

潘国祥车舒编著

ASP.NET

动态网页设计

完全手册



内容全面 涉及该领域的方方面面知识，
提供完整的制作流程讲解。

方便实用 讲解该领域目前流行的专业技巧和相关制作案例，
可以给你带来有价值的学习或应用帮助。

案头常备 通过书中目录，可以快速找到你需要了解的知识点和参考说明；
随时翻阅，能不断提高理论结合实践应用的本领。



清华大学出版社

陆扬
潘国祥
车舒
编著

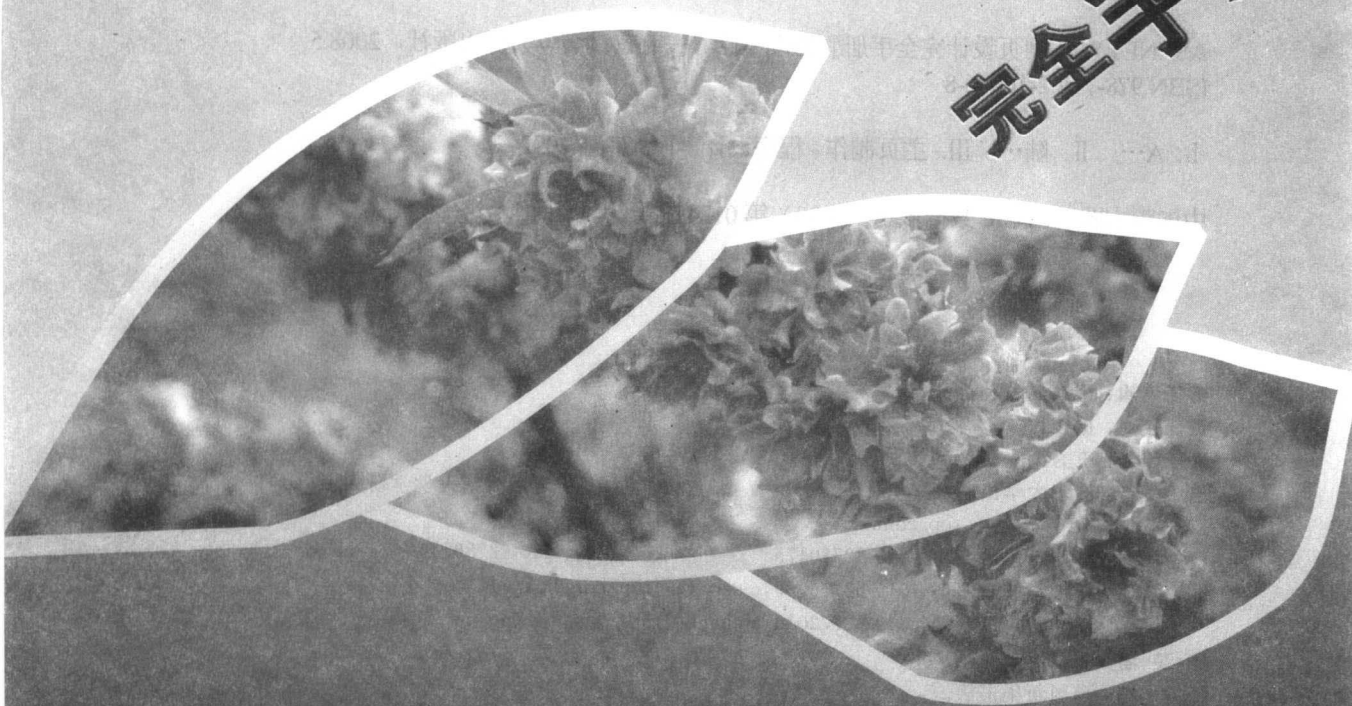
TP393.092/985

2008

ASP.NET

动态网页设计

完全手册



清华大学出版社
北京

内 容 简 介

本书循序渐进地介绍了 ASP.NET 的概念与主要应用技术, 以及动态网页开发的设计技巧和方法, 并且通过丰富、典型且实用的范例来讲解各个技术内容。主要包括 ASP.NET 2.0 基础知识简介、C#语言基础、HTML 控件和 Web 服务器控件、ASP.NET 的常用内置对象、ASP.NET 高级编程、ASP.NET 的其他 Web 控件、在 ASP.NET 中应用 XML、ASP.NET 数据库访问——ADO.NET、ASP.NET 的安全性和 ASP.NET 的配置和优化等。

本书语言简约、实例丰富, 可以帮助读者充分掌握开发一个完整的 Web 站点所需要的各方面技术, 从而设计出功能实用、界面优美的大型网站。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET 动态网页设计完全手册/陆扬等编著. —北京: 清华大学出版社, 2008.5
ISBN 978-7-302-16946-8

