

 普通高等教育计算机规划教材

ASP.NET

程序设计教程(C#版)

第 2 版

崔 淼 关六三 彭 炜 主编

提供电子教案



下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

ASP.NET 程序设计教程(C#版)

第2版

崔森 关六三 彭炜 主编



机械工业出版社

社址: 北京市西城区百万庄大街24号

邮编: 100037

电话: (010) 68335634

电传: (010) 68335634

网址: <http://www.cmpbook.com>

电子邮箱: custserv@cmpbook.com

发行部: (010) 68335634

编辑部: (010) 68335634

印刷部: (010) 68335634

装订部: (010) 68335634

本书以 Microsoft Visual Studio 2008 为开发平台, 兼容 Microsoft Visual Studio 2005。以 C# 为开发语言, 面向毫无程序设计基础的读者, 从零开始, 采用案例方式, 全面细致地介绍了 ASP.NET 的基础知识、特点和具体应用。本书在上一版基础上增加了 CSS + DIV 网页布局技术、母版页与主题技术、ASP.NET AJAX、基于角色的安全技术、Microsoft SQL Server 数据库常用操作、LINQ to SQL 数据库访问技术等内容, 对数据库应用程序设计部分进行了结构优化, 使其层次更加清晰, 更易于理解。本书每章均配有一定数量的习题, 以方便读者上机练习。

为配合教学, 本书的配套教材《ASP.NET 程序设计教程 (C# 版) 上机指导与习题解答》(第 2 版) 对教材中的习题做了详细解答, 同时每章增加了实训环节。

本书适合作为高等院校计算机专业教材, 同时也可作为广大计算机爱好者的自学用书和各类 ASP.NET (C#) 程序设计培训班的教学用书。

图书在版编目 (CIP) 数据

ASP.NET 程序设计教程: C# 版/崔森, 关六三, 彭炜主编. —2 版. —北京: 机械工业出版社, 2010. 8

(普通高等教育计算机规划教材)

ISBN 978-7-111-31223-9

I. ①A… II. ①崔… ②关… ③彭… III. ①主页制作—程序设计—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字 (2010) 第 129445 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 张宝珠

责任印制: 李妍

北京振兴源印务有限公司印刷

2010 年 8 月第 2 版·第 1 次印刷

184mm × 260mm · 19.75 印张 · 488 千字

0001-4000 册

标准书号: ISBN 978-7-111-31223-9

定价: 34.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

门户网: <http://www.cmpbook.com>

销售一部: (010) 68326294

教材网: <http://www.cmpedu.com>

销售二部: (010) 88379649

读者服务部: (010) 68993821

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基本素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- (4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- (5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前言

本书以 Microsoft Visual Studio 2008 为开发平台, 兼容 Microsoft Visual Studio 2005。以 C# 为开发语言, 面向毫无程序设计基础的读者, 从零开始, 采用案例方式, 全面细致地介绍了 ASP.NET 的基础知识、特点和具体应用。本书在上一版基础上增加了 CSS + DIV 网页布局技术、母版页与主题技术、ASP.NET AJAX、基于角色的安全技术、Microsoft SQL Server 数据库常用操作、LINQ to SQL 数据库访问技术等内容, 对数据库应用程序设计部分进行了结构优化, 使其层次更加清晰, 更易于理解。

本书在例题处理上采用“任务驱动”方式, 即先给出设计目标, 然后介绍为实现该目标而采取的设计方法。为初学者考虑, 程序设计中的操作以详尽的表述结合图例来说明, 以求读者对每一步操作都清清楚楚。在代码设计中尽可能多地给出注释, 以求读者对每一行代码的意义及其前后联系明明白白。针对初学者的特点, 本书在编排上注意做到简明扼要、由浅入深和循序渐进, 力求通俗易懂、简洁实用。本书概念清晰、逻辑性强、层次分明、例题丰富, 符合教师教学和学生学习的习惯。本教材在编写的主导思想上突出一个“用”字, 避免繁琐的, 长篇大论的理论阐述, 紧紧抓住培养学生基本编程技能这个纲, 以求达到学以致用用的目的。全书共分为 12 章, 主要包括 Visual Studio 开发环境简介, 静态网页设计基础, C#语法基础, 结构化程序设计方法, 常用 Web 服务器控件, 常用内置对象和状态管理, 数据库应用程序开发和基于角色的安全管理等方面的内容。本书每章均配有一定数量的习题, 以方便学生课后练习。

