

Mastering ASP.NET with C#

ASP.NET与C# 从入门到

〔美〕 A. Russell Jones 著

陈建春 白 雁 杨永平 等译

精通

- 全面涵盖使用Visual C#.NET进行服务器端编程
- 详细介绍利用ASP.NET和ADO.NET访问数据
- 强力构建C# Web应用程序和XML Web服务



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

Mastering ASP.NET with C#

ASP.NET与C# 从入门到精通

〔美〕 A. Russell Jones 著

陈建春 白 雁 杨永平 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

.NET是Microsoft公司在快速发展的软件技术世界中维持领先的保证, ASP.NET则是开发.NET应用程序的语言之一。本书旨在介绍如何使用ASP.NET框架和代码隐藏类的C#开发Web应用程序, 并通过大量翔实的实例说明用ASP.NET开发应用程序所涉及的原理及技术基础。本书首先从Web应用程序、HTML基础知识及动态Web应用程序开始讲述; 接着介绍System.Web命名空间、常用的基础对象以及ASP的调试和错误处理、文件和事件日志访问、消息处理; 然后是数据库访问工具和XML、应用程序的状态维护与缓存方法、访问控制与监控方法; 最后以Web服务、COM组件、SOAP工具箱以及如何创建自己的Web控件为结束。

本书适合于对C#和.NET框架有所了解的网络编程人员阅读, 也可作为高级开发人员的工具手册。



Copyright©2002 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版, SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号: 01-2002-2681

图书在版编目(CIP)数据

ASP.NET与C#从入门到精通/(美)琼斯(Jones, A.R.)著; 陈建春等译. —北京: 电子工业出版社, 2003.3

书名原文: Mastering ASP.NET with C#

ISBN 7-5053-8476-7

I. A… II. ①琼… ②陈… III. ①主页制作—程序设计 ②C语言—程序设计 IV. ①TP393.092 ②TP312

中国版本图书馆CIP数据核字(2003)第006314号

责任编辑: 李 莹

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 45.125 字数: 1150 千字

版 次: 2003年3月第1版 2003年3月第1次印刷

定 价: 72.00元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换, 若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

致 谢

在此我非常感谢Sybex编辑组的具有突出才干的编辑们，特别是Richard Mills、Tom Cirtin、Erica Yee、Gene Redding、Denise Santoro Lincoln、Mike Gunderloy等，感谢他们自始至终耐心细致地给予本书支持和帮助。同时感谢技术新闻组的工作人员，他们不计报酬，花费了很多时间来解答我的问题，而且他们的工作产生了卓有成效的结果。

译 者 的 话

王斌、王波、陈紫芊、王子逸、沈东升、周君、周敏、高镇宇、李勇、张跃平、宋朝阳等同志不同程度地参与了本书的翻译工作。在本书的翻译过程中，得到了曾学贵教授、魏庆朝教授等的大力帮助，杜薇、李洁等同志完成了本书的翻译工作，在此深表感谢。

简介

在过去的20年当中，程序设计的重点一直在服务器和客户机之间交换着。从主机的批处理到独立的应用程序，再到客户机-服务器直至网络程序，程序开发的焦点始终根据当前可获得的硬件、软件以及通信模式而来回改变。从电传打字机到终端，从大型主机到微机到现代服务器，从台式机到膝上电脑到手提设备，从固定线路直接连接到私有网络到因特网，程序员已经将他们的精力集中在改进用户界面或建立为运行用户界面的设备提供数据服务的后台系统上。在20世纪80年代和90年代初期，微型计算机的迅猛发展迫使开发人员把注意力转向后者；为此，如今的计算机购买者喜欢高分辨率多颜色的显示器、声音和语音设备、快速处理器、大容量数据存储设备、廉价内存以及功能强大的交互式图形操作系统。

