



计 算 机 类 本 科 规 划 教 材

ASP.NET 数据库 网站设计教程 (C#版)

◆ 刘瑞新 主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

计算机类本科规划教材

ASP.NET 数据库网站设计教程 (C#版)

刘瑞新 主编

朱 立 张治斌 刘桂玲 等编著



电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

微软公司的 ASP.NET+C#组合是网站开发采用的主流技术之一。本书以实际应用为目的,全面系统地介绍了开发 ASP.NET 数据库网站的方法及知识,包括 ASP.NET 基础,ASP.NET 服务器标准控件和验证控件,ASP.NET 常用对象、状态管理,SQL Server 数据库基础,使用 .NET 数据提供程序访问数据库,使用 DataSet 访问数据库,数据绑定与数据绑定控件,站点导航和母版页,新闻网站的设计,用 ASP.NET MVC 架构开发网站等内容。每章均有典型的演练和实训,以提供教师演示和学生练习。本书在 Visual Studio 2010 和 SQL Server 2008 环境下讲解,也完全可以运行在 Visual Studio 2005/2008 和 SQL Server 2005 环境下。本书概念清晰、重点突出、实例丰富,符合教师教学和学生学习习惯,是一本非常适合课堂教学的、用 Visual C#语言开发 Web 结构的数据库网站设计教材。

为了方便教师授课及读者的学习,本书提供了电子教案、源代码等,登录华信教育资源网(www.hxedu.com.cn)注册后免费下载。

本书可作为高等学校计算机类相关专业教材,同样适合作为高职高专院校计算机类相关专业的教材,也可作为网站开发人员的技术参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ASP.NET 数据库网站设计教程: C#版 / 刘瑞新主编. —北京: 电子工业出版社, 2015.1

计算机类本科规划教材

ISBN 978-7-121-24671-5

I. ①A… II. ①刘… III. ①网页制作工具—程序设计—高等学校—教材②C 语言—程序设计—高等学校—教材 IV. ①TP393.092②TP312

中国版本图书馆 CIP 数据核字(2014)第 254656 号

责任编辑: 冉 哲

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 22 字数: 592 千字

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

印 数: 3 000 册 定价: 45.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

微软公司的 ASP.NET+C#组合是网站开发采用的主流技术之一。本书以 Visual Studio 2010 和 SQL Server 2008 为运行环境,以 Visual C#为开发语言,比较完整地介绍了开发 ASP.NET 数据库网站所需要的内容和相关知识,主要内容包括 ASP.NET 基础,ASP.NET 服务器标准控件和验证控件,ASP.NET 常用对象、状态管理,SQL Server 数据库基础,使用 .NET 数据提供程序访问数据库,使用 DataSet 访问数据库,数据绑定与数据绑定控件,站点导航和母版页,新闻网站的设计,用 ASP.NET MVC 架构开发网站等内容。每章均有典型的演练和实训,以提供教师演示和学生练习。考虑到有些学校计算机配置较低,除最后一章外,其他章节内容完全可以运行在 Visual Studio 2005 和 SQL Server 2005 环境下。

在学习本教材前,要求学生具有静态网站的基础知识和程序设计语言基础。本书特别增加了一章介绍 SQL Server 数据库基础,介绍了学习本书所必需的数据库方面的知识。

在网站设计技术和技巧方面,本书大量采用微软和业界推荐与采用的方法,使得本书介绍的方法更贴近实际应用。值得一提的是,本书中所有变量的命名,均采用业界提倡采用的 Pascal、camel 命名法;另外,本书中的许多源代码来自富有经验的程序员,或经过简化而成,阅读这样的代码,有利于养成良好的代码编程风格。本书在编写风格上,力求深入浅出,尽量将知识融于浅显的案例之中,争取读者跨越最少的阻碍掌握知识。

在教学中我们发现,学生在设计 ASP.NET 网站时,往往要花费近乎一半的时间来设计静态网页,而无法集中精力练习本课程的 ASP.NET 技术。为此,我们在本书中首创把公司常用的设计方法引入到教材中,即前端设计师设计静态网页,后端设计师把静态网页改成动态网站。我们特意在本书配套下载教学包中提供了一套比较完整的用 CSS 设计的静态新闻网站,相当于前端设计师完成的网页,读者只需将静态网页中的静态元素替换成服务器控件,并编写相应的事件程序即可。这样读者就可以专注于学习和练习 ASP.NET 技术,一方面节省了大量设计静态网页的时间,又体现了公司采用的分工协作的工作方式。

本书在知识内容的细节介绍上,采用了符合认知规律的形式,即先引出概念,再介绍语法规则,然后介绍方法步骤,最后给出应用实例。之所以采用这种方式介绍知识,是因为 ASP.NET 及 C#都是人工语言,我们必须按照业界及微软所采用的形式和方法、步骤来设计教材,因为在 MSDN 和相关手册中都采用这种编写形式。我们必须适应这种学习形式,只有掌握了这种形式,才能很好地从 MSDN 等帮助中取得需要的知识和方法。也就是说,我们必须按照业界和 MSDN 提供的语法规则来“套用”,这种“套用”的方法,是学生必须掌握的。相反,有些所谓的基于工作过程或项目驱动的教材,只给出一段程序,省略了最重要的语法解析,读者只能看懂这段程序,而不知道这段程序为何要这样编写,变换一项要求更是不会编写,作者认为,这类教材舍本而求末,违反了认知规律。