I. A… II. 陆… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2008) 第 012411 号

责任编辑: 田在儒 林都嘉

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 26.75 字 数: 629 千字

版 次: 2008 年 5 月第 1 版 印 次: 2008 年 5 月第 1 次印刷

印 数: 1~4000

定 价: 49.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 019039-01

导读

读者对象

本书适用于以下读者：

- 想学习和迅速掌握 ASP.NET 2.0 的读者；
- 想迅速掌握一门网络编程语言的读者；
- 其他有相关知识需求的读者。

写作环境

本书在写作时基于 Windows XP 操作系统平台，建议读者采用相同的系统环境。如果读者在 Windows 98/2000/2003 操作系统下，或者使用比本书所用软件版本低的版本，则请参考相关软件的讲解。

学习提示

ASP.NET 2.0 是 Microsoft 公司动态服务器页面(简称 ASP)的最新版本，它是一种基于服务器的、功能强大的技术，用于为万维网站或企业的内部网创建动态的、交互式的 HTML 页。ASP.NET 可分为许多其他技术，如 Web 服务、ADO.NET、定制控件和安全性等。本书配合大量示例代码，全面系统地讲述了它们之间的关系，使读者对如何以可行的方式从 ASP.NET 使用中获益有比较深刻和全面的理解。

全书共分为 10 章，分别为 ASP.NET 2.0 基础知识简介、C#语言基础、HTML 控件和 Web 服务器控件、ASP.NET 的常用内置对象、ASP.NET 高级编程、ASP.NET 的其他 Web 控件、在 ASP.NET 中应用 XML、ASP.NET 数据库访问——ADO.NET、ASP.NET 的安全性和 ASP.NET 的配置和优化等。

在阅读完本书后，读者会很熟悉 ASP.NET 的构成，并能创建出灵活、安全和健壮的 Web 站点，以各种方式收集和处理信息，使自己和用户从中受益。本书适合于没有 ASP、C#、XML、面向对象编程或 .NET Framework 经验的初学者。如果读者对 HTML 有一定的了解，那么将对学习本书大有裨益。

本书作者

参与本书写作的作者除封面署名外，还有田高阳、步文海、杨明、逯阳、印霞、张婧晶等。

技术支持

如果在阅读本书的过程中有什么困难，可以登录到“黑魔方”专题网站，网址是 <http://www.heimofang.com>。这是大家共同交流的平台，在那里会有很多的作者、老师、读者、编辑一起交流，在相关的栏目中发求助帖子，问题会很快得到解答。除上述方法外，也可以使用下面的方式寻求技术支持（但可能会慢一些）。

- 发电子邮件到 tianzr@tup.tsinghua.edu.cn
- 打电话给 010-62773950 或发传真给 010-62771155。

目录

第1章 ASP.NET 概述

2	1.1	Web 基础知识
2	1.1.1	HTTP 协议
3	1.1.2	Web 服务器
4	1.1.3	静态网页
5	1.1.4	动态网页
6	1.2	ASP.NET 简介
6	1.2.1	基本概念
9	1.2.2	ASP.NET 的发展历史
10	1.2.3	ASP.NET 与 ASP 的区别
11	1.2.4	ASP.NET 的工作原理
12	1.3	建立 ASP.NET 的运行环境
12	1.3.1	IIS
13	1.3.2	MDAC
14	1.3.3	.Net Framework 和 ASP.NET
16	1.4	开始编写 ASP.NET 程序
16	1.4.1	开发环境的选择
16	1.4.2	Visual Studio.NET

第2章 C#语言基础

21	2.1	编程语言的选择
21	2.1.1	ASP.NET 支持的语言
21	2.1.2	为什么选择 C#
22	2.2	变量、常量和表达式
22	2.2.1	变量
26	2.2.2	数据类型
33	2.2.3	运算符
45	2.2.4	常量
45	2.2.5	表达式
47	2.2.6	数据类型之间的转换
53	2.3	构造类型
53	2.3.1	数组
55	2.3.2	结构
55	2.3.3	枚举
56	2.4	控制结构

56	2.4.1	顺序
57	2.4.2	分支
59	2.4.3	循环
62	2.5	函数
63	2.5.1	函数的定义
63	2.5.2	函数的调用
64	2.5.3	参数的传递
64	2.5.4	变量的作用域

第 3 章 HTML 控件和 Web 服务器控件

66	3.1	HTML 控件
68	3.1.1	Web 表单
73	3.1.2	Anchor 控件
74	3.1.3	Button 控件
76	3.1.4	InputButton 控件
77	3.1.5	InputCheckBox 控件
78	3.1.6	InputRadioButton 控件
80	3.1.7	InputImage 控件
81	3.1.8	InputFile 控件
82	3.2	Web 服务器控件
87	3.2.1	用于文本输入和显示的内部控件
93	3.2.2	用于控制传送的内部控件
100	3.2.3	用于选择的列表控件