对于初学程序设计的读者, 往往会出现“上课听得懂, 课后不会做”的现象。为方便上机练习和教师组织编程实训, 还编写了配套教材《ASP.NET 程序设计教程 (C#版) 上机指导与习题解答》(第 2 版) 对教材中的习题做了详细解答, 同时每章增加了实训环节, 最后还附带了一个基于三层架构的 BBS 论坛设计, 对面向对象的程序设计方法进行了介绍。

为配合教学, 本书提供教学课件及书中所有例题、习题的源代码打包压缩文件, 需要的读者可从机械工业出版社网站 www.cmpedu.com 下载。

本书由崔淼、关六三、彭炜主编, 参加编写的作者还有陈克坚、曾赞、王宁、陈红斌、朱一飞、郭璐青、田俊松、李晓娟、魏蔚、臧顺娟、张丽娜、刘克纯、彭春艳、翟丽娟、庄建新、彭守旺、崔瑛瑛、李建彬、马春锋、岳香菊、张国胜、李智、丁新旺。全书由崔淼统稿, 刘瑞新审。

由于编者水平有限, 本书难免会出现一些错误或不当之处, 恳请批评指正。

作者

目 录

出版说明

前言

第 1 章 ASP. NET 与 Visual Studio 开发平台	1
1.1 C/S 和 B/S 架构体系	1
1.1.1 C/S 与 B/S 架构的比较	1
1.1.2 B/S 架构编程技术	2
1.1.3 ASP. NET 网站的主要组成文件	3
1.2 Visual Studio 集成开发环境	5
1.2.1 Visual Studio 项目管理	5
1.2.2 使用 Visual Studio 的帮助系统	14
1.3 创建 ASP. NET 网站的基本步骤	18
1.4 习题	22
第 2 章 网页设计基础	23
2.1 HTML 和 XHTML 标记语言	23
2.1.1 HTML 的基本格式	23
2.1.2 常用 HTML 标记	25
2.1.3 XHTML 的基本格式	29
2.1.4 常用 XHTML 标记	30
2.2 在 Visual Studio 中设计网页	32
2.2.1 向 ASP. NET 网站中添加静态网页	32
2.2.2 .aspx 文件与 XHTML	33
2.2.3 Visual Studio 提供的网页设计环境	34
2.3 样式控制	35
2.3.1 CSS 的概念	35
2.3.2 向 ASP. NET 网站中添加样式表	38
2.3.3 使用样式生成器	39
2.3.4 创建和使用样式规则	40
2.4 页面布局	43
2.4.1 使用表格布局页面	43
2.4.2 使用 DIV 和 CSS 布局页面	45
2.4.3 页面元素的定位	51
2.5 习题	53
第 3 章 主题与母版页	55
3.1 使用主题和外观文件	55
3.1.1 使用主题	55

3.1.2	使用外观文件	56
3.1.3	利用主题实现页面换肤	59
3.2	母版页与内容页	62
3.2.1	母版页和内容页的概念	63
3.2.2	创建母版页和内容页	63
3.3	从内容页访问母版页的内容	67
3.3.1	从内容页访问母版页控件的属性	67
3.3.2	在内容页中响应母版页控件的事件	68
3.4	习题	69
第4章	C#语法基础与程序设计方法	70
4.1	C#变量	70
4.1.1	变量的命名规范	70
4.1.2	声明变量	71
4.1.3	给变量赋值	72
4.1.4	变量的作用域	72
4.2	数据类型及类型转换	73
4.2.1	数值类型	73
4.2.2	字符类型	74
4.2.3	布尔类型和对象类型	75
4.2.4	类型转换	75
4.3	运算符与表达式	77
4.3.1	运算符与表达式类型	78
4.3.2	运算符的优先级与结合性	82
4.4	C#常用方法与属性	83
4.4.1	日期时间类常用方法与属性	84
4.4.2	常用数学方法与属性	84
4.4.3	常用字符串方法与属性	85
4.4.4	随机方法	85
4.5	结构化程序设计方法	86
4.5.1	基本控件与顺序结构程序设计方法	86
4.5.2	选择结构程序设计方法	90
4.5.3	循环结构程序设计方法	95
4.6	习题	98
第5章	ASP.NET 常用控件	101
5.1	ASP.NET 控件概述	101
5.1.1	HTML 控件和服务端控件的概念	101
5.1.2	控件和页面的执行过程	102
5.2	常用标准服务器端控件	102
5.2.1	ImageButton 和 HyperLink 控件	102

