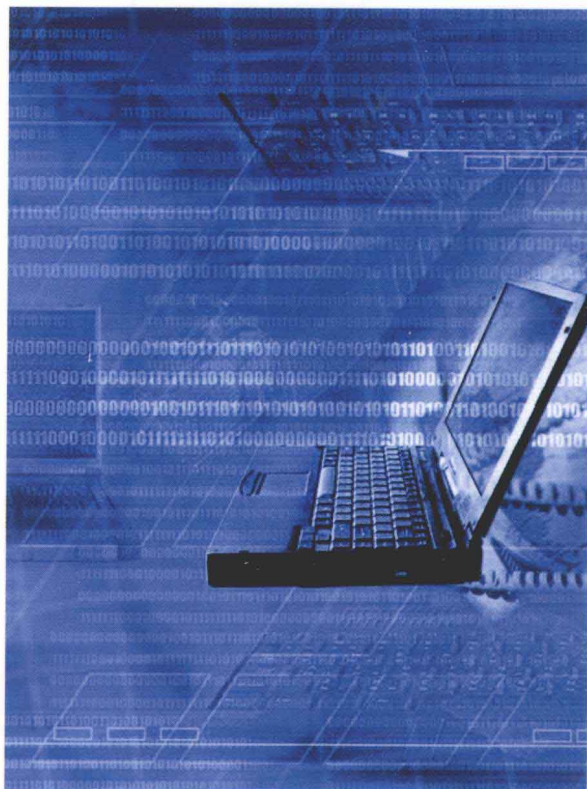


ASP.NET 4.0基础教程

(C# 2010篇)

- ◆ ASP.NET 4.0开发基础
- ◆ C# 语言基础知识
- ◆ Web控件与用户控件
- ◆ 主题、母版页和页面导航
- ◆ ASP.NET常用对象
- ◆ ADO.NET数据库编程
- ◆ 数据绑定和数据控件
- ◆ XML数据操作
- ◆ LINQ技术
- ◆ 配置ASP.NET应用程序
- ◆ 网络书店开发实例



王祥仲 朱艳丽 编著

清华大学出版社

内 容 简 介

本书包括 13 章,系统地介绍了如何使用 ASP.NET 开发动态网站,具体包括 ASP.NET 的运行环境、C#编程语言概述、ASP.NET 控件的使用、常用对象、数据绑定和数据控件、母版页、网站导航、数据库编程、XML 数据管理、LINQ 技术等内容,最后通过一个具体的动态网站开发项目演示了利用 ASP.NET 4.0 进行动态网站开发的方法和思路。

本书教学重点明确、结构合理、语言简明,而且每个实例均为作者在本领域工作中的真实案例,具有很强的实用性。

本书可作为高等学校计算机相关专业的教材,也可供广大网站开发人员参考。

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 4.0 基础教程(C# 2010 篇)/王祥仲,朱艳丽 编著. —北京:清华大学出版社,2012.6
(高等学校计算机应用规划教材)

ISBN 978-7-302-28929-6

I. ①A… II. ①王… ②朱… 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. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字(2012)第 111017 号

责任编辑:刘金喜

装帧设计:康 博

责任校对:蔡 娟

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.75

字 数:502 千字

版 次:2012 年 6 月第 1 版

印 次:2012 年 6 月第 1 次印刷

印 数:1~5000

定 价:33.00 元

产品编号:045611-01

前 言

ASP.NET 4.0 是一种基于服务器的功能强大的技术,用于为 Internet 或企业内部网创建动态的、交互式的 HTML 网页。ASP.NET 4.0 构建在 .NET Framework 4.0 之上,扩展了 ASP.NET 3.5 的功能,其内核是一个基于控件的、事件驱动的架构,这意味着只需要向页面添加少量的代码,就可以完成强大的功能。

本书面向初、中级用户,由浅入深、全面系统地介绍了使用 ASP.NET 和 C# 开发动态网站的基础知识、基本方法和具体应用。

本书由浅入深、层层深入地讲解了使用 ASP.NET 开发动态网站的技术,内容共分为 13 章,包括 ASP.NET 开发基础、C# 编程语言、ASP.NET 的服务器控件和核心对象、数据访问、母版页、网站导航等技术的讲解,最后通过一个综合实例展示了如何使用多种技术来开发动态网站。

第 1 章讲解 ASP.NET 的基础知识。首先对 ASP.NET 技术进行概括介绍,然后介绍 Visual Studio 2010 集成开发环境,最后通过一个实例介绍如何使用 ASP.NET 创建一个 Web 程序。

第 2 章讲解 C# 编程语言。首先介绍 C# 的数据类型和变量,然后介绍流程控制的相关知识和 C# 面向对象的特征——类和对象,最后介绍 C# 4.0 的新特性。