第 4 章 ASP.NET 的常用内置对象

118	4.1	Response 对象
119	4.1.1	输出字符串到网页上
120	4.1.2	重定向
121	4.1.3	缓存 HTML
122	4.1.4	输出文本文件的内容
123	4.2	Request 对象
125	4.2.1	get 方法
127	4.2.2	post 方法
128	4.2.3	使用环境变量
129	4.2.4	获取浏览器信息
130	4.3	Application 对象
130	4.3.1	使用 Application 对象
136	4.3.2	Contents 集合和 StaticObjects 集合
136	4.3.3	Lock 和 UnLock 方法的使用

137		4.3.4 Application 事件
138	4.4	Session 对象
138		4.4.1 Session 简介
139		4.4.2 Session 对象的属性
139		4.4.3 Session 对象的方法
141		4.4.4 Session 对象的事件
142		4.4.5 使用 Session 对象的注意事项
143	4.5	Server 对象
143		4.5.1 Server 对象的属性
143		4.5.2 Server 对象的方法
145	4.6	Cookie
145		4.6.1 什么是 Cookie
147		4.6.2 设置 Cookie
151		4.6.3 检索 Cookie
152		4.6.4 检测用户是否使用了 Cookie

第 5 章 ASP.NET 高级编程

155	5.1	命名空间
160	5.2	类库
160		5.2.1 类库和基库
161		5.2.2 集合
162		5.2.3 ArrayList
164		5.2.4 HashTable
165		5.2.5 SortedList
166		5.2.6 目录和文件的处理
171	5.3	Page 类
172		5.3.1 页面状态
173		5.3.2 页面指令
175	5.4	事件驱动
175		5.4.1 什么是事件驱动
176		5.4.2 ASP.NET 中的事件
180		5.4.3 ASP.NET 中的服务器控件事件

第 6 章 ASP.NET 的其他 Web 控件

182	6.1	Web 增强控件
182		6.1.1 日历控件
189		6.1.2 广告控件
191		6.1.3 验证控件
201	6.2	自定义控件

201	6.2.1	代码分离技术
204	6.2.2	编写自定义控件
205	6.2.3	用户控件

第 7 章 ASP.NET 数据库访问——ADO.NET

211	7.1	概述
211	7.1.1	ADO.NET 和 ADO
213	7.1.2	ADO.NET 特性
215	7.2	Connection 对象
215	7.2.1	Connection 对象概述
216	7.2.2	Connection 对象实例
218	7.3	Command 对象
218	7.3.1	Command 对象概述
218	7.3.2	Command 对象实例
224	7.4	DataReader 对象
224	7.4.1	DataReader 对象概述
224	7.4.2	DataReader 对象实例
230	7.5	DataAdapter 对象
230	7.5.1	DataAdapter 对象概述
230	7.5.2	DataAdapter 对象命令
233	7.6	DataSet 对象
233	7.6.1	DataSet 对象概述
235	7.6.2	DataSet 对象实例

第 8 章 在 ASP.NET 中应用 XML

247	8.1	什么是 XML
251	8.2	标记、元素以及元素的属性
252	8.3	创建 XML 文档
255	8.4	使用样式表显示 XML
257	8.5	在 ASP.NET 中使用 XML
257	8.5.1	写入 XML 数据
259	8.5.2	读取 XML 数据
260	8.5.3	编辑 XML 数据
262	8.5.4	将 XML 转化为字符串

第 9 章 ASP.NET 的配置和优化

264	9.1	ASP.NET 的配置
264	9.1.1	machine.config 文件和 web.config 文件

270		9.1.2 global.asax 文件
271		9.1.3 创建 Application 事件代码
274	9.2	ASP.NET 的优化
274		9.2.1 使用缓存
278		9.2.2 跟踪
281		9.2.3 监视 ASP.NET 进程

第 10 章 ASP.NET 的安全性

284	10.1	什么是安全性
285	10.2	身份验证和授权
285		10.2.1 Windows 提供的身份验证和授权
286		10.2.2 IIS 提供的身份验证和授权
289		10.2.3 ASP.NET 提供的身份验证和授权
299	10.3	SSL 加密方法简介