5.2.2	Image 和 ImageMap 控件	103
5.2.3	RadioButton 和 RadioButtonList 控件	105
5.2.4	CheckBox 和 CheckList 控件	107
5.2.5	ListBox 和 DropDownList 控件	107
5.2.6	FileUpload 控件	110
5.2.7	Table 控件	114
5.2.8	容器控件	116
5.3	数据验证控件	120
5.3.1	数据验证控件概述	120
5.3.2	必须项验证控件(RequiredFieldValidator)	121
5.3.3	比较验证控件(CompareValidator)	123
5.3.4	范围验证控件(RangeValidator)	124
5.3.5	正则表达式验证控件(RegularExpressionValidator)	125
5.3.6	自定义验证控件(CustomValidator)	126
5.3.7	验证摘要控件(ValidatorSummary)	126
5.4	用户自定义控件	128
5.4.1	用户自定义控件概述	128
5.4.2	创建用户控件	128
5.4.3	在设计视图中添加用户控件	129
5.4.4	程序运行时动态添加用户控件	131
5.4.5	公开构成控件的属性	131
5.5	习题	133
第6章	ASP.NET 内置对象和状态管理	136
6.1	Page 对象	136
6.1.1	Page 对象的常用属性、方法和事件	136
6.1.2	Web 窗体页面的生命周期	137
6.1.3	Page 对象的 Load 事件与 Init 事件比较	138
6.2	Response 对象	139
6.2.1	Response 对象的常用属性和方法	139
6.2.2	使用 Response 对象输出信息到客户端	140
6.2.3	使用 Redirect 方法实现页面跳转	141
6.3	Request 对象	142
6.3.1	Request 对象的常用属性和方法	142
6.3.2	使用 UserHostAddress 属性获取用户 IP 地址	143
6.3.3	通过查询字符串实现跨页数据传递	144
6.3.4	使用 Browser 属性获取客户浏览器信息	145
6.3.5	使用 Form 属性处理 HTML 表单	146
6.4	Server 对象	146
6.4.1	Server 对象的常用属性和方法	146

6.4.2	Execute 和 Transfer 方法	147
6.4.3	MapPath 方法	147
6.4.4	对字符串编码和解码	148
6.5	ASP.NET 的状态管理	149
6.5.1	创建和使用 Cookie 对象	149
6.5.2	创建和使用 Session 对象	152
6.5.3	创建和使用 Application 对象	157
6.6	习题	160
第7章	ASP.NET AJAX	162
7.1	基本概念	162
7.1.1	Ajax 的主要内容及原理	162
7.1.2	ASP.NET AJAX 概述	163
7.2	常用 ASP.NET AJAX 控件	163
7.2.1	ScriptManager 控件和 ScriptManagerProxy 控件	164
7.2.2	UpdatePanel 控件	166
7.2.3	UpdateProgress 控件	170
7.2.4	Timer 控件	171
7.3	ASP.NET AJAX 控件工具包	172
7.3.1	安装 ASP.NET AJAX 控件工具包	173
7.3.2	使用 ConfirmButtonExtender 控件弹出确认对话框	175
7.3.3	使用 RoundedCornersExtender 控件实现圆角效果	178
7.3.4	使用 CalendarExtender 控件	178
7.4	习题	179
第8章	数据库基础和数据访问控件	181
8.1	使用数据库系统	181
8.1.1	创建 Microsoft SQL Sever 数据库	181
8.1.2	常用 SQL 语句	184
8.1.3	Microsoft SQL Server 常用操作	186
8.1.4	创建 Microsoft Access 数据库	190
8.2	使用数据控件访问数据库	192
8.2.1	使用数据源控件	192
8.2.2	使用 GridView 控件	193
8.2.3	使用 FormView 控件	197
8.2.4	使用 Repeater 控件	199
8.2.5	使用 DetailsView 控件	203
8.3	习题	206
第9章	使用 ADO.NET 访问数据库	209
9.1	ADO.NET 和数据库通用接口	209
9.1.1	ADO.NET 概述	209

