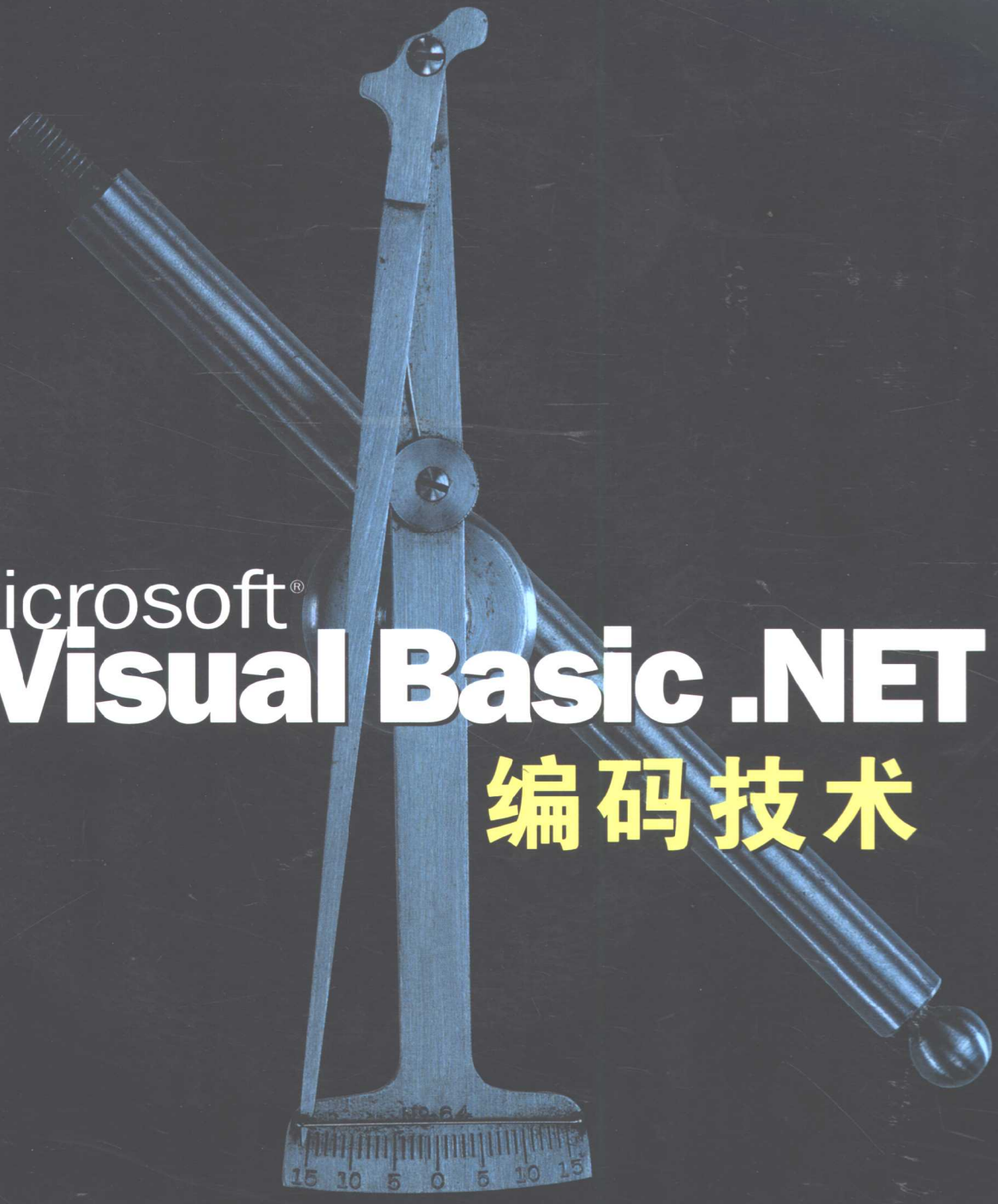
Microsoft® 程序设计系列



Microsoft® Visual Basic .NET 编码技术

Microsoft .net

北京大学出版社

Microsoft Visual Basic .NET
编码技术

[美] John Connell 著

莱恩工作室 译

北京大学出版社

·北 京·

著作权合同登记　图字：01-2001-3133 号

Coding Techniques for Microsoft Visual Basic .NET

John Connell

图书在版编目(CIP)数据

Microsoft Visual Basic .NET 编码技术/(美) 克内尔(Connell, J.)著；莱恩工作室译. —北京：北京大学出版社，2002.8

