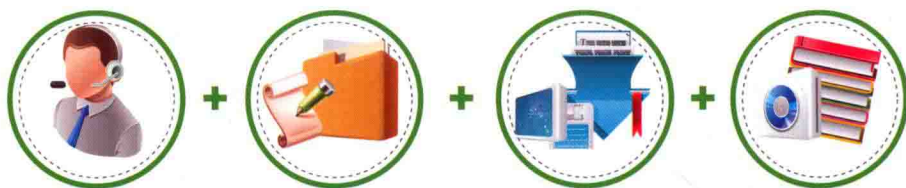


# ASP.NET

## 应用开发实例教程

卢守东 编著



- ◆ 以基础理论—实用技术—实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置“项目实训”，培养学生的实践能力
- ◆ 配备免费教学资源——电子课件及源代码



全国高等院校应用型创新规划教材 计算机系列

# ASP.NET 应用开发实例教程

卢守东 编著

清华大学出版社  
北京

## 内 容 简 介

本书以应用为导向,以实用为原则,以能力提升为目标,以典型实例与完整案例为依托,遵循程序设计与案例教学的基本思想,全面介绍基于 ASP.NET 的 Web 应用开发的主要技术。全书共分为 8 章,包括 ASP.NET 概述、C#编程基础、ASP.NET 服务器控件、ASP.NET 内置对象、SQL Server 数据库应用基础、ADO.NET 数据库访问技术、ASP.NET AJAX 编程技术与 ASP.NET 应用案例,并附有相应的思考题与实验指导。

本书内容全面,面向应用,示例翔实,解析到位,编排合理,结构清晰,循序渐进,准确严谨,注重应用开发能力的培养,可作为各高校本科或高职高专计算机、电子商务、信息管理与信息系统及相关专业 ASP.NET 程序设计、Web 程序设计、动态网站开发等课程的教材或教学参考书,也可作为 ASP.NET 程序设计人员的技术参考书以及初学者的自学教程。

本书所有示例的代码均已通过调试,并能成功运行,其开发环境为 Windows 7、Visual Studio 2010 与 SQL Server 2008。

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 应用开发实例教程/卢守东编著. —北京:清华大学出版社, 2019

(全国高等院校应用型创新规划教材 计算机系列)

ISBN 978-7-302-52506-6

I. ①A… II. ①卢… III. ①网页制作工具—程序设计—高等学校—教材 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 043706 号

责任编辑:孟 攀

封面设计:杨玉兰

责任校对:王明明

责任印制:李红英

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62791865

印 装 者:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.75 字 数:480 千字

版 次:2019 年 4 月第 1 版 印 次:2019 年 4 月第 1 次印刷

定 价:59.00 元

产品编号:079949-01

# 前 言

ASP.NET 是 Microsoft 公司推出的新一代 Web 应用程序或动态网站开发技术,也是目前 Web 应用开发领域的主流技术之一,其实际应用亦相当广泛。为满足社会不断发展的实际需求,并提高学生的专业技能与就业能力,多数高校的计算机、电子商务、信息管理与信息系统等相关专业均开设了 ASP.NET 程序设计、ASP.NET 开发技术等 ASP.NET 应用开发类课程。

为满足 ASP.NET 教学的实际需要,作者结合自身多年的教学实践经验,以所写讲义为基础编成本书,以应用为导向,以实用为原则,以能力提升为目标,以典型实例与完整案例为依托,遵循程序设计与案例教学的基本思想,结合教学规律与开发需求,按照由浅入深、循序渐进的原则,全面介绍基于 ASP.NET 的 Web 应用开发的主要技术。

本书共包括 8 章。第 1 章为 ASP.NET 概述,主要介绍 ASP.NET 的概况,讲解 ASP.NET 应用开发工具的基本用法与运行环境的搭建方法,并通过具体实例说明 ASP.NET 应用程序的创建与部署方法;第 2 章为 C#编程基础,主要介绍 C#的概况,讲解 C#的基本语法,并通过具体实例说明 C#基本语句的使用方法、类与对象的基本用法以及命名空间的定义与引用方法,同时介绍 C#中常用系统类的有关属性与主要方法;第 3 章为 ASP.NET 服务器控件,主要介绍 ASP.NET 服务器控件的概况,并通过具体实例讲解各类 ASP.NET 标准控件的主要用法、各种 ASP.NET 验证控件的基本用法以及 ASP.NET 用户控件的创建与使用方法;第 4 章为 ASP.NET 内置对象,主要介绍 ASP.NET 内置对象的概况,并通过具体实例讲解各种 ASP.NET 内置对象的基本用法;第 5 章为 SQL Server 数据库应用基础,主要介绍 SQL Server 的概况,并通过具体实例讲解 SQL Server 的安装与设置方法、SQL Server 数据库管理的基本技术以及常用 SQL 语句的基本用法;第 6 章为 ADO.NET 数据库访问技术,主要介绍 ADO.NET 的概况,并通过具体实例讲解 ADO.NET 常用对象的主要用法、常用服务器端数据访问控件的基本用法以及 DataSet 的典型应用模式;第 7 章为 ASP.NET AJAX 编程技术,主要介绍 AJAX 技术的概况、ASP.NET AJAX 的技术框架与 ASP.NET AJAX Extensions 的使用要点,并通过具体实例讲解各种 ASP.NET AJAX 服务器端控件的基本用法;第 8 章为 ASP.NET 应用案例,主要以一个简单的人事管理系统为例,分析系统的基本需求与用户类型,完成系统的功能模块设计与数据库结构设计,并采用 ASP.NET+SQL Server 模式加以实现。全书各章均有“本章要点”“学习目标”与“本章小结”,既便于抓住重点、明确目标,也利于“温故而知新”。书中的诸多内容均设有相应的“说明”“提示”“注意”等知识点,以便于读者的理解与提高。此外,各章均安排有相应的思考题,以利于读者的及时回顾与检测。书末还附有相应的实验指导,以便于读者的上机实践。综观全书,设计精心、结构清晰、编排合理、示例翔实、解析到位,集系统性与条理性于一身,融实用性与技巧性于一体,颇具特色。