9.1.2	数据库通用接口	209
9.1.3	ADO.NET 的数据模型	211
9.1.4	ADO.NET 中的常用对象	212
9.2	数据库连接对象(Connection)	212
9.2.1	Connection 对象概述	213
9.2.2	创建 Connection 对象	213
9.2.3	Connection 对象的属性和方法	213
9.2.4	数据库的连接字符串	215
9.3	数据库命令对象(Command)	219
9.3.1	创建 Command 对象	220
9.3.2	Command 对象的属性和方法	221
9.4	数据读取对象(DataReader)	224
9.4.1	DataReader 对象的常用属性及方法	224
9.4.2	使用 DataReader 对象	225
9.5	数据适配器对象(DataAdapter)	228
9.5.1	DataAdapter 对象概述	228
9.5.2	DataAdapter 对象的属性和方法	228
9.6	习题	231
第10章	使用 DataSet 访问数据库	233
10.1	DataSet 概述	233
10.1.1	DataSet 与 DataAdapter	233
10.1.2	DataSet 的组成	234
10.1.3	DataSet 中的对象、属性和方法	235
10.2	使用 DataSet 访问数据库	236
10.2.1	创建 DataSet	236
10.2.2	填充 DataSet	236
10.2.3	多结果集填充	238
10.2.4	添加新记录	239
10.2.5	修改记录	240
10.2.6	删除记录	241
10.3	DataSet 应用实例(网站用户管理)	241
10.3.1	用户管理模块应具有的功能	242
10.3.2	模块功能的实现	244
10.4	习题	255
第11章	LINQ to SQL 数据库操作	256
11.1	LINQ 的概念	256
11.1.1	LINQ 的构成	256
11.1.2	与 LINQ 相关的几个概念	257
11.2	使用 LinqDataSource 控件	262

11.2.1	创建 DataContext 类	262
11.2.2	配置 LinqDataSource	265
11.2.3	LinqDataSource 控件使用示例	266
11.3	使用 LINQ to SQL 的对象和方法	267
11.3.1	返回数据表中的全部记录(select 语句)	268
11.3.2	返回数据表中符合条件的部分记录(Where 语句)	269
11.3.3	返回数据集合的排序(OrderBy 方法)	271
11.3.4	连接不同数据表(Join 语句)	271
11.3.5	插入新记录	273
11.3.6	修改记录	274
11.3.7	删除记录	274
11.3.8	使用 LINQ to SQL 直接执行 SQL 语句	274
11.4	习题	276
第 12 章 ASP.NET 的安全管理		279
12.1	用户身份认证	279
12.1.1	传统的认证方法	279
12.1.2	基于角色的安全管理简介	280
12.2	使用网站管理工具进行安全设置	280
12.2.1	连接 SQL Server 数据库	280
12.2.2	使用 ASP.NET 网站管理工具	282
12.3	使用登录控件	286
12.3.1	Login 控件	286
12.3.2	CreateUserWizard 控件	288
12.3.3	LoginName 和 LoginStatus 控件	288
12.3.4	LoginView 控件	289
12.3.5	ChangePassword 和 PasswordRecovery 控件	292
12.4	使用 Membership API	295
12.4.1	用户管理	295
12.4.2	角色管理	302
12.5	习题	304
参考文献		305

第1章 ASP.NET 与 Visual Studio 开发平台

随着 Internet 技术的飞速发展, 基于 WWW 的网站开发技术和网页制作方法也有了长足的进步。从早期使用 HTML 标记语言设计的静态网页, 到使用 CGI、ASP、JSP、PHP、ASP.NET、ASP.NET 2.0、ASP.NET 3.5 开发设计的, 功能更加强大且具有良好交互性的动态网站, 使基于 WWW 服务的浏览器/服务器 (Browsers/Server, B/S) 架构应用程序得到了大幅度的发展。

1.1 C/S 和 B/S 架构体系

目前在应用程序开发领域中, 主要有两大编程体系: 一是基于操作系统平台的客户机/服务器 (Client/Server, C/S) 架构, 另一种是基于 WWW 服务的 B/S 架构。