附录 1 文章发布系统的设计

307	附 1.1	功能说明
308	附 1.2	功能模块

附录 2 文章管理系统

357	附 2.1	系统结构
357	附 2.2	文件代码

附录 3 留言本

398	附 3.1	系统结构
398	附 3.2	文件代码

附录 4 邮件列表

412	附 4.1	邮件列表作用和工作方式
412	附 4.2	邮件列表的设计

第 1 章

ASP.NET 概述

对于现代人来说,Internet 已经成为生活、学习和工作中不可缺少的一部分。Internet 是跨平台的,它不需要用户在自己的计算机上装载任何其他软件,只要有一个浏览器就可以浏览到各种信息。随着网络接入技术的不断发展,与 Internet 相连也变得越来越简单、越来越方便,以至于很多传统的行业现在也开始越来越依赖于网络。在这种情况下,很多单位和个人都建立了自己的网站,但是如果只使用 HTML 就只能保证网页的美观,却不能引入更多更强大的功能,于是采用动态网页设计就成为现在网站设计的主流,ASP 就是在这种情况下诞生的。

随着时间的推移,ASP 一方面为网站的设计者带来了便利,另一方面也使得网站的各种代码难于管理,这对于程序员来说,面对的将是大量的 HTML 代码和 VBScript、JavaScript 代码混合在一起的程序,这些程序代码的模块化和可重用性都很低。另外,由于 VBScript 这样的脚本语言的局限,使得很多功能都不能轻松实现,即使实现了也需要写大量的代码,为了解决这些问题,ASP.NET 就随之产生了。

微软作为全球软件业最大的公司,在 .NET 技术上投入了大量的人力物力,并且把公司未来战略重心放在了 .NET 上。从目前看来, .NET 技术无疑代表了未来 Internet 技术的发展方向。ASP.NET 是微软公司推出的网络编程开发框架,它是 Microsoft.NET 系统中的重要成员,利用 ASP.NET,程序员可以很方便快速地开发功能强大的网络应用程序。本章将对 ASP.NET 及其 Web 相关内容进行简要的介绍。

1.1 Web 基础知识

WWW (World Wide Web) 是一个大规模、在线式的信息储藏所,用户可以通过一个被称为浏览器 (Browser) 的交互式应用程序来查看。从技术上讲,WWW 是一个支持交互式访问的分布式超媒体 (HyperMedia) 系统,而所谓的超媒体系统则直接扩充了传统的超文本 (HyperText) 系统。在 Web 上可获得的超媒体文档被称为网页。由于网页可以包含许多内容,因此,每一个包含超媒体文档的网页都采用一个标准的表示方式,以便浏览器能够解释网页的内容。该标准被称为超文本标记语言 (HyperText Markup Language, HTML), 它是一个标记语言,并不包含详细的格式指令,在语句构成上,每个 HTML 文档都以一个包含标签和其他信息的文本文件来表示。

1.1.1 HTTP 协议

HTTP 协议即超文本传输协议 (Hypertext Transfer Protocol), 这个协议是在 Internet 中进行信息传送的协议,浏览器默认使用这个协议。当用户在浏览器的地址栏中输入网址 (如 www.mywebsite.com) 的时候,浏览器就会自动使用 HTTP 协议来搜索 <http://www.mywebsite.com> 网站的首页。

从浏览器向 Web 服务器发出的搜索某个 Web 网页的请求是 HTTP 请求,当 Web 服务器收到这个请求之后,就会按照请求的要求,找到相应的网页,如果可以找到这个网页,那么就把网页的 HTML 代码通过网络传回浏览器,如果没有找到这个网页,就发送一个错误信息给发出 HTTP 请求的浏览器,后面的这些操作称为 HTTP 响应。



HTTP 是一个无状态协议，也就是说，当使用这个协议的时候，所有的请求都是为搜索某一个特定的 Web 网页而发出的。它不知道现在的请求是第一次发出还是已经多次发出，也不知道这个请求的发送来源，当用户请求一个 Web 网页的时候，浏览器会与相关的 Web 服务器相连接，检索到这个页面之后，就会把这个连接断开。

从程序设计的角度来看，无状态的特点对于 HTTP 来说是一个缺点，因为这使得某些功能很难实现，但是由于网络本身的特点，这也是没有办法改变的。可以假设一下，如果 HTTP 协议是有状态的协议，那么就应该让一个连接长时间地存在下去，这样就可以判断一个用户到底使用了多长时间，在这段时间内都做了些什么事情。这样在 Internet 环境中，一个 Web 服务器要保存太多的连接（因为在 Internet 环境中，用户的数量是很难估计的），会导致服务器瘫痪，正因为如此，对于所有的 HTTP 请求，Web 服务器都会以同样的方式来对待。

同其他网络应用一样，Web 浏览采用的也是客户端/服务器（Client/Server）模式。当给定一个链接的资源定位器（Uniform Resource Locator, URL）时，浏览器就成为一个客户，与 URL 指定地址的服务器进行通信来请求网页文档，然后浏览器向该用户显示网页。与其他网络应用连接不同的是，Web 浏览器和服务器之间的连接只维持一小段时间，即浏览器建立连接、发送请求，然后接收请求的内容或者得到一个该内容不存在的信息，一旦文档或图像已被传输完毕，该连接则被关闭。

