

微软.NET程序设计系列

Microsoft Press

- ◆ 欧美读者评价 ★★★★★
- ◆ Web 权威专家精心编著
- ◆ 入围欧美电脑图书排行榜
- ◆ 大量可重用的范例代码
- ◆ 赏心悦目的版式设计
- ◆ 附开发职业规划蓝图

MICROSOFT®

ASP.NET

程序设计

[美] G. Andrew Duthie 著

李万伦 何 蕾 赵 海 译
Visual Studio .NET 产品组 审校

Microsoft
.net



清华大学出版社

微软.NET 程序设计系列

ASP.NET 程序设计

[美] G. Andrew Duthie 著
李万伦 何蕾 赵海 译
Visual Studio .NET 产品组 审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

Active Server Pages .NET(ASP.NET)提供了迄今为止最高级的 Web 开发平台。本书是《微软.NET 程序设计系列》丛书之一,循序渐进地介绍了这一革命性的编程框架及其底层技术。全书分为 6 个部分,分别介绍 ASP.NET 基础、ASP.NET Web 开发基础、ASP.NET Web 应用程序、ASP.NET Web 窗体、ASP.NET Web 服务和其他程序设计技巧。在讲解理论的同时,本书还利用初级开发者和经验丰富的开发者每天都要面对的实际问题进行范例讲解。

本书适合准备开发 ASP.NET Web 应用程序的开发者阅读。

Microsoft ASP.NET Step by Step

Microsoft Press

Copyright © 2002 by Microsoft Corporation

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation

All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

图书在版编目(CIP)数据

ASP.NET 程序设计/(美)杜西著;李万伦,何蕾,
赵海译.一北京:清华大学出版社,2002

(微软.NET 程序设计)

书名原文:Microsoft ASP.NET Step by Step

ISBN 7-302-05609-9

I .A... II.①杜...②李...③何...④赵...

III.主页制作—程序设计 IV.TP393.092

中国版本图书馆 CIP 数据核字(2002)第 045600 号

北京市版权局著作权合同登记号:图字 01-2000-3534 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:马云艳

印 刷 者:北京牛山世兴印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×960 1/16 印张:26 字数:660 千字 彩插页:4

版 次:2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

书 号:ISBN 7-302-05609-9/TP·3305

印 数:0001~5000

定 价:48.00 元

《微软.NET 程序设计系列》序

英文版 Visual Studio .NET 是 2002 年 2 月 13 日发布的，之所以选择这一天，原因很多，但其中一个很浪漫的因素是第二天是情人节，比尔想把 Visual Studio .NET 作为礼物送给全世界的程序员。

摩尔定律告诉我们，每过 10 年计算机系统性能将会提高 100 倍，通信带宽也会增加 100 倍，而需要的资金却不会增加。在过去的 30 年中，这种情况确实发生了。由于微软和计算机界的共同努力，PC 进入了家庭，软件也变成了一个巨大的产业。软件开发在过去的数十年里发生了巨大的变化，持续不断的技术进步和创新推动软件工业发展到了一个新的阶段。

这期间，软件开发工具所起的巨大作用是有目共睹的。早期的程序开发，所使用的编程语言是机器指令，编程人员只能使用二进制代码编写程序。由于硬件条件的限制，在编写程序时考虑最多的是如何节省资源，尽可能少的使用 CPU 时间和内存。因此，程序开发难度之大是当今年轻的程序员们难以想象的。后来，技术人员发明了程序设计语言，从汇编语言到高级程序设计语言，包括结构程序设计语言，逻辑/函数程序设计语言和面向对象的程序设计语言。尤其是软件危机所引发的软件工程的革命，软件重用、组件开发等大量新技术的涌现，软件开发的效率大大提高了，同时也大大增强了软件的功能。

大家可能还记得程序设计从字符模式转换到图形界面的情形。那时硬件的速度依然很慢，人们所习惯的是 MS-DOS 的编程规则和 MS Basic、Turbo Pascal 和 Quick Basic 等编程工具。由于没有与图形界面相适应的开发工具，软件开发变得十分困难。幸运的是，Visual Basic、PowerBuilder、Delphi 等几种突破性开发工具的诞生很快就改变了这种状态，使得图形界面下的应用程序开发和字符模式下的开发一样简单。