ISBN 7-301-05136-0

Ⅰ.M…　Ⅱ.①克…　②莱…　Ⅲ.BASIC 语言-程序设计　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 027475 号

书　　名：Microsoft Visual Basic .NET 编码技术

著作责任者：[美] John Connell 著　莱恩工作室　译

责 任 编 辑：邱淑清　范　彦　张济青　徐　涛

标 准 书 号：ISBN 7-301-05136-0/TP·0568

出　版　者：北京大学出版社

地　　址：北京市海淀区中关村北京大学校内　100871

网　　址：http://cbs.pku.edu.cn

电　　话：出版部 62754962　发行部 62754140　编辑室 62757065

电 子 信 箱：zpup@pup.pku.edu.cn

排　印　者：北京大学印刷厂

发　行　者：北京大学出版社

经　销　者：新华书店

787 毫米×1092 毫米　16 开本　38.25 印张　750 千字

2002 年 8 月第 1 版　2002 年 8 月第 1 次印刷

定　　价：82.00 元(含光盘)

简介

在过去的几年时间里，我曾经很幸运地、使用了目前正为美国某些“财富 500”(Fortune 500)公司所采用的 Microsoft Visual Basic 来设计和编写了一些产品程序。我还曾做过一些技术专家的教师、老板和管理者，这些工作使我吸取了许多 Visual Basic 程序员的思想。在我上大学期间，我又曾反复阅读了一些对程序员来说非常有意义的 Visual Basic 的编程知识，以及像基于对象的编程方式等这些使有些程序员感到迷惑的领域。在此基础上我撰写了《Microsoft Visual Basic .NET 编码技术》，该书将会带您进入程序设计即将到来的新的高度，在您巩固每天使用的编程知识的同时，此书将为您揭开您尚不确知的编程技术的神秘面纱。

本书的适用读者

本书是由一位 Visual Basic 程序员为其他 Visual Basic 程序员编写的。在描述如何使用 Visual Basic .NET 的过程中，我首先打了个“地基”，提供了一些在计算和软件开发方面的变化的背景知识，以了解 Microsoft .NET Framework 的有关知识，这对于程序员具有非常重要的意义，并且是他们在实践中必须掌握的。我介绍了使用 Visual Basic .NET 进行面向对象的编程的基础内容，并解释了如何构建您自己的类，以及如何使用 .NET Framework 类，如何使用数组和集合，以及如何调试和处理程序中的错误。站在这个地基上，我们可以攀登到更高一级。我详细介绍了如何使用 .NET 程序集，如何使用文件和数据流，以及如何通过网络监视文件，包括如何构建在服务器上运行的 Windows 服务应用程序。利用三个整章的篇幅，我介绍了使用 Visual Basic .NET 和 ADO.NET 编写进行数据访问的程序都有哪些变化。然后，我们将进入 Web 服务的世界——设计在 Internet 上运行的程序和组件。在最后一章中，我将对前面几章的内容做一个总结。通过这种方式，您将会看到大量非常有用和有意义的示例代码。

通过编写工作程序来学习 Visual Basic .NET

在我曾读过的大多数计算机图书中，不管它们所讲述的是哪种编程语言，作者都会提供一些理论上的代码片段，以说明其要点或构造。虽然这种方法很有帮助，但是它使读者不知如何将一段代码用于完整的工作程序的较大方案中。我曾发现，学习一种新的计算机语言（如 Visual Basic .NET），最好的方法是用这种语言编写完整的工作程序。在心中制定一个目标——并编写一个解决某个问题的程序——涉及编程语言的多个方面，并努力使各个部分适于在一起工作。我在本书中就采用了这种方法，通过一些示例应用程序，向您演示了 Visual Basic .NET 的一些要点。