1.1.1 C/S 与 B/S 架构的比较

在 2000 年以前 C/S 架构占据着开发领域的主流地位, 其结构如图 1-1 所示。通常程序员将开发完成的软件安装在某计算机 (客户机) 中, 将数据库安装在专用的服务器 (数据库服务器) 中, 用户通过安装在客户机中的软件和网络进行各种数据库操作。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。

再有, 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, 以保证能高速处理由服务器端发送过来的大量原始数据。常用的聊天工具如 QQ、MSN, 及一些网络游戏都属于 C/S 架构的应用程序。

B/S 架构体系如图 1-2 所示, 由客户机、应用服务器和数据库服务器 3 个部分组成。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计算机中。在大型应用系统中, 也可以将同一应用程序的不同数据处理业务分别安装在不同的应用服务器中 (两台或多台), 以实现分布式数据处理的需要。

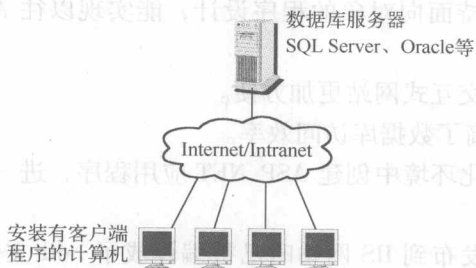


图 1-1 C/S 架构体系

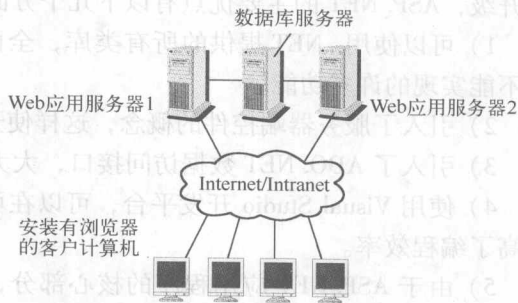


图 1-2 B/S 架构体系

与 C/S 架构相比, B/S 应用程序不需要在客户机上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上的 Internet 浏览器访问指定的网页即可。目前绝大多数 PC 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, 即可正常使用 B/S 架构的应用程序。

此外, 在 B/S 架构的应用程序中, 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。

1.1.2 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 种编程技术: ASP、ASP.NET、PHP 和 JSP。

1. ASP

ASP (Active Server Pages) 所使用的 VBScript 脚本语言直接来源于 Visual Basic 语言, 秉承了 Visual Basic 语言的简单易用特点, 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, 所以不存在浏览器兼容的问题。

但因为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。而且, ASP 不能运行在 Linux、FreeBSD 等操作系统中。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。

2. ASP.NET

2002 年 Microsoft 公司发布了 .NET Framework 1.0, 其中的 ASP 版本为 ASP.NET 1.0; 2003 年又发布了 .NET Framework 1.1, 其中的 ASP 版本为 ASP.NET 1.1, 开发平台为 Visual Studio 2003; 2005 年推出了 .NET Framework 2.0 版, 其中的 ASP 版本为 ASP.NET 2.0, 开发平台为 Visual Studio 2005; 2008 年推出了最新的 .NET Framework 3.5 版, 其中的 ASP 版本为 ASP.NET 3.5, 开发平台为 Visual Studio 2008。

ASP.NET 目前主要支持的语言有 C# (读作 C Sharp) 和 Visual Basic。其中本教材重点介绍的 C# 语言是 Microsoft 公司专门为 .NET 量身定做的编程语言, 它与 .NET 有着密不可分的联系。

与早期的 ASP 相比, ASP.NET 有了本质上的变化, 不能将 ASP.NET 看成是 ASP 的简单升级, ASP.NET 的主要优点有以下几个方面:

1) 可以使用 .NET 提供的所有类库, 全面支持面向对象的程序设计, 能实现以往 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) 使用 Visual Studio 开发平台, 可以在可视化环境中创建 ASP.NET 应用程序, 进一步提高了编程效率。

5) 由于 ASP.NET 应用程序的核心部分, 在发布到 IIS 网站时已被编译成了 .dll 文件, 所以执行速度更快。但 ASP.NET 目前只能运行在 Windows 操作系统的 IIS (Internet Information Server) 环境中, 可移植性较差。