Internet 的诞生和流行使软件设计发生了质的变化。运行在服务器上的应用程序将结果以 HTML 格式发送给浏览器，而客户端可以随时随地连接到任意一个站点，得到想要的信息。应用程序不需要同时存在于服务器和客户机。主要的应用程序都运行在服务器上，客户机只需安装浏览器。大量的软件开发工具迅速地应运而生。随着大量 Web 应用程序的开发，软件设计人员意识到，仅在一端(服务器端)有智能是不够的，人们真正需要的关键应用程序应该是两端都具有智能，即让不同的计算机针对同一问题协同工作。应该让用户看到来自许多不同 Web 站点的信息，同时容许用户对这些信息进行编辑和加以注释。

事实上，其典型应用即是电子商务。那么，如何为一个复杂的机构建立电子商务呢？比如说，如何将 A 公司连接到 B 公司？A 公司内部可能运行着许多不同的系统，有大量的员工需要与不同的公司分别从事不同的业务。而那些公司的情况也同样复杂。如何来设计这样的软件，使得有关双方可以通过数字化的方案了解彼此的进展和状态，从而与当今基于电话和纸张的沟通相比，大大地提

高工作效率呢？

这就是分布式计算。分布式计算的基本思想是将不同的系统互联互通，这在学术界已经存在很长时间了。今天，分布式计算已成为一种实实在在的需求。因此，必须有一种有效的开发工具来帮助开发人员建立这种分布式计算系统。

1996 年，微软公司和业界其他公司一起，将 SGML 结构化文档标准归纳、概括为一种新的标准——XML。这一标准使数据共享获得新的动力，但仅有 XML 还不能实现互联网上的计算机之间的“真正”相联——不能发现其他计算机所提供的服务，不能彼此发送消息，通过消息共享服务。现实的情况是，计算机之间的连接不一定可靠，双方的软件开发人员也未协调过彼此的开发，甚至另一端的计算机上可能正运行着恶意代码。即便面对这样的现实，开发人员依然必须确保他们的计算机安全可靠。这是一个巨大的挑战，一个计算机行业必须齐心协力方能面对的挑战。

迎接挑战的武器是 XML Web 服务，即建立完全基于开放标准的服务。这些标准确保在不同的系统上生成相同格式的 HTML 文件。鉴于 XML 标准已得到广泛认可，更大的竞争变成谁能以最优的方式实现这些标准。如同在图形界面世界有 Macintosh、OS2、Windows 和其他系统在竞争。众多计算机公司将竞相建立各种 Web 服务，而且这些 Web 服务是可以互操作的。为了避免 Web 服务不兼容的问题，微软公司联合 IBM 及其他在全球计算机界最有影响的 50 家公司一起组成了 WSI，即 Web 服务协调工作组织，共同制定有关 Web 服务的标准。由于这些公司在全球的代表地位和影响，从而确保了基于这些标准的 Web 服务能够很好地协调工作。

这种真正的分布式计算需要新的开发工具来支持，而它不应该只提供一种编程语言。更为重要的是，用户可以不必重新建立他们已有的软件应用系统。

.NET 不仅仅表现为 Web 服务，其根本的目的是通过 XML Web 服务使人们可以在任何时间、任何地点、通过任何设备获取所需要的信息。通过 XML Web 服务实现计算机与手持设备、车载设备、电视等家用电器等的互联互通。

让计算机编程变得更容易，为全球软件开发人员提供功能更加强大、更加好用的开发工具，是微软公司长期坚持不懈为之努力的一项使命。Visual Studio .NET 正是基于这种蓝图而设计的，它为开发下一代以 XML Web 服务为中心的应用程序而设计，是软件开发有史以来功能最强大、最受欢迎的软件开发工具。它有许多激动人心的新功能，提供统一的集成开发环境，支持在同样的开发环境里用 Visual Basic、Visual C++、Visual C#、Visual J# 和其他数十种编程语言，编写、调试和部署各种应用程序。Visual Basic 进化为完全面向对象的编程语言；Visual C#作为一种新的编程语言集中了 C 和 C++的优点；Visual C++在保留原有十分强大的编程能力的基础上，增加了开发托管代码(Managed Code)功能。2002 年初，C# 和 CLI(Common Language Infrastructure, 通用语言架构)已被批准为 ECMA(欧洲计算机制造协会)的标准。

正是基于这些实际的需求，经微软独家授权，清华大学出版社引进并隆重推出了《微软.NET 程

序设计系列》丛书。本丛书从最基础的知识讲起,使用浅显的语言、生动的实例演示,全面而系统地介绍了如何使用 Visual Studio .NET 和 .NET 框架包含的各种功能。旨在循序渐进地带领想了解、掌握 .NET 技术的读者进入 Visual Studio .NET 的美妙世界。一书在握,您将发现,程序开发原来是如此轻松且充满乐趣!