第 3 章讲解 Web 控件。首先介绍基本的 Web 控件,然后介绍 ASP.NET 控件类,接着介绍了列表控件、表控件和验证控件,最后介绍两个比较复杂的控件:Calendar 控件和 AdRotator 控件。

第 4 章讲解用户控件,主要介绍用户控件的创建和使用。

第 5 章讲解主题和母版页,利用这两个工具就能够整合页面到一个统一网站。首先介绍主题的创建和应用,接着介绍母版页的创建和应用。

第 6 章讲解页面导航的有关知识。首先介绍站点地图,接着介绍两个导航控件:TreeView 控件和 Menu 控件。

第 7 章讲解 ASP.NET 中的核心对象。主要包括 Response 对象、Request 对象、Server 对象、Session 对象、Cookie 对象和 Application 对象。

第 8 章讲解 ADO.NET 数据库编程。首先概括介绍 ADO.NET,然后介绍如何连接数据库和获取数据,最后介绍如何使用 DataReader 和 DataAdapter。

第 9 章讲解如何进行数据绑定。重点介绍数据源控件 SqlDataSource 和数据绑定控件,包括 GridView 控件、DetailsView 控件、ListView 控件和 Chart 控件。

第 10 章介绍 XML 数据管理。首先介绍 XML 的基础知识,然后介绍如何使用流模式操作 XML 数据,接着介绍使用 XmlDocument 类编辑 XML 数据的方法,最后介绍 DataSet 和 XML 的相互转换。

第 11 章介绍 LINQ 技术。首先介绍基于 C# 的 LINQ 技术,然后介绍 LINQ 到 ADO.NET 的知识。

第 12 章介绍 ASP.NET 程序的配置。主要包括使用 web.config 进行配置和使用 global.asax 进行配置。

第 13 章通过开发一个“网络书店”系统来演示如何综合使用多种技术开发 Web 网站。本章除介绍 ASP.NET 的具体技术之外,对于需求分析和系统设计、数据库设计以及功能模块的划分也有比较详细的介绍,有利于读者了解一个实际项目的开发流程。

对于每个知识点,本书都有实例演示。为了便于读者快速了解 ASP.NET 技术,本书的实例没有纠缠于具体而又烦琐的程序细节,而是简单明了地说明了如何运用知识点。本书最后一章是一个综合案例,取自现实生活,希望读者通过这个案例,能够对如何使用 ASP.NET 编程有比较深刻的了解。本书适用对象为 ASP.NET 的初、中级读者,可以作为高等学校计算机相关专业的教材,也适合一些工程技术人员或者工科高校学生自学使用。

本书应用到的所有例子的源代码可通过 <http://www.tupwk.com.cn/downpage> 网站下载。要运行这些代码,读者需要配置一个运行环境,需要安装 Visual C# 2010 专业版,详细的配置可参考本书具体章节中的介绍。

本课程总学时为 65 学时,各章学时分配见下表(供参考):

课 程 内 容	学 时 数			
	合 计	讲 授	实 验	机 动
第 1 章 ASP.NET 4.0 开发基础	2	2		
第 2 章 C#语言快速掌握	6	4	2	
第 3 章 Web 控件	6	4	2	
第 4 章 用户控件	4	2	2	
第 5 章 主题和母版页	4	3	1	
第 6 章 页面导航	3	2	1	
第 7 章 ASP.NET 常用对象	8	6	2	
第 8 章 ADO.NET 数据库编程	7	5	2	
第 9 章 数据绑定和数据控件	6	4	2	
第 10 章 XML 数据操作	3	2	1	
第 11 章 LINQ 技术	6	4	2	
第 12 章 配置 ASP.NET 应用程序	3	2	1	
第 13 章 网络书店	7	3	3	1
合计	65	43	21	1

本书除封面署名作者外，参与编写的人员还有林丹、李辉、田芳、王建国、赵海峰、刘勇、徐超、周建军、徐兵、黄飞、林海、马建华、孙明、高峰、郑勇、刘建、李彬、彭丽、许小荣等。在此，编者对以上人员致以诚挚的谢意！

由于本书涉及的范围比较广泛，加之作者的经验有限，时间仓促，书中难免有不足之处，敬请广大读者、专家提出宝贵意见。

编 者

2012年2月

目 录

第 1 章 ASP.NET 4.0 开发基础	1
1.1 ASP.NET 简介	1
1.1.1 .NET 简介	1
1.1.2 ASP.NET 页面与 Web 服务器的交互过程	2
1.2 Visual Studio 2010 集成开发环境	3
1.2.1 集成开发环境简介	3
1.2.2 解决方案资源管理器	4
1.2.3 属性对话框	5
1.2.4 工具箱	6
1.3 创建第一个 ASP.NET 4.0 应用程序	6
1.3.1 创建 Web 站点	6
1.3.2 编写 ASP.NET 4.0 应用程序	8
1.3.3 编译和运行应用程序	9
1.4 Visual Studio 2010 新增功能	10
1.5 习题	11
第 2 章 C#语言快速掌握	13
2.1 数据类型	13
2.1.1 值类型	14
2.1.2 引用类型	18
2.1.3 装箱和拆箱	21
2.2 变量和常量	21
2.2.1 变量	21
2.2.2 常量	22
2.3 运算符	23
2.3.1 算术运算符	23
2.3.2 赋值运算符	24
2.3.3 关系运算符	25
2.3.4 逻辑运算符	26
2.3.5 条件运算符	27