如果您是从其他编程语言（如 C、C++、Java，甚至 COBOL）转移到 Visual Basic .NET，那么您不需要花费太多的时间就可以适应这一切。Microsoft .NET Framework 是未来的潮流，并且 Visual Basic 程序最适合利用这种新的技术。《Microsoft Visual Basic .NET 编码技术》将使您熟悉 .NET 技术的基础知识，我确信，您将会很快看到 .NET 的强大和方便，并且将会开始以一种新的令人激动的方式进行编程。最后需要说明的是，您将会发现这一过程也是非常有趣的。

本书的具体内容

下面我将介绍每章的重点，并概述在阅读本书的过程中您将会学到的知识。

- **第一章“Visual Basic .NET 基础”** 我从介绍 .NET 开始，并解释了为什么说它是 21 世纪编程的革命性方法。.NET Framework 一个主要的好处是，它能够只编写一次程序，就可以自动确定任意的硬件或操作系统。当程序员需要创建用于桌面 PC 的应用程序及用于 Internet 的应用程序时，这种灵活性是非常重要的。我回顾了一下 Visual Basic 的发展历程，从一种带动 Windows 编程潮流的计算机语言发展到全面成为主流的 Visual Basic .NET。我从较高的角度介绍了 .NET Framework 的一些主要功能，例如，类框架、公共语言运行库、Web 服务、程序集，以及新的 Visual Studio .NET 交互式的开发环境（IDE）——使用这些新功能的场所。您将首先看到一些 Visual Basic .NET 代码，以对编写 Visual Basic .NET 程序的需要有初步的了解。阅读完第一章之后，您会很好地理解我们将要进入的世界，以及什么是重点。
- **第二章“使用 Visual Basic .NET 进行面向对象的编程”** Visual Basic .NET 现在是一种完全面向对象的语言，因此，它最终将会加入到像C++ 和 Java 那样所谓的专业化语言的行列。如果对于面向对象的编程方式，您是一名新手，那么您将会发现本章很容易使您熟悉这一切。为了说明对象是如何从类派生而来

的，我使用了一个简单的 Visual Basic .NET 窗体，并说明了属性和方法、继承性，以及命名空间。我还介绍了共享变量、重载、多态，以及封装。因为 .NET 的大部分功能都来自于 .NET Framework 提供的基类，所以我展示了如何使用各种命名空间，以及如何访问这种内置的功能。

- **第三章"编写您的第一个类"** 在第二章的基础上，我在第三章中解释了如何编写您自己的类，然后如何创建从这个基类继承得到的第二个类。您还将了解有关 *imports* 指令的内容、如何为项目添加程序集、如何使用共享的成员变量，以及为什么要使用和什么时候使用 *Option Strict* 和 *Option Explicit* 指令。在构建类的同时，您将会使用到允许您在实例化基础上初始化对象的重载的构造函数。在总结第三章的过程中，您将对面向对象的编程方式有清楚的理解，并且会惊奇地发现，它实际上是多么引人注目。
- **第四章"Visual Basic .NET 的数据类型和特性"** 在理解 Visual Basic 早期版本中引用和值（原始）数据类型是非常有用的同时，由于比较和初始化对象（在 .NET 中，一切都是以对象的形式出现）的方式，因而理解 Visual Basic .NET 中的这些概念也是非常重要的。在 .NET 中，数据类型是严格分类的（每个变量必须有一个特定的数据类型），并且它们还是类型安全的（只能通过它的数据类型来访问变量）。当不再需要某个变量时，将通过一种不确定的结束算法把它标记为将要被删除，也称为"垃圾回收"。如果您总是将您的引用变量设置为 *Nothing*，那么您将会对 Visual Basic .NET 处理释放的资源和内存的方式感兴趣。完成第四章之后，您将会很好地理解在 Visual Basic .NET 中变量是如何被调用、初始化和放弃的。
- **第五章"使用文件和字符串介绍 .NET 类框架"** .NET 类框架的面向对象的、分层的类库是 .NET 强大的后盾。从命名空间开始，我从基础讲述了框架，解释了框架的组织方式，并使用一些具体的示例来准确地阐明了它的使用方法。对于作为面向对象的编程方式新手的您来说，本章将准确地向您展示如何在框架中查找您需要的内容，以及如何恰当地使用它。使用内置的工具 Windows Class Viewer（Windows 类查看器），您将会全面地掌握框架，并快速地掌握您所需要的技术。在这个过程中，我描述了用来指定参数、重载构造函数和方法，以及返回类型的 C# 类表示法。在本章主要的示例中，我展示了如何通过框架来访问文件和流类，以及它们如何用来读写磁盘。我还介绍了一些字符串及其新的、不变的特性。本章还演示和说明了一些复制、克隆和格式化字符串的技术。
- **第六章"Visual Basic .NET 中的数组和集合"** 正如您可能期望的，数组在 Visual Basic .NET 中与在 Visual Basic 的早期版本中的处理方式是不同的。.NET Framework 类 *System.Array* 是用于各种数组类型的基类。因为数组都是对象，所