本书的另一个特点是合理取舍,因为受到课时的限制,课堂没有过多的时间讲授全部内容,本书选取 ASP.NET 中应用最多的知识来介绍,舍去很少使用的内容(例如,在工程中很少用到的数据源控件,本书略去不再介绍)。我们在教学中知道,ASP.NET 技术的重点是 ADO.NET 数据访问技术,而难点在数据绑定与数据绑定控件,所以本书加大了这两部分的篇幅。对于 ASP.NET 其他技术,按照本书的思路和方法,通过查询 MSDN 等帮助,即便没有介绍的内容,

也可以很快掌握。

本书的主要作者是具有丰富教学经验的教师与经验丰富的企业程序开发工程师，优势互补保障了教材的质量，使得教材更贴近实际，是校企结合的结晶和范例。

教学课时安排可参考下表：

序 号	学 习 任 务	教 学 方 法	参考学时（包括讲授、练习）
1	第1章 ASP.NET 基础	讲授、演练	2~4
2	第2章 ASP.NET 常用服务器标准控件	讲授、演练	10~12
3	第3章 ASP.NET 验证控件	讲授、演练	4
4	第4章 ASP.NET 常用内置对象	讲授、演练	2
5	第5章 ASP.NET 的状态管理	讲授、演练	2
6	第6章 SQL Server 数据库基础	讲授、演练	2
7	第7章 使用.NET 数据提供程序访问数据库	讲授、演练	8~10
8	第8章 使用 DataSet 访问数据库	讲授、演练	4~8
9	第9章 数据绑定与数据绑定控件	讲授、演练	8~10
10	第10章 站点导航和母版页	讲授、演练	4
11	第11章 ASP.NET 网站实例——新闻网站	演练	10~12
12	第12章 用 ASP.NET MVC 架构开发网站	讲授、演练	4
		机动、考核	4
		合 计	64~78

本书由刘瑞新主编，朱立、张治斌、刘桂玲等编著，参加编写的作者有：刘瑞新（第1、9、11章），张治斌（第2章），沈淑娟（第3章），吴遥（第8章），崔淼（第4、5章），刘桂玲（第7章），朱立（第10、12章），第6章及课件的制作由王如雪、曹媚珠、陈文焕、刘有荣、李刚、孙明建、李索、刘大学、刘克纯、沙世雁、缪丽丽、田金凤、陈文娟、田同福、徐维维、徐云林完成，教材中的许多代码由沈宇峰编写并提供技术支持，全书由刘瑞新主编、统稿。由于编著者水平有限，书中错误与疏漏之处在所难免，敬请师生批评指正。

本书可作为高等学校计算机类相关专业教材，同样适合作为高职高专院校计算机类相关专业的教材，也可作为网站开发人员的技术参考书。

为了方便教师授课及读者的学习，本书提供了电子教案、源代码等，登录华信教育资源网（www.hxedu.com.cn）注册后免费下载。

作 者

目 录

第 1 章 ASP.NET 基础	1	2.5 实训	58
1.1 C/S 和 B/S 架构体系	1	第 3 章 ASP.NET 验证控件	64
1.2 静态网页与动态网页	2	3.1 验证控件概述	64
1.2.1 静态网页技术	2	3.2 必须项验证控件	65
1.2.2 动态网页技术	2	3.3 比较验证控件	67
1.3 .NET Framework 简介	5	3.4 范围验证控件	70
1.4 ASP.NET 网站的开发过程	6	3.5 正则表达式验证控件	72
1.4.1 ASP.NET 开发工具	6	3.6 自定义验证控件	74
1.4.2 新建和运行 ASP.NET 网站	7	3.7 验证摘要控件	79
1.4.3 打开和编辑 ASP.NET 网站	9	3.8 指定验证组	81
1.4.4 保存或关闭 ASP.NET 网站	12	3.9 禁用验证控件	81
1.5 ASP.NET Web 窗体模型	13	3.10 实训	82
1.5.1 ASP.NET Web 窗体的概念	13	第 4 章 ASP.NET 常用内置对象	84
1.5.2 ASP.NET Web 窗体的模型	13	4.1 Page 对象	84
1.5.3 ASP.NET 网页的代码模型	14	4.1.1 Page 对象的常用属性、 方法和事件	84
1.6 ASP.NET 网站的组成文件	17	4.1.2 Web 页面的生命周期	86
1.7 实训	18	4.1.3 Page 对象的 Load 事件与 Init 事件比较	86
第 2 章 ASP.NET 常用服务器标准控件	21	4.2 Response 对象	87
2.1 常用标准控件	21	4.2.1 Response 对象的常用属性和 方法	87
2.1.1 文本输入/输出控件	21	4.2.2 使用 Response 对象输出信息 到客户端	88
2.1.2 按钮控件	23	4.2.3 使用 Redirect 方法实现页面 跳转	90
2.1.3 超链接控件	27	4.3 Request 对象	90
2.1.4 图像控件	28	4.3.1 Request 对象的常用属性和 方法	90
2.1.5 选择控件	28	4.3.2 通过查询字符串实现跨页 数据传递	91
2.1.6 容器控件	41	4.4 Server 对象	92
2.1.7 其他专用控件	44		
2.1.8 动态生成控件	49		
2.2 Web 用户控件	50		
2.2.1 创建用户控件	50		
2.2.2 把 Web 窗体转换为用户控件	52		
2.3 ASP.NET 网站中资源的路径	53		
2.4 ASP.NET 控件的类型和通用属性	55		