2.3.6 运算符的优先级	27
2.4 流程控制	28
2.4.1 条件语句	29
2.4.2 循环语句	34
2.4.3 跳转语句	36
2.5 类和对象	38
2.5.1 类	38
2.5.2 属性和方法	40
2.5.3 对象的创建和回收	41
2.5.4 继承和多态	44
2.6 委托与事件	49
2.6.1 概述	49
2.6.2 使用委托进行回调	50
2.6.3 动态注册和移除事件	51
2.7 C# 4.0 的新特性	53
2.7.1 大整数类型 BigInteger	53
2.7.2 动态数据类型 dynamic	54
2.7.3 命名参数和可选参数	55
2.8 习题	57
第 3 章 Web 控件	60
3.1 基本的 Web 控件	60
3.2 Web 控件类	64
3.2.1 Web 控件的基本属性	65
3.2.2 单位	67
3.2.3 枚举	67
3.2.4 颜色	67
3.2.5 字体	68
3.3 Web 控件的事件	69
3.3.1 Web 控件的事件模型	69
3.3.2 Web 控件事件的绑定	70
3.4 列表控件	71
3.4.1 ListBox 控件	71

3.4.2 DropDownList 控件.....	73	第 6 章 页面导航.....	133
3.4.3 CheckBoxList 控件.....	75	6.1 站点导航.....	133
3.4.4 RadioButtonList 控件.....	77	6.1.1 基于 XML 的站点地图.....	133
3.5 表控件.....	79	6.1.2 SiteMapDataSource 服务器 控件.....	135
3.5.1 表控件对象模型.....	79	6.2 TreeView 服务器控件.....	135
3.5.2 向页面中添加表控件.....	81	6.3 Menu 服务器控件.....	137
3.5.3 动态操作表控件.....	82	6.4 习题.....	139
3.6 验证控件.....	83	第 7 章 ASP.NET 常用对象.....	141
3.6.1 RequiredFieldValidator 控件.....	84	7.1 基本输出对象 Response.....	141
3.6.2 CompareValidator 控件.....	85	7.1.1 Response 对象的属性 和方法.....	141
3.6.3 RangeValidator 控件.....	87	7.1.2 输出字符串.....	143
3.6.4 RegularExpressionValidator 控件.....	88	7.1.3 输出文件.....	143
3.6.5 CustomValidator 控件.....	89	7.1.4 网页重定向.....	144
3.7 Rich 控件.....	92	7.2 基本输入对象 Request.....	146
3.7.1 Calendar 控件.....	93	7.2.1 Request 对象的属性.....	146
3.7.2 AdRotator 控件.....	98	7.2.2 获取浏览器信息.....	146
3.8 习题.....	101	7.2.3 获取 HTTP 中的信息.....	148
第 4 章 用户控件.....	104	7.3 Server 对象.....	149
4.1 用户控件概述.....	104	7.3.1 Server 对象的属性和方法.....	149
4.2 创建用户控件.....	105	7.3.2 利用 Server 对象进行 HTML 编码和解码.....	152
4.3 用户控件的使用.....	111	7.3.3 利用 Server 对象进行 URL 编码和解码.....	153
4.4 用户控件事件.....	114	7.4 Session 对象.....	155
4.5 习题.....	116	7.4.1 Session 对象的方法和事件.....	155
第 5 章 主题和母版页.....	118	7.4.2 Session 对象的唯一性和有效 时间.....	155
5.1 主题.....	118	7.5 Cookie 对象.....	160
5.1.1 主题概述.....	118	7.5.1 Cookie 对象的属性.....	160
5.1.2 主题创建.....	120	7.5.2 访问 Cookie.....	161
5.1.3 主题的应用.....	121	7.6 Application 对象.....	161
5.1.4 SkinID 的应用.....	123	7.6.1 如何使用 Application 对象.....	162
5.1.5 主题的禁用.....	124	7.6.2 同步 Application 状态.....	162
5.2 母版页.....	124	7.6.3 网站的访问计数.....	163
5.2.1 母版页概述.....	124		
5.2.2 创建母版页.....	126		
5.2.3 在母版页中放入网页的方法.....	129		
5.3 习题.....	131		