到目前为止,《微软.NET 程序设计系列》共包括 6 本书:

- 《Visual Basic .NET 程序设计》

介绍了 Visual Basic .NET 面向对象编程语言的特点和结构化异常处理机制,如何利用 Windows 窗体和 Web 窗体及各种控件开发用户界面,如何利用 ADO.NET 访问各种数据源,以及如何利用 ASP.NET 技术开发 Web 应用程序;在附录部分,还专门介绍了如何将用 Visual Basic 以前版本开发的应用程序升级到 Visual Basic .NET。

- 《Visual C++ .NET 程序设计》

作者在介绍 C++ 编程特点时,重点介绍了托管代码的扩展方面,如值类型、结构异常处理、委托和事件的关系,如何使用 .NET 框架提供的各种功能开发托管代码,同时也介绍了如何访问 COM 对象和 Win32 API。

- 《ASP.NET 程序设计》

重点描述了如何建立 Web 应用和 XML Web 服务。本书的作者 G. Andrew Duthie 是资深的网络开发技术方面的专家,长期从事 ASP 和 ASP.NET 有关的培训和咨询工作,具有丰富的建立 Web 应用的经验。本书首先介绍了建立 Web 应用的基础知识,以大量的实例和练习,帮助读者迅速掌握如何通过创建和应用 Web 窗体和服务器控件,建立 Web 应用和 XML Web 服务。对于有志从事开发 Web 应用和 XML Web 服务的人士,本书是一本很好的入门教材。

- 《ADO.NET 程序设计》

ADO.NET 是微软 .NET 框架中的数据访问组件,是一种新的数据访问对象模型。本书重点描述了如何访问各种数据源和建立数据集成系统。具体内容还包括:如何在 Windows 窗体和 Web 窗体中,通过 ADO.NET 绑定数据,如何与 XML 交互,以及如何访问早期的 ADO。

- 《Visual Basic .NET 和 Visual C# .NET 面向对象程序设计》

通过 Microsoft .NET 框架中的两个功能最为强大的编程语言: Visual Basic .NET 和 Visual C# .NET,全面阐述面向对象编程(OOP)思想,并借助具体示例和代码来说明面向对象编程的各种概念。

- 《Web 数据库程序设计(.NET 版)》

基础开发类书籍,介绍了如何借助 .NET 技术,构建更智能化、更高效的 Web 数据库。

ASP.NET 和 ADO.NET 都是当前最新的技术。Visual Basic 和 Visual C++ 也是全球最受欢迎的编程语言,拥有最大的用户群,《Visual Basic .NET 程序设计》和《Visual C++ .NET 程序设计》侧重介绍了这两种编程语言,以及如何用它们构建各种 Windows 桌面应用程序和 Web 应用程序。全套书的

写作遵循了循序渐进的原则，从最新技术的基本概念介绍出发，配有大量的实例和练习，逐步引导读者从一个程序设计门外汉变成程序设计高手；可读性很强，是学习掌握 Visual Studio .NET 的最佳教材。

作为 Visual Studio .NET 产品组的开发人员，我们欣喜地看到，《微软.NET 程序设计系列》将.NET 技术更快、更广地带到每个人的身边。同时也衷心希望，所有读者都能从本丛书中充分享受到.NET 的无穷魅力！

微软中国研究开发中心

Visual Studio .NET 产品组

2002 年 6 月

前言

《ASP.NET 程序设计》一书对使用 ASP.NET 开发 Web 应用程序进行了综合的介绍。本书的目标是帮助您掌握创建和使用 ASP.NET 应用程序所需的基本技能。全书分为 6 个部分，每个部分包括若干章，分别介绍特定的主题。学习了这些章节后，您将了解全新的 .NET 平台以及 ASP.NET 在其中扮演的角色，同时还可以掌握创建和使用 Web 窗体、服务器控件和 XML Web 服务的必需技能。

