



高等学校**应用型特色**规划教材

ASP.NET程序设计与 应用开发 (第2版)

周永匡 沈国荣 马继军 主 编
汤小燕 周嫚嫚 副主编



赠送
电子教案

- 以基础理论—实用技术—实训为主线
- 用任务来驱动，按教与学的实际需要取材谋篇
- 每一章都精心设置“小型案例实训”
- 配备丰富的免费教学资源——电子教案与程序源代码



清华大学出版社

高等学校应用型特色规划教材

ASP.NET 程序设计与应用开发 (第2版)

周永臣 沈国荣 马继军 主 编
汤小燕 周嫚嫚 副主编

清华大学出版社
北 京

内 容 简 介

本书系统全面地介绍了 ASP.NET 的有关知识及其相关的数据库技术、HTML 技术等。全书共 9 章, 内容包括 ASP.NET 基础、教务管理系统分析、教务管理系统网站总体结构设计、用户登录及用户管理模块设计、基础数据管理模块设计、业务数据管理模块设计、业务处理模块设计、报表打印模块设计、ASP.NET 应用程序状态管理与配置。另外, 附录中还提供了 CSS、C# 语言以及正则表达式等相关知识。

本书配有电子教案, 并提供程序源代码, 以方便读者自学。

本书易教易学、学以致用、注重能力, 对初学者容易混淆的内容进行了重点提示和讲解。本书适合作为普通高等院校应用型本科(含部分专科、高职类)各相关专业的程序设计教材, 也适合编程开发人员培训、自学使用。

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 程序设计与应用开发/周永臣等主编. —2 版. —北京: 清华大学出版社, 2014
(高等学校应用型特色规划教材)

ISBN 978-7-302-35966-1

I. ①A… II. ①周… III. ①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2014)第 066041 号

责任编辑: 章忆文 郑期彤

封面设计: 杨玉兰

责任校对: 周剑云

责任印制: 刘海龙

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>, 010-62791865

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 17 字 数: 413 千字

版 次: 2009 年 1 月第 1 版 2014 年 5 月第 2 版 印 次: 2014 年 5 月第 1 次印刷

印 数: 1~3000

定 价: 34.00 元

前 言

作为企业级应用开发的两大主流技术体系之一，.NET 技术近年来发展异常迅速，越来越受到国内外 IT 企业的认可，在各行各业都得到了广泛的应用，已成为每一个计算机技术及相关专业毕业生必须掌握的基本技能之一。

本书以“项目化”教学为指导，以微软的 Microsoft Visual Studio 2012 为开发平台，通过典型的“教务管理系统”的开发过程，系统介绍 Web 开发过程，将面向对象思想、设计模式和软件架构的知识融入各章节的教学之中。教师可通过课堂教学，展示本书项目的实际开发与操作步骤，剖析技术要点，并组织学生共同学习和进行开发实践，培养学生的软件开发能力。

全书共分 9 章，主要内容如下。

第 1 章简要介绍 ASP.NET，并通过创建 ASP.NET 网站，讲述 ASP.NET 网站的开发和应用环境、Web 窗体、服务器控件、网站的运行和调试等基本内容。

第 2 章以教务管理系统为例，讲解利用面向对象思想对系统进行分析的过程，以及教务管理系统数据库的设计和实现方法，并对教务管理系统的体系结构和功能结构进行设计。

第 3 章以教务管理系统为例，讲解网站的总体结构设计，包括网站的总体布局设计及网站的总体布局实现(网站目录结构、母版页、CSS 和网站导航)，并讲解相关的服务器控件的使用方法。

第 4 章以教务管理系统为例，讲解网站用户登录和用户管理模块设计，主要包括用户登录和用户管理建模、用户管理三层模型类的实现，以及表示层界面的设计和相关控件的使用方法，最后完成登录设计、用户管理和角色管理。

第 5 章以教务管理系统的系部管理为例，讲解基础数据管理模块的设计，主要包括基础数据管理业务需求建模、系部管理类设计、相关控件讲解、界面设计过程以及数据的备份与恢复界面设计。

第 6 章以教务管理系统的教师管理为例，讲解业务数据管理模块的设计，主要包括业务数据管理业务需求建模、业务数据管理类设计、界面设计。

第 7 章以教务管理系统的培养方案排课管理等为例，讲解业务处理模块的设计，主要包括培养方案排课管理、选课管理、成绩管理等模块的设计及需求建模，相关控件的使用方法和界面设计。