本书以适度与实用为原则,内容全面,面向应用,语言流畅,准确严谨,通俗易懂,贴近实际应用开发的技术需求,注重具体应用开发的能力培养,可充分满足课程教学的实



际需要,适合各个层面、各种水平的读者,既可作为各高校本科或高职高专计算机、电子商务、信息管理与信息系统及相关专业 ASP.NET 程序设计、Web 程序设计、动态网站开发等课程的教材或教学参考书,也可作为 ASP.NET 程序设计人员的技术参考书以及初学者的自学教程。

本书所有示例的代码均已通过调试,并能成功运行,其开发环境为 Windows 7、Visual Studio 2010 与 SQL Server 2008。作为参考,也为便于教学,本书提供了相应的教学资源(包括教学大纲、PPT 课件以及完整的示例源码等),可从出版社的官网免费下载。

本书的写作与出版,得到了作者所在单位及清华大学出版社的大力支持与帮助,在此表示衷心感谢。在紧张的写作过程中,自始至终也得到了家人、同事的理解与支持,在此也一起深表谢意。

由于作者经验不足、水平有限,书中不妥之处在所难免,恳请广大读者多加指正、不吝赐教。

编 者

# 目 录

|                               |    |                                           |    |
|-------------------------------|----|-------------------------------------------|----|
| 第 1 章 ASP.NET 概述.....         | 1  | 2.5.2 命名空间的定义 .....                       | 43 |
| 1.1 ASP.NET 简介 .....          | 2  | 2.6 常用系统类 .....                           | 43 |
| 1.2 ASP.NET 应用的开发工具 .....     | 2  | 2.6.1 DateTime 类 .....                    | 43 |
| 1.2.1 Visual Studio 简介 .....  | 2  | 2.6.2 Math 类 .....                        | 44 |
| 1.2.2 Visual Studio 的安装 ..... | 3  | 2.6.3 Random 类 .....                      | 44 |
| 1.2.3 Visual Studio 的使用 ..... | 3  | 2.6.4 String 类 .....                      | 45 |
| 1.3 ASP.NET 应用的运行环境 .....     | 8  | 2.7 程序设计实例 .....                          | 45 |
| 1.3.1 IIS 的安装 .....           | 8  | 本章小结 .....                                | 47 |
| 1.3.2 .NET 框架的安装 .....        | 9  | 思考题 .....                                 | 47 |
| 1.4 ASP.NET 应用程序的创建 .....     | 10 | 第 3 章 ASP.NET 服务器控件 .....                 | 49 |
| 1.5 ASP.NET 应用程序的部署 .....     | 13 | 3.1 服务器控件简介 .....                         | 50 |
| 本章小结 .....                    | 14 | 3.1.1 服务器控件的分类 .....                      | 50 |
| 思考题 .....                     | 14 | 3.1.2 服务器控件的添加与删除 .....                   | 51 |
| 第 2 章 C#编程基础 .....            | 15 | 3.1.3 服务器控件的属性、方法与事件 .....                | 51 |
| 2.1 C#简介 .....                | 16 | 3.2 标准控件 .....                            | 55 |
| 2.2 语法基础 .....                | 16 | 3.2.1 标签控件 .....                          | 55 |
| 2.2.1 数据类型 .....              | 16 | 3.2.2 文本框控件 .....                         | 55 |
| 2.2.2 常量 .....                | 18 | 3.2.3 按钮类控件 .....                         | 57 |
| 2.2.3 变量 .....                | 19 | 3.2.4 选择类控件 .....                         | 61 |
| 2.2.4 类型转换 .....              | 20 | 3.2.5 图形类控件 .....                         | 75 |
| 2.2.5 运算符与表达式 .....           | 22 | 3.2.6 链接类控件 .....                         | 79 |
| 2.2.6 数组 .....                | 24 | 3.2.7 日历控件 .....                          | 80 |
| 2.3 基本语句 .....                | 27 | 3.2.8 文件上传控件 .....                        | 82 |
| 2.3.1 分支语句 .....              | 27 | 3.2.9 表格控件 .....                          | 85 |
| 2.3.2 循环语句 .....              | 30 | 3.2.10 容器控件 .....                         | 87 |
| 2.3.3 跳转语句 .....              | 33 | 3.3 验证控件 .....                            | 89 |
| 2.3.4 异常处理语句 .....            | 35 | 3.3.1 RequireFieldValidator 控件 .....      | 89 |
| 2.4 类与对象 .....                | 37 | 3.3.2 RangeValidator 控件 .....             | 90 |
| 2.4.1 类的声明 .....              | 37 | 3.3.3 CompareValidator 控件 .....           | 90 |
| 2.4.2 类的成员 .....              | 38 | 3.3.4 RegularExpressionValidator 控件 ..... | 91 |
| 2.4.3 对象的创建与使用 .....          | 40 | 3.3.5 CustomValidator 控件 .....            | 91 |
| 2.5 命名空间 .....                | 42 | 3.3.6 ValidationSummary 控件 .....          | 92 |
| 2.5.1 命名空间的引用 .....           | 43 |                                           |    |

