



工业和信息化普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

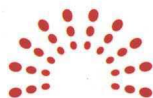
21st Century University Planned Textbooks of Computer Science

# ASP.NET程序设计 案例教程

## ASP.NET Programming Case Tutorial

杨树林 胡洁萍 编著

- 涵盖主流技术和方法，体现教学需求
- 融合典型案例和知识，适应案例教学
- 突出项目引导和驱动，贴近实际应用



高校系列



人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# ASP.NET程序设计 案例教程

ASP.NET Programming Case Tutorial

杨树林 胡洁萍 编著



高校系列

人民邮电出版社

北京

## 图书在版编目(CIP)数据

ASP.NET程序设计案例教程 / 杨树林, 胡洁萍编著  
— 北京: 人民邮电出版社, 2011.4  
21世纪高等学校计算机规划教材  
ISBN 978-7-115-24827-5

I. ①A… II. ①杨… ②胡… III. ①主页制作—程序设计—高等学校—教材 IV. ①TP393.092

中国版本图书馆CIP数据核字(2011)第019389号

## 内 容 提 要

ASP.NET 是微软公司推出的全新的互联网应用程序开发技术,是当今最主流的 Web 程序开发技术之一。本书共分 9 章,内容包括 ASP.NET 概述、C#语言基础、ASP.NET 网页、用户界面设计、数据库访问技术、状态管理与数据缓存、ASP.NET 常用技术、LINQ 数据库技术和 BBS 综合案例。每章内容都与案例相结合,有助于学生理解知识,应用知识,达到学以致用。书中引进一些新知识和新方法,内容实用,重点突出,讲解精练,案例典型,既方便学习,又便于应用。

本书内容丰富,实例典型,知识讲解系统,可作为大中专院校计算机及其相关专业的教材,也可供软件开发人员及其他有关人员学习参考。

21 世纪高等学校计算机规划教材

### ASP.NET 程序设计案例教程

- 
- ◆ 编 著 杨树林 胡洁萍  
责任编辑 邹文波
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
大厂聚鑫印刷有限责任公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 20.25 2011 年 4 月第 1 版  
字数: 534 千字 2011 年 4 月河北第 1 次印刷

---

ISBN 978-7-115-24827-5

定价: 37.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223  
反盗版热线: (010)67171154



2. 注意新方法、新技术的引用，突出实用内容（主要指市场开发常用的）。
3. 以案例为线索组织知识，围绕知识设计案例，二者相辅相成。
4. 突出案例的典型性，针对性强。将系统性（案例之间尽可能有一定的联系，组合在一起能形成完整系统）的案例融于知识中。
5. 处理好具体实例与思想方法的关系，局部知识应用与综合应用的关系。
6. 知识讲解循序渐进，难度适宜，便于教学和学习。

本书共分 9 章，内容包括 ASP.NET 概述、C#语言基础、ASP.NET 网页、用户界面设计、数据库访问技术、状态管理与数据缓存、ASP.NET 常用技术、LINQ 数据库技术和 BBS 综合案例。书中除包含许多配合知识学习的例子外，还包含 29 个案例，其中除了大型的论坛案例外，其他案例主要是网络书城项目的分解案例，随知识的讲解逐渐展开。