根据您开发和使用 Web 应用程序的水平，您可以从自己感兴趣的部分开始学习，或者完全跳过某些部分。您可以根据下表内容决定从哪里开始学习本书。

阶 段	学习步骤
刚刚开始编程	1. 根据“安装和使用练习文件”部分的指示安装练习文件。 2. 阅读第1~4章，了解.NET框架和ASP.NET的背景知识。 如果了解更多的ASP.NET底层技术，阅读第III部分。 如果想直接学习编程，阅读第IV部分。 3. 根据自己的兴趣学习其余章节。
从传统 ASP 或其他相似技术向 ASP.NET 转换	1. 根据“安装和使用练习文件”部分的指示安装练习文件。 2. 如果对.NET框架和ASP.NET的背景知识感兴趣，阅读或浏览第I和II部分。 3. 阅读第III、IV和V部分，学习基本的Web窗体、Web服务技术和自定义服务器控件开发技能。 4. 阅读第VI部分，了解其他的ASP.NET应用程序技能。
阅读几章后，将本书作为参考	1. 通过目录查找常规信息。 2. 阅读每章后面的“本章小结”部分，简要复习每章的主要内容。

安装和使用练习文件

为了帮助读者节省学习时间，作者提供了本书中所有操作需要用到的练习文件。通过书中的指导，自己动手操作，学用并举，这显然是获取新知识最简单有效的途径，能达到事半功倍的学习效果。请访问 <http://www.wenyuan.com.cn/> 下载资源，找到本丛书的相关链接，下载打包的练习文件。



注意

适合与本书配套使用的软件是 Microsoft Visual Studio .NET Professional，请在学习之

前确保安装了适当版本的 Visual Studio。要了解版本信息，请查看软件的产品包装，或启动 Visual Studio，然后在【帮助】菜单中选择【关于 Microsoft Development Environment】，从中查找相关信息。

系统要求

在使用练习文件前，要确保您已经安装了 .NET 框架的正确版本，并且您的操作系统满足运行 ASP.NET 应用程序的最低要求。

本书是为 ASP.NET 或 .NET 框架的 beta 2 及其以后的版本编写的。如果您安装了 beta 1 或更早的版本，需要更新它们以有效地使用本书的示例文件。示例文件大概需要 500 KB 的硬盘空间。

ASP.NET 应用程序只能运行在安装了 IIS 的 Windows 2000 及以后版本的操作系统上。本书中的范例是在 Windows 2000 中编写和测试的，但是它们应该无需进行修改即可在 Windows XP Professional 或 Windows .NET Server 上正确运行。

安装练习文件

请按以下步骤安装练习文件：

1. 用 WinZip 或其他解压缩软件对下载的练习文件进行解压缩。
2. 确保已经安装了 IIS。要检查是否安装了 IIS，在【控制面板】中打开【服务】小程序，如果已安装了 IIS，IIS Admin 和 World Wide Web Publishing 服务应该都已经被启动。如果已安装了这些服务并且启动了它们，跳至第 3 步。
3. 如果没有安装 IIS，选择【控制面板】|【添加/删除程序】|【添加/删除 Windows 组件】安装它。如果第 2 步中的服务没有被启动，右击该服务，选择【启动】。也可以将这些服务设置为自动启动，只要右击服务，选择【属性】，然后将【启动类型】改变为【自动】。



要点

IIS 的默认安装并没有配置为可以安全地在 Internet 上使用。如果第一次安装 IIS，应先阅读第 8 章的内容，并根据第 8 章建议的步骤，在将服务器连接到 Internet 之前为服务器进行安全设置。如果没有遵循第 8 章建议的步骤，您的服务器可能会被攻击。

4. 运行练习文件根目录下的 StartCD.exe 程序。
5. 开始菜单将为您提供多个选项。选择【安装示例文件】。
6. 根据屏幕指示安装示例文件。
7. 使用本书的练习后，可以卸载示例文件。如果要卸载示例文件，在【控制面板】的【添加/删除程序】中删除“Microsoft ASP.NET SBS”即可。