因为人们可以在自己的机器上使用各自的文件进行工作，因此微机的迅速发展导致了相应的数据分割。有趣的是，正是这种分割导致了网络能力的迅速增长，因为公司业务需要员工们共享信息，同时他们也需要将数据集中起来，以进行安全控制。这些需要促使了客户机-服务器计算模式的发展，使人们可以利用快速集中的数据库将大量的图形用户界面与快速的微机处理过程结合起来。

遗憾的是，正如人们开始设想的一样，客户机-服务器计算模式引出了几个问题。“胖”客户程序在部署、安装、维护、升级等方面都很困难。而企业需要的是一个不同种类的客户应用程序，用于从集中服务器上接收数据和程序代码，然后像以往使用桌面应用程序一样，显示数据并与之进行交互。万维网和浏览器技术的出现似乎解决了这个问题。

在过去的几年中，我们已经看到了“瘦”客户的复苏。人们为此通常采用的做法是，让浏览器或者小的可执行程序依照命令从中央服务器检索数据，这和早期大型机终端所做的工作差不多。尽管新的瘦客户比早期的大型机终端有更多的功能，但仍旧不能完全满足公众对其早已熟悉的商业应用程序（例如Microsoft Office、Quicken以及成千上万的定制的客户机-服务器应用程序）的需求，因为它们是那样的丰富多彩。

然而，尽管有这些缺陷，但是基于HTML前端代码的浏览器已经改变了这个世界。人们和企业越来越依赖于地点无关性，希望可以在任何地方、任何时间访问任何服务器，而且能够在实现这种愿望的道路上越走越顺畅。地点无关性导致了使用上的简易性，使浏览器和其他远程客户端普遍存在。

遗憾的是，浏览器并没有完全代替丰富的桌面客户程序，而且使得许多人感到似乎又回到了几十年前。浏览器在提交静态数据（例如报告、文档以及图形）的时候表现得非常好，但是将其用到客户机-服务器、窗体驱动、数据接口中时，表现得就相当不好了。因为它在人们使用指点设备浏览网页时，以及应应用程序的要求输入数据时，总显得有些磕磕绊绊，不太平稳。

我坚信.NET有改变这种情形的能力。使用.NET框架，就有可能创建交互性更强、反应更快的集中处理式应用程序，同时，改进实用工具并简化构建多功能客户的过程。另外，通

过使用.NET，还能在集中存放和管理数据以及便于提供多功能和瘦客户（记住，也不能功能太多或太瘦）之间架起一座桥梁，这意味着用户可以继续使用他们熟悉的图形控件和操作习惯，而开发人员可以按照需要进行动态更新，以实现对应应用程序的集中管理。

本书有哪些内容

这是一本探究新知识的书，同时也是一本详细说明概念的书。微软的.NET框架是为非常庞大和复杂的实体而精心设计的，但它也太庞大、太复杂了。在写本书期间，我面临的最大问题不是应该去写什么，而是哪些东西可以略去不写，并且这个令人头疼的问题一直困扰着我。本书中我想要囊括的东西太多，但是时间、篇幅、前期写作时.NET框架和Visual Studio的动态变动、以及本人对.NET框架不成熟的认识都阻止了我实现这个目标。

本书的核心思想是，介绍.NET提供的一种建立Web应用程序的全新模式、实现这种模式的两种新语言（C#和VB.NET）以及服务器端JScript的扩展版本——JScript.NET。

对于那些从VB转到C#的程序员们，我想说几句题外话。我个人以为，VB.NET是一种全新的语言，它与传统VB（以前所有版本的VB）相关的只是名称和一些共享的语法，除了这些以外，其他的东西都已经改变了。然而，人们将发现C#比其他任何使用类C语法的程序语言都更接近于VB的理念，同时Visual Studio .NET（VS.NET）使得应用C#更加直接。事实上，在使用了VB许多年以后，我开始憎恨编码时区分大小写的规定，但由于Visual Studio .NET（VS.NET）中增加了实时智能选择技术，人们可以不再那么操心了（C#是一种区分大小写的语言）。