由于编者水平有限，加上时间仓促，书中难免存在疏漏和不足之处，恳请读者批评指正，使本书得以改进和完善。

编者

2011 年 1 月于北京

# 目 录

|   |                                       |
|---|---------------------------------------|
| <b>第 1 章 ASP.NET 概述</b> .....1          | <b>2.3 流程控制与异常处理</b> .....39          |
| 1.1 ASP.NET 简介.....1                    | 2.3.1 分支结构.....39                     |
| 1.1.1 .NET 技术简介.....1                   | 2.3.2 循环语句.....42                     |
| 1.1.2 什么是 ASP.NET.....3                 | 2.3.3 异常处理.....44                     |
| 1.1.3 .NET 3.5.....4                    | <b>2.4 C#面向对象编程</b> .....45           |
| 1.2 ASP.NET 开发环境.....5                  | 2.4.1 类和对象.....45                     |
| 1.2.1 安装 Visual Studio 2008.....5       | 2.4.2 类的成员.....48                     |
| 1.2.2 创建 Web 项目.....8                   | 2.4.3 继承.....50                       |
| 1.2.3 管理 Web 项目中的资源.....9               | 2.4.4 案例 2-3 网络书城中的实体<br>模型类设计.....51 |
| 1.2.4 创建 ASP.NET 网页.....10              | 2.4.5 抽象类、接口与多态性.....56               |
| 1.3 ASP.NET 应用程序的构成.....13              | 2.4.6 案例 2-4 网络书城中的<br>接口设计.....58    |
| 1.3.1 文件类型.....13                       | <b>2.5 数组和集合</b> .....61              |
| 1.3.2 文件夹类型.....14                      | 2.5.1 声明与访问数组.....61                  |
| 1.3.3 网站全局文件.....15                     | 2.5.2 集合.....62                       |
| 1.3.4 ASP.NET 配置.....15                 | 2.5.3 案例 2-5 网络书城中的<br>购物车类设计.....65  |
| 1.4 ASP.NET 应用中的分层架构.....19             | <b>2.6 C# 3.5 的新特征</b> .....66        |
| 1.4.1 分层架构模式.....19                     | 2.6.1 隐型局部变量.....66                   |
| 1.4.2 ASP.NET 中的 3 层结构.....20           | 2.6.2 扩展方法.....67                     |
| 1.4.3 案例 1-1 网络书城系统<br>分析与设计.....20     | 2.6.3 Lambda 表达式.....67               |
| 本章小结.....24                             | 2.6.4 对象和集合初始化.....69                 |
| 习题与实验.....25                            | 2.6.5 匿名类型.....69                     |
| <b>第 2 章 C#语言基础</b> .....26             | 本章小结.....69                           |
| 2.1 C#语言的基本语法.....26                    | 习题与实验.....70                          |
| 2.1.1 基本编码规则.....26                     | <b>第 3 章 ASP.NET 网页</b> .....71       |
| 2.1.2 数据类型.....27                       | 3.1 概述.....71                         |
| 2.1.3 案例 2-1 为书城网站定义用户<br>权限枚举类型.....29 | 3.1.1 ASP.NET 网页及其存储模式.....71         |
| 2.1.4 运算符和表达式.....30                    | 3.1.2 ASP.NET 网页生命周期.....74           |
| 2.2 字符串、日期和时间.....31                    | 3.1.3 内置对象.....75                     |
| 2.2.1 字符串.....31                        | 3.2 ASP.NET Web 服务器控件.....78          |
| 2.2.2 DateTime 和 TimeSpan.....33        | 3.2.1 ASP.NET 服务器控件类型.....78          |
| 2.2.3 案例 2-2 日期操作工具类设计.....35           | 3.2.2 Web 服务器控件概述.....79              |
| 2.2.4 数据类型的转换.....38                    | 3.2.3 常用的 Web 控件.....81               |

|                                |            |                                      |            |
|--------------------------------|------------|--------------------------------------|------------|
| 3.2.4 案例 3-1 图书反馈网页的设计         | 85         | 5.1.3 案例 5-1 连接书城数据库                 | 128        |
| 3.2.5 验证控件                     | 90         | 5.1.4 数据更新操作                         | 129        |
| 3.2.6 案例 3-2 实现图书反馈网页的<br>数据验证 | 93         | 5.1.5 数据查询操作                         | 130        |
| 3.3 页面切换与数据传递                  | 98         | 5.2 应用程序结构与数据操作                      | 133        |
| 3.3.1 页面切换                     | 98         | 5.2.1 两层应用结构                         | 133        |
| 3.3.2 页面间的数据传递                 | 99         | 5.2.2 三层应用结构                         | 141        |
| 本章小结                           | 101        | 5.2.3 案例 5-2 书城网站的 DAL 层与<br>BLL 层实现 | 144        |
| 习题与实验                          | 101        | 5.3 数据绑定与数据绑定控件                      | 153        |
| <b>第 4 章 用户界面设计</b>            | <b>102</b> | 5.3.1 数据绑定简介                         | 153        |
| 4.1 主题                         | 102        | 5.3.2 GridView 控件                    | 153        |
| 4.1.1 概述                       | 102        | 5.3.3 DetailsView 控件与 FormView<br>控件 | 156        |
| 4.1.2 创建主题                     | 103        | 5.3.4 案例 5-3 实现书城网站的<br>图书管理         | 157        |
| 4.1.3 应用主题                     | 104        | 5.4 其他数据绑定控件                         | 165        |
| 4.1.4 案例 4-1 书城网站的主题设计         | 105        | 5.4.1 DataList 控件与 Repeater<br>控件    | 165        |
| 4.2 母版页                        | 108        | 5.4.2 案例 5-4 实现书城网站图书<br>分类菜单        | 166        |
| 4.2.1 母版页的基础知识                 | 108        | 5.4.3 ListView 控件与 DataPager<br>控件   | 167        |
| 4.2.2 创建母版页和内容页                | 109        | 5.4.4 案例 5-5 实现书城网站的<br>主界面          | 169        |
| 4.2.3 内容页和母版页的交互               | 112        | 本章小结                                 | 173        |
| 4.2.4 案例 4-2 书城网站的<br>母版页设计    | 113        | 习题与实验                                | 174        |
| 4.3 用户控件                       | 115        | <b>第 6 章 状态管理与数据缓存</b>               | <b>175</b> |
| 4.3.1 用户控件简介                   | 115        | 6.1 ASP.NET 状态管理概述                   | 175        |
| 4.3.2 用户控件的创建                  | 115        | 6.1.1 什么是状态管理                        | 175        |
| 4.3.3 用户控件的使用                  | 116        | 6.1.2 状态管理的类型                        | 176        |
| 4.3.4 案例 4-3 书城网站的<br>用户控件设计   | 117        | 6.2 基于客户端的状态管理                       | 176        |
| 4.4 网站地图与页面导航                  | 120        | 6.2.1 视图状态                           | 176        |
| 4.4.1 网站地图                     | 121        | 6.2.2 控件状态                           | 178        |
| 4.4.2 使用导航地图实现网站导航             | 122        | 6.2.3 隐藏域                            | 179        |
| 4.4.3 案例 4-4 书城的网站站点<br>导航设计   | 123        | 6.2.4 Cookie                         | 180        |
| 本章小结                           | 124        | 6.2.5 查询字符串                          | 182        |
| 习题与实验                          | 124        | 6.2.6 案例 6-1 完善书城网站用户<br>登录程序        | 182        |
| <b>第 5 章 数据库访问技术</b>           | <b>126</b> | 6.3 基于服务器的状态管理                       | 185        |
| 5.1 数据库访问基础                    | 126        |                                      |            |
| 5.1.1 ADO.NET 简介               | 126        |                                      |            |
| 5.1.2 数据库的连接                   | 127        |                                      |            |