4.4.1	Server 对象的常用属性和方法	92	7.1.1	ADO.NET 的数据模型	134
4.4.2	Execute 和 Transfer 方法	92	7.1.2	ADO.NET 的两种访问数据的方式	135
4.4.3	MapPath 方法	93	7.1.3	ADO.NET 中的常用对象	136
4.4.4	对字符串编码和解码	93	7.1.4	.NET 数据提供程序概述	137
4.5	实训	94	7.2	数据库的连接字符串	138
第 5 章	ASP.NET 的状态管理	97	7.2.1	数据库连接字符串的常用参数	138
5.1	状态管理概述	97	7.2.2	连接到 SQL Server 的连接字符串	139
5.2	创建和使用 ViewState 对象	97	7.2.3	连接字符串的存放位置	139
5.2.1	ViewState 对象概述	97	7.2.4	用数据源控件生成连接字符串	141
5.2.2	使用 ViewState	98	7.3	连接数据库的 Connection 对象	144
5.3	创建和使用 Cookie 对象	100	7.3.1	Connection 对象概述	144
5.3.1	创建 Cookie	100	7.3.2	创建 Connection 对象	145
5.3.2	读取 Cookie	101	7.3.3	Connection 对象的属性和方法	145
5.3.3	使用多值 Cookie	101	7.3.4	连接到数据库的基本步骤	146
5.4	创建和使用 Session 对象	104	7.3.5	关闭连接	148
5.4.1	Session 的工作原理	104	7.4	执行数据库命令的 Command 对象	148
5.4.2	Session 对象的常用属性及方法	104	7.4.1	Command 对象概述	148
5.4.3	使用 Session 对象	105	7.4.2	创建 Command 对象	148
5.5	创建和使用 Application 对象	108	7.4.3	Command 对象的属性和方法	149
5.5.1	Application 对象与 Session 对象的区别	108	7.4.4	增加、修改、删除记录操作	150
5.5.2	Application 对象的属性、方法和事件	109	7.4.5	统计数据库信息操作	153
5.5.3	使用 Application 对象	109	7.5	读取数据的 DataReader 对象	154
5.6	实训	111	7.5.1	DataReader 对象概述	154
第 6 章	SQL Server 数据库基础	116	7.5.2	创建 DataReader 对象	154
6.1	数据库的操作	116	7.5.3	DataReader 对象的属性和方法	155
6.2	表的操作	120	7.4.4	查询记录操作	156
6.3	记录的操作	124	7.7	实训	161
6.4	查询的操作	127	第 8 章	使用 DataSet 访问数据库	172
6.5	数据表脚本的生成和执行	129	8.1	DataSet 的基本构成	172
6.6	数据库的分离和附加	131	8.1.1	DataSet、DataAdapter 和数据源之间的关系	172
6.7	实训	133	8.1.2	DataSet 的组成结构和工作过程	173
第 7 章	使用 .NET 数据提供程序访问数据库	134			
7.1	ADO.NET 简介	134			

8.1.3 DataSet 中的常用子对象	174	第 10 章 站点导航和母版页	246
8.1.4 DataSet 对象常用属性和方法	174	10.1 ASP.NET 站点导航	246
8.2 DataAdapter 对象	175	10.1.1 概述	246
8.2.1 创建 DataAdapter 对象	175	10.1.2 ASP.NET 站点地图	247
8.2.2 DataAdapter 对象的属性和方法	175	10.1.3 SiteMapPath 控件	249
8.3 使用 DataSet 访问数据库	177	10.1.4 SiteMapDataSource 控件	251
8.3.1 创建 DataSet	177	10.1.5 TreeView 控件	252
8.3.2 填充 DataSet	177	10.1.6 Menu 控件	253
8.3.3 多结果集填充	179	10.2 ASP.NET 母版页	256
8.3.4 添加新记录	180	10.2.1 概述	257
8.3.5 修改记录	182	10.2.2 使用 ASP.NET 母版页的实例	258
8.3.6 删除记录	183	10.3 实训	263
8.3.7 DataTable 对象	184	第 11 章 ASP.NET 网站实例——新闻网站	272
8.4 实训	186	11.1 新闻网站的功能和设计	272
8.4.1 用户管理模块应具有的功能	186	11.1.1 新闻网站的功能	272
8.4.2 模块功能的实现	188	11.1.2 新闻网站的数据库	273
第 9 章 数据绑定与数据绑定控件	199	11.2 简化对数据库的操作	276
9.1 数据绑定	199	11.2.1 配置项	276
9.1.1 简单数据绑定和复杂数据绑定	199	11.2.2 SqlHelper 类中的方法	276
9.1.2 数据绑定控件概述	199	11.2.3 创建 SqlHelper 类	277
9.1.3 使用数据绑定表达式实现数据绑定	200	11.3 后台页面的设计	282
9.1.4 调用 DataBind() 方法实现数据绑定	206	11.3.1 后台管理主页和登录页	282
9.2 简单绑定控件	208	11.3.2 后台管理员的添加、编辑页	287
9.2.1 DropDownList 控件	208	11.3.3 新闻的添加	292
9.2.2 ListBox 控件	210	11.4 前台新闻首页、栏目页、内容页面的设计	297
9.3 Repeater 控件	212	11.4.1 前台新闻母版页	297
9.4 DataList 控件	216	11.4.2 新闻首页	298
9.5 GridView 控件	218	11.4.2 新闻内容页	300
9.5.1 GridView 控件的语法	218	11.4.3 新闻栏目页	302
9.5.2 GridView 控件的使用示例	221	11.6 实训	304
9.5.3 自定义列和模板列的使用	229	第 12 章 用 ASP.NET MVC 架构开发网站	305
9.6 DetailsView 控件	235	12.1 ASP.NET MVC 概述	305
9.7 FormView 控件	241	12.1.1 MVC 编程模型	305
9.8 实训	244		