第 8 章以教务管理系统的学生成绩打印为例，讲述报表设计的相关知识，并给出学生成绩打印的报表设计过程。

第 9 章以教务管理系统的在线人数统计为例，讲解状态管理与配置的相关知识，以显示在线人数、上次登录时间和登录次数为例，讲解状态管理的使用方法。



为方便教学,本书提供电子课件和程序源代码,如有需要,可到清华大学出版社网站下载。

本书由周永臣、沈国荣、马继军任主编,汤小燕、周嫚嫚任副主编。此外,参与本书编写工作的还有周跃进、黄兴荣、王琛、梁双华、李继良、董爱民、李志球等老师,全书由周永臣统稿。其他参与收集资料的人还有王珊珊、周海霞、卢振侠、何光明、石雅琴等。

由于编者水平有限,书中错误之处在所难免,欢迎读者提出宝贵意见和建议。联系邮箱: Book21Press@126.com。

编 者

目 录

第 1 章 ASP.NET 基础.....1	本章小结 36
1.1 ASP.NET 概述.....1	习题 36
1.1.1 ASP.NET 的发展.....1	
1.1.2 ASP.NET 的特点.....2	
1.1.3 .NET Framework.....3	
1.2 创建 ASP.NET 网站.....4	
1.2.1 ASP.NET 的运行环境.....4	
1.2.2 新建 ASP.NET 网站的过程.....4	
1.2.3 Web 窗体7	
1.2.4 Web 服务器控件12	
1.2.5 创建事件处理程序.....16	
1.2.6 运行和调试 ASP.NET 网页.....17	
1.3 小型案例实训：创建网站.....19	
本章小结20	
习题20	
第 2 章 教务管理系统分析.....21	
2.1 教务管理系统的业务需求分析.....21	
2.2 教务管理系统的参与者及用例.....22	
2.3 教务管理系统的数据库分析与设计.....23	
2.3.1 数据库的概念结构分析.....24	
2.3.2 数据库的逻辑结构设计.....25	
2.3.3 数据库的物理结构设计及实施.....26	
2.3.4 数据库代码设计.....28	
2.4 教务管理系统的功能模块结构分析.....31	
2.4.1 体系架构设计.....31	
2.4.2 系统功能模块结构设计.....33	
2.5 小型案例实训：PowerDesigner 数据库建模.....35	
第 3 章 教务管理系统网站总体结构设计..... 37	
3.1 总体结构布局设计 37	
3.2 总体结构布局实现 38	
3.2.1 网站目录结构实现 38	
3.2.2 网站布局实现 39	
3.2.3 系统导航设计 45	
3.3 相关服务器控件的使用方法 51	
3.3.1 Image 控件 51	
3.3.2 HyperLink 控件 51	
3.4 小型案例实训 52	
3.4.1 CSS 格式设计 52	
3.4.2 母版页设计 55	
3.4.3 导航控件的使用 56	
本章小结 58	
习题 59	
第 4 章 用户登录及用户管理模块设计..... 60	
4.1 用户登录需求建模 60	
4.2 用户登录实现 62	
4.2.1 启用成员资格和角色管理..... 62	
4.2.2 登录窗体设计 63	
4.2.3 登录后的其他设计 65	
4.3 修改密码页设计实现 66	
4.4 用户管理模块 67	
4.4.1 用户管理模块需求建模 67	
4.4.2 ADO.NET 69	
4.4.3 创建通用数据库操作类 75	

4.4.4 创建教务管理员实体类.....	79	5.3.5 ObjectDataSource 数据源 控件.....	140
4.4.5 创建教务管理员数据访问 层类.....	79	5.4 系部管理界面设计.....	141
4.4.6 创建教务管理员业务处理 层类.....	84	5.5 数据的备份与恢复界面设计.....	147
4.5 添加管理员表示层使用控件介绍.....	86	5.6 小型案例实训.....	152
4.5.1 RadioButton 和 RadioButtonList 控件.....	86	5.6.1 Gridview、DetailView 控件的使用.....	152
4.5.2 DropDownList 控件.....	87	5.6.2 验证控件的使用.....	154
4.5.3 MultiView 和 View 控件.....	87	本章小结.....	155
4.5.4 CreateUserWizard 控件.....	88	习题.....	155
4.5.5 SqlDataSource 数据源控件.....	91	第6章 业务数据管理模块设计	156
4.6 添加管理员表示层实现.....	93	6.1 业务数据管理业务需求建模.....	156
4.7 编辑管理员表示层使用控件介绍.....	97	6.2 教师管理类设计.....	158
4.7.1 CheckBox 和 CheckBoxList 控件.....	97	6.3 教师管理界面设计.....	159
4.7.2 GridView 控件.....	97	6.3.1 FormView 控件介绍.....	159
4.7.3 成员资格管理类.....	102	6.3.2 教师管理界面设计过程.....	161
4.7.4 角色管理类.....	104	6.4 小型案例实训.....	167
4.8 编辑管理员表示层设计.....	105	6.4.1 创建数据操作帮助类、 实体类.....	167
4.9 角色管理.....	113	6.4.2 数据访问层类实现.....	168
4.10 小型案例实训.....	116	本章小结.....	169
4.10.1 登录设计.....	116	习题.....	169
4.10.2 用户管理设计.....	118	第7章 业务处理模块设计	170
本章小结.....	118	7.1 培养方案排课管理模块设计.....	170
习题.....	119	7.1.1 培养方案排课管理模块 需求建模.....	170
第5章 基础数据管理模块设计	120	7.1.2 培养方案排课管理类设计.....	171
5.1 基础数据管理业务需求建模.....	120	7.1.3 培养方案排课管理界面 设计.....	172
5.2 系部管理类设计.....	122	7.2 选课管理模块设计.....	177
5.3 界面设计知识点.....	123	7.2.1 教师选课管理模块需求 建模.....	177
5.3.1 DetailsView 控件.....	123	7.2.2 教师选课管理类设计.....	179
5.3.2 TreeView 控件.....	125	7.2.3 教师选课管理界面设计.....	180
5.3.3 ListBox 控件.....	131	7.3 成绩管理模块设计.....	184
5.3.4 验证控件.....	133		