3. JSP

JSP 页面是由 HTML 代码和嵌入其中的 Java 代码组成的,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 JPS 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,但因 PHP 语言更新较慢,加上没有很好的技术支持,目前 PHP 的市场占有率正在逐步下降。

1.1.3 ASP.NET 网站的主要组成文件

Internet 的快速发展推动了网页设计方法的进步,从早期静态的 HTML 到动态的 CGI,再到 ASP、JSP 和 PHP。虽然基于 B/S 架构的 ASP 易学好用,深得用户的支持,但在语言编写环境、代码重用以及安全性和可靠性等方面仍然存在着一些明显的不足,为了弥补这一不足,Microsoft 公司推出了全新的 ASP.NET。

ASP.NET 由 Microsoft 公司的 .NET 技术细化而来,目前最高版本为 ASP.NET 3.5,采用 Visual Studio 2008 作为可视化开发环境。它跟 ASP 技术有着紧密的联系,但又有着本质的不同。ASP.NET 完全基于模块与组件,具有更好的可扩展性与可定制性,它在数据处理方面引入了许多具有革新意义的新技术,其功能远远超越了 ASP,从而具有更强的灵活性,有效地缩短了 Web 应用程序的开发周期。

ASP.NET 为用户提供了一个全新而强大的服务器控件结构。它几乎完全是基于组件和模块化的,每个页面、对象和 HTML 元素都是一个可运行的组件对象。在开发语言上,ASP.NET 抛弃了 VBScript 和 JavaScript,而使用 .NET Framework 所支持的 C#、VB 等作为开发语言,这些语言生成的网页在后台被转换成类(Class),并被编译成一个动态链接库文件(.dll),也正因为如此,ASP.NET 的执行效率比 ASP 更高。

在开始创建 ASP.NET 网站之前,有必要先了解一下 ASP.NET 应用程序的组成及几个重要文件的作用。

一个使用 Visual Studio 开发出来的 ASP.NET 网站,通常包含以下一些主要文件:

- 1) 一个或多个扩展名为 .aspx 的网页文件,网站中也允许包含 .htm 或 .asp 文件。
- 2) 一个或多个 web.config 配置文件。
- 3) 一个以 Global.asax 命名的全局文件(可选)。
- 4) App_Code、App_Data 等专用目录。

1. 网页文件(.aspx)

网页文件,也称为 Web 窗体。它是 Web 网站应用程序运行的主体。在 ASP.NET 中的基

本文件就是这些以 .aspx 为扩展名的网页文件。一个 ASP.NET 网站可以看做是由众多 .aspx 文件通过超链接集合而成的。

在大家熟悉的 Windows 应用程序中，主程序文件是以 .exe 为扩展名的可执行文件，这种文件可以直接在 Windows 操作系统中运行。与 Windows 应用程序不同，ASP.NET 的 .aspx 文件只能运行在 Windows Server 提供的 IIS 服务器中，用户可以通过浏览器打开 .aspx 文件。

当用户通过浏览器请求一个 .aspx 文件时，IIS 服务器首先要创建程序中包含的服务器端控件，运行服务器端代码，然后将程序执行结果转换成标准 HTML 代码发送给客户端。

当然，并不是每次接到客户端请求时，都要在服务器端重新解读和运行，对于那些曾经被访问过，并且没有改变的 .aspx 网页，服务器会从缓冲区中读取上次的运行结果直接发送给客户端，从而提高了程序的运行效率。

在 ASP.NET 网站中可以包含 .html、.asp、.css 和其他类型的文件，服务器在处理这些文件时仍采用原有的处理方式不变。对 .html 文件不做任何处理直接发送到客户端，对 .asp 文件则需要转换成标准的 HTML 文件后再发送到客户端。

如果将一个标准 HTML 网页文件的扩展名改为 .aspx 也是可以的，但服务器在接到客户端请求后，会按 ASP.NET 程序进行解读，当服务器发现其中并不包含服务器端代码时，也会将 HTML 文件发送到客户端，这样做将会无端增加服务器的资源开销，是一种不可取的设置方法。