|  |     |   |     |
|--|-----|---|-----|
| 6.3.1 应用程序状态 .....                           | 185 | 7.5.4 案例 7-5 实现书城网站的<br>图书添加 .....          | 232 |
| 6.3.2 Session 状态管理 .....                     | 187 | 本章小结 .....                                  | 236 |
| 6.3.3 案例 6-2 实现书城网站的<br>用户统计 .....           | 189 | 习题与实验 .....                                 | 237 |
| 6.4 数据缓存 .....                               | 190 | <b>第 8 章 LINQ 数据库技术</b> .....               | 238 |
| 6.4.1 缓存概述 .....                             | 190 | 8.1 LINQ 概述 .....                           | 238 |
| 6.4.2 页输出缓存 .....                            | 192 | 8.1.1 什么是 LINQ .....                        | 238 |
| 6.4.3 使用应用程序缓存 .....                         | 195 | 8.1.2 基本的查询操作 .....                         | 239 |
| 6.4.4 缓存依赖 .....                             | 196 | 8.1.3 LINQ 查询表达式 .....                      | 241 |
| 6.4.5 案例 6-3 在书城网站中应用<br>缓存技术 .....          | 198 | 8.1.4 使用 LINQ 进行数据转换 .....                  | 243 |
| 本章小结 .....                                   | 199 | 8.2 LINQ to ADO.NET .....                   | 246 |
| 习题与实验 .....                                  | 200 | 8.2.1 创建对象模型 .....                          | 246 |
| <b>第 7 章 ASP.NET 常用技术</b> .....              | 201 | 8.2.2 查询和更改数据库 .....                        | 248 |
| 7.1 成员资格与角色管理 .....                          | 201 | 8.2.3 案例 8-1 使用 LINQ 实现书城<br>网站的数据访问层 ..... | 249 |
| 7.1.1 验证方式及其配置 .....                         | 201 | 8.2.4 存储过程 .....                            | 252 |
| 7.1.2 成员资格管理及其配置 .....                       | 202 | 8.2.5 案例 8-2 使用 LINQ 实现书城<br>网站的结账 .....    | 253 |
| 7.1.3 ASP.NET 登录控件 .....                     | 207 | 8.3 LINQ to XML .....                       | 261 |
| 7.1.4 案例 7-1 基于成员资格管理<br>实现书城网站用户管理 .....    | 208 | 8.3.1 LINQ to XML 概述 .....                  | 261 |
| 7.2 个性化用户服务 .....                            | 212 | 8.3.2 创建 XML 树 .....                        | 262 |
| 7.2.1 个性化服务简介 .....                          | 212 | 8.3.3 序列化 XML 树 .....                       | 265 |
| 7.2.2 个性化服务配置 .....                          | 212 | 8.3.4 查询 XML 树 .....                        | 265 |
| 7.2.3 个性化数据操作 .....                          | 214 | 8.3.5 修改 XML 树 .....                        | 266 |
| 7.2.4 为匿名用户实现个性化服务 .....                     | 215 | 本章小结 .....                                  | 268 |
| 7.2.5 案例 7-2 实现网络书城<br>购物车功能 .....           | 215 | 习题与实验 .....                                 | 269 |
| 7.3 验证码功能实现 .....                            | 218 | <b>第 9 章 BBS 综合案例</b> .....                 | 270 |
| 7.3.1 绘图的基本知识 .....                          | 218 | 9.1 系统分析与设计 .....                           | 270 |
| 7.3.2 案例 7-3 实现书城网站验证码 .....                 | 219 | 9.1.1 系统分析 .....                            | 270 |
| 7.4 ASP.NET AJAX .....                       | 221 | 9.1.2 总体设计 .....                            | 271 |
| 7.4.1 ASP.NET AJAX 概述 .....                  | 221 | 9.1.3 创建对象模型 .....                          | 276 |
| 7.4.2 创建 AJAX 应用 .....                       | 223 | 9.1.4 接口设计 .....                            | 278 |
| 7.4.3 案例 7-4 在书城网站中<br>使用 ASP.NET AJAX ..... | 225 | 9.2 数据访问层实现 .....                           | 280 |
| 7.5 文件操作 .....                               | 228 | 9.2.1 版块数据访问类 .....                         | 280 |
| 7.5.1 文件的管理 .....                            | 228 | 9.2.2 主题数据访问类 .....                         | 281 |
| 7.5.2 文件的 I/O 操作 .....                       | 229 | 9.2.3 帖子数据访问类 .....                         | 284 |
| 7.5.3 文件上传 .....                             | 231 | 9.3 业务逻辑层实现 .....                           | 286 |
|  |     | 9.3.1 版块业务逻辑类 .....                         | 286 |