如果各位曾经开发过Web应用程序，则不管采用的是什么技术，都已经领先于一名普通的程序员了，因为通过开发你们已经知道网络是如何工作的了。微软在Visual Studio和ASP.NET应用程序中花费了很多时间（或许非常成功），致力于将网络是如何工作的隐藏起来，与此对应，我在本书中也花了相当多的时间，尽力解释ASP.NET应用程序是如何简化这项工作的。在某些方面，ASP.NET和C#很像传统的VB，VB适于建立一个规模适度、效率平平的视窗程序，而这二者适于建立适度规模、平平效率的Web程序。

要明白，尽管Visual Studio .NET和.NET框架改变了网络编程，但网络本身并不因为.NET而发生丝毫的改变，它仍旧是面向网页的无状态的通信机制。使用C#建立网络应用程序很容易忘记这一点，我认为使用.NET进行Web应用程序设计时最大的威胁就是.NET已成功地将复杂性隐藏在丰富的编程模式之下。然而，复杂性并不会因为通过Visual Studio的过滤之后就会消失。它们仍旧躲在某一隐蔽的地方，在人们不注意的时候出来咬人一口。

幸运的是，.NET不仅使以前的复杂工作变得容易，而且提供了接近重点和抓住关键的工具，以利于暴露复杂问题，进而实现简单化。在编写本书的过程中，我使用.NET工作了近一年的时间，我彻底相信.NET和类似的系统实现了编程的伟大变革。虽然本书并不绝对要求使用Visual Studio来创建项目，但是如果没有Visual Studio的内容，那么读者必定会非常不满。

尽管Visual Studio将大多数网络编程技术集成到了一个独立界面，能够辅助与简化HTML代码以及其他一些文件格式的编写，但要成为一名完全合格的网络程序员，还需要知道很多的相关技术，而且这些技术都比较复杂。其中包括：

C# 用来创建类、检索和操纵数据以及处理事件的语言。

超文本标记语言 (HTML) 一种格式化布局语言，用于设计用户界面。

级联样式表 (CSS) 一种用以说明网页对象显示样式的方法，它具有健壮性、可扩展性以及分级的特点。

JavaScript/JScript/ECMAScript 在客户浏览器中用来管理网页对象的编程语言。JScript是微软版权所有的ECMAScript。JavaScript的名字最初由Netscape采用。

说明：不要将客户端的JScript与微软新的JScript.NET语言相混淆。JScript与JScript.NET的关系就像C#与C++的关系一样，它们的语法相似但语言不同。

可扩展标记语言 (XML) 在Visual Studio和.NET中，可以把标记语言看做是用来存储和管理数据库检出数据的一种方式；指定应用程序配置信息的一种格式；保持数据和对象的一个方法；以及从一个组件或层向另一个组件或层传递信息，对象和数据的一个通用容器。

扩展样式表语言 (转换用) (XSL/XSLT) 为了将XML文档从一种状态转换到另外一种状态而专门创建的XML词汇表。这种转换可以从XML到XML、从XML到文本或者从XML到任何其他类型的文件。

XML模式 (XSD) 为了将XML文档从一种状态转换到另外一种状态而专门创建的XML词汇表。这种转换可以从XML到XML、从XML到HTML、从XML到PDF文档、从XML到文本或者从XML到任何其他类型的文件。

文档对象模型 (DOM) 该模型用于操纵创建在文档树结构中的对象。这个文档可以是XML或者HTML。例如，可以使用.NET XML命名空间类来操纵使用XML文档存储的对象，然而按照传统，需要使用JavaScript来操纵构成一个HTML网页的DOM对象。在某些情况下，可能甚至需要使用老版本的基于COM组件的MSXML解析器来操纵在Internet Explorer (IE) 中作为数据岛存储的XML。这个解析器也提供DOM对象和方法，虽然它们与在.NET中的稍微有点不同。