7.7 习题	164	9.4 DetailsView 控件	210
第 8 章 ADO.NET 数据库编程	167	9.4.1 DetailsView 控件的属性	210
8.1 ADO.NET 的基本对象	167	9.4.2 在 DetailsView 控件中显示数据	212
8.1.1 ADO.NET 简介	167	9.4.3 在 DetailsView 控件中操作数据	214
8.1.2 ADO.NET 组件结构	168	9.5 ListView 控件	216
8.2 连接数据库	169	9.5.1 ListView 控件的属性	216
8.2.1 建立 SQL Server 数据库	169	9.5.2 ListView 控件的方法	218
8.2.2 连接 SQL Server 数据库	171	9.5.3 为 ListView 控件创建模板	219
8.2.3 连接 Access 数据库	172	9.6 Chart 控件	222
8.3 读取数据	174	9.7 习题	225
8.3.1 使用 SqlCommand 类	174	第 10 章 XML 数据操作	228
8.3.2 使用 OleDbCommand 类	175	10.1 XML 概述	228
8.3.3 使用存储过程	177	10.1.1 XML 的语法	228
8.4 使用 DataReader	179	10.1.2 文档类型定义	230
8.5 填充数据集	182	10.1.3 可扩展样式语言	232
8.5.1 使用 DataAdapter	182	10.1.4 XPath	235
8.5.2 使用 DataTable、DataColumn 和 DataRow	185	10.2 .NET 中实现的 XML DOM	235
8.5.3 访问数据集	187	10.2.1 创建 XML 文档	237
8.6 习题	190	10.2.2 将 XML 读入文档	237
第 9 章 数据绑定和数据控件	194	10.2.3 创建新节点	238
9.1 数据绑定简介	194	10.2.4 修改 XML 文档	239
9.1.1 简单数据绑定和复杂数据 绑定	194	10.2.5 删除 XML 文档的节点、 属性和内容	239
9.1.2 用于简单数据绑定的控件	195	10.2.6 保存 XML 文档	240
9.2 数据源控件	196	10.2.7 使用 XPath 导航选择节点	240
9.2.1 SqlDataSource 控件概述	197	10.3 DataSet 与 XML	241
9.2.2 SqlDataSource 控件的属性	199	10.3.1 把 XML 数据读入 DataSet 对象	241
9.2.3 SqlDataSource 控件的功能	201	10.3.2 把 DataSet 写出为 XML 数据	242
9.2.4 使用 SqlDataSource 控件	202	10.4 XML 数据绑定	243
9.3 GridView 控件	203	10.5 习题	247
9.3.1 GridView 控件概述	203	第 11 章 LINQ 技术	249
9.3.2 在 GridView Web 服务器 控件中分页	205	11.1 LINQ 概述	249
9.3.3 对 GridView Web 服务器 控件中的数据进行排序	208		

11.2 基于 C# 的 LINQ	250	13.2 系统设计	292
11.2.1 LINQ 查询简介	251	13.2.1 系统模块的划分	292
11.2.2 基本查询操作	252	13.2.2 系统框架设计	295
11.3 LINQ 到 ADO.NET	254	13.2.3 系统程序结构设计	301
11.3.1 LINQ 到 SQL 基础	255	13.2.4 数据库设计	303
11.3.2 对象模型和对象模型的 创建	256	13.3 数据访问和存储层(DAL 层) 的实现	308
11.3.3 查询数据库	258	13.3.1 ADO.NET 数据访问 组件	308
11.3.4 更改数据库	261	13.3.2 LINQ 到 SQL 数据访问 组件	315
11.3.5 存储过程	265	13.4 业务逻辑层的实现	316
11.4 LinqDataSource 控件	268	13.4.1 Book 类	316
11.5 QueryExtender 控件	269	13.4.2 Category 类	319
11.6 习题	272	13.4.3 Comment 类	320
第 12 章 配置 ASP.NET 应用程序	274	13.4.4 Cart 类	323
12.1 使用 web.config 进行配置	274	13.4.5 Order 类	324
12.1.1 身份验证和授权	276	13.4.6 Folders 类和 Mails 类	324
12.1.2 在代码中获取 web.config 应用程序设置	277	13.4.7 User 类	325
12.2 使用 global.asax 进行配置	280	13.5 界面表示层的实现	327
12.2.1 编写 Application_Start 和 Application_End 事件 处理代码	281	13.5.1 书籍信息浏览功能	327
12.2.2 编写 Session_Start 和 Session_End 事件处理 代码	284	13.5.2 书籍评论功能	329
12.2.3 编写错误处理程序	286	13.5.3 购物车功能	331
12.3 习题	288	13.5.4 订单生成与修改功能	334
第 13 章 网络书店	291	13.5.5 站内邮件功能	336
13.1 功能分析	291	13.6 小结	337