|                               |     |                    |     |
|-------------------------------|-----|--------------------|-----|
| 9.3.2 主题业务逻辑类·····            | 286 | 9.5.1 主题设计·····    | 290 |
| 9.3.3 帖子业务逻辑类·····            | 287 | 9.5.2 母版及主页设计····· | 291 |
| 9.4 系统配置·····                 | 288 | 9.5.3 主题视图设计·····  | 294 |
| 9.4.1 连接字符串及环境变量配置·····       | 288 | 9.5.4 帖子视图设计·····  | 303 |
| 9.4.2 验证模式、成员及角色<br>管理配置····· | 289 | 9.5.5 用户视图设计·····  | 313 |
| 9.5 表现层设计·····                | 290 | 本章小结·····          | 316 |
|                               |     | 习题与实验·····         | 316 |

# 第 1 章

## ASP.NET 概述

### 本章要点

- .NET 技术及 ASP.NET 概述
- ASP.NET 应用开发环境及其使用
- ASP.NET 应用程序组成
- 分层架构及其设计原则
- 利用分层架构思想分析和设计网络书城

ASP.NET 是统一的 Web 应用程序平台，它提供了为建立和部署企业级 Web 应用程序所必需的服务。ASP.NET 为能够面向任何浏览器或设备的更安全的、更强的可升级性，以及更稳定的应用程序提供新的编程模型和基础结构。

## 1.1 ASP.NET 简介

ASP.NET 并不是一门编程语言，而是一个统一的 Web 开发模型，它支持以可视化的方式创建企业级网站。ASP.NET 是 .NET 框架（.NET Framework）的一部分，可以利用 .NET 框架中的类进行编程，可使用 VB.NET、C#、J# 和 JScript.NET 等编程语言来开发 Web 应用程序。

### 1.1.1 .NET 技术简介

#### 1. 什么是 .NET

.NET 技术是微软公司推出的一个全新概念，它代表了一个集合、一个环境和一个可以作为平台支持下一代 Internet 的可编程结构。.NET 的目的就是将互联网作为新一代操作系统的基础，对互联网的设计思想进行扩展，用户在任何地方、任何时间，以及利用任何设备都能访问所需要的信息、文件和程序。

.NET 平台包括 .NET 框架和 .NET 开发工具等组成部分。.NET 框架是整个开发平台的基础，包括公共语言运行时（Common Language Runtime, CLR）和 .NET 类库。公共语言运行时类似于 Java 虚拟机，负责内存管理和程序执行，是 .NET 的基础。.NET 开发工具包括 Visual Studio .NET 集成开发环境和 .NET 编程语言。.NET 编程语言包括 Visual Basic、Visual C++ 和新的 Visual C#，用来创建运行在公共语言运行时上的应用程序。.NET 框架结构如图 1-1 所示。

|                             |                 |             |                           |                          |                            |                         |
|-----------------------------|-----------------|-------------|---------------------------|--------------------------|----------------------------|-------------------------|
| Visual Basic.NET            | C#              | 托管 C++      | Jscript.NET               | 其他语言                     |                            |                         |
| 公共语言规范 (CLS)                |                 |             |                           |                          |                            |                         |
| ASP.NET<br>Web 应用<br>Web 服务 | Windows<br>应用程序 | 控制台<br>应用程序 | WCF<br>(Windows 通<br>信基础) | WPF<br>(Windows<br>呈现基础) | WWF<br>(Windows 工<br>作流基础) | WCS<br>(Windows<br>卡空间) |
| .NET 框架基础类库                 |                 |             |                           |                          |                            |                         |
| 公共语言运行时                     |                 |             |                           |                          |                            |                         |
| 操作系统                        |                 |             |                           |                          |                            |                         |

图 1-1 .NET 框架结构

## 2. 公共语言运行时

公共语言运行时是 .NET Framework 的基础。它管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务，还能监视程序的运行，强制实施代码访问安全，通过严格类型验证和代码验证加强代码的可靠性。此外，它的托管环境消除了许多常见的软件问题。

公共语言运行时不仅提供了多种软件服务，而且对以往的软件提供了支持。托管和非托管代码之间的互操作性，使开发人员能够继续使用所需的 COM 组件和动态链接库 (Dynamic Link Library, DLL)。

公共语言运行时提高了开发人员的工作效率，如程序员可以用他们选择的开发语言编写应用程序，却仍能充分利用其他开发人员用其他语言编写的运行库、类库和组件。以 .NET 框架为目标的语言编译器，使得用该语言编写的现有代码可以使用 .NET 框架的功能，这大大减轻了现有应用程序迁移过程的工作负担。

## 3. .NET 框架类库

.NET 框架类库是一个与公共语言运行时紧密集成的可重用的类型集合。该类库是面向对象的，并提供用户自己的托管代码可从中导出功能的类型。这不但使 .NET 框架类型易于使用，而且还减少了学习 .NET 框架的新功能所需要的时间。此外，第三方组件可与 .NET 框架中的类无缝集成。