7.3.1 成绩管理模块需求建模.....	184	9.1.1 状态管理概述	218
7.3.2 成绩管理类设计	185	9.1.2 基于客户端的状态管理	222
7.3.3 成绩管理界面设计	186	9.1.3 基于服务器端的状态管理	229
7.4 小型案例实训	191	9.2 ASP.NET 应用程序配置	236
7.4.1 业务处理层类实现	191	9.2.1 Global.asax 文件	236
7.4.2 表示层实现	192	9.2.2 Web.config 文件	238
本章小结	196	9.3 在线人数统计及属性文件的使用	242
习题	197	9.3.1 在线人数统计实现	242
第 8 章 报表打印模块设计	198	9.3.2 显示上次访问时间 & 访问 次数的实现	245
8.1 学生成绩报表设计相关知识	198	9.4 小型案例实训：网站配置	246
8.1.1 ReportViewer 控件	198	本章小结	248
8.1.2 RDLC 报表设计	200	习题	248
8.2 成绩打印设计	209	附录 A CSS	249
本章小结	217	附录 B C#语言	252
习题	217	附录 C 正则表达式	260
第 9 章 ASP.NET 应用程序状态 管理与配置	218	参考文献	264
9.1 ASP.NET 状态管理	218		

第 1 章 ASP.NET 基础

本章要点

- .NET 的发展
- .NET 的体系结构
- ASP.NET 网站的组成

能力目标

- 了解.NET 的发展
- 熟悉.NET 体系结构
- 使用 Visual Studio 2012 创建并运行一个 ASP.NET 应用程序
- 掌握 Web 窗体的组成和常见属性、相关服务器控件的使用方法

ASP.NET 是微软主推的新技术，是微软.NET 战略中的一部分。ASP.NET 是运行于 Windows 平台.NET 框架下的一种新型的、功能强大的 Web 编程语言，一经推出就备受关注。发展至今，ASP.NET 经过十几年的改进和优化，已成为成熟、稳定的一种 Web 编程语言。

ASP.NET 是一个统一的 Web 开发模型，它作为.NET Framework 的一个组成部分提供。当创建 ASP.NET Web 应用程序时，可以使用 Visual Basic、C#、JScript 以及任何与公共语言运行库(CLR)兼容的语言来编写代码，同时能够充分利用公共语言运行库的各项功能，并对.NET Framework 中的类进行访问。

1.1 ASP.NET 概述

ASP.NET 提供了一种新的编程模型和结构，可以生成更安全、更稳定、可伸缩的 Web 应用程序。要理解什么是 ASP.NET，就必须先了解 ASP.NET 的发展以及几个基本概念。

1.1.1 ASP.NET 的发展

ASP(动态服务器页面)的第一个版本是 0.9 测试版，它能够将代码直接嵌入 HTML，使得设计 Web 页面变得更简单，并且能够通过内置的组件实现强大的功能。之后是 ASP 1.0，它作为 IIS 的附属产品免费发送，并且不久就在 Windows 平台上得到广泛应用。ASP 与 ADO(ActiveX 数据对象)的结合使开发者可以很容易地在数据库中建立和打开一个记录集。

1998 年，微软发布了 ASP 2.0。ASP 1.0 和 ASP 2.0 的主要区别是外部的组件是否需要实例化。有了 ASP 2.0 和 IIS 4.0，就可以开发 ASP 应用了，而且每个组件都有了自己单独的内存空间。接着，微软开发了 Windows 2000 操作系统，发布了 IIS 5.0 以及 ASP 3.0。此