12.1.2 建立第一个 MVC 应用程序	306	12.3.1 控制器	318
12.1.3 MVC 程序的结构	307	12.3.2 视图	323
12.2 路由和 URL 导向	308	12.4 模型与模型状态	328
12.2.1 MVC 路由	309	12.4.1 强类型视图	328
12.2.2 进站路由——从 URL 到路由	312	12.4.2 视图和模型	329
12.2.3 出站路由——从路由到 URL	316	12.4.3 ModelState	332
12.3 控制器和视图	318	12.4.4 验证规则	333
		12.5 实训	334
		参考文献	344

第1章 ASP.NET 基础

本章内容: C/S 和 B/S 架构体系, 静态网页与动态网页, .NET Framework 简介, Visual Studio 简介, ASP.NET 网站的创建过程。

本章重点: ASP.NET 网站的创建。

1.1 C/S 和 B/S 架构体系

目前在程序开发领域中, 主要有两大编程体系: 一是基于操作系统平台的 C/S 结构, 二是基于浏览器的 B/S 结构。

1. C/S 架构体系

在 2000 年以前, C/S (Client/Server, 客户机-服务器) 架构体系占据着开发领域的主流地位, 如图 1-1 所示。通常, 程序员将开发完成的软件安装在某台微机 (客户机) 中, 将数据库安装在专用的服务器 (数据库服务器) 中, 这样就可以利用两端的硬件资源, 将任务合理分配到客户端和服务端, 降低了系统的通信开销。这种架构要求客户机中必须安装客户端程序, 否则无法工作。另外, 在 C/S 架构中, 主要的数据分析处理工作需要在客户机中完成, 这就要求客户机有较高的硬件配置。C/S 架构的应用程序有: QQ、MSN、Foxmail、Outlook、浏览器及一些网络游戏等。



图 1-1 C/S 架构体系

2. B/S 架构体系

B/S (Browser/Server, 浏览器/服务器) 架构体系如图 1-2 所示, 由客户机、Web 服务器和数据库服务器三部分组成。在中小型应用系统中, Web 服务器可以与数据库服务器安装在同一台服务器中。与 C/S 架构相比, 它不需要在客户机中安装专门的客户端软件, 用户在使用程序时仅需要通过安装在客户机中的浏览器访问指定的 Web 服务器即可。目前, 绝大多数微机都在使用集成了 Internet Explorer (IE 浏览器) 的 Windows 操作系统, 也就是说, 只要客户机能够通过网络访问指定的 Web 服务器, 即可使用 B/S 架构的应用程序。此外, 在 B/S 架构中, 主要的数据分析处理工作是在应用服务器中完成的, 客户端主要用来下达指令和接收结果, 所以客户机的配置要求不高。B/S 架构非常适合“瘦客户端”的运行环境。

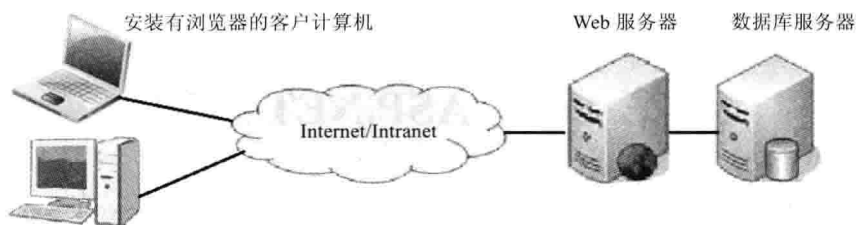


图 1-2 B/S 架构体系

1.2 静态网页与动态网页

随着 Internet 技术的发展，基于 Internet 的 Web 应用程序开发已经成为当今软件技术发展最快的应用领域，B/S 结构的应用程序已经成为应用软件的开发主流。

1.2.1 静态网页技术

在动态网页技术出现之前，所有的网页都是静态的。静态网页是指由网页编写者用纯 HTML 代码编写的网页，以 .html 或 .htm 文件格式保存。静态网页制作完成并发布后，网页的内容（文本、图像、声音、超链接等）和外观是保持不变的，无论哪个浏览者、在何时、以何种方式访问这个网页，它的外观总是保持不变。静态网页中不包含任何客户交互的动态内容，其优点是访问效率很高，网站的开发和架设相当容易。目前仍然有很多网站使用静态网页技术。