例如，.NET 框架集合类实现一组可用于开发用户自己的集合类的接口，用户的集合类将与 .NET 框架中的类无缝地混合。

.NET 框架类型使用户能够完成一系列常见编程任务 (包括诸如字符串管理、数据收集、数据库连接、文件访问等任务)。除这些常见任务之外，类库还包括支持多种专用开发方案的类型。例如，可使用 .NET 框架开发下列类型的应用程序和服务。

- 控制台应用程序。
- Windows GUI 应用程序 (Windows 窗体)。
- Windows Presentation Foundation (WPF) 应用程序。
- ASP.NET 应用程序。
- Web 服务。
- Windows 服务。
- 使用 Windows Communication Foundation (WCF) 的面向服务的应用程序。
- 使用 Windows Workflow Foundation (WWF) 的启用工作流程的应用程序。

例如，Windows 窗体类是一组综合性的可重用的类型，它们大大简化了 Windows GUI 的开发。如果要编写 ASP.NET 应用程序，可使用 Web 窗体类。

## 4. Visual Studio

Visual Studio 是一套完整的开发工具，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C# 和 Visual C++ 都使用相同的集成开发环

境 (IDE), 这样就能够进行工具共享, 并能够轻松地创建混合语言解决方案。另外, 这些语言使用 .NET 框架的功能, 它提供了可简化 ASP Web 应用程序和 XML Web Services 开发的关键技术。

Visual Studio 提供丰富的开发环境, 包含如下特性。

- 页面设计。使用 Web 表单设计器, 可以通过拖曳的方式设计页面, 省去了很多编写 HTML 代码的麻烦。

- 自动错误检测。Visual Studio 能够自动地报告出代码编写中的错误, 这样不用经过调试就可以发现那些诸如语法的错误, 可以节省代码调试时间。

- 调试工具。Visual Studio 提供了强大的调试工具, 使用这些调试工具可以查看运行中的代码和跟踪变量的内容。

- 智能感知。在代码编辑过程中, Visual Studio 能够识别变量并自动列出该对象的信息, 以方便代码的编辑。

## 5. C#语言

C#是微软公司在 2000 年 7 月发布的一种全新的简单、安全、面向对象的程序设计语言。它是专门为 .NET 的应用而开发的语言。它吸收了 C++、Visual Basic、Delphic、Java 等语言的优点, 体现了当今最新的程序设计技术的功能和精华。C#继承了 C 语言的语法风格, 同时又继承了 C++ 面向对象特性。不同的是, C#的对象模型已经面向 Internet 进行了重新设计, 使用的是 .NET 框架的类库; C#不再提供对指针类型的支持, 使得程序不能随便访问内存地址空间, 从而更加健壮; C#不再支持多重继承, 避免了以往类层次结构中由于多重继承带来的可怕后果。 .NET 框架为 C# 提供了一个强大的、易用的、逻辑结构一致的程序设计环境。同时, 公共语言运行时为 C#程序语言提供了一个托管的运行时环境, 使程序比以往更加稳定、安全。

## 1.1.2 什么是 ASP.NET

ASP.NET 是微软公司推出的用于编写动态网页的一项功能强大的新技术, 它建立在公共语言运行时基础上, 是一个已编译的、基于 .NET 的环境, 可以用任何与 .NET 兼容的语言创作应用程序。

ASP.NET 的语法在很大程度上与 ASP 兼容, 同时它还提供一种新的编程模型和结构, 用于生成更安全、可伸缩和稳定的应用程序。与以前的动态网页开发技术相比, 其优点体现在以下几方面。

- 可管理性。ASP.NET 使用基于文本的、分级的配置系统, 简化了将设置应用于服务器环境和 Web 应用程序的工作。因为配置信息是存储为纯文本的, 因此, 可以在没有本地管理工具的帮助下应用新的设置。配置文件的任何变化都可以自动检测到并应用于应用程序。

- 安全。ASP.NET 为 Web 应用程序提供了默认的授权和身份验证方案。开发人员可以根据应用程序的需要很容易地添加、删除或替换这些方案。

- 易于部署。通过简单地将必要的文件复制到服务器上, ASP.NET 应用程序即可以部署到该服务器上。不需要重新启动服务器, 甚至在部署或替换运行的已编译代码时也不需要重新启动。

- 增强的性能。ASP.NET 是运行在服务器上的已编译代码。与传统的 Active Server Pages (ASP) 不同, ASP.NET 能利用早期绑定、实时 (JIT) 编译、本机优化和全新的缓存服务来提高性能。

- 灵活的输出缓存。根据应用程序的需要, ASP.NET 可以缓存页数据、页的一部分或整个页。缓存的项目可以依赖于缓存中的文件或其他项目, 或者可以根据过期策略进行刷新。