次并不是简单地对 ASP 进行补充, ASP 3.0 与 ASP 2.0 核心的不同是把很多的事情交给了 COM(组件对象模型)来做。在 Windows 2000 中, 微软结合 MTS(微软事务服务器)与 COM 核心环境做出了 COM+, 这让主机有了一种新的方法来使用组件, 同时给主机带来了更多的稳定性, 使其成为一个可以升级的高效率工作平台。IIS 5.0 在表面上似乎没有什么改动, 但是在接口上有较大变化。在内部, 它使用 COM+组件服务来为组件提供一个更好的执行环境。

2000 年 ASP.NET 1.0 正式发布, 它不是 ASP 的简单升级, 而是微软推出的新一代 ASP。2003 年 ASP.NET 升级为 1.1 版本。ASP.NET 1.1 的发布更加激发了 Web 应用程序开发人员对 ASP.NET 的兴趣, 并且对网络技术有着巨大的推动作用。2005 年 11 月微软又发布了 ASP.NET 2.0。ASP.NET 2.0 的发布是 .NET 技术走向成熟的标志, 它在使用上增加了方便、实用的新特性, 具有高安全性、易管理性和高扩展性等特点, 使 Web 开发人员能够更加快捷、方便地开发 Web 应用程序, 不但执行效率大幅度提高, 对代码的控制也做得更好。2007 年微软推出了 ASP.NET 3.5 版本, 使网络程序开发更倾向于智能开发, 网络程序运行起来像 Windows 下的应用程序一样流畅。2010 年 ASP.NET 4.0 发布, 2012 年 ASP.NET 4.5 发布。

ASP.NET 实现了页面代码与应用程序的分离。简单地说, 一个 ASP.NET 应用程序其实是由两个文件.aspx 和.cs 构成的。其中, 页面代码通常放置 HTML 语言、HTML 控件和 Web 控件, 扩展名为.aspx; 而主程序代码则放在另一个扩展名为.cs 或.bs 的类文件中。另外, 开发者可以把所有相关的类代码编译成扩展名为.dll 的文件, 也就是把一些重用性高的、由用户自己开发的控件集成在一起, 一次编译成.dll 文件, 这个.dll 文件不能反编译回程序代码, 从而可以很有效地保护代码, 并且能用来开发商业控件。

1.1.2 ASP.NET 的特点

1. 执行效率大幅度提高

以前的 ASP 是即时解释程序, ASP.NET 是把基于通用语言的程序在服务器上运行, 即将程序在服务器端首次运行的时候进行编译, 这样的执行效果, 明显优于一条一条地解释程序。

2. 世界级的工具支持

ASP.NET 构架可以用微软公司的 Visual Studio .NET 开发环境进行开发, 这个开发工具最大的特点就是所见即所得, 功能强大的图像化开发界面能够有效地缩短开发时间。

3. 强大的适应性

由于 ASP.NET 是基于通用语言编译运行的程序, 它几乎可以运行在所有的平台上。通用语言的基本库、消息机制、数据接口的处理都能无缝地整合到 ASP.NET 的 Web 应用中。ASP.NET 同时也是语言无关性的, 所以, 用户可以选择一种自己最擅长的语言来编写程序, 或者用多种语言编写程序, 现在已经支持的有 C#(C++和 Java 的结合体)、Visual Basic、JScript。

4. 简单易学性

ASP.NET 使运行一些普通的任务(如表单的提交、客户端的身份验证、分布系统和网站配置等)变得非常简单。另外,通用语言简化了开发,使把代码组合成软件变得相当简单。

5. 可管理性

ASP.NET 使用一种基于字符的、分级的配置系统,使服务器环境和应用程序的设置更加简单。因为配置信息都保存在简单的文本中,新的设置不需要启动本地的管理工具就可以实现。

1.1.3 .NET Framework

.NET Framework 是支持生成和运行下一代应用程序和 XML Web 服务的内部 Windows 组件,旨在实现一种一致的面向对象的编程环境,使开发人员的经验在面对 Windows 应用程序和 Web 应用程序等类型不同的应用程序时保持一致。.NET 框架的体系结构如图 1-1 所示。