除了使用安装程序安装示例文件，也可以直接浏览这些文件。为了易于查找，示例文件按照章号

组织。



注意

有些示例文件需要使用 Visual Basic .NET 或 C# 的命令行编译器。为简化编译过程，练习文件中也包括了批处理文件，但是为了使用这些批处理文件，您需要将包含命令行编译器的文件夹的路径添加到 PATH 环境变量中。在默认情况下，编译器位于 %windir%\Microsoft.NET\Framework\%version% 文件夹中，其中的 %windir% 是 Windows 目录，而 %version% 是 .NET 框架的版本号。要找到您的系统中的这些值，在【资源管理器】中打开 Windows 目录，依次打开 Microsoft.NET 节点及其 Framework 子节点。版本号应该是 Framework 文件夹下的文件夹的名称。

在 Windows 2000 中添加该文件夹的路径的步骤如下：

1. 在桌面上右击【我的电脑】图标，选择【属性】。
2. 选择【高级】标签，单击【环境变量】按钮。
3. 在【环境变量】对话框中【系统变量】中选择 Path 变量，单击【编辑】按钮。
4. 将包含 vbc.exe 和 csc.exe 程序的文件夹的路径添加到【变量值】文本框的后面，用分号隔开。
5. 依次单击【确定】按钮，关闭各个对话框。

也要注意，使用其他命令行工具(如 wsdl.exe)时，可能也需要向 PATH 环境变量中添加路径。如果在运行任何命令行工具时出现问题，可以使用 Windows 2000 的【搜索】工具(选择【开始】|【搜索】|【文件或文件夹】)查找这些工具的位置。然后将它们的路径添加到 PATH 环境变量中。

本书中的约定和特色段落

本书在版式上力求紧凑美观、便于阅读与学习。书中采用了一些统一的格式规范，以及几种特色段落，提供更简便快捷的学习指南。

约定

书中使用了一些特殊的文本格式，从而有助于更加快捷地找到所需信息：

- 练习中的中文屏幕项(如菜单名、对话框名、标题名及按钮名)均用黑体方括号括起来，以与其他文字区分，如“【文件】菜单”、“【确定】按钮”等等。
- 为了表达更清晰，使用了简写的菜单命令。如“选择【文件】|【新建】|【项目】”，表示首先单击【文件】菜单，指向【新建】，然后单击【项目】。
- 键名之间的加号(+)表示同时按下这些键。如“使用组合键 Alt+Tab”表示同时按下 Alt 键和

Tab 键。

- 键名之间的逗号(,)表示依次按下这些键。如“使用组合键 Alt, F4”表示依次按下 Alt 键和 F4 键。
- 此外, 本书使用一些特殊的字体格式来组织内容。下表描述了本书中使用的字体格式。

字体格式	表 示	例 子
黑体	首次出现的术语、关键词汇	生成是编译项目中所有代码模块的过程。
加粗字体	需要读者输入的内容	输入 XvectorA 。
Courier New 字体	键入的程序代码	<pre>` Visual Basic Public Class Book End Class // Visual C# public class Book { }</pre>
正体英文	布尔值	True, true, False, false
	类名	Book, Library, Train
	控件类型	ListBox, TextBox
	数据类型	String, string, Integer, int
	事件	Click
	窗体	Form1
	命名空间	ReadBooks
	参数类型	String, string, Integer, int
	属性	Name

特色段落

书中有两类特色段落, 一类带有图标, 有如下几种:



提示 对可能的结果或替代方法进行说明。



要点 提供完成一项任务所需的重要信息。



注意 提供与所讨论主题相关的其他信息。

另一类特殊段落没有图标, 用楷体表示。这类特殊段落提供了正文之外更深层次的信息, 包括调

试信息、设计提示信息或读者可能希望进一步了解的主题。

本书每章都以“本章小结”部分结束。小结提供了每章所介绍的语法与技巧的简要回顾。

勘误、建议和帮助

为了保证本书内容及附加练习文件的准确性，微软出版社通过多种渠道提供支持。请到微软支持网站查询勘误和附加信息，网址为：<http://mspress.microsoft.com/support/>。