- 国际化。ASP.NET 在内部使用 Unicode 以表示请求和响应数据。可以为每台计算机、每个目录和每页配置国际化设置。

- 移动设备支持。ASP.NET 支持任何设备上的任何浏览器。开发人员使用与用于传统的桌

面浏览器相同的编程技术，处理新的移动设备。

- 扩展性和可用性。ASP.NET 被设计成可扩展的、具有特别专有的功能，以提高群集的、多处理器环境的性能。此外，Internet 信息服务（IIS）和 ASP.NET 运行时密切监视和管理进程，以便在一个进程出现异常时，可在该位置创建新的进程使应用程序继续处理请求。

- 跟踪和调试。ASP.NET 提供了跟踪服务，该服务可在应用程序级别和页面级别调试过程中启用。可以选择查看页面的信息，或者使用应用程序级别的跟踪查看工具查看信息。在开发和应用程序处于生产状态时，ASP.NET 支持使用 .NET Framework 调试工具进行本地和远程调试。当应用程序处于生产状态时，跟踪语句能够留在产品代码中而不会影响性能。

- 与 .NET Framework 集成。ASP.NET 是 .NET Framework 的一部分，整个平台的功能和灵活性对 Web 应用程序都是可用的。也可从 Web 上流畅地访问 .NET 类库以及消息和数据访问解决方案。ASP.NET 是独立于语言之外的，所以开发人员能选择最适于应用程序的语言。另外，公共语言运行时的互用性还保存了基于 COM 开发的现有投资。

- 与现有 ASP 应用程序的兼容性。ASP 和 ASP.NET 可并行运行在 IIS Web 服务器上而互不冲突；不会发生因安装 ASP.NET 而导致现有 ASP 应用程序崩溃的可能。ASP.NET 仅处理具有 .asp 文件扩展名的文件。具有 .asp 文件扩展名的文件继续由 ASP 引擎来处理。然而，应该注意的是会话状态和应用程序状态并不在 ASP 和 ASP.NET 页面之间共享。

ASP.NET 启用了分布式应用程序的两个功能：Web 窗体和 XML Web 服务。

- Web 窗体技术可建立强大的基于窗体的网页。Web 窗体页面使用可重复使用的内建组件或自定义组件以简化页面中的代码。

- 使用 ASP.NET 创建的 XML Web 服务可使用户远程访问服务器。使用 XML Web 服务，商家可以提供其数据或商业规则的可编程接口，之后可以由客户端和服务端应用程序获得和操作。以任何语言编写的且运行在任何操作系统上的程序，都能调用 XML Web 服务。

### 1.1.3 .NET 3.5

.NET 框架 3.5 版本，在传承以往版本的优良性能的同时，有如下改变。

- 深度集成 LINQ 和数据感知。这项新功能能够让程序员使用 LINQ 语言，为几种不同的数据类型（SQL 数据、集合、XML 和数据集）以相同的语法编写代码来进行操作。

- ASP.NET AJAX 技术可以建立更有效率、更具有互动性和高度个性化的 Web 体验，而且这些都可以在最流行的浏览器中实现。

- 新的 Web 协议支持创建包括 AJAX、JSON、REST、POX、RSS、ATOM 和几个新的 WS-\* 标准的 WCF 服务。

- Visual Studio 2008 开发工具支持 WWF、WCF、WPF 以及 workflow 服务技术的开发。

- .NET 3.5 类库中的新类能够满足很多客户的需求。

.NET 3.5 的安装过程将在后面介绍 Visual Studio 2008 安装时进行讲解，它同该工具的安装过程一起进行。.NET 3.5 运行的系统需求如下。

- 支持的操作系统：Windows Server 2003、Windows Server 2008、Windows Vista 和 Windows XP。

- 处理器：至少 400MHz 的奔腾处理器，而 1GHz 以上的奔腾处理器的效果更好。

- RAM：至少 96MB，推荐 256MB。

- 硬盘：至少有 500MB 的可利用空间。

- 显示器：至少 800 像素 × 600 像素、256 色，推荐 1024 像素 × 768 像素、32-bit 色以上。

此外，在安装 .NET 3.5 以前，请先删除 .NET 的早期版本。

## 1.2 ASP.NET 开发环境

每一个正式版本的 .NET 框架都会有一个与之对应的高度集成的开发环境，微软公司称之为 Visual Studio，也就是可视化工作室。随同 .NET 3.5 一起发布的开发工具是 Visual Studio 2008，它对基于 .NET 3.5 的项目开发有很大帮助，使用 Visual Studio 2008 可以很方便地进行各种项目的创建，具体程序的设计，程序调试和跟踪，以及项目发布等。

### 1.2.1 安装 Visual Studio 2008

Visual Studio 2008 目前有 3 个版本：Visual Studio 2008 Professional 版本、Visual Studio 2008 Standard 版本和 Visual Studio Team System 2008 Team Suite 版本，其中前两种用于个人和小型开发团队，采用最新技术开发应用程序和实现有效的业务目标，前两种又统称为 Visual Studio 2008；第 3 种是为体系结构、设计、数据库开发以及应用程序测试等多任务的团队提供的集成的工具集，在应用程序生命周期的每个步骤，团队成员都可以继续协作并利用一个完整的工具集与指南。

首先，读者可以到 <http://mmsdn.microsoft.com/zh-cn/vs2008/default.aspx> 下载 Visual Studio 2008 的试用版，也可以去购买正版安装程序。