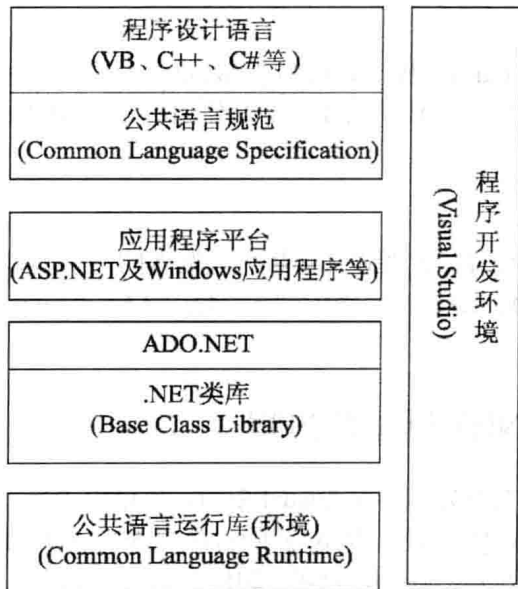


图 1-1 .NET 框架的体系结构

其中,公共语言运行库(CLR)和.NET 类库为两个主要组件。CLR 是.NET Framework 的基础,它是一种多语言执行环境,支持众多的数据类型和语言特性,管理着代码的执行。用户编写的程序不直接编译为 Windows 可以执行的二进制文件,而是编译为 CLR 能够执行的中间语言(Intermediate Language, IL 或 MSIL),然后由 CLR 来运行。

CLR 的执行模型如下。

(1) 将源代码(vb、cs)编译成托管代码块。托管代码块由中间语言和元数据组成,是在 CLR 的环境中执行的代码,脱离 CLR 环境则不能正常运行。托管代码生成的应用程序称为托管应用程序。



- (2) 将托管代码块合并成程序集。
- (3) 加载公共语言运行库。
- (4) 执行程序集的代码。
- (5) 生成本地代码。

.NET 类库是一个综合性的面向对象的可重用类型集合,使用该库可以开发多种应用程序,包括传统的命令行、图形界面应用程序、基于 ASP.NET 的 Web 应用程序和分布式 Web 服务。.NET 类库主要包括系统框架服务、ADO.NET 组件、XML 数据组件、Windows 表单组件、ASP.NET 应用服务、ASP.NET Web 表单和 XML Web 服务。

1.2 创建 ASP.NET 网站

本节介绍 ASP.NET 的运行环境、Visual Studio 2012 的界面,以及新建一个网站的方法,关于 Visual Studio 2012 的安装条件和过程在此不做介绍。

1.2.1 ASP.NET 的运行环境

1. 操作系统支持

虽然 ASP.NET 在 Windows 操作系统和非 Windows 操作系统中都能运行,但我们推荐在 Windows 操作系统中运行 ASP.NET。本书使用的是 ASP.NET 4.5,其需要 Windows Vista SP2 以上的操作系统。

2. .NET 环境

要想正常运行 ASP.NET,就要在计算机中安装 .NET 运行环境,即 .NET Framework。目前其版本较多,有 1.0、1.1、2.0、3.0、3.5、4.0、4.5,均可以从微软的网站上下载。本书使用的版本为 4.5。

1.2.2 新建 ASP.NET 网站的过程

当前 ASP.NET 程序的开发大多在 Visual Studio 环境中进行,Visual Studio 是一套完整的开发工具集,用于生成 ASP.NET Web 应用程序、XML Web 服务、桌面应用程序和移动应用程序。本书使用的版本是 Visual Studio 2012。Visual Basic、Visual C++、Visual C# 等全部使用相同的集成开发环境(IDE)。

使用 Visual Studio 2012 可以创建两种类型的 ASP.NET 网站:本地 IIS 网站和文件系统网站。本地 IIS 网站是计算机上的 IIS Web 应用程序,要求计算机必须安装并运行 IIS 5.0 或更高版本。文件系统网站可以通过 Visual Studio 2012 内置的开发服务器来测试,而不需要在计算机上安装 IIS,这对于在计算机上进行本地开发十分有用。

启动 Visual Studio 2012 的方法是,在桌面上选择“开始”→“程序”→Microsoft Visual Studio 2012→Microsoft Visual Studio 2012 命令。