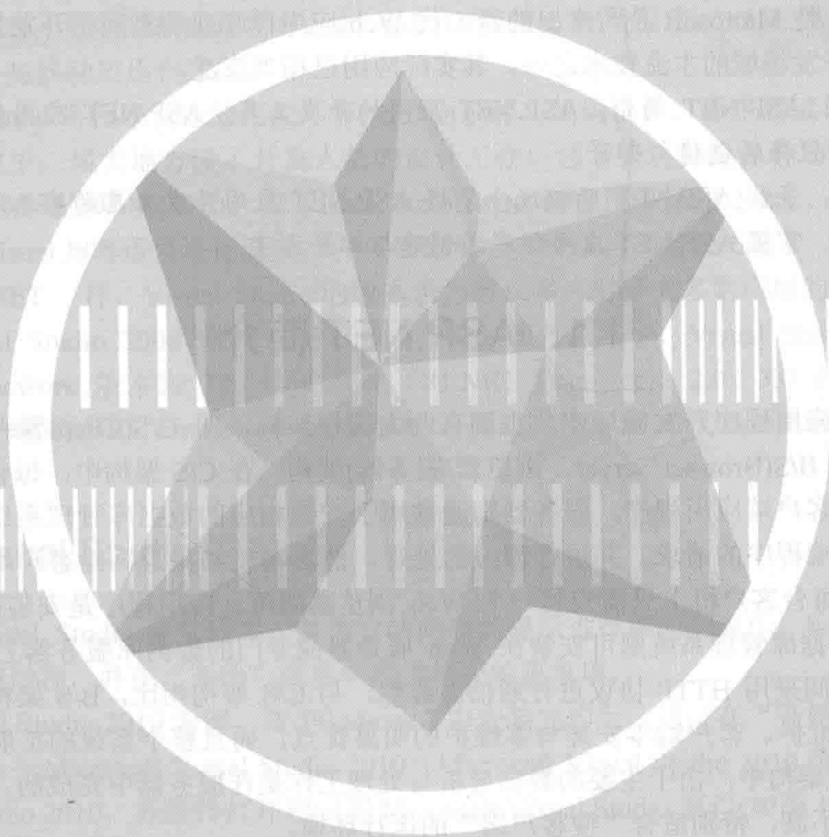
|       |                          |     |
|-------|--------------------------|-----|
| 3.4   | 用户控件.....                | 97  |
| 3.4.1 | 用户控件的创建.....             | 97  |
| 3.4.2 | 用户控件的添加.....             | 100 |
| 3.4.3 | 构成控件的属性访问.....           | 102 |
|       | 本章小结.....                | 103 |
|       | 思考题.....                 | 103 |
| 第4章   | ASP.NET 内置对象.....        | 105 |
| 4.1   | 内置对象简介.....              | 106 |
| 4.2   | Page 对象.....             | 106 |
| 4.2.1 | Page 对象的常用属性.....        | 106 |
| 4.2.2 | Page 对象的常用方法.....        | 106 |
| 4.2.3 | Page 对象的常用事件.....        | 107 |
| 4.2.4 | Page 对象的应用实例.....        | 107 |
| 4.3   | Response 对象.....         | 108 |
| 4.3.1 | Response 对象的常用属性.....    | 108 |
| 4.3.2 | Response 对象的常用集合.....    | 109 |
| 4.3.3 | Response 对象的常用方法.....    | 109 |
| 4.3.4 | Response 对象的应用实例.....    | 109 |
| 4.4   | Request 对象.....          | 114 |
| 4.4.1 | Request 对象的常用属性.....     | 114 |
| 4.4.2 | Request 对象的常用集合.....     | 116 |
| 4.4.3 | Request 对象的常用方法.....     | 122 |
| 4.4.4 | Request 对象的应用实例.....     | 122 |
| 4.5   | Application 对象.....      | 123 |
| 4.5.1 | Application 对象的常用集合..... | 124 |
| 4.5.2 | Application 对象的常用属性..... | 125 |
| 4.5.3 | Application 对象的常用方法..... | 126 |
| 4.5.4 | Application 对象的常用事件..... | 128 |
| 4.5.5 | Application 对象的应用实例..... | 129 |
| 4.6   | Session 对象.....          | 131 |
| 4.6.1 | Session 对象的常用集合.....     | 131 |
| 4.6.2 | Session 对象的常用属性.....     | 133 |
| 4.6.3 | Session 对象的常用方法.....     | 134 |

|       |                      |     |
|-------|----------------------|-----|
| 4.6.4 | Session 对象的常用事件..... | 136 |
| 4.6.5 | Session 对象的应用实例..... | 138 |
| 4.7   | Server 对象.....       | 140 |
| 4.7.1 | Server 对象的常用属性.....  | 140 |
| 4.7.2 | Server 对象的常用方法.....  | 140 |
| 4.7.3 | Server 对象的应用实例.....  | 141 |
|       | 本章小结.....            | 145 |
|       | 思考题.....             | 145 |