Visual Studio 2008 的安装步骤如下。

- (1) 打开安装程序后，首先进入图 1-2 所示的安装向导界面。在此界面，可以选择安装的产品，单击【安装 Visual Studio 2008】即可进入资源复制过程。
- (2) 在资源复制完毕后，进入图 1-3 所示界面。



图 1-2 安装向导界面

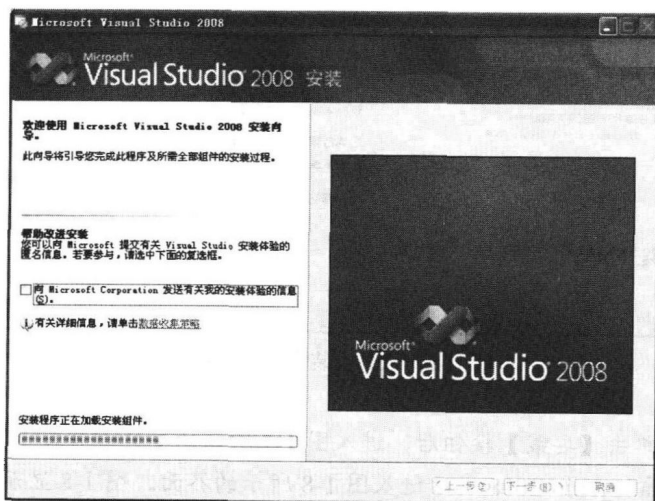


图 1-3 组件加载过程

(3) 组件加载完毕后,【下一步】按钮被激活,如图 1-4 所示。



图 1-4 组件加载完毕时的界面

- (4) 在图 1-4 所示界面,单击【下一步】按钮,进入图 1-5 所示的软件安装许可认证界面。
- (5) 在图 1-5 中选中【我已阅读并接受许可条款】单选按钮,输入用户名称后【下一步】按钮会被激活,单击【下一步】按钮,进入图 1-6 所示的安装界面。图 1-6 所示为安装功能选择和安装路径选择的界面,选中【默认值】单选按钮,并选择安装路径,然后单击【安装】按钮。

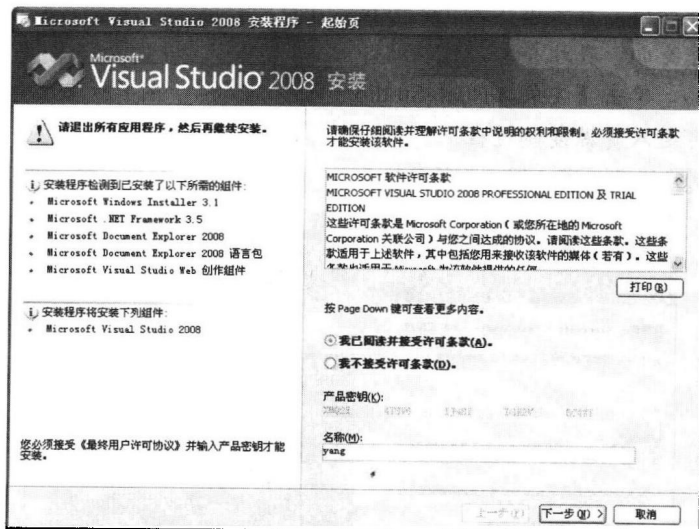


图 1-5 软件安装许可认证界面

- (6) 在图 1-6 中单击【安装】按钮后,进入图 1-7 所示的安装过程显示界面。
- (7) 在安装 Visual Studio 2008 之后进入图 1-8 所示的界面。图 1-8 显示了该软件已经成功安装在本地计算机上。