HTTP（HyperText Transport Protocol, HTTP）是 Web 操作的基础，它是一个使信息能通过 Web 交换的客户端/服务器协议。HTTP 定义了浏览器能提出的请求的类型以及服务器返回的响应类型，通过 HTTP 用户可以从远程服务器中获取网页，而且如果用户有特定权限，还可以将网页文档存储在服务器上，HTTP 还提供向网页上添加新信息或删除信息的能力。

1.1.2 Web 服务器

当提到 Web 服务器的时候，很多人都会认为这是一台物理的计算机，但实际上，Web 服务器是一种软件，可以管理各种 Web 文件，并为提出 HTTP 请求的浏览器提供 HTTP 响应。大多数情况下，Web 服务器和浏览器处于不同的计算机，但是它们也可以并存在同一计算机上。

如图 1-1 所示，当 HTTP 客户端向远程 Web 服务器发送一个访问网页的请求时，服务器将网页传输给客户端作为响应，最终在客户端的屏幕上显示网页内容。用户还可以在客户端从这个新网页转向某一链接而调用另一个网页，另一个远程 Web 服务器以相同的方式响应，并将所请求的网页送回客户端浏览器。

从上述的 Web 服务过程来看，网页脚本是在 Web 服务器上运行的，而不是在客户端运行，传送到客户端浏览器上的 Web 页是在服务器上生成的。所以不必担心客户端浏览器是否能够处理脚本，因为 Web 服务器已经完成了所有的脚本处理，并将标准 HTML 传输到客户端浏览器上。由于只有脚本的结果返回到浏览器，因此，服务器端的原脚本文件不易被复制，用户也看不到创建他们正在浏览的 Web 页的脚本命令。

比较常见的 Web 服务器有 Apache 和 IIS，由于 ASP.NET 只能在 IIS 上运行，因此本书把介绍的重点放在 IIS 上。IIS 是微软公司的操作系统 Windows 2000/XP 所提供的，



在后面的章节将会进行详细介绍。

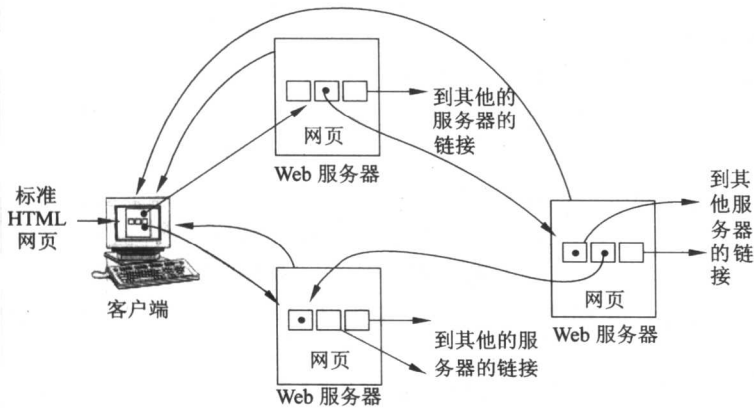


图 1-1

1.1.3 静态网页

在动态网页出现之前，所有的网页都是静态的，静态网页就是用纯 HTML 代码编写的网页，这些网页的代码是用一些编辑器输入的，或者是用一些网页设计程序生成的，保存为 .html 或 .htm 文件的形式。由于这些网页中没有任何与用户相关的部分，因此，在设计完成之后，无论是哪个用户、在什么时候、以何种方式访问这个页面，它的样子都不会发生任何变化。

静态 Web 文档是一个存储在 Web 服务器上的标准 HTML 脚本文件，静态网页在 Web 页面设计时就确定了网页的内容。由于文档内容不会变化，因此，对静态 Web 文档来说，每次访问都会返回相同的结果。

静态 Web 文档的主要优点在于它的简单、可靠性和性能。由于静态 Web 文档是直接指定的格式，因此它可以由不懂编程的人创建，更重要的是，在已经创建和测试完成之后，静态 Web 永远是正确的。最后浏览器可以快速地存取文档，同时通过把文档放在本地缓冲区内来加快以后对这些文档的访问速度。静态 Web 文档的主要缺点则是不够灵活，当文档信息需要改变时就必须重新设计，而且这种改变是很耗费时间的，因为这种改变主要靠人工修改 Web 服务器上的文件，所以静态 Web 文档不适合频繁变化的报告信息。

下面是一个静态网页的例子，用于显示一个红色的“Hello World.”字符串。

```
<html>
<body>
  <font color=red> Hello World.<br></font>
</body>
</html>
```