以您所创建的每个数组都有它自己的内涵——它所包含的元素数目、它所包括的维数、它的边界，等等。最好的是，通过使用 *System.Array Sort*，现在可以自动地排序和逆向排序 Visual Basic 数组。不仅如此，还可以使用各种方法（如内置的二进制搜索）来搜索 .NET 数组。我将会创建一个将数字转换成罗马数字的计算器程序，以说明数组的使用方法。数组实际上是一个简单的集合，而集合是一组对象。集合可以从 .NET Framework 的 *System.Collection* 命名空间继承得到。*Collection* 命名空间包含用来创建新对象（如数组列表、散列表、队列、堆栈，以及字典等）的接口和类。在这些数据结构中，其中一些可能对于 Visual Basic 程序员来说是新内容。在第六章中，我编写了一个模拟不确定的罗杰兰（Rogerian）心理学家的程序。我使用了 .NET 数组的一些高级功能来完成这项工作。用户可以使用 Visual Basic .NET 运行这个有趣的项目，来查看人与机器的心理交互过程。

- **第七章“处理错误和调试程序”** 虽然错误不会使程序崩溃，但是不处理的错误可能会使程序崩溃。（语法、运行时或逻辑）错误随时都可能出现，并且它们会产生一些异常情况。Visual Basic .NET 使用一个结构化的 *Try…Catch…Finally* 的构造，取代了在 Visual Basic 早期版本中使用的非结构化的 *Goto ErrorHandler*。由于结构化的错误处理被内置到 .NET Framework 的核心部分，所以我们可以立即利用它的这种功能。在本章中，我使用了在第六章中创建的计算器程序，展示了在该程序中可能出现的各种错误，以及如何使用结构化的错误处理来处理各种潜在的错误。我还介绍了调试器。Visual Studio .NET IDE 为程序员提供了一些调试窗口，使他们可以查看所运行的程序中从变量值到程序集代码的各项内容。我还编写了一个通用的错误处理类 ErrorTrace.vb，它可以跟踪日志，并且可以将它添加到任意的 Visual Basic .NET 程序中。最后，我展示了如何通过 Visual Basic .NET 程序写入 Windows NT 或 Windows 2000 的事件日志。
- **第八章“程序集的详细内容”** 程序集是 Visual Basic .NET 程序的构造块。它们是用于部署、版本控制、重用和安全性的基本单元。在本章中，我构建了一个名为“AssemblySpy”的程序，它检查了用与 .NET 兼容的语言编写的 .NET 程序集的内部结构。该程序在它的用户界面中使用了一些新的图形化的 .NET 控件，并提供了一些有关静态和实例字段、属性、事件、方法和构造函数的信息。我介绍了私有和共享程序集的一些好处，以及为便于版本控制和共享而创建“严格命名”的程序集的原因。严格命名的程序集允许一个接一个地执行，以使在同一个目录下具有相同名称的两个程序集都可以运行。Microsoft .NET 程序是在具有严格命名的程序集知道将要使用哪个程序集的基础上编译的，从而消除了给 Windows 编程带来麻烦的 DLL 冲突。