第1章 ASP.NET 4.0开发基础

ASP.NET 是 Microsoft .NET Framework 中一套用于生成 Web 应用程序和 XML Web Services 的技术。ASP.NET 页面在服务器上执行并生成发送到桌面或移动浏览器的标记(如 HTML、WML 或 XML)。该页面使用一种已编译的、由事件驱动的编程模型,这种模型可以提高性能并支持将应用程序逻辑同用户界面相隔离。

本章重点:

- Web 和 ASP.NET 的基本概念
- ASP.NET 开发环境
- Visual Studio 2010 开发环境

1.1 ASP.NET 简介

ASP.NET 是微软公司为了迎接网络时代的来临,提出的一个统一的 Web 开发模型。ASP.NET 是建立在公共语言运行库上的编程框架,可用于在服务器上生成功能强大的 Web 应用程序。

1.1.1 .NET 简介

.NET 是微软公司发布的新一代的系统、服务和编程平台,主要由 .NET Framework 和 Microsoft Visual Studio .NET 开发工具组成。

.NET Framework 是一种新的计算平台,它包含了操作系统上软件开发的所有层,简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 主要包括两个最基本的内核,即公共语言运行库(Common Language Runtime, CLR)和 .NET Framework 基本类库,它们为 .NET 平台的实现提供了底层技术支持。下面将分别进行详细的介绍。

1. 公共语言运行库

公共语言运行库是 .NET Framework 的基础,是 .NET Framework 的运行时环境。公共语言运行库是一个在执行时管理代码的代理,以跨语言集成、自描述组件、简单配置和版本化及集成安全服务为特点,提供核心服务(如内存管理、线程管理和远程处理)。公共语言运行库还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。公共语言运行库遵循公共语言架构(简称 CLI)标准,使 C++、C#、Visual Basic 以及 JScript 等多种语言能够深度集成。在 .NET Framework 中,用一种语言所写的代码能继承用另一种

语言所写的类的实现，用一种语言所写的代码抛出的异常能被用另一种语言写的代码捕获。

2. .NET 基类库

.NET Framework 的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，如 ADO.NET、ASP.NET 等。.NET 基类库位于公共语言运行库的上层，与 .NET Framework 紧密集成在一起，可被 .NET 支持的任何语言所使用。这也就是为什么 ASP.NET 中可以使用 C#、VB.NET、VC.NET 等语言进行开发。.NET 类库非常丰富，提供数据库访问、XML、网络通信、线程、图形/图像、安全、加密等多种功能服务。类库中的基类提供了标准的功能，如输入/输出、字符串操作、安全管理、网络通信、线程管理、文本管理和用户界面设计功能。这些类库使得开发人员更容易地建立应用程序和网络服务，从而提高开发效率。

1.1.2 ASP.NET 页面与 Web 服务器的交互过程

ASP.NET 是一个统一的 Web 开发模型，它包括使用尽可能少的代码生成企业级 Web 应用程序所必需的各种服务。ASP.NET 作为 .NET Framework 的一部分提供。

ASP.NET 网页在任何浏览器或客户端设备中向用户提供信息，并使用服务器端代码来实现应用程序逻辑。使用 ASP.NET 网页可以为网站创建动态内容。通过使用静态 HTML 页(.htm 或.html 文件)，服务器读取文件并将该文件按原样发送到浏览器，以此来满足 Web 请求。相比之下，当用户请求 ASP.NET 网页(.aspx 文件)时，该页则作为程序在 Web 服务器上运行。该页运行时，可以执行网站要求的任何任务，包括计算值、读写数据库信息或者调用其他程序。该页动态地生成标记(HTML 或另一种标记语言中的元素)，并将该标记作为动态输出发送到浏览器。

ASP.NET 页面作为代码在服务器上运行。因此，要得到处理，页面必须在用户单击按钮(或者当用户选中复选框或与页面中的其他控件交互)时提交到服务器。每次页面都会提交回自身，以便它可以再次运行其服务器代码，然后向用户呈现其自身的新版本。传递 Web 页面的过程如下：

(1) 用户请求页面。使用 HTTP GET 方法请求页面，页面第一次运行，执行初步处理(如果已通过编程让它执行初步处理)。

(2) 页面将标记动态呈现到浏览器，用户看到的网页类似于其他任何网页。

(3) 用户输入信息或从可用选项中进行选择，然后单击按钮。如果用户单击链接而不是按钮，页面可能仅仅定位到另一页，而第一页不会被进一步处理。

(4) 页面发送到 Web 服务器。浏览器执行 HTTP POST 方法，该方法在 ASP.NET 中称为“回发”。更明确地说，页面发送回其自身。例如，如果用户正在使用 Default.aspx 页面，则单击该页上的某个按钮可以将该页发送回服务器，发送的目标则是 Default.aspx。