这只是一个简单的 HTML 静态网页，结果是在浏览器中用红色显示“Hello World.”字符串，只要这个文件存在，无论什么用户、在什么时候、以何种方式访问这个页面，



都会显示同样的结果。

使用静态网页也存在很大的局限性,如果希望为用户显示一些个性化的信息,使用静态网页就无法实现了。例如,如果当前的时间是新年的开始,就可以在网页上显示一个“新年好!”的信息;如果当前时间是圣诞节,就可以在同样的位置显示一个“Merry Christmas!”信息。除此之外,静态网页无法防止用户复制 HTML 代码,因为每个用户都可以采用浏览器的“查看源文件”命令来查看这个网页的 HTML 代码。

1.1.4 动态网页

动态网页可以为不同的用户提供个性化的服务,从前面关于静态网页的介绍中可以看出,使用这种网页的优点是不言而喻的。而为了实现这种动态性,就需要进行程序设计,随着技术的不断发展,在动态网页的实现过程中,一般采用客户端编程和服务器端编程两种程序设计方法。

客户端编程是采用下载到浏览器上的程序来完成所有有关动态服务的工作的,通常的情况是程序员把客户端代码编写到 HTML 文件中,当用户提出对这个网页的请求时,这些客户端代码(即可以实现动态内容的程序)和 HTML 文件的代码一起以响应的方式返回给提出请求的浏览器,又因为所有的代码(包括程序和 HTML 标记等)都被浏览器接收,所以这些程序的执行是由浏览器来实现的。常见的客户端编程技术有 JavaScript、VBScript 和 Java applet 等。

在动态网页刚刚出现的时候,多数是使用客户端编程的方法来实现网页的动态服务,因为这样做可以减少服务器的负担,充分利用客户端计算机的资源,但是现在客户端编程技术已经越来越不受欢迎。首先,由于所有的代码都要下载到客户端来执行,因此相对而言下载的时间就会增加,尤其当程序的代码量很大的时候,下载时间的延长会十分明显;其次,由于所有的客户端代码都是由浏览器来执行的,因此在编程的时候,需要针对不同的浏览器进行测试,以保证代码的正确执行。这是因为现在流行的浏览器很多,一个程序能在 IE 上正确执行,但是在 Netscape Navigator 或者 Opera 上就不一定可以正确执行,这为代码的快速编写设置了很大的障碍;再次,如果需要使用服务器端的资源(如数据库中的记录),那么采用客户端编程就无法实现;最后,采用客户端编程无法保证代码的安全性,因为所有可以访问到这个页面的用户都可以采用浏览器的“查看源代码”命令来查看网页的所有代码(包括 HTML 和客户端程序)。

由于客户端编程有这些缺点,而现在的服务器硬件的飞速发展,相应可以使用的资源也就越来越多,因此客户端编程可以节省服务器资源的优势已经大大丧失。所以服务器端编程已经成为动态网页编程的主流,服务器端编程的原理是:程序员编写的代码被保存在服务器上,当用户对某个动态网页提出 HTTP 请求的时候,这个请求所要访问网页的代码都在服务器端执行完成,并把执行结果以 HTML 的形式传回浏览器。这样由于浏览器接收到的只是程序执行结果,因此上面提到的所有问题都迎刃而解。常见的服务器端编程技术有:CGI、PHP、ASP、JSP 和 ASP.NET。

相对于静态 Web 文档来说,动态 Web 文档不是以一个预先定义的格式存在,而是在客户端访问 Web 服务器时由服务器解释创建的,服务器返回程序的解释结果作为应答。由于每次访问都要创建新的文档,因此动态 Web 文档的内容是变化的,其主要优



点在于它具有报告当前信息的能力。例如，一个动态 Web 文档可以用来报告股市行情、新闻更新或者售票情况等。当客户端申请信息时，Web 服务器运行一个应用程序，访问所需要的信息，并创建一个文档，服务器最终将这个文档的 HTML 脚本返回给客户端浏览器。

动态 Web 文档把任务放在服务器一端，客户端浏览器采用和静态 Web 同样的方法访问动态 Web，实际上，从浏览器的角度来看，动态 Web 和静态 Web 是没有区别的，因为浏览器看到的都是标准的 HTML 脚本，它并不知道 Web 服务器是从一个磁盘文件还是从一个计算机程序中取得文档的。