- **第九章"文件系统的监视"** *System.IO* 命名空间中的 *FileSystemWatcher* 类被内置到.NET Framework 中。当更改、创建、删除或重新命名文件或目录时,该类将会触发事件。通过将它添加到从 *FileSystemWatcher* 继承得来的、名为"*SystemObserver*"的类中,我将详细地介绍这个类。我还将解释一些有关委托的新的意见,它们使程序员可以定义和响应他们的事件。我将一些委托添加到响应 *FileSystemWatcher* 的事件的 *SystemObserver* 类中。我还构建了一个名为"File Sentinel"的 Windows 程序,并导入了这个 *SystemObserver* 类。该程序提供了一个允许用户选择文件或目录的用户界面。File Sentinel 可以监视任意的文件或目录(如 cookie 文件),并在发生一些重要的事情时通知您。在本章的最后,我向您展示了如何开发 Windows Service(Windows 服务)应用程序(严格地说是"NT 服务")。我将把 *SystemObserver* 类转换成 Windows 服务,以说明我们的代码的重用性。
- **第十章"使用 ADO.NET 进行数据访问"** ADO.NET 组件得到重新设计,以提供更加统一的对象模型,同时还为 Internet 编程提供了更大的可伸缩性。可以通过使用新的断开的 *DataSet* 对象来做到这一点,它为在客户端展示和操纵数据提供了一种常见的方法。传统数据访问的编程需要保持与数据库的连接,而 ADO.NET 则完全是在断开状态下进行的。这种方法使用的是一些被称为"受管理的提供者"(managed provider),它们包括连接对象、命令对象、*DataReader* 对象,以及各种 *DataAdapter* 对象等。ADO.NET 最吸引人的地方是,内存中的数据集现在可以以基于文本的 XML 形式来表示它的内容,这使它可以通过任意的 HTTP 端口 80 防火墙来回地传递,从而使在不同类型的系统和非 Windows 系统之间共享数据成为现实。
- **第十一章"数据集的详细内容"** 在本章中,我深入研究了 ADO.NET 对象模型。我从编写一个说明数据集如何以 XML 形式表示它们的内容的程序开始,介绍了方案和数据的 XML 表示法。我还展示了数据是如何在本地得到操纵,以及是如何在之后将所做的更改写回到数据源中。我还演示了如何使用 Visual Basic .NET 提供的工具 Xsd.exe,它可以采用 XML 文件,并根据它生成一个 XML 方案定义(XSD)文件。然后,这个 XSD 文件可以被用来创建 Visual Basic .NET 类,这个类知道如何添加、删除和修改以前不知道的 XML 文件。接下来,我们以编程的方式在数据集中构建了一个 *DataTable* 对象,并动态地在字段间添加了关系,以建立一种父/子关系。来自数据集的数据将以 XML 的形式保持在磁盘中,然后数据网格控件被用来从 XML 文件重新构成数据。数据网格不知道或不关心信息是来自数据库还是来自 XML 文件——无论是哪种情况,都包含构建自身所需的全部信息。新的 *DataView* 对象向您展示了如何为一个数据表创建两个不同的视图。

- **第十二章“ADO.NET 中的数据绑定”** 本章通过回顾 *BindingContext* 对象完成了我们对 ADO.NET 的介绍。因为断开记录集的 ADO.NET 没有游标的明确实现，所以我描述了如何使用 *BindingContext* 对象及其相关的 *CurrencyManager* 对象在记录间定位。我利用一个在单个表的记录间定位的程序说明了这项工作的完成方法。在本章中，我还以编程的方式构造了一个表，并为它添加了一些记录。我们还在该表中搜索了一些特定的记录，并对这些记录进行了一些操纵。
- **第十三章“ASP.NET 和 Web 服务”** 在 Visual Basic .NET 中，您可以像构建 Windows Form(Windows 窗体)一样方便地构建 Web Form(Web 窗体)。同样，这也使用了统一的对象模型。我还将介绍 ASP.NET 是如何提供简化的开发过程，并同时提供改善的可伸缩性。我介绍了一些新的服务器端的图形化控件，以及一个新的“代码背后”的概念，它使您可以区分用于商务逻辑的代码和用于用户界面的代码。我们还构建了一个投入使用的 ASP.NET 贷款计算器程序，以完整地回顾 Web Form、对象状态、服务器端的控件、回递、字段验证器、变量缓存、用来动态构造数据表的数据绑定，等等。进入到 Web Services(Web 服务)，您将会看到 SOAP(Simple Object Access Protocol，简单对象访问协议)是如何通过发送触发远程服务器上的 API 方法的纯文本，来突破指定协议的障碍。Web 服务可以通过利用“Web 服务描述语言”(Web Services Description Language，WSDL)来宣传它们的 API 和数据类型。我使用一个 Magic 8(魔术 8)球的示例程序和一个享用 Magic 8 球 Web 服务的 Windows 程序说明了这些概念。创建和享用 Web 服务是我最喜欢的 Visual Basic .NET 的部分。
- **第十四章“可视继承和自定义控件”** 正如您在前面已经发现的，Visual Basic .NET 中的一切都是以对象的形式出现的，包括 Windows 窗体。因为我们可以从现有的对象继承，所以在本章中，我们将构建一个标准的窗体，它提供了一致的外观和感觉，然后从基础窗体继承以构建一致的子窗体。我们还将构建一个自定义的 Visual Basic .NET 控件。我还包括了一个有趣的项目，它采用了您在本书中学到的知识，并应用了这些知识——一个在您的计算机屏幕上放置黄色的电子便笺的程序。该程序将自动保存便笺的内容、大小和位置，并在下次运行该程序时重新构建所有现有的便笺。我介绍了如何部署 Visual Basic .NET 程序，并创建了一个在 Windows 2000、Windows Me 或 Windows XP 机器上部署便笺项目的安装项目，即使在这些计算机中还没有安装.NET 公共语言运行库。