(5) 在 Web 服务器上，该页再次运行，并且可在页上使用用户输入或选择的信息。

(6) 页面执行通过编程所要实行的操作。

(7) 页面将其自身呈现回浏览器。

只要用户在该页面中工作,此循环就会继续。用户每次单击按钮时,页面中的信息会发送到 Web 服务器,然后该页面再次运行。每个循环称为一次“往返行程”。由于页面处理发生在 Web 服务器上,因此页面可以执行的每个操作都需要一次到服务器的往返行程。在下一节可以通过例子进一步了解这个过程。

此外,ASP.NET 网页是完全面向对象的。在 ASP.NET 网页中,可以使用属性、方法和事件来处理 HTML 元素。ASP.NET 页框架为响应在服务器上运行的代码中的客户端事件提供统一的模型,从而不必考虑基于 Web 的应用程序中固有的客户端和服务器隔离的实现细节。该框架还会在页面处理生命周期中自动维护页及该页上控件的状态。

使用 ASP.NET 页和控件框架还可以将常用的 UI 功能封装成易于使用且可重用的控件。控件只需编写一次,即可用于许多页面并集成到 ASP.NET 网页中。这些控件在呈现期间放入 ASP.NET 网页中。

1.2 Visual Studio 2010 集成开发环境

Visual Studio 2010 是一个功能强大的集成开发环境,在该开发环境中可以创建 Windows 应用程序、ASP.NET 应用程序、ASP.NET 服务和控制台程序等。

1.2.1 集成开发环境简介

打开 Visual Studio 2010 集成开发环境,如图 1-1 所示,可以看到界面主要由几个不同的部分组成。

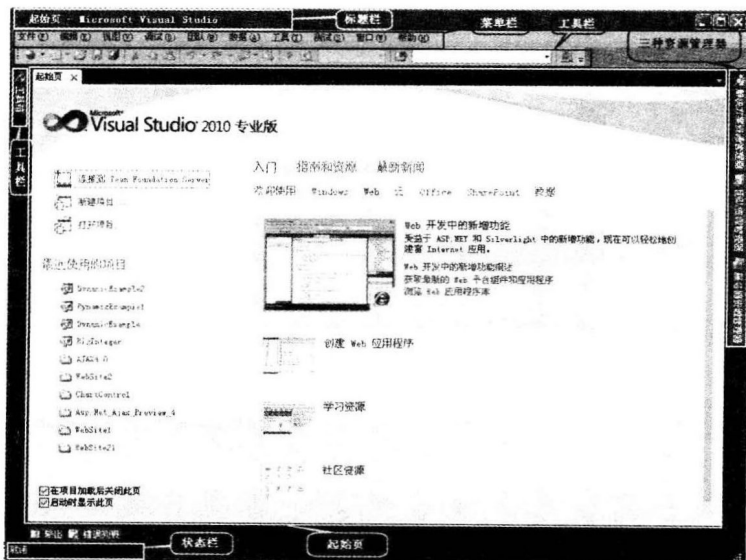


图 1-1 Visual Studio 2010 界面

Visual Studio 2010 的主界面各组成部分如下：

- 标题栏：位于主界面的顶部，用于显示页面的标题。
- 菜单栏：位于标题栏的下方，包含了实现软件所有功能的选项。
- 工具栏：位于菜单栏的下方，包含了软件常用功能的快捷按钮。
- 状态栏：位于界面的底部，用于显示软件的状态信息。
- 起始页：主界面中工具栏和状态栏之间的显示部分，占据了主界面的绝大部分位置。显示的内容包括：连接到 Team Foundation Server(团队服务器)、新建项目和打开项目的快捷按钮；最近使用的项目列表和 Visual Studio 2010 入门、指南和新闻列表的选项卡等。
- 工具箱：位于主界面的左侧边，提供了设计页面时常用的各种控件，只要简单地将控件拖动到设计页面即可方便地使用。
- 解决方案资源管理器：位于主界面的右侧边最上部，用于对解决方案和项目进行统一的管理，其主要组成是各种类型的文件目录。
- 团队资源管理器：位于解决方案资源管理器的下方，是一个简化的 Visual Studio Team System 2010 环境，专用于访问 Team Foundation Server 服务。
- 服务器资源管理器：位于团队资源管理器的下方，用于打开数据连接、登录服务器、浏览数据库和系统服务。

1.2.2 解决方案资源管理器

在 Visual Studio 2010 中，选择“视图”|“解决方案资源管理器”命令，就可以利用资源管理器对网站项目进行管理。通过资源管理器，可以浏览当前项目所包含的所有的资源(.aspx 文件、.aspx.cs 文件、图片等)，也可以向项目中添加新的资源，并且可以修改、复制和删除已经存在的资源。解决方案资源管理器如图 1-2 所示。



图 1-2 解决方案资源管理器

在添加一个 Web 页面后，可以使用 Visual Studio 对它进行编辑，在资源管理器中双击某个要编辑的 Web 页面文件，该页面文件就会在图 1-3 所示左边的窗口中打开。