动态 Web 文档的代价要比静态 Web 文档高，因为动态 Web 的创建者必须要懂得如何写程序，此外还必须仔细编写和广泛测试，以保证输出的合法性。

1.2 ASP.NET 简介

ASP 是一种功能强大而且易于学习的服务器端的脚本编程环境，它是 Microsoft 从 NT Server 操作系统开始附带的脚本编程环境，并且在 NT Workstation、Windows 98 和 Windows 2000/XP/2003 中也都附带这个脚本编程环境。在 2001 年，微软在前面 3 个版本的 ASP 基础上，推出了全新的 ASP.NET，它开始抛弃前面 3 个版本都使用的脚本语言，而是选择 Visual Basic.NET 作为它的默认语言，但是，无论如何变化，使用这种环境，都可以方便地创建动态、快速、交互性强的 Web 站点。

ASP.NET 技术就是由微软的 .NET 技术细化来的，ASP.NET 又叫 ASP+，它并不是 ASP 的简单升级，而是 Microsoft 推出的新一代 Active Server Pages 脚本语言。ASP.NET 是微软发展的新型体系结构 .NET 的一部分，它的全新技术架构会让每一个人的网络生活都将变得更简单。

ASP.NET 是完全基于模块与组件的，具有更好的可扩展性与可定制性，数据处理方面更是引入了许多激动人心的新技术，正是这些具有革新意义的新特征，才让 ASP.NET 远远超过了 ASP，同时也提供给 Web 开发人员更好的灵活性，有效缩短了 Web 应用程序的开发周期。ASP.NET 与 Windows 2000 Server/Advanced Server 的完美结合，为中小型乃至企业级的 Web 商业模型提供了一个更为稳定、高效、安全的运行环境。

1.2.1 基本概念

1. Client和Server

在了解 ASP.NET 之前，先来了解一下客户端 (Client) 和服务器 (Server) 间的关系。在计算机的世界里，凡是提供服务的一方就称为服务器端 (Server)，而接受服务的另一方称为客户端 (Client)。例如人们最常接触的局域网里的打印服务器提供打印服务的例子，在这里把提供打印服务的计算机叫打印服务器，而使用打印服务器提供打印服务的另一方，则称为客户端。但是谁是客户端、谁是服务器端也不是绝对的，如果原来提供服务的服务器端要使用其他计算机提供的服务，那么它就转变为客户端了。



在网络上用户在浏览器中输入网址, 通过 HTTP 协议向 Web 服务器提出浏览网页的请求 (Request), 服务器收到用户的请求后, 立即把用户所要求浏览的网页数据传输给客户端, 这个动作称为响应 (Response)。

2. 请求和响应

在了解请求和响应的过程之前, 先来了解一下程序的编译和解释有什么不同。

由于计算机能够执行的只是二进制的代码, 因此所有用助记符来编写的程序都需要使用一个程序把程序语句翻译成计算机可以执行的二进制代码。如果这个翻译的过程是在程序执行之前预先进行的, 那么就叫编译; 如果这个翻译的过程是在程序的执行过程中进行的, 那么这个过程就是解释。由于编译是在程序执行之前进行的, 因此可以对代码进行优化来保证编译的结果可以最好地利用计算机硬件的各种性能。而解释是在程序的执行过程中进行的, 所以没有办法对程序进行相关的优化。

早期的 ASP 是 IIS 的一种开放式的无需进行编译的应用程序环境, 也就是说, ASP 程序是解释执行的。IIS 是服务器上安装的 Internet 信息服务器 (Internet Information Server), 它是 Microsoft 公司开发的一个网络文件和应用程序服务器 (即 Web 服务器), 这个服务器包含在操作系统中, 在 Windows 2000 中, 它的版本是 5.0。IIS 支持 HTTP、FTP 和 Gopher 协议, 由于 ASP 是服务器端的脚本编程环境, 而所有的程序都是解释执行的, 因此, 这意味着这个环境中的所有程序在每次被访问的时候都需要 IIS 进行一次解释, 而从客户端会得到一个执行结果。

在 ASP.NET 中, 所有的程序执行都是经过服务器编译的, 在这一点上, ASP.NET 与早期的 ASP 版本有很大的不同, 因此在程序执行的效率上也有很大的提高。具体的方法是: 在 ASP.NET 中所有的程序仍然是保存在服务器端的, 当一个程序第一次被执行的时候进行编译, 所以当这个程序被再次执行的时候会直接在服务器上执行它的已经编译好的可执行二进制代码, 然后把执行结果通过网络返回给客户端。因此, 与 ASP 相比, ASP.NET 程序的执行速度会快很多。

ASP 请求的处理过程如图 1-2 所示, 需要注意的是, 由于 ASP.NET 对 ASP 请求的处理过程与早期版本的 ASP 的处理有所不同, 因此在服务器端并没有详细表示出请求的处理过程。

