscript)均在 Web 服务器端执行,用户端的浏览器不需要执行这些脚本语言(!注:如果输送到浏览器里的 HTML 代码里有脚本语言,则用户端的浏览器必须能够执行这些脚本语言)。

(5) Active Server Pages 能与任何 ActiveX Script 语言相容。除了可使用 VBScript 或 JScript 语言来设计外,还可通过 Plug-In 的方式,使用由第三方所提供的其他脚本语



言, 譬如 REXX、Perl、Tel 等。脚本引擎是处理脚本程序的 COM (Component Object Model) 物件。

(6) ASP 可使用服务器端的脚本来产生客户端的脚本。

(7) ActiveX Server Components (ActiveX 服务器元件) 具有无限可扩充性。可以使用 Visual Basic、Java、Visual C++、COBOL 等多种语言来编写。

JSP 的技术特点

1. 将内容的生成和显示进行了分离

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, 而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台组件 (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提供服务。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 页面开发人员不一定都熟悉脚本语言。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 标识中进行动态内容生成时所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件, 设置或检索组件属性, 下载 Applet, 以及执行用其他方法更难于编码或耗时的功能。

通过开发定制化标识库, JSP 技术是可以扩展的。以后, 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。

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。作为采用 Java 技术家族的一部分, 以及 Java 2 (企业版体系结构) 的一个组成部分, JSP 技术能够支持高度复杂的基于 Web 的应用。

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 页面具有 Java 技术的所有优点, 包括强大的存储管理及安全性。

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, 用户可以自己选择、更改服务器或工具, 而并不影响当前的应用。



1.3 JSP 与 ASP 的比较

ASP 由于其功能简单实用受到广泛的欢迎,再加上 MS 的强有力支持,可以说是时下建设网站最为流行的语言,它借助 MS 的 Com+ 技术,几乎可以实现在 C/S 下能够实现的所有功能,可以说其具有相当的竞争力。ASP 的代码实现也很简单,MS 提供的 InterDev 在 RAD 的基础上,使 ASP 的开发一夜之间火遍了大江南北。但是其缺点是,由于 MS 长久以来的垄断,而只能运行在 MS 的平台上,因此在关键性的应用上使人难以接受。虽然它可以通过增加控件而在 LINUX 下使用,但是其功能最强大的 DCOM 控件在 LINUX 下却不能使用。而 SUN 公司在 Java 的基础下开发出的 JSP(Java Server Pages)实现了动态页面与静态页面的分离,脱离了硬件平台的束缚,以及编译后的运行方式大大提高了其执行效率。所以可以说今后 JSP 将是 INTERNET 编程的首选工具。为了使广大应用 ASP 的程序员们能尽快地学会 JSP,现把其主要特性对比如下:

1.3.1 结构

JSP 和 ASP 在结构上类似,都是以“<%”和“%>”作为标记符,不同的是在标记符之间的代码 ASP 为 JavaScript 或 VBScript 脚本,而 JSP 为 Java 脚本。JSP 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 程序)。Servlet 支持 HTTP 协议的请求和响应。当 JSP 被转换成纯 Java 代码、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,这和 ASP 的线程管理器功能类似。同 CGI 为每个请求创建一个进程的模式比较,这种方式效率高得多。

1.3.2 性能和平台无关性

与 C++、VB 等语言相比较,Java 是个效率不太高的语言,但它以牺牲效率换来的平台无关性,使 Java 可以在大多数操作系统上运行而不需要被重新编译,而这正是许多人所希望的。Java 的编译代码是一种字节代码,它可以在所有平台上迁移,而不需要做任何改动。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(servlet 不需要任何图形界面)。Servlet 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,所以尽管首次调用 Servlet 时会有几秒钟的加载时间,但后续的请求响应非常迅速,因为服务器已经缓存了运行的 Servlet。当前的 JSP 服务器,都带有 Java 即时编译器(JIT),因此,JSP 的执行比每次都要编译运行的 ASP 代码快,尤其是在代码中存在循环操作时,JSP 的速度要快 1~2 个数量级。