静态网页技术的工作过程（见图 1-3）如下。

① 浏览者在浏览器地址栏中输入 HTTP 请求或链接到该网页地址，该请求通过网络从浏览器传送到 Web 服务器中。

② Web 服务器在服务器中定位该 .html 或 .htm 文件，将其转化为 HTML 流。

③ Web 服务器将 HTML 流通过网络传送到浏览者的浏览器中。

④ 浏览器解析 HTML，并显示网页。



图 1-3 静态网页技术的工作过程

静态网页的主要缺点是，当网页中的内容需要改变时，必须重新制作网页，所以静态网页不适合需要频繁改变内容的网页。

1.2.2 动态网页技术

动态网页技术主要分为两种：客户端动态网页技术和服务器端动态网页技术。

1. 客户端动态网页技术

客户端动态网页技术是指 Web 服务器把原始的 HTML 页面及一组包含了页面逻辑的脚本、

组件等一起发送到客户端，这些脚本和组件包含了如何与浏览者交互并产生动态内容的指令，由客户端的浏览器及其插件解析 HTML 页面并执行这些指令。典型的客户端动态网页技术包括 JavaScript、VBScript、ActiveX 控件、Java Applet、Ajax 等。

客户端动态网页由网页制作者用 HTML 语言编写，并将其以.html 或.htm 文件格式保存。同时也可使用其他语言编写指令，这些指令嵌入 HTML 语言中，或者以单独的文件保存。

客户端动态网页技术的工作过程（见图 1-4）如下。

- ① 用户在客户端浏览器中输入一个 HTTP 网页请求，通过网络传送给 Web 服务器。
- ② Web 服务器在服务器中定位该.html 或.htm 文件，以及 HTML 文件指令中包含的其他文件，并将其转化为 HTML 流。
- ③ Web 服务器将 HTML 流和其他指令，通过网络传送到浏览者的浏览器中。
- ④ 浏览器插件解析指令，并将其转换为 HTML 文件。
- ⑤ 浏览器解析 HTML，显示网页。

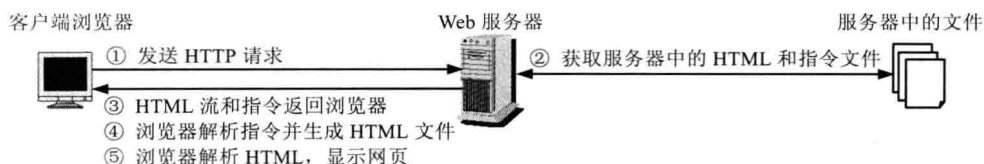


图 1-4 客户端动态网页技术的工作过程

客户端动态网页技术的主要优点是，充分利用了客户端的计算机资源，减轻了服务器和网络上的计算机压力，同时可以方便地实现基于图形的用户交互界面。然而，客户端动态网页技术需要把语言脚本和组件下载到客户端的计算机中，如果脚本或者组件较大，下载速度就会变慢。其次，现在的每种客户端浏览器可能存在兼容问题，不能完整地解析代码。还有，将脚本和组件下载到客户端的计算机中后，源代码不便于保密。另外，有些脚本和组件可能含有恶意代码。所以，客户端网页技术在 Web 应用程序上的应用一般局限在显示动画、验证用户输入等方面。

2. 服务器端动态网页技术

服务器端动态网页技术是指在 Web 服务器端根据客户端浏览器的不同请求，动态地生成相应的内容，然后发送给客户端浏览器。

服务器端动态网页技术的工作过程（见图 1-5）如下。

- ① 用户在客户端浏览器中输入一个 HTTP 网页请求，通过网络传送到 Web 服务器中。
- ② Web 服务器在服务器中定位指令文件。
- ③ Web 服务器根据指令生成 HTML 流。
- ④ Web 服务器将生成的 HTML 流通过网络传送到浏览者的浏览器中。
- ⑤ 浏览器解析 HTML，显示网页。

使用服务器端动态网页技术，所有指令都先在服务器中进行处理，并根据不同浏览者的请求生成不同的 HTML 静态网页，然后把静态网页传送到客户端的浏览器中，再由浏览器解析并显示出来。服务器端动态网页技术把原始页面代码始终隐藏在服务器中，浏览者无法看到原始代码，起到了保密作用。缺点是，由于页面是在浏览者请求时临时生成的，因此，首次显示网页时速度较慢。

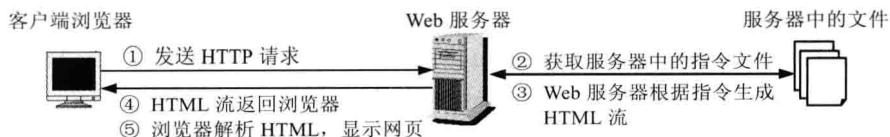


图 1-5 服务器端动态网页技术的工作过程

3. B/S 架构编程技术

目前 B/S 架构应用程序开发，主要使用 4 种技术：ASP、ASP.NET、JSP 和 PHP。

(1) ASP

ASP (Active Server Pages) 使用 VBScript 脚本语言，可以将脚本语言直接嵌入 HTML 文档中，不需要编译就可以直接运行。由于 ASP 程序是在服务器端运行的，当客户端浏览器访问 ASP 网页时，服务器将网页解释成标准的 HTML 代码发送给客户端，因此，不存在浏览器兼容的问题。

但是，每当客户端打开一个 ASP 页面时，服务器都会将该 ASP 程序解释一遍，最后生成标准的 HTML 代码发送到客户端，从而影响了 ASP 程序的运行速度。再有，ASP 不能运行在 Linux、FreeBSD 等操作系统中。在一般情况下，用 ASP 开发的程序只能运行在 Windows 操作系统的 IIS 环境中。目前新开发的系统已经不再使用 ASP。