在尚未发布的本地网站中存在有一些扩展名为 .cs (C#) 或 .vb (Visual Basic) 的文件，这些文件是对应网页 (.aspx) 的源程序文件 (C#或 Visual Basic)，设计完毕并发布到远程服务器中的 ASP.NET 网站通常不再包含此类文件，程序中的源文件经编译后会生成一些 .dll 文件。网站发布到远程 Web 服务器后，在站点文件夹中系统会创建一个 bin 文件夹，这些 .dll 文件就存放在该文件夹内。

2. 网站配置文件 (web.config)

web.config 文件是一个基于 XML 的配置文件，它用来储存 ASP.NET 网站的配置信息，它可以出现在应用程序的每一个目录中。当用户通过 Visual Studio 开发平台新建一个 ASP.NET 网站后，默认情况下会在根目录自动创建一个默认的 web.config 文件，其中包括了默认的配置信息，网站中所有子目录都将继承这些配置。在某些特定的情况下也可以为子目录设置单独的 web.config 文件，该文件的设置将覆盖根站点 web.config 文件的设置。

web.config 文件的作用是对 ASP.NET 应用程序进行统一的配置，如客户端的认证方式；基于角色安全技术的策略；数据绑定的方法；远程处理对象等。在 web.config 中定义的元素及对应的属性值，在 web.config 所覆盖的范围内自动生效，用户无需在所有页面代码中逐一加以设置。用户自定义属性可以在程序的任何部分通过代码调用，无需重复添加代码。

3. ASP.NET 目录

ASP.NET 允许向网站中添加一些具有特殊用途的目录，其中最常用的有：App_Code 和 App_Data，它们在需要时可由系统自动创建。

(1) App_Code

代码共享目录 App_Code 用来存放应用程序中所有网页都可以使用的共享文件。如将类文件存放在该目录下,就可以被程序中所有网页调用。在构建三层架构应用程序时,可以将中间层代码文件放在该目录下以实现共享。

(2) App_Data

数据共享目录 App_Data 用来存放数据库文件。为了实现客户管理和个性化服务,系统将一些专用的数据库或数据表文件存放在该目录下。需要说明的是,存放在 App_Data 目录下的文件无法通过 URL 地址直接访问,所以也可在该目录下存放那些需要受到保护的文件。

1.2 Visual Studio 集成开发环境

Visual Studio 是 Microsoft 推出的用于软件开发的重要平台,目前其最高版本为 Visual Studio 2008,内置 .NET Framework 版本为 3.5,同时提供对 .NET Framework 2.0、3.0 的支持。本教材中除特殊声明的内容外,均同时适用于 Visual Studio 2005 和 Visual Studio 2008 两个版本。

Visual Studio 开发平台将程序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)集成在同一个窗口中,极大地方便了开发人员的设计工作。通常将这种集多项功能于一身的开发平台称为集成开发环境 (IDE)。

1.2.1 Visual Studio 项目管理

在 Visual Studio 开发环境中,所有的程序均属于一个“项目”(Windows 应用程序)或“网站”(ASP.NET 应用程序),其中通常包括了 Web 窗体文件和其他一些与程序相关的文件,所以项目管理是在 Visual Studio 集成开发环境中进行 ASP.NET 网站应用程序设计工作的基础。

1. Visual Studio 的初始设置

初次运行 Visual Studio 时系统将显示如图 1-3 所示的“选择默认环境设置”对话框,要求进行开发环境的初始配置。选择不同的项目会导致 Visual Studio 的菜单命令、工具箱等具有不同的内容。进行 ASP.NET 网站开发应在“选择默认环境设置”列表中,选择“Web 开发设置”后单击“启动 Visual Studio”按钮进入开发环境。

如果在使用的过程中想改变默认环境设置,可以在 Visual Studio 主界面中执行“工具”菜单下的“导入和导出设置”命令,通过如图 1-4 所示的“导入和导出设置向导”对话框对当前设置进行必要地修改。在对话框中,选择“重置所有设置”单选钮,单击“下一步”按钮,在如图 1-5 所示的“保存当前设置”对话框中选择“否,仅重置设置,从而覆盖我的当前设置”单选钮,单击“下一步”按钮。

在如图 1-6 所示的“选择一个默认设置集合”对话框中,在“要重置为哪个设置集合”列表框中,选择希望的选项后单击“完成”按钮即可将 Visual Studio 开发环境按新的选择进行重新配置。