1.3.3 Session 的管理

为了跟踪用户的操作状态,ASP 应用了名为 Session 的对象,而 JSP 使用一个叫 Http-



Session 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,当客户停止活动超过这个期限后自动失效。Session 中保存和检索的信息不能是基本数据类型(primitive data types),如 int,double 等,而必须是 Java 相应的对象如 Integer,Double。

例如:

```
HttpSession session = request.getSession(); //得到一个 Session 对象或创建一个  
session.getId() //得到 SessionID 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 must 简单类型
```

1.3.4 Application 管理

有时服务器需要管理面向整个应用的参数,使得每个客户都能获得同样的参数值。和 Session 一样,ASP 使用 Application 对象,而 JSP 使用 ServletContext 对象,操作的方法和 Session 一样。

例如:

```
getServletContext().setAttribute("Item", ItemValue); //设置一个应用变量  
Integer id = (Integer) getServletContext().getAttribute("ItemName"); //得到 item
```

1.3.5 ServerSideIncludes

在服务器端的引用上 ASP 和 JSP 有着相同之处。ASP 和 JSP 都可以在支持此功能的服务器(IIS, APACHE)上实现服务器端包含虚文件。但 JSP 是将动态页面的结果包含进来,而不是包含文件的代码本身。当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。

例如:

```
ASP: <!-- #include file = "subfile.asp" -->  
JSP: <% @include file = "subfile.jsp" -->
```

1.3.6 Java 组件:JavaBeans

JavaBeans 是一些完成预定义功能的封装的对象数据。JavaBeans 和 JSP 与 COM 和 ASP 相比如下:

COM 对象常用来封装商业逻辑和为 ASP 页面完成高强度计算。重用的组件使得页面简单快速,因为组件由编译语言(C++, VB)构成,而不是解释型的 scripting 语言(VBScript, JScript)。

JavaBeans 只能用 Java 语言开发,COM 可以由符合标准对象模型的任何语言开发(包括 Visual J++)。另一方面,JavaBeans 更容易开发,一旦掌握了 Java,了解 JavaBeans 的结构非常容易。而 COM 的复杂性,使得它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。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。JavaBeans 不需要重新注册,同时开发者不需要得到对服务器的完全访问权限,这是其巨大的优点。

可以创建满足商业逻辑的完整的 JavaBeans 库,让非编程人员使用库来开发动态网站。Beans 可以在服务器端管理数据库连接。

JavaBeans 符合结构化对象模型;每个 Bean 有一个不带参数的构造函数,控制它的 servlet 可以使用内省(introspection)来设置其属性。要设置 Bean 的内置属性,必须使用带有属性名的 setProperty 标签。如果一个属性可以设置,Bean 需要有一个 setXxxx 方法,(Xxxx 用实际的属性名来代替)。

1.3.7 数据库的连接

ASP 使用 ODBC 通过 ADO 连接数据库,而 Java 通过 JDBC 连接数据库。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,桥可以访问任何数据库产品。

1.3.8 结论

JSP 模型是在 ASP 之后定义的,它借用了 ASP 的许多优点,如 Session, Application 等对象。同时 JSP 使用灵活而强大的 Java 语言,而不是低效的 Scripting 语言。ASP 开发者只能使用基于 windows 平台的技术,而 Java 和 Jsp 是跨平台的。

总的来讲,JavaServer Pages™(JSP)和 微软的 Active Server Pages(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提供的技术环境支持。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与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 使网站建设与发展变得较为简单快捷。

尽管 JSP 技术和 ASP 在许多方面都有相似的,但仍然存在很多不同之处,其中最本质上的区别在于:两者是来源于不同的技术规范组织,其实现的基础——WEB 服务器平台要求不相同。

与 ASP 相比,JSP 有其特有的优点,下面分别进行介绍。

• JSP 技术——开放的技术

JSP 和 ASP 技术明显的不同点是开发人员在对其各自软件体系设计的方式不同。JSP 技术基于平台和服务器是相互独立的,输入支持来自各种工具包,服务器的组件和数据库产品则由开发商所提供。相比之下,ASP 技术则是主要依赖微软的支持。