|       |                         |     |
|-------|-------------------------|-----|
| 第5章   | SQL Server 数据库应用基础..... | 147 |
| 5.1   | SQL Server 简介.....      | 148 |
| 5.2   | SQL Server 的安装与设置.....  | 148 |
| 5.2.1 | SQL Server 的安装.....     | 148 |
| 5.2.2 | SQL Server 的设置.....     | 148 |
| 5.3   | SQL Server 的数据库管理.....  | 153 |
| 5.3.1 | 数据库的基本操作.....           | 154 |
| 5.3.2 | 表的基本操作.....             | 157 |
| 5.4   | 常用的 SQL 语句.....         | 161 |
| 5.4.1 | SQL 语句的编写与执行.....       | 162 |
| 5.4.2 | 插入(INSERT)语句.....       | 163 |
| 5.4.3 | 更新(UPDATE)语句.....       | 163 |
| 5.4.4 | 删除(DELETE)语句.....       | 163 |
| 5.4.5 | 查询(SELECT)语句.....       | 164 |
|       | 本章小结.....               | 168 |
|       | 思考题.....                | 168 |

|       |                      |     |
|-------|----------------------|-----|
| 第6章   | ADO.NET 数据库访问技术..... | 169 |
| 6.1   | ADO.NET 简介.....      | 170 |
| 6.1.1 | ADO.NET 的结构.....     | 170 |
| 6.1.2 | ADO.NET 的命名空间.....   | 172 |
| 6.2   | ADO.NET 常用对象.....    | 172 |
| 6.2.1 | Connection 对象.....   | 172 |
| 6.2.2 | Command 对象.....      | 176 |
| 6.2.3 | DataReader 对象.....   | 178 |
| 6.2.4 | DataAdapter 对象.....  | 182 |
| 6.2.5 | DataSet 对象.....      | 182 |
| 6.3   | 服务器端数据访问控件.....      | 184 |
| 6.3.1 | GridView 控件.....     | 184 |

|                                      |     |                        |     |
|--------------------------------------|-----|------------------------|-----|
| 6.3.2 DataList 控件 .....              | 195 | 8.1.1 基本需求 .....       | 224 |
| 6.4 DataSet 典型应用实例 .....             | 202 | 8.1.2 用户类型 .....       | 224 |
| 本章小结 .....                           | 210 | 8.2 系统的设计 .....        | 224 |
| 思考题 .....                            | 210 | 8.2.1 功能模块设计 .....     | 224 |
| <b>第 7 章 ASP.NET AJAX 编程技术</b> ..... | 211 | 8.2.2 数据库结构设计 .....    | 225 |
| 7.1 ASP.NET AJAX 基础 .....            | 212 | 8.3 系统的实现 .....        | 226 |
| 7.1.1 AJAX .....                     | 212 | 8.3.1 数据库的创建 .....     | 226 |
| 7.1.2 ASP.NET AJAX .....             | 213 | 8.3.2 网站的创建 .....      | 227 |
| 7.1.3 ASP.NET AJAX Extensions ...    | 213 | 8.3.3 素材文件的准备 .....    | 227 |
| 7.2 ASP.NET AJAX 服务器端控件 .....        | 214 | 8.3.4 登录功能的实现 .....    | 228 |
| 7.2.1 ScriptManager 控件 .....         | 214 | 8.3.5 系统主界面的实现 .....   | 236 |
| 7.2.2 ScriptManagerProxy 控件 .....    | 215 | 8.3.6 当前用户功能的实现 .....  | 242 |
| 7.2.3 UpdatePanel 控件 .....           | 215 | 8.3.7 用户管理功能的实现 .....  | 246 |
| 7.2.4 UpdateProgress 控件 .....        | 218 | 8.3.8 部门管理功能的实现 .....  | 260 |
| 7.2.5 Timer 控件 .....                 | 220 | 8.3.9 职工管理功能的实现 .....  | 273 |
| 本章小结 .....                           | 222 | 本章小结 .....             | 293 |
| 思考题 .....                            | 222 | 思考题 .....              | 293 |
| <b>第 8 章 ASP.NET 应用案例</b> .....      | 223 | <b>附录 A 实验指导</b> ..... | 295 |
| 8.1 系统的分析 .....                      | 224 |                        |     |



# 第 1 章

## ASP.NET 概述

ASP.NET 是 Microsoft 公司推出的新一代 Web 应用程序或动态网站开发技术，也是目前 Web 应用开发领域的主流技术之一，其实际应用已相当广泛。

**本章要点：**ASP.NET 简介；ASP.NET 应用的开发工具；ASP.NET 应用的运行环境；ASP.NET 应用程序的创建与部署。

**学习目标：**了解 ASP.NET 的概况；掌握 ASP.NET 应用开发工具的基本用法与运行环境的搭建方法；掌握 ASP.NET 应用程序的创建与部署方法。

## 1.1 ASP.NET 简介

在当前的应用程序开发领域中，主要有两大编程架构，即 C/S(Client/Server，客户机/服务器)架构与 B/S(Browser/Server，浏览器/服务器)架构。在 C/S 架构中，每台客户机都必须安装专门的客户端应用程序，服务器则通常用于安装相应的数据库管理系统，其任务主要是接收客户端程序的请求，并进行相应的处理，然后再将结果返回给客户端程序。而在 B/S 架构中，每台客户机上只需安装一个 Web 浏览器即可，应用程序是安装在 Web 服务器上的(至于数据库管理系统则可安装在 Web 服务器或专门的数据库服务器上)，浏览器与 Web 服务器之间采用 HTTP 协议进行通信。显然，与 C/S 架构相比，B/S 架构具有应用程序易于升级与维护、客户端零安装与零维护的明显优点，而且整个系统的扩展非常容易。此外，在 B/S 架构中，由于主要的数据分析与处理工作是在服务器中完成的，因此对客户机的配置要求不高，特别适合“瘦客户端”的运行环境。