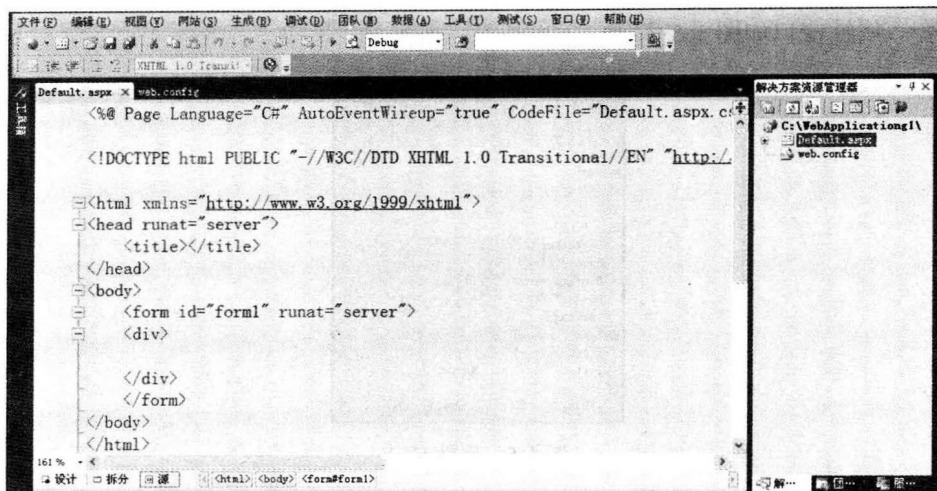


图 1-3 视图设计器

在图 1-3 页面文件编辑窗口可以通过底部的“设计”、“拆分”和“源”三个按钮来进行三种视图的编辑。其中，“设计”视图用来显示设计的效果；“拆分”视图同时显示“设计”视图和“源”视图；“源”视图显示设计源码，可以在该视图中直接通过编写代码来设计页面。

在 Web 页面的设计视图下，双击页面的任何地方即可打开隐藏的后台代码文件。在此界面中，开发者可以编写与页面对应的后台逻辑代码。通过双击网站目录下的文件名，也可以进入图 1-4 所示的后台隐藏的代码文件。

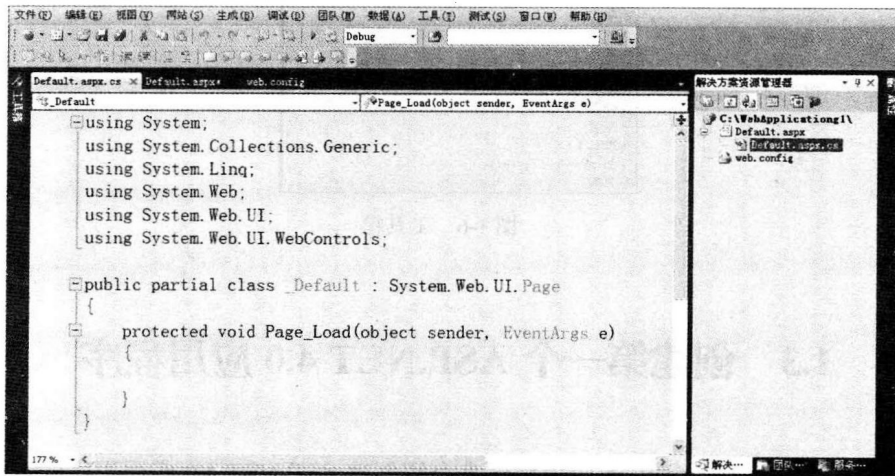


图 1-4 后台隐藏的代码文件

1.2.3 属性对话框

在进行页面设计时需要使用到“属性”对话框，在此对话框中，用户可以对页面的一些属性值进行设置，这些设置的属性值会自动添加到源代码中，属性值会随着标签值的改

变而改变。属性窗口如图 1-5 所示。

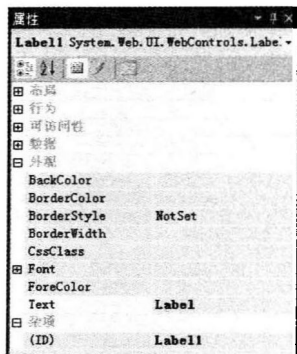


图 1-5 属性窗口

1.2.4 工具箱

在 Visual Studio.NET 的窗口左侧有一个隐藏的工具箱,当用户将鼠标放置在“工具箱”按钮上时会弹出一个“工具箱”对话框,如图 1-6 所示。在此对话框中列出了开发 ASP.NET Web 窗体的多种控件,用户可以直接使用这些控件,节省了编辑代码的时间,加快了程序开发的进度。