如果读者对本书内容或附加练习有疑问、意见或建议，请通过下面的方法反馈到微软出版社：

电子邮件：

mspinput@microsoft.com

邮政地址：

Microsoft Press

Attn: Microsoft ASP.NET Step by Step Editor

One Microsoft Way

Redmond, WA 98052-6399

此外，也可向本书中文版编辑室发送电子邮件，以反馈意见。邮件地址请见本书封底上的“读者服务邮箱”。

请注意，以上地址并不提供对 Visual Studio .NET 软件的产品支持。若要获取有关产品支持的帮助信息，请访问：<http://support.microsoft.com>。

微软出版社网站

微软出版社网站的网址为：<http://www.microsoft.com/mspress>。

微软出版社在网站上提供的信息有：微软出版社书籍的完整列表、相关事件，以及图书的附加内容等。

此外，微软公司还在网站上提供了关于 Visual Studio .NET 软件的最新信息，网址为：

<http://msdn.microsoft.com/vstudio/nextgen/>。

经历了三个月的紧张翻译，这本书终于完成了。严谨、求实和高质量一直是我们的工作目标，但软件的时效性不容我们精雕细琢，错误之处在所难免，欢迎广大读者批评指正。

在此特别感谢微软(中国)公司，以及陈志全先生与燕晓群女士，正是由于他们的大力支持，本书才得以顺利出版。

目 录

前言	xi
----------	----

第 I 部分

ASP.NET 基础

第 1 章 ASP.NET 概述	3
1.1 Microsoft .NET 简介	3
1.1.1 .NET 框架	4
1.1.2 企业版服务器	5
1.1.3 语言和语言工具	8
1.2 ASP.NET 体系结构	11
1.2.1 原有特性	11
1.2.2 新特性	11
1.3 本章小结	12
第 2 章 使用 ASP.NET 开发工具	13
2.1 Notepad “.NET”	13
2.1.1 最便宜的开发工具	14
2.1.2 创建虚拟目录	14
2.1.3 其他文本编辑器	16
2.2 Visual Studio .NET	17
2.2.1 创建 ASP.NET Web 应用程序	17
2.2.2 创建新 ASP.NET 页 (Web 窗体)	18
2.2.3 添加控件	19
2.2.4 保存和浏览页	20
2.3 本章小结	21

第 II 部分

ASP.NET Web 开发基础

第 3 章 ASP.NET 开发概述	25
3.1 ASP.NET 项目类型	25
3.2 ASP.NET 文件类型	27
3.3 本书中使用的语言	28
3.3.1 Visual Basic .NET	28
3.3.2 C#	29
3.3.3 使用多种语言	29
3.4 Visual Studio .NET 特性	29
3.4.1 IDE 的增强特性	29
3.4.2 新特性	30
3.4.3 窗口	31
3.4.4 工具栏	34
3.4.5 菜单	35
3.4.6 选项	36
3.5 本章小结	37
第 4 章 编程基础	38
4.1 表达式、变量和常量	39
4.1.1 表达式	39
4.1.2 变量	39
4.1.3 变量声明	41
4.1.4 常量	45
4.2 过程	45
4.2.1 Sub 过程	46
4.2.2 Function 过程	47
4.3 流控制	48
4.3.1 If 语句	48

4.3.2	Select Case 语句	49
4.3.3	循环语句	50
4.4	错误处理	52
4.4.1	使用 On Error 语句	53
4.4.2	使用结构性异常处理语句: Try...Catch...Finally	54
4.5	面向对象编程的基础	55
4.5.1	将类作为代码容器	55
4.5.2	使用继承	56
4.6	本章小结	58

第III部分

ASP.NET Web 应用程序

第 5 章 创建 ASP.NET Web 应用程序 61

5.1	用 Visual Studio .NET 创建 Web 应用程序	61
5.2	用 .NET SDK 创建 Web 应用程序	62
5.2.1	创建应用程序根目录	62
5.2.2	创建子目录	65
5.2.3	创建.aspx 页	65
5.2.4	创建 Global.asax 文件	68
5.2.5	创建 Web.config 文件	70
5.3	本章小结	71

第 6 章 管理应用程序状态

6.1	使用应用程序状态	72
6.1.1	同步访问应用程序状态	74
6.1.2	对应用程序状态的建议	75
6.1.3	应用程序状态的局限性	75
6.2	使用会话状态	76
6.2.1	启用会话状态	78
6.2.2	在 Visual Studio .NET 里 启用会话状态	78
6.2.3	会话状态管理建议	79
6.2.4	会话状态的局限性	79