与基于 C/S 架构的客户端应用程序不同，Web 应用程序是基于 B/S 架构的。在 Web 应用程序开发方面，目前常用的技术主要有 ASP、JSP、PHP、ASP.NET 等。其中，ASP.NET 是 Microsoft 公司所推出的新一代 Web 应用程序或动态网站开发技术。

ASP.NET 是 .NET 框架的一部分，可以使用任何 .NET 兼容的语言(如 Visual Basic、C# 等)编写 ASP.NET 应用程序。与 JSP、PHP、ASP 等 Web 应用程序开发技术相比，ASP.NET 具有方便、灵活、性能优、效率高、安全性强以及完全面向对象等特性，已成为目前主流的 Web 编程技术之一。

ASP.NET 发展迅猛，版本众多。其第一个版本为 ASP.NET 1.0，于 2002 年 1 月与 Visual Studio .NET 2002 一起由 Microsoft 公司正式发布。此后，Microsoft 公司不断推出更高的 ASP.NET 版本，包括 ASP.NET 1.1、ASP.NET 2.0、ASP.NET 3.0、ASP.NET 3.5、ASP.NET 4.0、ASP.NET 4.5、ASP.NET 4.5.1、ASP.NET 4.5.2 等。在开发 ASP.NET 应用程序时，应注意选定相应的 ASP.NET 版本(通常不应低于 ASP.NET 2.0 版本)。

## 1.2 ASP.NET 应用的开发工具

对于 ASP.NET 应用程序的开发来说，目前最为常用的开发工具就是 Visual Studio。

### 1.2.1 Visual Studio 简介

Visual Studio(VS)是 Microsoft 公司推出的一种功能十分强大的软件开发工具或平台，

可用于开发多种不同类型的应用程序,如 ASP.NET Web 应用程序、XML Web Services、桌面应用程序与移动应用程序等。

Visual Studio 将程序设计的相关环节(如界面设计、代码设计、程序调试与发布等)集成在同一个窗口中,极大地方便了开发人员的设计工作。通常将这种集多种功能于一体的开发平台称为集成开发环境(IDE)。在 Visual Studio 中,可使用 Visual Basic、Visual C++、Visual C#、Visual J#等语言进行开发,并可创建混合多种语言的解决方案。

与 ASP.NET 一样,Visual Studio 的版本也为数众多。对于大多数应用的开发来说,只需选用 Visual Studio 2008 或 Visual Studio 2010 即可。其中,Visual Studio 2008 内置的 .NET Framework 版本为 3.5,同时提供了对 .NET Framework 2.0、3.0 的支持。Visual Studio 2010 内置的 .NET Framework 版本为 4.0,同时提供了对 .NET Framework 2.0、3.0、3.5 的支持。

## 1.2.2 Visual Studio 的安装

启动 Visual Studio 安装程序,并按安装向导的提示进行相应的操作,即可完成 Visual Studio 的安装过程,并在“开始”菜单中添加相应的菜单项。

以 Visual Studio 2010 为例,在 Windows 7 中安装完毕后,通过在“开始”菜单中选择“所有程序”→Microsoft Visual Studio 2010→Microsoft Visual Studio 2010 菜单项,即可启动 Visual Studio 2010,并最终打开相应的 Microsoft Visual Studio 窗口(如图 1-1 所示)。