启动 Visual Studio 2012 后,在菜单栏中选择“文件”→“新建”→“网站”命令,弹出如图 1-2 所示的“新建网站”对话框。

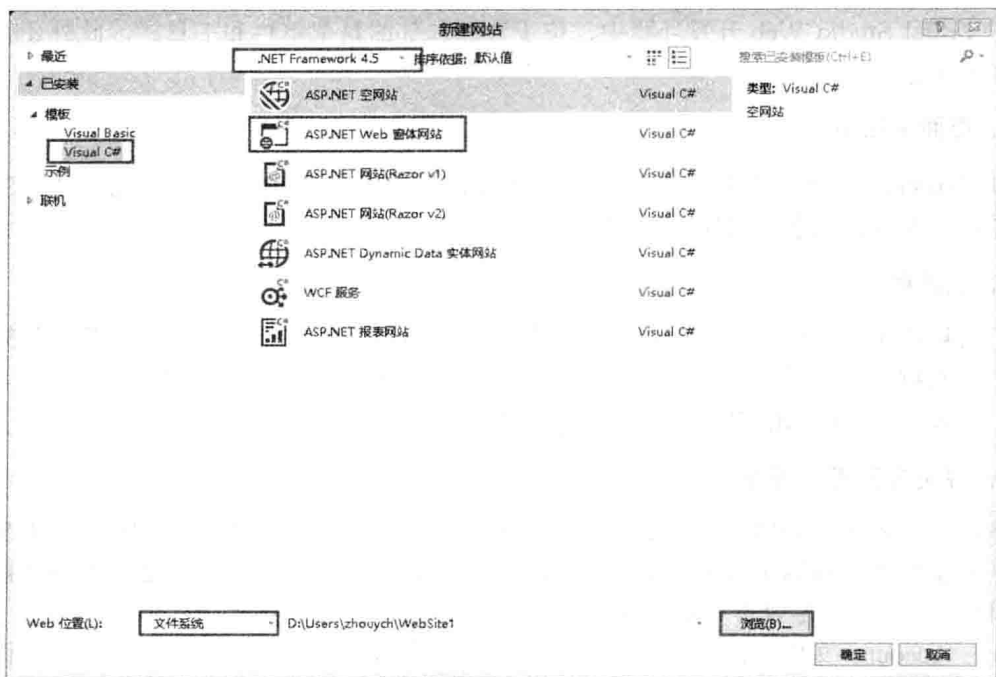


图 1-2 “新建网站”对话框

在对话框中依次选择 Visual C#、.NET Framework 4.5、“ASP.NET Web 窗体网站”选项，然后在“Web 位置”下拉列表框中选择“文件系统”选项，再单击“浏览”按钮，确定网站的位置，如图 1-2 所示。设置完成后单击“确定”按钮，这样就创建了一个基于 Visual C# 的新网站。网站创建完成以后，Visual Studio Web 开发环境界面如图 1-3 所示。

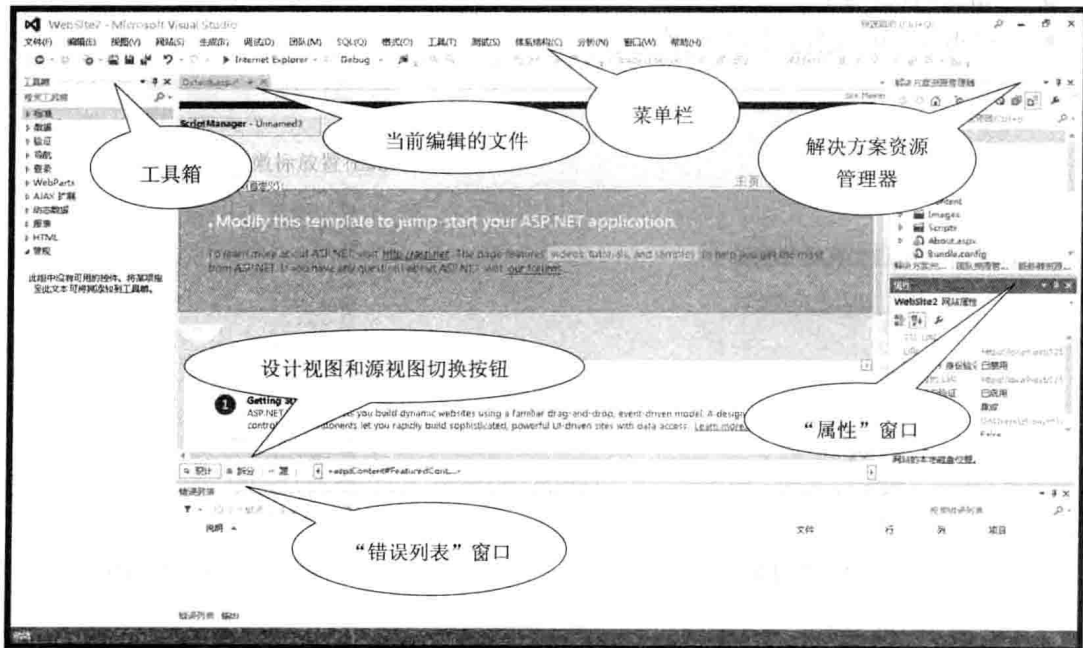


图 1-3 Visual Studio Web 开发环境界面

在 Visual Studio Web 开发环境中, 位于界面顶部的是菜单栏和工具栏, 此外还有以下几个主要部分。

1. 页面编辑窗口

页面编辑窗口位于界面的中间区域, 能以设计视图、源视图或拆分视图显示当前文档, 并可以在各视图之间切换。

2. 工具箱

工具箱提供建立应用程序时可用的控件, 默认情况下, 位于界面左侧, 其中包括“标准”、“数据”、“验证”、“导航”、“登录”、WebParts、“AJAX 扩展”、“动态数据”“报表”、HTML 和“常规”等选项卡, 如图 1-4 所示。