关于附带的光盘

所有的示例代码都包含在本书附带的光盘中。这些代码已经使用 Microsoft Visual

Studio .NET 的测试版 2,在运行安装有 Service Pack 2 的 Microsoft Windows 2000 的计算机上进行了测试。要使用 Web Forms 示例,必须安装"Internet 信息服务"(Internet Information Service,IIS)。您应该在安装 Visual Studio .NET 之前安装 IIS。如果您先安装了 Visual Studio .NET,那么您可能会看到下面的错误消息:

Error while trying to run project: Unable to start debugging on the Web server. The server does not support debugging of ASP.NET or ATL Server applications. Run setup to install the Visual Studio .NET server components. If setup has been run verify that a valid URL has been specified.
You may also want to refer to the ASP.NET and ATL Server debugging topic in the online documentation. Would you like to disable future attempts to debug ASP.NET pages for this project?

要解决这个问题,请试着按照下面的步骤进行操作(要了解更多的信息,请参见 Visual Studio .NET 联机帮助的有关内容):

1. 确保在"添加/删除程序"中使用"添加/删除 Windows 组件"安装了 IIS 和 Microsoft FrontPage 2000 Server Extensions。
2. 在"添加/删除程序"中卸载.NET Framework。
3. 重新运行 Visual Studio .NET 的安装程序,然后重新安装.NET Framework。

在第十章、第十一章和第十二章开发的有关 ADO.NET 的示例程序还需要 Microsoft SQL Server 2000,其中一个程序还需要 Microsoft Access 2000 或 Microsoft Access 2002。您必须拥有安装和运行这些应用程序的足够权限。

需要注意的是,在本书出版时,Visual Studio .NET 的最终版本还没有发布。虽然这些程序都是在 Visual Studio .NET 测试版 2 中测试的,但是,最终版本中的变化可能需要对示例程序做一些小的修改。

系统要求

您需要具有下面的软件才能运行本书附带光盘中的示例:

- Microsoft Visual Studio .NET;
- Microsoft Windows 2000 或 Microsoft Windows XP;
- IIS 5 或更高版本;
- Microsoft SQL Server 2000;
- Microsoft Access 2000 或 Microsoft Access 2002。

支持帮助

我们所付出的每一份努力都是为了确保本书及其附带光盘内容的准确性。如果您

遇到任何问题或麻烦，可以参考下面的资源：

Microsoft 出版社通过 World Wide Web 提供的图书修订内容：

http://www.microsoft.com/mspress/support/

如果您对本书或附带光盘有什么意见、问题或想法，那么请使用下面的联系方式将它们发送给 Microsoft 出版社：

电子邮件：

mspinput@microsoft.com

邮政地址：

Microsoft Press
Attn: Coding Techniques for Microsoft Visual Basic .NET Editor
One Microsoft Way
Redmond, WA 98052-6399

请注意，以上地址不提供任何产品方面的支持。

目录