动态HTML (DHTML) 一种技术的名称，用来操纵创建在浏览器中的对象，并响应由这些对象或用户初始化引发的事件。使用支持DHTML的浏览器，例如IE和Netscape，可以指定位置、内容并显示网页中的每一个对象的特征。换句话说，DHTML采用的是不同于静态HTML的显示方式，使得它就像一个独立的Windows应用程序一样能够快速响应。

在微软以前的网络编程系统 (VB 6中的WebClasses以及Visual InterDev中的ASP) 中，程序员必须编写原始的HTML。虽然这一版本的Visual Studio在消除必须要知道HTML方面做了一个大胆的尝试，但它仍然没有完全成功。因为程序员需要在创建C#应用程序的同时具备少量的HTML知识，因此在本书中我包含了对HTML的简单介绍。如果读者一直在使用FrontPage或Dreamweaver来设计网页，以避免学习如何编写原始的HTML代码，那么我建议你应该仔细学习关于HTML的介绍，直至完全习惯于用文本编辑器来编写HTML，否则在使用编程语言直接生成HTML时就会感到很困难，而很多Web应用程序要求必须这么做。

谁应该阅读本书

本书主要面向对C#和.NET框架知之较少的网络编程初学者。我们虽然并不要求阅读本书的读者是有经验的C#网络程序员，但也不能是十足的初学者。除了那些与ASP.NET和网络编程有关的内容之外，本书没有解释C#语言或者框架本身。如果读者已经听过介绍C#编程的课程，建立过几个C#窗口或者控制台应用程序，甚至已经通读了指定的C#编程书籍，那么就对理解本书的代码不会感到太困难。

除了具备一点C#的知识外，读者不必知道任何有关因特网、内联网、浏览器、HTML、JavaScript、VBScript、XML、XSLT、DOM或者其他与网络相关的技术。这是一本启蒙书籍，详细讲述了使用C#和ASP.NET进行网络编程的基础知识，介绍了其他网络技术的相关内容，以利于读者使用C#创建出健壮性、可扩展性兼备的Web应用程序。

我为什么要写本书

我之所以写这本书是因为我对程序设计和编写代码的过程非常入迷。我已经写了另外两本有关网络编程的书籍：一本是关于使用Visual Basic 6进行WebClass编程的书籍，叫做《Visual Basic开发指南——ASP和IIS篇》（Sybex, 1999），另外一本是《Active Server Pages 3从入门到精通》（Sybex, 2000）。这两本书都卖得相当好，但这不是我写书的原因。在编写本书的实践过程中，我有理由和借口去学习更加广泛的技术，而不只是将.NET看成创建Web应用程序的工具。同时，这样广泛的学习为我提供了既有广度又有深度的信息，使我解决了用其他任何方法都无法解决的一些疑难问题。由于我坚信.NET和类似的环境是编程未来发展的方向，所以我想要把这种信念传播给大众（其实也是为自己从它的试用版本到最后的发布版都一直用这种技术编程而找理由）。

我喜欢学习计算机语言。到目前为止我已经从事程序设计20多年了，并且在传统的ASP出现以前就开始编写网络程序。沿着这条路，我学习并使用了大量的计算机语言。尽管我决不是也不想成为任何编程语言或技术上的专家，但我在Visual Basic、数据库、网络编程、XML、XSLT以及本书中讨论的其他技术方面都有广泛的经验。

我的教育背景是科学和生物、音乐、计算机培训（CBT）、交互式视频培训（IVT），近期又参加了基于Web的培训（WBT）、数据库应用以及通用人力资源（HR）网络应用。在决定不从事医学领域的工作以前，我是一个预科生；而且我在田纳西州的诺克斯维尔动物园工作了几年，在动物园馆长John Arnett的带领下我最终成了一个保护爬行动物的首领，那时整天在第十大爬行动物集散地工作。但是我对爬行动物的一丝好奇最终被微薄的薪水耗完，转而对音乐感兴趣了，然后我返回学校主修音乐，学习钢琴和音乐理论。