3. 解决方案资源管理器

解决方案资源管理器位于界面右上方, 其中显示当前站点中的所有文件。图 1-5 所示的解决方案资源管理器中显示的文件是 Visual Studio 2012 自动创建的构建一个站点所需要的文件(夹), 其中包括以下主要文件夹。

- Account 文件夹: 账户管理文件夹, 含有系统自动生成的注册、登录、修改密码等窗体。
- App_Data 文件夹: 系统数据文件夹。
- Scripts 文件夹: 含有 JScript 文件的文件夹。
- Content 文件夹: 含有网站样式的 CSS 文件夹。
- About.aspx、Default.aspx 文件: Web 窗体文件。
- Global.asax 文件: 全局设置文件。
- Site.master 文件: 网站母版。
- Web.config 文件: 网站配置文件。



图 1-4 工具箱



图 1-5 解决方案资源管理器

4. “属性”窗口

在设计页面时, 可以通过“属性”窗口查看和修改当前选中对象的属性。除了可以方

便地管理对象的属性外,“属性”窗口还具有事件的管理功能,可以方便地管理窗口及其控件的事件。

5. “错误列表”窗口

“错误列表”窗口显示在页面编辑窗口下方,用于显示代码中的错误。单击错误列表中的错误项,可以定位到页面编辑窗口中的错误代码处。

1.2.3 Web 窗体

Web 窗体(Web Form)及其架构是基于.NET 通用运行环境的可扩展编程模型,类似于C#程序设计中的 Windows 窗体,主要用来生成与用户交互的界面。用户在 Visual Studio 环境中,可以用熟悉的 Windows 窗体应用程序的设计方式来设计 Web 页面,这极大地提高了 Web 应用程序的开发效率。Web 窗体采用事件驱动、面向对象的模式编写程序。它是一个容器对象,不仅有自己的属性、方法、事件,而且能容纳 HTML 控件、Web 服务器控件等对象。

Web 窗体模型由两部分组成:用户界面(UI)和实现逻辑(或称为代码)。用户界面含有页面布局信息和 ASP.NET 服务器控件的 HTML 模板,它负责对浏览器上的 Web 窗体进行表示;实现逻辑部分是对 Web 窗体进行逻辑处理的 ASP.NET 程序代码,负责生成 Web 窗体上动态显示的内容。逻辑部分和用户界面生成的 HTML 协同工作,可以实现完全动态的 Web 页面。

ASP.NET 很容易做到用户界面和内容的分离,因为 Web 窗体模型是基于 Web 用户界面和实现逻辑分离技术实现的。ASP.NET 可以通过代码内联和后台编码两种形式实现用户界面和实现逻辑的联系。代码内联是将用户界面和实现逻辑两部分放在同一个文件中;后台编码是将用户界面和实现逻辑分别存在两个不同的文件中。

1. Web 窗体的组成文件

在 1.2.2 小节创建的 WebSite 网站中,自动创建了 Default.aspx 窗体,这个窗体由两个文件组成:包含 HTML 内容的用户界面文件(扩展名为.aspx)和后台代码文件(扩展名为.aspx.cs)。Visual Studio 默认将窗体文件分为扩展名为.aspx 和扩展名为.aspx.cs 的文件,这样把界面和逻辑(代码)分开,方便应用程序的维护。

当页面被请求时,服务器将联合 Default.aspx 和 Default.aspx.cs 两个文件生成 Default 类的一个实例并执行,类代码的执行结果是将 HTML 代码传回客户端,浏览器负责将 HTML 代码生成用户看到的页面。

在图 1-3 所示的界面中,选择“网站”→“添加新项”命令,弹出如图 1-6 所示的“添加新项”对话框,选择“Web 窗体”选项,并选中右下方的“将代码放在单独的文件中”复选框,然后单击“添加”按钮,添加一个新的名为“Default2.aspx”的 Web 窗体。在解决方案资源管理器中,双击此窗体文件将其打开,再单击页面编辑窗口下方的“源”按钮,即可切换到源视图,查看 Web 窗体界面的源代码。Default2.aspx 文件的内容如下:



```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
</div>
</form>
</body>
</html>
```