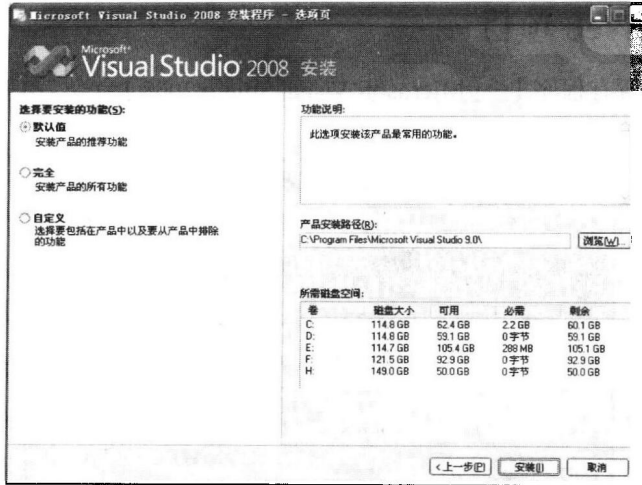


图 1-6 安装路径选择

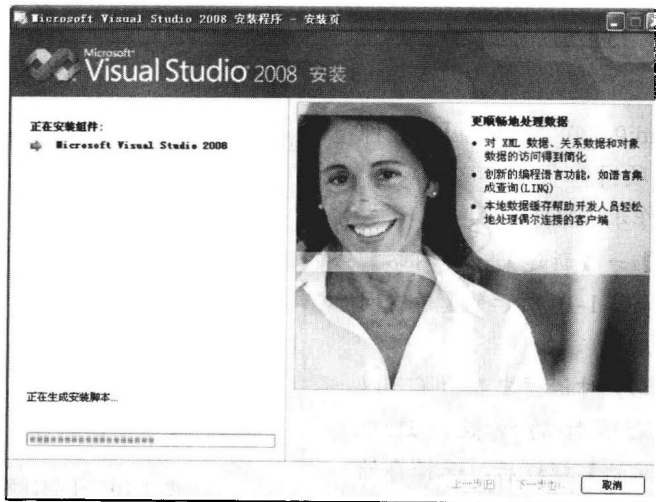


图 1-7 安装过程的显示

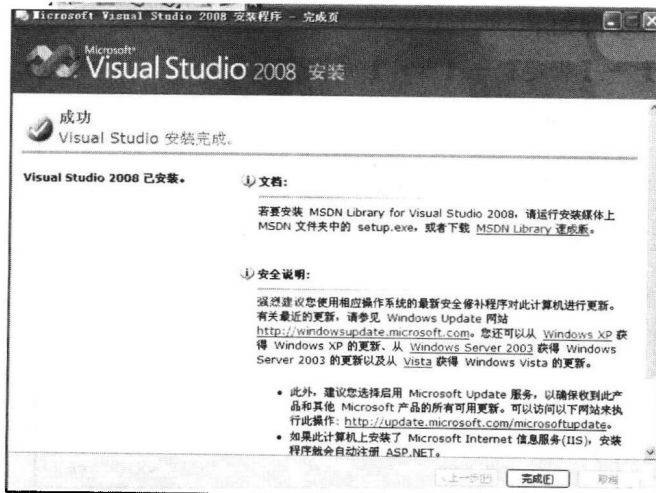


图 1-8 成功安装提示界面



## 1.2.2 创建 Web 项目

选择【开始】→【所有程序】→【Microsoft Visual Studio 2008】命令，打开【Visual Studio 2008】主界面，如图 1-9 所示。

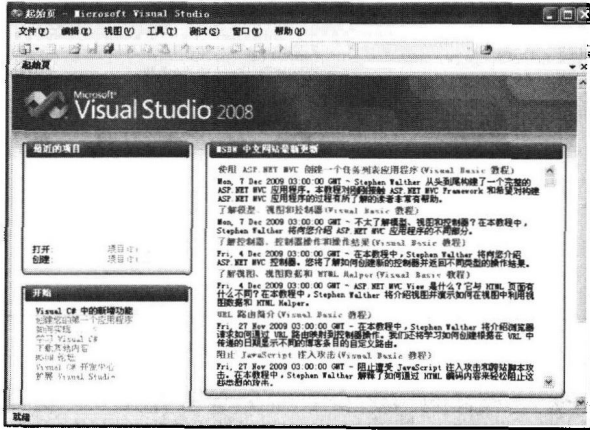


图 1-9 Visual Studio 2008 的主界面

在启动 Visual Studio 2008 之后，可用 3 种方式来创建一个 Web 项目。

第 1 种方式的操作步骤如下。

(1) 选择【文件】→【新建】→【网站】命令，打开如图 1-10 所示的【新建网站】对话框。

(2) 在【新建网站】对话框中显示了可以创建的网站项目的模板，选中【ASP.NET 网站】模板后，选择存储位置并输入名称，然后在【语言】下拉列表中选择“Visual C#”选项，单击【确定】按钮，一个新的网站项目即可创建。

第 2 种方式的操作步骤如下。

(1) 选择【文件】→【新建】→【项目】命令，打开如图 1-11 所示的【新建项目】对话框。

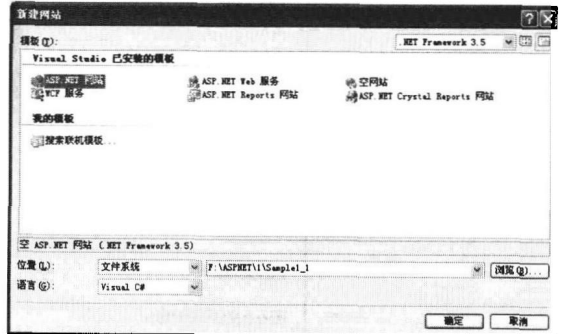


图 1-10 【新建网站】对话框

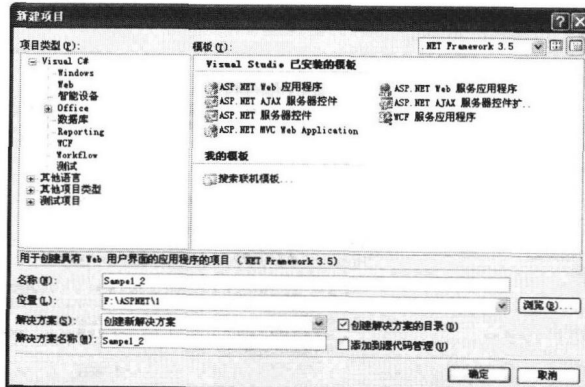


图 1-11 【新建项目】对话框