我第一次涉足计算机是在1979年，那时我还是田纳西大学钢琴班的一名大学生，我发现Donald Pederson博士的音乐理论电脑实验室里放满了8K主存的新款苹果II微型计算机（信不信由你）。从那里回来后，我就决定要开始学习电脑。我的第一个真正的程序是告诉人们如何识别和书写乐谱——这是音乐理论课的一部分。

这项工作激起了我对计算机的痴迷，而且这份痴迷一直保持到现在。在获得了音乐理论的硕士学位后，我进入伊利诺斯大学攻读中等教育方面的博士学位。这个学校最先使用了

一个重要的计算机教学系统，叫做PLATO。作为科研助手，我与Esther Steinberg博士（《Teaching Computers to Teach》一书的作者）一起工作，深入研究了各种界面特征与初学者和计算机使用专家之间的相关性。毕业以后，我在InterCom, Inc.公司工作，用了大约12年的时间建立起基于计算机的培训程序和人力资源（HR）应用程序。直到那个阶段的后期，我才开始编写技术论文，我的第一篇技术论文是为Fawcette的Visual Basic程序员期刊和XML杂志写的，然后我又开始为Sybex写书。从2000年起，我主要为VF公司的Playwear部门效力，它是世界上最大的服装制造商之一；现在又为DevX, Inc. (www.devx.com) 公司工作，开始时作为网络开发员，现在是执行编辑，负责编写、审阅以及编辑涉及到所有网络相关技术的程序设计论文。

从本书中读者将学到什么

本书将介绍如何以特定的方式（通过使用代码隐藏类来建立Web应用程序）使用C#和ASP.NET框架。在传统的ASP中，程序员可以将执行代码和HTML混合在同一个文件中。在ASP.NET中，虽仍然可以这样做，但本书描述的技术更像VB6的WebClasses，使用HTML模板与VB编译代码创建的DLL相关联。DLL可以访问HTML模板并为它们填充数据，从而实现用户界面（HTML）和代码的分割。

C#中的代码隐藏类遵循相同的逻辑，但相对要容易使用一些。最简单地，可以创建一个叫做Web表单的HTML模板，其中包含用户界面元素。在Web表单中，可以引用隐藏代码文件中类的代码；然后，从这个C#类中编写出HTML元素的内容。像WebClasses一样，从模板中分离出激活HTML模板的代码可以实现更进一步的分离。举例来说，假设我们定义了一批用户界面元素，希望在不影响页面正常工作的情况下，使用HTML设计器构建这个用户界面，并且通过增加静态元素或改变这些元素的位置或外观来修改这个界面，毫无疑问，使用C#很容易完成这种操作。同样地，可以重复使用用户界面模板，使用不同的数据填充它们或者将它们从一个应用程序拷贝到下一个应用程序而不必重新创建用户界面。

由于这些原因，使用ASP.NET框架和代码隐藏类来编制C# Web应用程序是本书采用的基本技术。本书大致有一半的内容都在致力于解释如何使用与探究Web表单，但我也已经提到过，有几种用来建立Web应用程序的辅助技术大家必须要知道，例如HTML和CSS，或者知道但至少是听说过诸如使用ADO.NET进行数据访问、网络服务、高速缓存数据、编写组件和服务、XML以及使用XSLT来转换XML文档等。

如何阅读本书

真正的网络初学者通过阅读本书的前几章应该能够受益，这几章将讨论网络如何工作并简单介绍HTML。已经知道HTML和CSS或者具备了传统ASP编程经验的读者可以跳过这些技术，而且不会影响学习后面的章节。

本书不是一本参考书，而是一本陈述性、探索性的书籍。随着本书的进程，我们将建立一个相当大的Web应用程序和几个相关的应用程序，每一章都将包含整个项目的一个子目录。我们对这些应用程序没有做成型的规划，除了要求它们提供一个用来进行说明的框架外别无所求。到本书结束时，我们将拥有一大批Web以及其他一些.NET特性，例如用户控件、