(2) ASP.NET

ASP.NET 是一种用于创建动态 Web 页的强大的服务器端新技术，它可为 WWW 站点或企业内部互联网创建动态的、可进行交互的 HTML 页面。

ASP.NET 是微软 .NET 体系结构的一部分，并不是 ASP 的升级版本，所以在学习 ASP.NET 前并不需要先学习 ASP。ASP.NET 的主要优点如下。

① 使用 .NET 提供的所有类库，全面支持面向对象的程序设计，可以实现以往 ASP 所不能实现的许多功能。

② 引入了服务器端控件的概念，这样使开发交互式网站更加方便。

③ 引入了 ADO.NET 数据访问接口，大大提高了数据库访问效率。

④ 在 Visual Studio 可视化开发环境中创建 ASP.NET 应用程序，可以采用 C#、Visual Basic、C++、J# 等语言，进一步提高了编程效率。

⑤ 因为 ASP.NET 应用程序的核心部分在发布到 IIS 网站前就已被编译成 .dll 文件，所以执行速度更快。但是，ASP.NET 目前只能运行在 Windows 操作系统的 IIS 环境下。

(3) JSP

JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成，具有良好的跨平台性。在页面被客户端请求时，服务器对其中的 Java 代码进行处理，然后将生成的标准 HTML 页面发送到客户端。与 ASP 不同的是，JSP 页面第一次被访问时，服务器将 JSP 编译成二进制代码，并保存起来，以后当客户机再次访问该页面时，这些二进制代码将被直接调用，所以 JSP 较 ASP 具有更高的执行效率。

(4) PHP

PHP 程序最初是用 Perl 语言编写的简单程序，后来经其他程序员不断完善，于 1997 年发布了功能基本完善的 PHP3。PHP 程序可以运行在 UNIX、Linux 和 Windows 操作系统中，对客户端浏览器也没有特殊的要求。PHP 也是将脚本语言嵌入到 HTML 文档中，它大量采用了 C、Java 和 Perl 语言的语法，并加入了 PHP 自己的特征。

PHP 在 1999—2000 年期间应用较为普遍，由 Linux + PHP + MySQL 构成的完全开源而且非常稳定的应用平台曾经风靡一时，但由于 PHP 语言更新较慢再加上没有很好的技术支持，目前 PHP 正在逐步退出 B/S 架构的开发领域。

1.3 .NET Framework 简介

Microsoft .NET Framework（简称.NET）是一种新的开发平台，是微软公司为适应 Internet 发展的需要而推出的一种特别适合网络编程和网络服务开发的平台。对于软件开发人员来说，.NET 是继 DOS 开发平台（如 BASIC、Fortran、Pascal 等）、Windows 开发平台（Visual Basic、Visual FoxPro 等）之后，以计算机网络为背景的新一代软件开发平台。

1. .NET Framework 结构

Microsoft .NET Framework 是一个用于 Windows 应用程序、Web 应用程序、控制台应用程序和智能设备应用程序的平台。.NET Framework 提供丰富的类库，程序员可以使用类库来减少编写、测试和维护的代码量。

.NET 技术的核心是 .NET Framework，它是构建于计算机网络基础上的开发工具。.NET Framework 的基本结构如图 1-6 所示。



图 1-6 .NET Framework 结构

从图 1-6 中可以看出，.NET Framework 的最上层是开发完成的应用程序，分为基于 ASP.NET 的网络应用程序和基于 Windows 系统的应用程序。前者由 Web 窗体和 Web 服务（Web Service）组成，用户通过浏览器访问存放在服务器中的应用程序；后者由窗体和控件组成，用户可在 Windows 环境下直接运行程序。这两类应用程序均可使用 C#、Visual Basic、C++、J#等语言编写，而且在同一程序内允许使用不同的编写语言。

2. .NET Framework 的组件

.NET Framework 有两个主要组件：公共语言运行时环境和 .NET Framework 类库。

（1）公共语言运行时环境

.NET 框架的底层是公共语言运行时环境（Common Language Runtime, CLR），它提供了程序代码可以跨平台执行的机制。此外，.NET 的公共语言运行时环境还提供了系统资源统一管理和安全机制。

公共语言运行时环境管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。公共语言运行时环境是 .NET Framework 的基础，可以将运行时环境看作一个在执行时

管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且还强制实施严格的类型安全及可提高安全性和可靠性的其他形式的代码准确性。以运行时环境为目标的代码称为托管代码，而不以运行时环境为目标的代码称为非托管代码。

(2) .NET Framework 类库

.NET Framework 的中间一层是基础类库 (Basic Class Library)，它提供一个可以被不同程序设计语言调用的、分层的、面向对象的函数库。在传统的程序开发环境中，各种语言都有自己独立的函数库，互不通用，这样就使得跨语言编程十分困难。随着计算机及网络技术的发展，软件开发也进入了一个功能更强大、应用范围更广的时代，此时团队开发就显得尤为重要了。在 .NET Framework 的基础类库中提供了大量的基础类，如窗体控件、通信协议、网络存取等，并以分层的结构加以区分，这就使得各种语言的编程有了一个一致的基础，减小了各语言之间的界限。