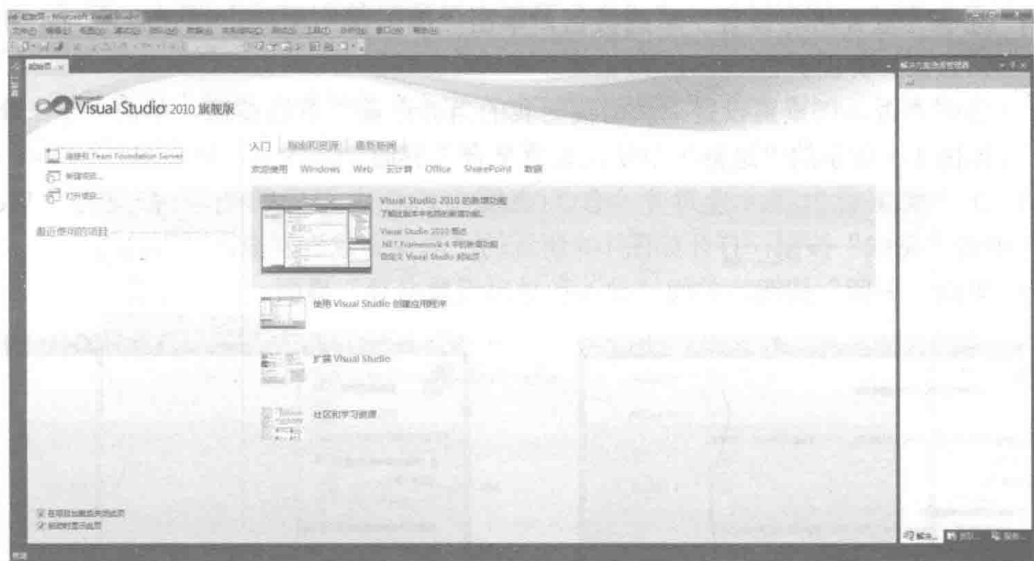


图 1-1 Microsoft Visual Studio 窗口

## 1.2.3 Visual Studio 的使用

在此,仅简要介绍 Visual Studio 2010 的常用操作与基本用法。

### 1. 默认环境设置

在首次运行 Visual Studio 2010 时,可根据需要选定相应的默认环境设置(如图 1-2 所

示)。此外，在使用过程中，也可根据需要随时改变默认环境设置。为开发 ASP.NET 应用程序，通常应将默认环境设置为“Web 开发”。



图 1-2 “选择默认环境设置”对话框

**【实例 1-1】**设置 Visual Studio 2010 的默认开发环境。

操作步骤：

- (1) 在 Visual Studio 2010 的主窗口中选择“工具”→“导入和导出设置”菜单项，打开如图 1-3 所示的“导入和导出设置向导”对话框。
- (2) 选中“重置所有设置”单选按钮，单击“下一步”按钮，打开如图 1-4 所示的“保存当前设置”界面。
- (3) 选中“否，仅重置设置，从而覆盖我的当前设置”单选按钮，单击“下一步”按钮，打开如图 1-5 所示的“选择一个默认设置集合”界面。
- (4) 在“要重置为哪个设置集合”列表框中，选中相应的选项(在此为“Web 开发”)，单击“完成”按钮，打开如图 1-6 所示的“重置完成”界面。
- (5) 单击“关闭”按钮，关闭“导入和导出设置向导”界面。



图 1-3 “导入和导出设置向导”界面



图 1-4 “保存当前设置”界面

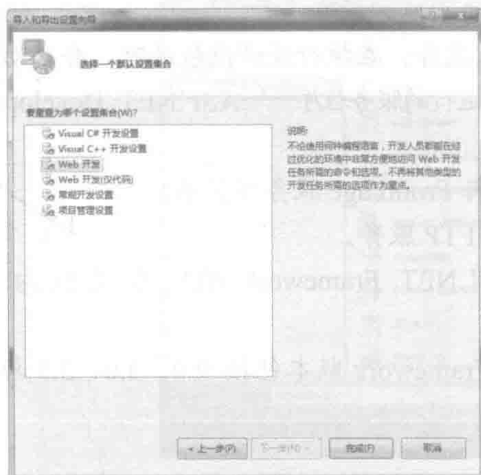


图 1-5 “选择一个默认设置集合”界面



图 1-6 “重置完成”界面

## 2. 创建网站

要使用 Visual Studio 2010 开发 ASP.NET 应用程序，首先要创建一个相应的 ASP.NET 网站。

**【实例 1-2】**在 Visual Studio 2010 中创建 ASP.NET 网站 WebSite01。

操作步骤：

- (1) 在 Visual Studio 2010 的主窗口中选择“文件”→“新建网站”菜单项，打开如图 1-7 所示的“新建网站”对话框。
- (2) 在左侧的“模板”列表框中选中“已安装的模板”下的 Visual C#(相当于将开发语言选定为 Visual C#)，并在中部的列表框中选中“ASP.NET 网站”。

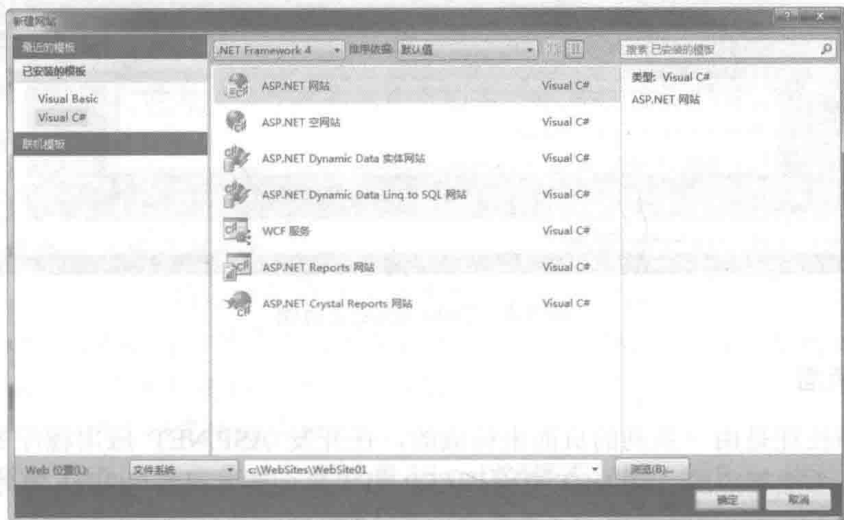


图 1-7 “新建网站”对话框

- (3) 在“Web 位置”下拉列表框中选择“文件系统”，并指定网站文件的保存位置(在此为 C:\WebSites\WebSite01)。

【说明】在 Visual Studio 中，ASP.NET 网站的创建方式有 3 种。

- 文件系统。使用该方式时，不必安装 IIS 服务。在执行或调试程序时，会自动启动 Visual Studio 自带的用于支持 ASP.NET 程序运行的服务程序——ASP.NET Development Server(ASP.NET 开发服务器)。
- HTTP。使用该方式时，需安装 IIS 服务与 FrontPage 服务器扩展。
- FTP。使用该方式时，需安装 IIS 服务与 FTP 服务。