合成控件以及网络服务等，它们都包含了基本的功能，可以在开发应用程序过程中加以利用。

虽然可以从Sybex网站（www.sybex.com）安装这些示例代码，但我建议不要通过这种方法使用本书。相反，应该手工输入每一章的代码，如果遇到不能解决的问题或错误，再进行拷贝。通过这种方式，可能会发现解决问题的捷径和更好的方法，并且找到自己的工作方式。在通读本书的过程中，应注意书中一些代码的变化。因为随着内容的推进，完成某项任务的代码（例如一个循环）可能会发生变化。在有些情况下这些变化是故意的；因为有许多方法可以解决同一个问题，所以我在代码中使用了不同的例子，但并不区分哪一个是最有效的方法或最完美的语法。有些人喜欢这种语法，另外一些人可能喜欢另外的语法。在其他情况下，代码的变化反映了我自己认识上的变化和使用.NET框架与C#语言经验的增加。还有极个别情况是因为在编写本书的过程中，.NET框架本身完善并变化了。

本书中没有涉及什么

本书探讨的是一种很具体的技术（使用C#后台代码类开发ASP.NET Web表单），而且直接面向Web开发的初学者。书中的代码并未完全成型，不能在应用产品开发过程中直接拷贝并重用，设计这些代码只是为了介绍.NET是如何工作的以及编程细节，因此读者要自己建立并调试应用产品的代码。

本书并未全面列出所有方法和属性，虽给出了一些这样的列表，但是不多。读者可以通过在线的.NET框架和Visual Studio文档以及其他书籍找到它们。

本书中没有的东西想必会在其他的书中出现，并且可能已经出现了。因为我想集中讲述一些基础性的东西：为浏览器客户建立Web应用程序。然而即使这样，由于篇幅所限，我也不得不删除许多有趣的相关主题。例如，高级的DataGrid操纵技术，如何建立商业定制控件的方法，关于使用.NET进行电子商务应用或网页设计的具体指导，以及如何使Web应用程序全球化或者如何集成微软其他的.NET技术（例如Passport和MyServices）等，本书都未加论述。但是如果读者想从隐藏代码类的观点学习.NET Web表单，则我认为本书不但有趣，而且会给大家提供一些思路。

目 录

第一部分 Web程序设计基础	1
第1章 后台环境的介绍: Web应用程序如何工作	1
Web请求是如何工作的	1
客户机请求的内容	3
Web服务器如何响应——准备阶段	3
Web服务器如何响应——执行阶段	5
客户机如何处理响应	6
动态Web页面	8
C#可以做什么	10
C#在Web应用程序中的优势	12
小结	14
第2章 HTML基础	15
什么是HTML	15
语法: 标志和属性	17
格式化文本	25
在Web站点中加入图像	35
超链接	39
格式化表格	46
使用图像映像	51
了解框架	54
控制元素定位	57
级联样式表	62
小结	66
第3章 动态Web应用程序	68
什么是动态Web应用程序	69
什么是数据以及数据在什么地方	73
如何检索数据	74
如何将数据打包	76
什么是用户界面	79
小结	80

第二部分 用Visual C#进行服务器端Web编程	81
第4章 ASP.NET简介	81
什么是ASP.NET	81
为什么需要使用ASP.NET	85
ASP.NET可以做什么	87
为什么要在关于C# Web的书籍中介绍ASP.NET	88
创建第一个Web表单	91
小结	97
第5章 Web表单	98
Web表单是服务器端对象	98
表单的有效性验证	107
代码隐藏设计	117
页面/表单布局	119
小结	121
第6章 System.Web命名空间	122
从用户处获取信息	122
向客户端发送信息	145
HttpApplication和HttpApplicationState对象	152
小结	165
第7章 SessionState对象	166
介绍会话状态	166
cookie、URL再生和ASP.NET会话对象	169
Session变量	175
ASP.NET会话状态	180
将数据与个体联系起来	180
小结	189
第8章 HttpServerUtility对象	190
CreateObject和CreateObjectFromClsID方法	190
调用服务器对象方法和#include命令执行外部代码	192
Server.HtmlEncode、HtmlDecode、UrlEncode和UrlDecode方法	200
Server.GetLastError和ClearError方法	206
小结	207