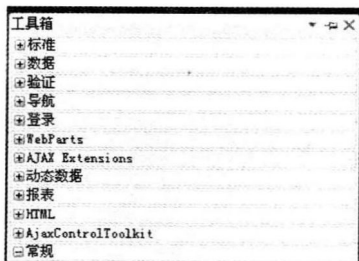


图 1-6 工具箱

1.3 创建第一个 ASP.NET 4.0 应用程序

下面通过本书的第一个 Web 应用程序来介绍创建 ASP.NET 4.0 应用程序的过程。本练习将实现在运行程序后,在浏览器中显示“欢迎进入 ASP.NET 的世界”。

1.3.1 创建 Web 站点

在使用 Microsoft Visual Studio 2010 开发 ASP.NET 4.0 应用程序时并不需要把应用程序存储到一定的虚拟目录中即可对应用程序进行运行和调试,但为了能够以站点的形式来浏

览创建好的应用程序,需要先在 IIS 管理器中创建一个 Myroot 虚拟目录,并把这个虚拟目录指定到一个文件夹,而本书创建的所有示例均将存储在这个文件夹中。

使用 Microsoft Visual Studio 2010 创建 Web 项目的方法有两种。

第一种方法是:选择“文件”|“新建项目”命令,打开“新建项目”对话框,如图 1-7 所示。在该对话框的“项目类型”列表框中选中 Web 类型;在“模板”列表框中将显示 Microsoft Visual Studio 2010 提供的 Web 程序模板。模板中的 ASP.NET Web 应用程序和 ASP.NET 空 Web 应用程序是用得最多的。其中要说明的是 ASP.NET 空应用程序模板创建后的目录下只有一个简单的网站配置文件 web.config;而 ASP.NET Web 应用程序和 Visual Studio 2008 中创建的 ASP.NET Web 应用程序有了很大的不同,它提供许多新增的内容,包括基础的身份验证功能、默认的网站项目母版页、默认的 css 样式文件 site.css、精简的 web.config 配置文件,只存放站点设置的数据信息,把大部分不需要在网站应用程序中使用的配置设置信息放在了 machine.config 文件中,以及 JQuery 的自动智能提示。



图 1-7 “新建项目”对话框

这里在其中选中“ASP.NET Web 应用程序”模板;在“名称”文本框中输入要创建的项目名称;在“位置”文本框中输入创建的项目存储的位置,或者通过“浏览”按钮进行路径选择。单击“确定”按钮即可创建一个 Web 项目。

第二种方法是:选择“文件”|“新建网站”命令,打开“新建网站”对话框,如图 1-8 所示。在该对话框中显示了可以创建的网站项目的模板,提供的模板有 ASP.NET 网站、ASP.NET 空网站、WCF 服务、ASP.NET Reports、ASP.NET Crystal Reports 等。模板中的 ASP.NET Web 应用程序和 ASP.NET 空 Web 应用程序是用得最多的。这里选择“ASP.NET 网站”模板;在“Web 位置”列表框中选择“文件系统”,在文本框中输入存储位置,也可以通过“浏览”按钮选择路径,并在路径后面输入要创建的项目的名称。这里也可以在“Web 位置”列表框中选择 HTTP 选项,然后输入要创建的网站名称 <http://localhost/Myroot/Sample1-1/>,这样就会在虚拟目录 Myroot 下创建一个新的网站项目。单击“确定”按钮即可创建一个 Web 项目。



图 1-8 “新建网站”对话框

这里利用上面所述的其中一种方法创建一个名为 Sample1-1 的 Web 项目，这时，Microsoft Visual Studio 2010 会自动在虚拟目录对应的文件夹里生成名为 Sample1-1 的文件夹和一个名为 Default.aspx 的新页面，以及其他一些文件和文件夹，如图 1-9 所示。Default.aspx 页面包括两个文件，一个是 Default.aspx.cs 文件，用于编写程序的后台代码；另一个是 Default.aspx.designer.cs，存放的是一些页面控件中控件的配置信息。

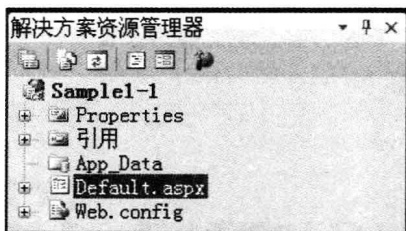


图 1-9 设计视图

1.3.2 编写 ASP.NET 4.0 应用程序

下面演示如何在该页面中添加文本。步骤如下：

- (1) 选择“设计”选项卡切换到“设计”视图。
- (2) 将插入点放在第一行，输入“欢迎进入 ASP.NET 的世界”。
- (3) 在屏幕右下角的“属性窗口”中可以设置所输入的文本的字体、颜色和字体大小等属性。这里将设置该网页的 title 属性为“你好”，其余采用默认属性。

此时该网页的代码如下所示：

```
1.< %@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
2.<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```