具体的处理过程是这样的: 在客户端有一个用于浏览网页的浏览器, 用户在这个浏览器中输入 HTTP 请求, HTTP 请求通过 Internet 找到相应的 Web 服务器, 并把这个请求传给这个服务器相应的处理模块。由这个处理模块负责找到相应的 ASP 程序或 ASP.NET 程序, 进行相应的执行 (对于 ASP 程序来说, 是通过一个名为 asp.dll 的 ISAPI DLL 进行程序的解释; 而对于 ASP.NET 程序来说, 则是通过 aspnet_isapi.dll 进行处理的), 执行结果通过 Internet 返回给客户端, 形成 HTTP 响应。

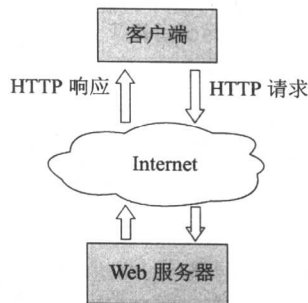


图 1-2

不论 ASP 文件还是 ASP.NET 文件都是一个可以用任何文本编辑器编辑的纯文本文件, 只要将这个文件的扩展名保存为 .asp 或者 .aspx 就可以了。现在也有很多的开发工具可以进行 ASP.NET 程序的开发, 例如 Microsoft 的 Visual InterDev 等开发工具。不

过也有很多程序员就选择使用记事本开发 ASP 程序或 ASP.NET 程序,当然,对于 ASP.NET 来说,还可以使用最新的 Visual Studio.NET 进行各种控件的开发。

3. CGI和ISAPI

几乎所有的 Web 服务器都支持 CGI,在 UNIX 计算机上,大部分的 CGI 程序都是 Perl 语言和 C 语言编写的,在 IIS 上,用 CGI 程序可以调整操作系统所提供的所有功能,并使用 ODBC 建立数据库连接来使用数据库功能。

但是,从开发人员的角度来看,这些应用程序在维护和调试的时候都十分困难,并且从 Web 服务器的性能上来看,使用 CGI 还有一个更大的缺点,就是对于每一个客户的请求 CGI 都要产生一个进程来进行处理,这样当服务器的访问量很大的时候,系统内部用于处理用户响应的进程就十分多,这么多的进程会让服务器的资源很快地消耗掉,很容易导致计算机的崩溃。由此可以看出,使用 CGI 的优点是可以创建功能强大的应用程序;缺点是难以维护和调试,系统资源消耗大。

使用 ISAPI 的优点是可以避免像 CGI 程序那样,对每一个客户请求都产生一个进程,所以它成为一种既可以产生功能强大的应用程序,又能减少性能损失的一个程序设计环境。ISAPI 应用程序是通过 DLL 实现的,并且可以加载到服务器的进程空间,以保证程序执行得更快,用 ISAPI 可以创建两种不同的应用程序,即过滤程序和扩展程序。

过滤程序对于客户来说是透明的,它用于监视请求、自定义身份验证方案、随机数据转换以及更多的事情,在服务器打开的时候,过滤程序加载到 Web 的进程空间,并一直驻留在内存中,直到服务器关闭为止。扩展程序与 CGI 应用程序的工作类似,ISAPI 扩展程序用于处理表单、从数据库中检索数据、执行业务逻辑,与过滤程序一样的是,扩展程序也是加载到 Web 服务器的进程空间,但是只在第一个用户从扩展程序请求服务时才发生,而不是在服务器启动时发生,扩展程序也可以用来获得操作系统级别的访问。

由于 ISAPI 的 DLL 时被加载到 Web 服务器的进程空间的,因此这种方法的最大缺点是,如果编写的 ISAPI 应用程序对性能的考虑不是很好,就会导致服务器的崩溃,此外它仍然要使用 C 语言这样复杂的语言来进行编写,这对程序员来说,调试和维护也不是很容易的事。

4. Microsoft.NET

2000 年 6 月 22 日,不论对 Microsoft 还是对整个 IT 业界来说都将成为值得纪念的一天,因为在这一天微软正式推出了其下一代计算计划——Microsoft.NET。这项计划将使微软现在的软件在 Web 时代不仅适用于传统的 PC,而且也能满足目前呈强劲增长势头的新设备,如蜂窝电话以及 PDA (Personal Digital Assistant) 等的需要。微软还计划通过创建新的工具来吸引软件开发人员和合作伙伴对 Microsoft.NET 的认同,并且开发出其他基于 Internet 的服务。

那么,究竟什么是.NET 呢?请听来自微软的声音:“……因特网的革命……从微软的角度来讲,我们就是要建设一个平台来创建并支持新一代的应用。……我们必须有一套通用系统服务来支持这样的操作。这种观点就说明,我们还有下一个层次的发展,也就是说因特网下一步的发展,它将使因特网的作用远远超越展现一个网站”。

Microsoft.NET 首先是一个开发平台,它包括用于创建和操作新一代服务的.NET 基