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类型集合，程序员可以使用它开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面 (GUI) 应用程序，也包括基于 ASP.NET 所提供的创新的应用程序 (如 Web 窗体和 XML Web Services)。

图 1-6 显示了公共语言运行时环境和类库与应用程序之间以及与整个系统之间的关系。

3. .NET Framework 的版本

2002 年，微软引入了建立在 .NET Framework (1.0 版) 框架上的托管代码机制以及一种新的语言 C# (读作 C Sharp)，发布 Visual Studio .NET，其中的 Web 开发为 ASP.NET 1.0。

2003 年，微软发布 Visual Studio 2003，.NET 框架也升级到 1.1 版，其中的 Web 开发为 ASP.NET 1.1。

2005 年，微软发布 Visual Studio 2005，.NET 框架升级为 2.0 版。

2008 年，微软发布 Visual Studio 2008，.NET 框架为 2.0、3.0、3.5 版，可以创建面向 2.0、3.0 或 3.5 版的项目。

2010 年，发布 Visual Studio 2010，.NET 框架支持 2.0、3.0、3.5、4.0 版。

2012 年，微软发布 Visual Studio 2012，.NET 框架支持 2.0、3.0、3.5、4.0、4.5 版，适用于开发 Windows 8 专用程序。

2013 年 11 月 13 日，微软发布 Visual Studio 2013，.NET 框架支持 2.0、3.0、3.5、4.0、4.5、4.5.1 版。

1.4 ASP.NET 网站的开发过程

1.4.1 ASP.NET 开发工具

开发 ASP.NET 网站最好的工具是使用微软的 Visual Studio (简称 VS)，VS 是一套完整的集成开发工具，使用 VS 能够快速构建 ASP.NET 应用程序。VS 开发工具把开发设计中需要的各个环节 (界面设计、程序设计、运行和调试程序等) 集成在一个窗口中，极大地方便了开发人员的设计工作。通常，将这种集多种功能于一身的开发平台称为集成开发环境 (Integrated Development Environment, IDE)。Visual Studio 是一个家族产品系列，主要有：Professional (专业版)、Premium (企业版)、Ultimate (旗舰版) 等版本。对于初学者，任何版本都能满足学习。

使用 VS 开发环境，开发人员能够高效地开发 ASP.NET 应用程序。VS 开发环境为开发人

员提供了诸多控件,使用这些控件能够实现复杂的功能,极大地简化了开发人员的工作。使用 VS 开发环境进行 ASP.NET 应用程序开发还能够直接编译和运行 ASP.NET 应用程序。

SQL Server 是微软为开发人员提供的数据库工具,所以微软把 Visual Studio 和 SQL Server 紧密地集成在一起。通常,使用 Visual Studio 进行 ASP.NET 应用程序的开发,用 SQL Server 负责应用数据的存储。使用 SQL Server 进行 .NET 应用程序数据开发能够提高 .NET 应用程序的数据存储效率。

本书以 Visual Studio 2010 旗舰版+SQL Server 2008 为开发环境,讲授用 Visual C# 创建 ASP.NET 网站的开发方法。

1.4.2 新建和运行 ASP.NET 网站

基于浏览器的 B/S 结构的应用程序统称为 Web 应用程序,采用微软 ASP.NET 框架设计的程序称为 ASP.NET Web 应用程序,在 VS 中简称为网站。在 VS 集成开发环境 (IDE) 中,网站名就是文件夹名,也就是 ASP.NET Web 应用程序名,包括一系列多种类型的文件和文件夹。

在 VS 中创建一个 ASP.NET 网站,一般需要经过以下 5 个步骤。

① 新建网站。创建一个新的 ASP.NET 网站,并命名网站名称。

② 添加 Web 窗体。向网站中添加 Web 窗体,然后根据需要更改 Web 窗体名称。

③ 添加控件。设计网站中包含的所有 Web 窗体的外观,设置 Web 窗体中所有控件对象的初始属性值。

④ 编写事件代码。编写用于响应系统事件或响应用户事件的代码。

⑤ 运行网站。试运行并调试程序,纠正存在的错误,调整 Web 窗体。

本节通过一个简单 ASP.NET 网站的创建过程,介绍在 VS 中使用 Visual C# 语言创建 ASP.NET 网站的基本步骤。

【演练 1-1】 在 VS 中创建一个 ASP.NET 网站,网站中只有一个网页,其功能是显示两行文字。

① 启动 Visual Studio 2010,依次单击“开始”→“所有程序”→“Microsoft Visual Studio 2010”→“Microsoft Visual Studio 2010”,进入 VS,显示起始页。

② 依次单击“文件”菜单→“新建”→“网站”,显示“新建网站”对话框,在左侧窗格中选中“Visual C#”,在中间窗格中选中“ASP.NET 空网站”;在对话框下部的“Web 位置”框中选择默认的“文件系统”,在文本框中把默认的位置改为“C:\ex1-1”,表示在 C 盘根文件夹下创建一个“ex1-1”文件夹,如图 1-7 所示。最后单击“确定”按钮。

③ 向网站中添加一个 Web 窗体。在 VS 主窗体右侧的解决方案资源管理器中,右击网站名称“C:\ex1-1”,显示快捷菜单,如图 1-8 所示,单击“添加新项”。显示“添加新项”对话框,在左侧框中选择“Visual C#”,在中间“模板”框中选择“Web 窗体”,本例题不用更改默认的网页名称 Default.aspx,如图 1-8 所示。最后单击“添加”按钮。