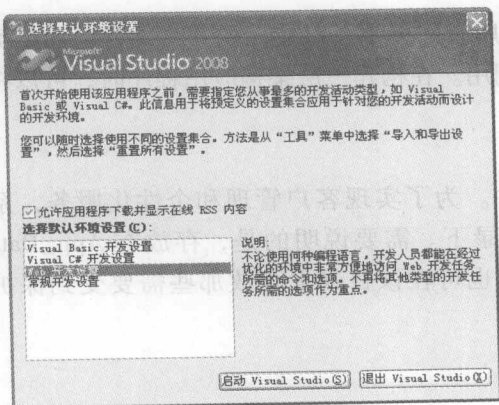


图 1-3 选择默认环境设置

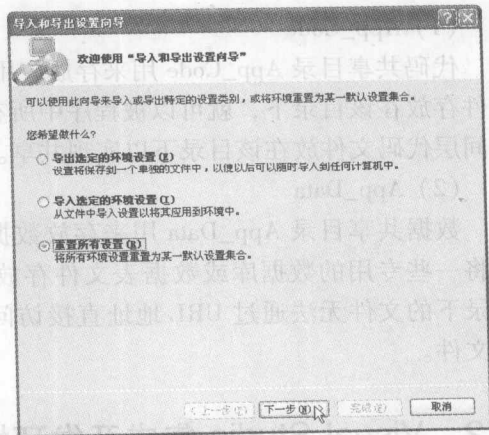


图 1-4 重置所有设置

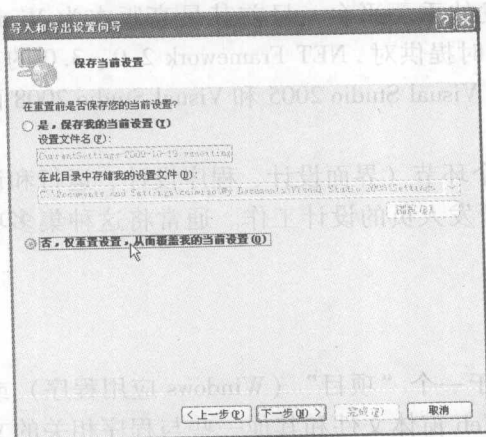


图 1-5 保存设置

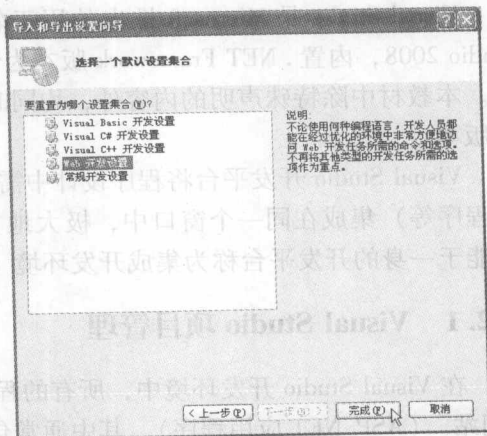


图 1-6 重新选择默认设置集合

2. 新建网站项目

在 Visual Studio 中, 一个完整的应用系统可能会包含在若干个“项目”中, 与应用系统所有相关项目的集合被称为“解决方案”。

ASP.NET 网站是最常用的项目类型。创建一个 ASP.NET 网站应用程序, 首先需要在 Visual Studio 环境中创建一个新网站项目。在系统启动后, 可在如图 1-7 所示的“起始页”界面“最近的项目”子窗口的“创建”栏中单击“网站”, 打开如图 1-8 所示的对话框。在“模板”列表中选择“ASP.NET 网站”, 在“位置”下拉列表框中选择“文件系统”, 在“语言”下拉列表框选择 Visual C#, 并在保存位置栏中指定网站文件的保存位置 (如本例的 D:\ASP.NET3.5\code\WebSite1) 及 .NET Framework 版本 (.NET 2.0、3.0 或 3.5), 设置完毕后单击“确定”按钮。注意, 在 Visual Studio 2005 中仅提供了对 .NET Framework 2.0 的支持。

在 Visual Studio 中可以使用“文件系统”方式, 将新建的 ASP.NET 网站存放在本地硬盘的任意位置, 也可以通过 HTTP 或 FTP 方式将站点直接创建在本地或远程 IIS 服务器中。