6.3	状态和可伸缩性	79
6.4	配置对会话状态的存储	80
6.4.1	存储进程内会话状态	81
6.4.2	存储进程外会话状态	81
6.4.3	在 SQL Server 里存储 会话状态	82
6.4.4	使用无 cookie 会话	83
6.4.5	为无 cookie 会话设置 格式	83
6.5	用客户端 cookie 存储状态	84
6.6	ASP.NET 服务器控件状态	85
6.7	本章小结	85

第 7 章 配置 ASP.NET 应用程序

7.1	了解 ASP.NET 配置	87
7.1.1	Web.config 简介	88
7.1.2	编辑配置文件	89
7.1.3	配置 ASP.NET 应用程序	90
7.1.4	重写子目录下的配置设置	91
7.1.5	锁定配置设置	91
7.2	ASP.NET 配置元素	92
7.2.1	<trace>	92
7.2.2	<globalization>	94
7.2.3	<httpRuntime>	95
7.2.4	<compilation>	97
7.2.5	<pages>	101
7.2.6	<customErrors>	103
7.2.7	<authentication>	105
7.2.8	<identity>	108
7.2.9	<authorization>	109
7.2.10	<machineKey>	110
7.2.11	<securityPolicy>	112
7.2.12	<trust>	112
7.2.13	<sessionState>	113
7.2.14	<httpHandlers>	115

7.2.15	<httpModules>	116
7.2.16	<processModel>	118
7.2.17	<webControls>	123
7.2.18	<clientTarget>	123
7.2.19	<browserCaps>	124
7.3	设置和检索自定义应用	
	程序设置	125
7.4	本章小结	126
第 8 章	ASP.NET 中的安全性	127
8.1	安全的重要性	127
8.2	安全基础	128
	8.2.1 服务器设置和应用	
	程序设计	129
	8.2.2 密码	135
	8.2.3 限制账户	138
	8.2.4 不安装示例	138
	8.2.5 使用验证	139
	8.2.6 打补丁	140
	8.2.7 访问控制	141
	8.2.8 审计和日志	141
	8.2.9 用 SSL 保护通信	142
8.3	启用验证	144
	8.3.1 选择验证类型	145
	8.3.2 使用基于 Windows 的	
	验证	145
	8.3.3 使用 Passport 验证	146
	8.3.4 使用基于窗体的验证	147
8.4	使用授权	151
	8.4.1 使用 NTFS ACL 授权	151
	8.4.2 使用基于 URL 的授权	152
8.5	使用模拟	154
8.6	代码访问安全性	154
8.7	安全资源	155
8.8	本章小结	156

第 IV 部分

ASP.NET Web 窗体

第 9 章	创建 Web 窗体	159
9.1	ASP.NET Web 窗体解析	159
	9.1.1 理解页面元素	160
	9.1.2 了解页面生存期	162
	9.1.3 使用指令	163
	9.1.4 @ Page 示例	167
	9.1.5 @ Control 指令	169
	9.1.6 Page 类	173
	9.1.7 在 Web 窗体中编写代码	174
	9.1.8 创建和使用用户控件	176
	9.1.9 使用服务器控件	181
9.2	事件处理	183
	9.2.1 处理页面事件	184
	9.2.2 处理控件事件	186
	9.2.3 处理页面错误	187
9.3	页面运行时结构	187
	9.3.1 查看页面的控件树	187
	9.3.2 在运行时添加和操纵	
	控件	188
9.4	在 Web 窗体中使用代码隐藏	188
9.5	本章小结	191
第 10 章	使用服务器控件	192
10.1	控件的类型	192
	10.1.1 HTML 控件	192
	10.1.2 Web 控件	198
	10.1.3 向页面添加服务器控件	198
	10.1.4 其他 Web 控件	203
	10.1.5 特殊控件	217
	10.1.6 验证控件	223
	10.1.7 数据绑定控件	228
	10.1.8 其他控件	228