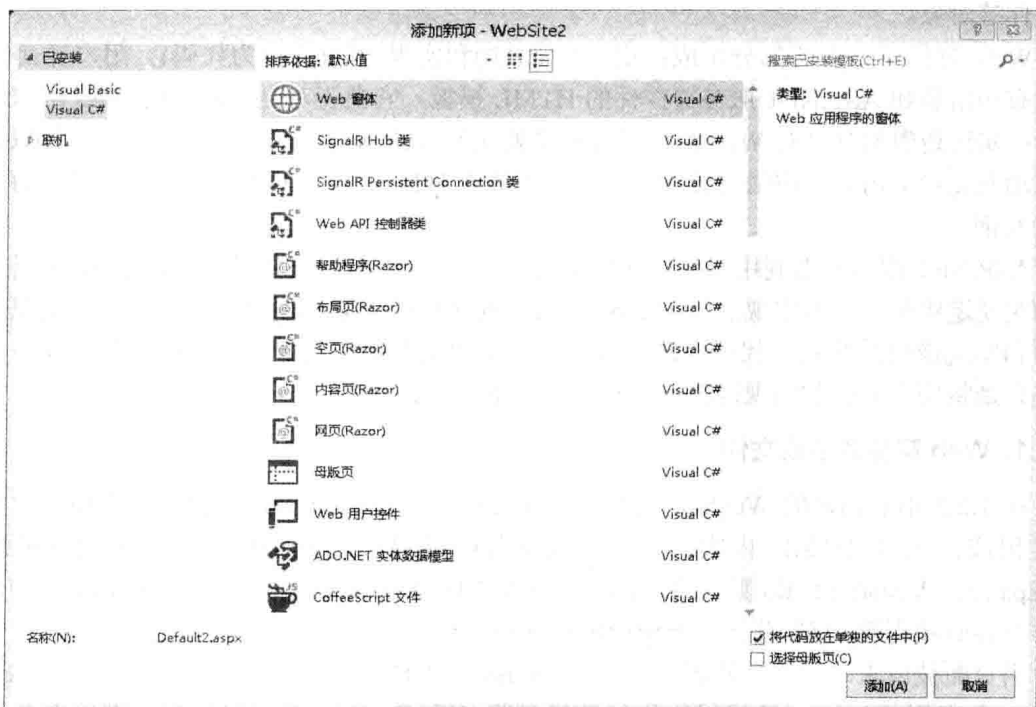


图 1-6 “添加新项”对话框

1) ASP.NET 窗体的.aspx 文件组成

(1) 可选的@Page 指令或其他指令, 适合正在创建的页面类型。

Default 2.aspx 文件中第一行的@Page 指令用于定义 ASP.NET 窗体页分析器和编译器使用的页文件的特定属性。

- Language 属性: 指定页面所使用的语言, 常用的有 VB 和 C#, 本网页为 C#, 每页只能使用和指定一种语言。
- AutoEventWireup 属性: 指定页的事件是否自动绑定, 若启用了事件绑定, 则为

true, 否则为 false。

- **CodeFile** 属性: 指定指向该页面后台代码文件的路径, 此处为"Default2.aspx.cs", 它是自动生成的文件。
- **Inherits** 属性: 定义该页继承的隐藏代码类, 可以是 **Page** 类派生的任何类, 此处该页继承于 **Default2** 类。

(2) HTML 标记。

在 **@Page** 指令下方是一些标准的 HTML 文档标记。在 **<body>** 与 **</body>** 之间包含一对 **<form></form>** 标记, 在 **<form>** 与 **</form>** 标记之间包含一对 **<div></div>** 标记。

这对 **<form></form>** 标记用于创建一个服务器端 **HtmlForm** 控件, 在网页上添加的文本和 HTML 元素以及通过工具箱添加的 Web 服务器控件, 均位于 **<div></div>** 标记之间。



注意: ① 页面只能包含一个 form 元素。

② form 元素必须包含 **runat** 属性, 其值为 **server**。

③ 可执行回发的 Web 服务器控件必须位于 form 元素内。

2) ASP.NET 窗体的.aspx.cs 文件组成

查看 **Default2.aspx.cs** 文件, 其内容如下。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

(1) 在文件的开头部分, 通过 **using** 语句引入一些经常用到的命名空间。

(2) **public partial class Default2 : System.Web.UI.Page** 定义了一个继承自 **Page** 类的页面类 **Default2**。

(3) 这个类中添加了一些事件处理过程, 如 **Page_Load**、**Button1_Click** 等, 默认情况下系统只给出了 Web 窗体的加载事件 **Page_Load** 的模板。

2. Web 窗体的处理过程及事件

ASP.NET 基于 Web 窗体的事件驱动交互模型, 用户可以用熟悉的 C/S 模式的 Windows 应用程序设计方法, 采用面向对象的技术来实现 Web 世界的 B/S 模式的应用程序。要学习 Web 窗体的事件驱动编程, 必须掌握窗体的处理过程及各阶段发生的事件。通过前面的学习, 已经知道 Web 窗体的扩展名是 **.aspx**, 当浏览器第一次请求一个 **.aspx** 文件时, Web 窗体将被 CLR 编译器编译成一个 **Page** 类并实例化, 以后每次请求这个页面时, 这个动态创建的 **Page** 类就会实例化, 从而得到一个可输出的 HTML 页面的 **Page** 对象。这