(4) 在 .NET Framework 下拉列表框中选定 .NET Framework 的版本(在此为 .NET Framework 4)。

【说明】Visual Studio 2010 所支持的 .NET Framework 版本包括 2.0、3.0、3.5 与 4，可在创建网站时根据需要进行选择。

(5) 单击“确定”按钮。

【说明】在新建的 ASP.NET 网站中，包含有一个自动创建的 ASP.NET 页面(或 Web 窗体)Default.aspx，与其相关联的程序代码文件为 Default.aspx.cs。与以前的版本(如 Visual Studio 2008)不同，Visual Studio 2010 自动创建的 Default.aspx 已经不是一个空白的页面，而是一个应用了母版 Site.master 的页面(如图 1-8 所示)。

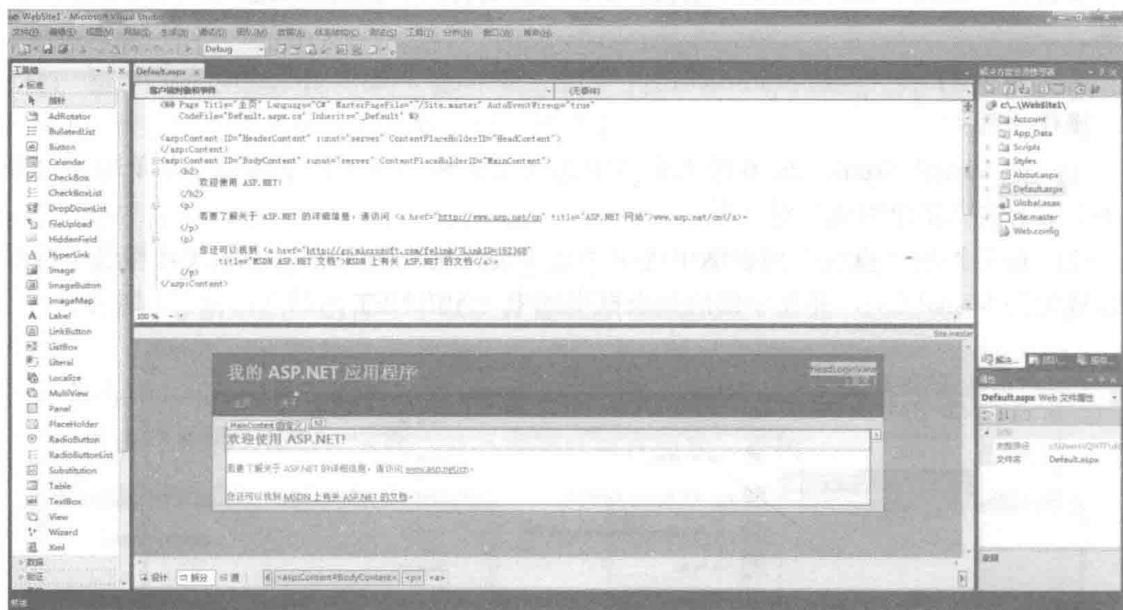


图 1-8 Default.aspx 页面

### 3. 添加页面

一个网站往往是由一系列的页面来构成的。在开发 ASP.NET 应用程序的过程中，可根据需要逐一添加相应的 ASP.NET 页面。ASP.NET 页面通常又称为 Web 窗体。

【实例 1-3】在网站 WebSite01 中添加一个 ASP.NET 页面 Abc.aspx。

操作步骤：

(1) 在“解决方案资源管理器”子窗口中，右击网站项目，并在其快捷菜单中选择“添加新项”菜单项，打开“添加新项”对话框(如图 1-9 所示)。

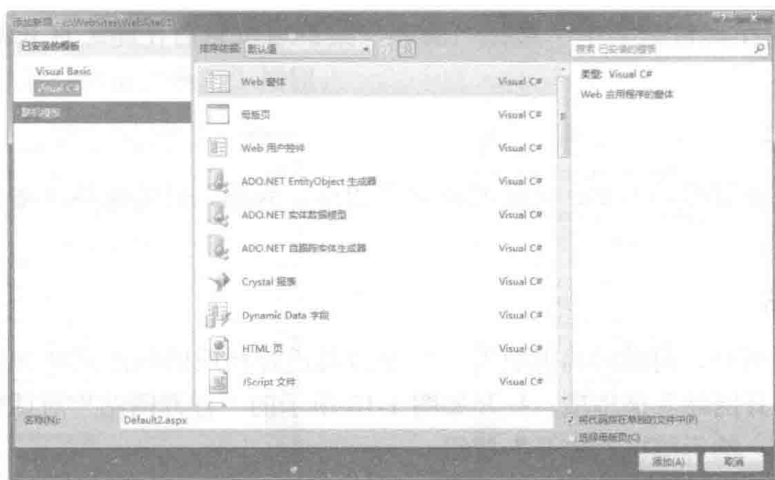


图 1-9 “添加新项”对话框

(2) 在左侧的“模板”列表框中选中“已安装的模板”下的 Visual C#(相当于将开发语言选定为 Visual C#), 并在中部的列表框中选中“Web 窗体”。