10.2 本章小结	229
第 11 章 访问和绑定数据	231
11.1 了解 ADO.NET	231
11.1.1 了解 .NET 数据提供程序	232
11.1.2 了解数据集	233
11.2 创建和打开连接	234
11.2.1 SqlConnection	234
11.2.2 OleDbConnection	237
11.2.3 使用信任连接	238
11.3 用命令读取和更新数据	239
11.3.1 SqlCommand	240
11.3.2 OleDbCommand	243
11.3.3 使用存储过程	246
11.4 使用数据集	247
11.4.1 使用数据适配器	248
11.4.2 读取 XML 数据	248
11.4.3 使用 DataTables、DataColumns 和 DataRows	249
11.4.4 更新数据集	254
11.4.5 类型化数据集	255
11.4.6 使用数据视图	258
11.5 用数据阅读器读取数据	259
11.5.1 SqlDataReader	259
11.5.2 OleDbDataReader	259
11.6 数据绑定	260
11.6.1 简单的数据绑定	260
11.6.2 绑定控件	261
11.6.3 使用 DataBinder.Eval	262
11.6.4 使用数据绑定控件	262
11.7 本章小结	276
第 12 章 创建自定义服务器控件	278
12.1 创建控件	279
12.1.1 创建命名空间	279
12.1.2 创建类	280

12.1.3 从基类继承	280
12.1.4 呈现控件输出	281
12.1.5 编译控件	282
12.1.6 小结：创建新控件	283
12.1.7 注册控件	284
12.1.8 在页面中使用控件	284
12.1.9 小结：使用自定义控件	285
12.2 添加功能	287
12.2.1 添加属性和方法	287
12.2.2 创建、触发和处理事件	291
12.2.3 处理回发	293
12.2.4 维护状态	301
12.3 通过撰写创建自定义控件	303
12.3.1 重写 CreateChildControls	303
12.3.2 实现 INamingContainer	303
12.4 创建模板控件	304
12.5 扩展已有控件	309
12.6 本章小结	312

第 V 部分

ASP.NET Web 服务

第 13 章 创建和使用 Web 服务	317
13.1 理解基于 XML 的 Web 服务	317
13.1.1 SOAP 和 Web 服务	319
13.1.2 Web 服务的其他要求	319
13.2 创建 Web 服务	320
13.2.1 声明 Web 服务	320
13.2.2 创建 Web 服务类	320
13.2.3 公布 Web 服务	328
13.2.4 Web 服务的安全性	330
13.3 使用 Web 服务	331
13.3.1 查找 Web 服务	332
13.3.2 理解 WSDL 文件	334
13.3.3 创建代理类	334

13.3.4 创建客户端 Web 窗体页	336	15.3.1 用【复制项目】部署 Web 应用程序	366
13.3.5 创建客户端控制台应用 程序	338	15.3.2 用【Web 安装项目】部署 Web 应用程序	367
13.4 本章小结	340	15.4 本章小结	368
第VI部分		第 16 章 跟踪和调试 ASP.NET	
其他程序设计技巧		应用程序	
第 14 章 使用缓存提高性能		370	
14.1 了解缓存	343	16.1 跟踪	370
14.2 使用输出缓存	344	16.1.1 启用页面级别的 跟踪输出	371
14.2.1 使用@ OutputCache 指令	344	16.1.2 启用应用程序级别的 跟踪输出	371
14.2.2 使用 Response.Cache	351	16.1.3 写入跟踪输出	372
14.3 缓存任意数据	351	16.1.4 理解跟踪输出	374
14.4 本章小结	357	16.2 调试	375
第 15 章 部署 ASP.NET 应用程序		375	
15.1 ASP.NET 应用程序的结构	358	16.2.1 错误类别	375
15.1.1 物理路径与 URL	360	16.2.2 预防错误	376
15.1.2 存储应用程序特有 的配置设置	362	16.2.3 在调试模式下编译 Web 窗体页	376
15.2 手动部署 Web 应用程序	363	16.2.4 理解 ASP.NET 的 错误消息	377
15.2.1 设置目标部署目录	363	16.2.5 调用运行库调试器	378
15.2.2 把文件复制到目标目录	363	16.3 本章小结	381
15.2.3 为子目录设置 IIS 权限	365	附录 A 将 ASP 迁移到 ASP.NET	
15.3 Visual Studio .NET 中的部署 选项	366	383	
		附录 B 其他程序代码	
		392	