第9章 调试ASP.NET和错误处理	208
安装一个调试站点	208
使用调试器	211
ASP.NET事件的顺序	214
保护性编码：如何避免错误	225
小结	236
第10章 用ASP.NET访问文件和事件日志	237
访问文件	237
使用文件	238
创建文件和目录	252
写入文件	255
发送和接收文件	263
访问Windows注册表	270
小结	279
第11章 用ASP.NET发送和接收消息	280
消息类型	280
小结	310
第三部分 用ASP.NET访问数据	311
第12章 关系型数据库和SQL	311
数据库与文件的对比	311
表、索引、主关键字和外部关键字	317
基于集合的数据检索	320
介绍SQL	322
小结	330
第13章 ADO.NET	331
System.Data命名空间	331
ADO.NET的数据访问	347
小结	358
第14章 访问数据	359
确定性的最终化	359
只检索需要的数据	362
介绍存储过程	364
改进数据访问代码	377

DataList和Repeater Web控件	382
删除、更改和增加数据	388
如何写应用程序	403
小结	405
第15章 在Web应用程序中使用XML	406
介绍System.Xml命名空间	406
读XML文件	414
使用XmlDataDocument类	424
XmlException类	427
在程序中执行XSLT转换	428
转到应用程序	434
小结	435
第四部分 C# Web应用程序	437
第16章 C# Web应用程序	437
Web应用程序与Web站点的比较	437
再次介绍web.config和machine.config文件	444
站点层次与目录层次的比较	451
内部配置文件	451
创建定制的配置节	453
配置文件的存放定位和锁定设置	459
小结	462
第17章 状态维护和高速缓存	463
状态维护选择	463
使用缓存对象	485
缓存ASP.NET页面	490
众多的选择	494
小结	495
第18章 访问控制和监视	496
实现用户名/密码的安全性	496
使用SSL确保服务器安全	512
监视站点——超越页面计数	516
停止应用程序进行维护	522
小结	523

第19章 规划应用程序	524
进行设想	524
确定应用程序的用户	525
确定应用程序的需求	526
将自己的设想与别人共享	528
规划用户界面	530
规划数据存储和检索	531
规划对象的功能和界面	532
规划管理功能	532
创建数据库	533
创建数据访问组件	535
创建业务组件	535
建立用户界面	537
小结	538
 第五部分 高级Visual C# Web应用程序	 539
第20章 平衡浏览器客户	539
哪种客户端脚本更可行	539
将脚本发送到浏览器	541
文档对象模型	563
从脚本访问DOM	564
使用ActiveX控件	567
小结	573
 第21章 Web服务	 574
介绍Web服务	574
创建Web服务	579
使用Web服务	589
简单对象访问协议	598
查找Web服务	602
小结	603
 第22章 Web服务、COM组件和SOAP工具包	 604
为什么要担心旧的技术	604
创建客户应用程序	605
创建服务器应用程序	612
小结	616

第23章 创建自己的Web控件	617
创建用户控件	617
创建复合控件	625
创建自定义服务器控件	633
在ASP.NET Web表单和组件之间传递数据	642
小结	642
第24章 效率和可扩展性	643
始终要考虑效率	643
响应时间与开发代价的比较	648
硬件与软件的比较	650
关于效率和可扩展性的忠告	651
小结	653
后记	654
网络开发的时效性强	654
浏览器不兼容导致的问题	654
浏览器很好，但不是长久之计	654
XML很重要	655
现在应该学习什么	655
第六部分 附录	657
附录A 快速HTML参考	657
附录B JScript 5.5参考	678