(1) 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。取代过去依附于单一平台或开发商的技术,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工具包。

由于 ASP 是基于 Activex 控件技术提供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。ASP 主要工作环境是微软的 IIS 应用程序结构,又因 Activex 对象具有平台特性,所以 ASP 技术不能很容易地实现跨平台的 WEB 服务器工作。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。

(2) 开放的开发过程,开放的原代码

SUN 公司应用 Java 社团性过程开发 JSP 技术。自 1995 年,SUN 公司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 公司授权了工具提供商(如 Macromedia)、结盟公司(如 Apache, Netscape)、最终用户及协作商等。最近,SUN 公司将最新版本的 JSP 和 Java™ Servlet(JSP 1.1, Java SERVLET 2.2)的源代码发放给 Apache,以求 JSP 与 Apache 紧密的相互发展。Apache、SUN 和许多其他公司及个人公开成立一个强大的咨询机构以便任何公司和个人都能免费取得信息(详见 <http://jakarta.apache.org>)。

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,其发展是建立在封闭的开发过程基础之上。

跨平台访问需要第三方 ASP 的引入产品 支持 Web 信息机构环境中不同系列的计算机群即保证用户在当前软硬件及人力资源上的投资完全兼容,JSP 技术提供灵活的开放选择:可以使用各种各样的工具提供商提供的工具,高度体现工业化标准输入与配置。

(3) 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的、动态内容的 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,所以网页制作者可以充分利用与 XML 兼容的标签技术的强大功能,大大减少了对脚本语言的依赖。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复杂程度。

JSP 跨平台的可重用性:JSP 的开发人员在开发过程中一直注重可重用性。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了简便的、与 XML 兼容的接口,即共享网页的打包功能使其完全的工业标准化。

这种基于组件的模式能够有效提高应用程序的开发效率,因为这种模式能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便可使用。像这样有效的方法在 JSP 中无处不在,并可将其打包成一个 JavaBean 或一个工业标准化的 JavaBean 组件。

● 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 是成熟的、强大的、易扩充的编程语言,远优于基于 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 Serv-



lets, 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他方面也变的一样简单。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。加之,JSP 为应用提供了强大的意外事件处理机制。而且 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。

(1) 易于维护性

基于 JSP 技术的应用程序比基于 ASP 的应用程序易于维护和管理。

脚本语言都能很好服务于小的应用程序,但不能适应大型的、复杂的应用程序。而 Java 是结构化的,它比较容易创建和维护庞大的组件化的应用程序。

JSP 突出的组件技术使修改内容不影响逻辑或修改逻辑不影响内容变得很容易实现。企业级的 JavaBeans 结构符合了企业逻辑,例如数据库的访问、数据库安全、事务完整性及独立性。

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,及其他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 JSP 能够适应现实世界的各种 Web 应用程序不断的变化和发展。

表 1-1 JSP 与 ASP 比较一

	ASP 技术	JSP 技术
可重用,跨平台组件	无	企业级 JavaBeans,定制 JSP 标签
安全防范系统崩溃	无	有
内存泄露保护	无	有
脚本语言	VBScript, JScript	Java
定制标签	无	有

(2) 企业产品的多样性

Java2 平台适用于多企业应用程序的 Java 结构,作为 J2EE 的部分,通过 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,所以它们同为服务器端 Java 应用程序。J2EE 平台内容不仅包括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。

JSP 网页可以访问标准的 J2EE 服务,包括:

- Java 名称和目录界面 API;
- JDBC API(与关联的数据库通信);
- JavaMailTM(支持基于 Java 邮件和消息应用程序的类);
- JavaTM 消息服务。

通过 J2EE, JSP 网页能够用许多方式同企业系统交互访问。J2EE 支持两种 CORBA 规范技术:Java IDL 和 RMI-IIOP。在企业级 JavaBeans 技术支持下, JSP 网页通过运用高级的,对象映射的方式访问数据库。