(3) 在“名称”文本框中输入页面的文件名(在此为 Abc.aspx)。

(4) 单击“添加”按钮。

**【提示】**ASP.NET 页面默认采用代码隐藏模型, 即其程序代码单独保存在相关联的隐藏代码文件中。根据所使用的编程语言的不同, 代码文件的扩展名也有所不同。例如, 使用 Visual C#时扩展名为.cs, 使用 Visual Basic 时扩展名为.vb。实际上, ASP.NET 页面也支持单文件模型, 即其程序代码与页面标记一起保存在同一个.aspx 文件中。使用“添加新项”对话框添加 Web 窗体时, 若取消“将代码放在单独的文件中”复选框的选中状态, 则以单文件方式创建之。

**【说明】**在 ASP.NET 页面文件中, 通常包含有一些以“<%@”开头、“%>”结尾的页面指令。页面指令包含一个或多个与其值成对出现的属性, 用于指定页面的设置信息, 一般放在文件的开头, 必要时也可放在文件内的任何位置。@ Page 指令是最常用的一个页面指令, 其基本格式为:

```
<%@ Page 属性名="属性值" [属性名="属性值"...] %>
```

@ Page 指令的常用属性如下。

- Language: 用于指定所使用的编程语言, 如 C#、VB 等。
- AutoEventWireup: 用于指定是否自动与事件相关联(其默认值为 true)。
- CodeFile: 用于指定相应的程序代码文件。
- Inherits: 用于指定所继承的类。

对于新建的 Default.aspx 页面, 若使用 Visual C#语言, 则其@ Page 指令通常为:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="Default" %>
```

#### 4. 删除页面

对于不再需要的页面, 可随时将其删除掉。为此, 可在“解决方案资源管理器”子窗

口中右击之，并在其快捷菜单中选择“删除”菜单项，然后在随之打开的对话框中单击“确定”按钮。

5. 关闭项目

对于当前正在开发的网站，可随时将其关闭掉。为此，只需选择“文件”→“关闭项目”菜单项即可。

6. 打开网站

对于已有的网站，可随时将其打开，以便对其进行相应的修改或完善。为此，可选择“文件”→“打开网站”菜单项，打开如图 1-10 所示的“打开网站”对话框，并在其中选定要打开的网站，然后单击“打开”按钮。



图 1-10 “打开网站”对话框

1.3 ASP.NET 应用的运行环境

ASP.NET 应用程序通常在 Windows 系统平台上运行，且必须预先在系统中安装好 IIS(Internet 信息服务)以及相应版本的.NET 框架(.NET Framework)。例如，对于 ASP.NET 4.0 的应用程序来说，所需的.NET 框架为.NET Framework 4.0。

1.3.1 IIS 的安装

IIS 其实就是一种 Web 服务器。作为 Windows 操作系统中的一个组件，IIS 的安装是极为简单的。

1. 安装 IIS

以 Windows 7 为例，安装 IIS 的主要步骤为：打开“控制面板”，单击“程序和功能”图标，单击“打开或关闭 Windows 功能”链接，选中“Internet 信息服务”下的“Web 管理工具”复选框（如图 1-11 所示），再选中“万维网服务”下“应用程序开发

功能”下的“.NET 扩展性”“ASP.NET”“ISAPI 扩展”与“ISAPI 筛选器”复选框（如图 1-12 所示），最后单击“确定”按钮。



图 1-11 “Windows 功能”对话框(Web 管理工具)      图 1-12 “Windows 功能”对话框(万维网服务)

2. 测试 IIS

打开 IE 浏览器，在地址栏中输入“http://localhost”或“http://127.0.0.1”，若能显示如图 1-13 所示的 IIS 7 页面，则说明 IIS 一切正常。

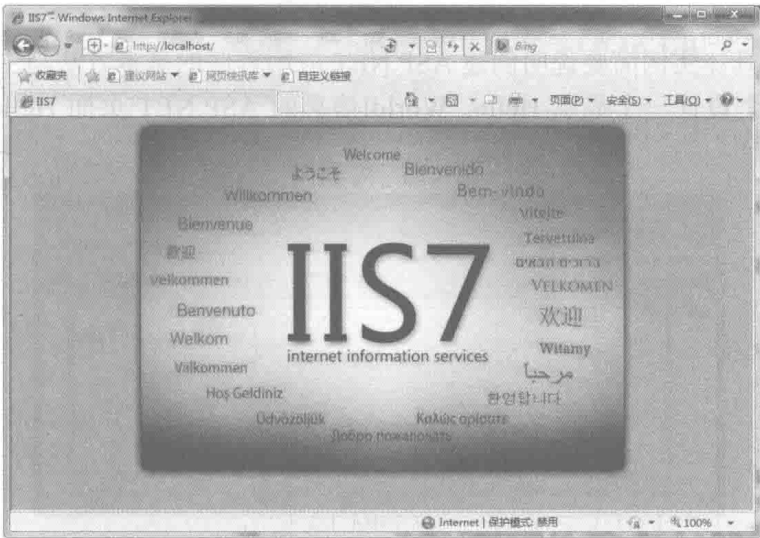


图 1-13 IIS 7 页面

1.3.2 .NET 框架的安装

为安装 .NET 框架，应先下载 .NET Framework 安装程序，然后再运行之。.NET Framework 安装程序可从微软官方网站或其他有关网站免费下载。

以 .NET Framework 4.0 为例，启动其安装程序后，将打开如图 1-14 所示的“Microsoft .NET Framework 4 安装程序”对话框。按照安装向导的提示完成相应的操作后，即可完成安装过程。