④ 在工作区“Default.aspx”选项卡中默认显示源视图,如图 1-9 左图所示。单击“Default.aspx”选项卡底部的“设计”标签,切换到设计视图,如图 1-9 右图所示。

⑤ 向 Web 窗体中添加两行静态文本。在设计视图中,当输入文本时,会出现一个蓝色框,表示 div 标记,输入的文本在 div 标记中。在蓝色框中输入“我的第一个 ASP.NET 网站”,按 Enter 键换行,插入换行符
,再输入“开启 ASP.NET 之旅”,如图 1-10 所示。如果要使文本在 div 标记中居中,以及改变字体、字号、颜色等外观,在 VS 工具栏中单击相应按钮即可。

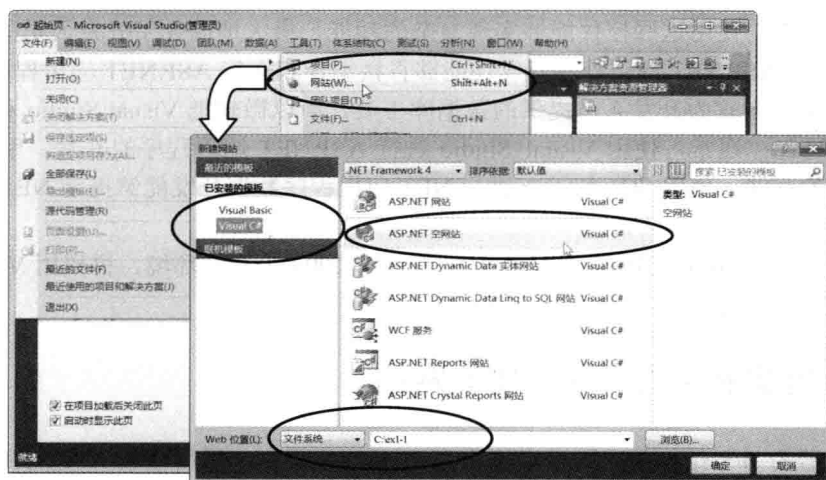


图 1-7 新建一个空网站

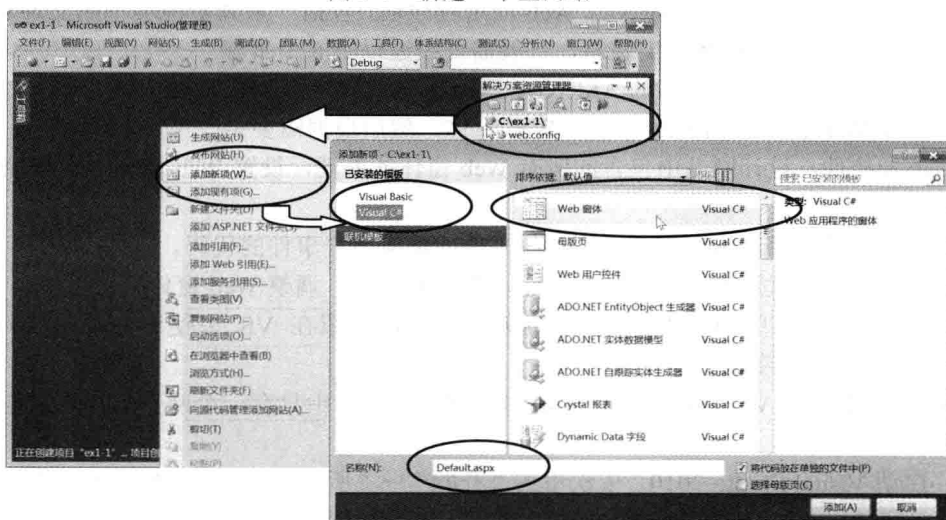


图 1-8 添加一个 Web 窗体

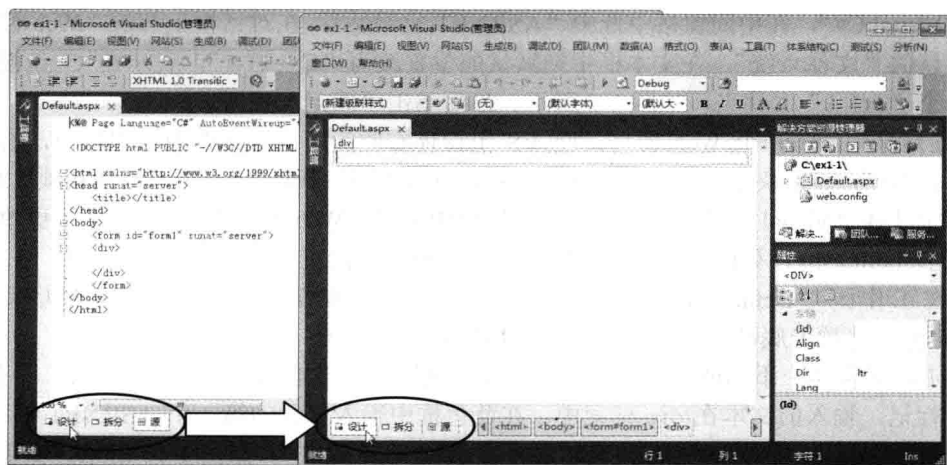


图 1-9 Web 窗体的“源”视图和“设计”视图