



Fast Track Visual Basic .NET

Written and tested for final release of .NET v1.0

高效掌握 Visual Basic .NET

Billy S. Hollis

Rockford Lhotka 著 康博 译



清华大学出版社

<http://www.tup.com.cn>

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



高效掌握 Visual Basic .NET

Billy S. Hollis 著

Rockford Lhotka

康 博 译

清 华 大 学 出 版 社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3205

内 容 简 介

Visual Basic .NET 是世界上最为流行的编程语言之一 Visual Basic 的最新版本。Visual Basic .NET 对原有的 Visual Basic 6 语言作了许多重要的改进和完善,增添了 VB 程序员所殷切期望的“完全面向对象”、“结构化错误处理”和“新的线程模型”等新特性。

本书的主要目标是使 VB6 开发人员能够快速掌握 Visual Basic .NET 的新特征。书中首先概述 .NET 和引入 .NET Framework;接着介绍 .NET IDE 的新特点、语言和语法变化、用户界面的设计、新的面向对象特征、Web 功能的实现、数据访问;最后介绍了 Visual Basic .NET 的一些高级主题,如应用程序的安装与部署,以及 .NET 的互操作性与迁移等。

本书读者对象主要是针对具有 VB6、ASP 以及 VBScript 丰富应用知识的广大开发人员,和想从其他平台或者语言快速转变为使用 Visual Basic .NET 的程序员。

Billy S. Hollis, Rockford Lhotka: Fast Track Visual Basic .NET

EISBN: 1-861007-12-4

Copyright© 2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

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。

图书在版编目(CIP)数据

高效掌握 Visual Basic .NET/(美)罗克福特等著;康博译. —北京:清华大学出版社, 2002

书名原文: Fast Track Visual Basic .NET

ISBN 7-302-05961-6

I. 高... II. ①罗...②康... III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 060529 号

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

责任编辑: 于平

封面设计: 康博

版式设计: 康博

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 25 字数: 640 千字

版 次: 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号: ISBN 7-302-05961-6/TP·3548

印 数: 0001~5000

定 价: 45.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

作为构成最近新发布的 .NET Framework 的一部分, Visual Basic .NET 无疑是近年来该平台上使用的最流行的开发工具。结果, 很多现有的 Visual Basic 开发人员都需要认真对待该语言的新特征, 以及改进过的老功能。

本书的主要对象是那些富有经验的 Visual Basic 6 开发人员, 他们正在期待未来出现的新事物——Microsoft 公司的 .NET 平台以及与其相关的 Framework。为了把 Visual Basic 移植到这个新的平台上, Microsoft 公司不得不对该语言进行了许多重要的修改, 并添加了如下的新特征:

- 完全面向对象, 包括类继承、方法重载、参数化的构造函数以及共享的类成员
- 结构化的错误处理
- 新的线程模型

本书旨在培养 VB6 开发人员使其能够快速掌握 Visual Basic .NET 新的重要特征。

本书读者对象

本书不仅涉及到了 Visual Basic 的一个新版本, 而且还介绍了一个新的能够运行该语言的编程平台。它主要面向那些具有 VB、ASP 以及 VBScript 丰富应用知识的开发人员。我们没有试图将此书作为 VB 语言的一个指南, 此书在 VB6 和 Visual Basic .NET 代码技术之间作了一些比较, 因此具有良好的 VB6 基础(或者至少是 VBA 及 VBScript)对完全掌握本书的主要内容是很有必要的。

Visual Basic .NET 被作为首要的 Web 开发工具, 这不仅适用于传统的 Web 应用程序, 而且还适用于 Web 服务的创建以及基于 Web 的移动应用程序。

本书还将吸引那些使用 ASP.NET 的 Web 开发人员在开发他们的 ASP.NET 站点时使用 VB 语言。ASP.NET 和 Visual Basic .NET 之间的集成程度远远超出了 ASP 和 VB6 之间的集成程度, 并且充分理解 Visual Basic .NET 的知识对一些使用 ASP.NET 的 Web 开发人员来说也是很重要的。

实际上, 由 Microsoft 公司为 ASP 开发所提供的 Visual InterDev 开发工具已经被取代。在 .NET 环境下 Visual InterDev 的功能等同于 Visual Studio .NET IDE 的功能。

.NET Framework 为 Windows 和 Internet 的程序设计提供了一种新的功能强大的方法, 而且还可以把经验丰富的开发人员从其他的平台和语言吸引到 .NET 和 Visual Basic .NET 语言上来。由于本书提供了经常执行的编程任务和技术的范例, 并允许从其他平台或者语言快速转变为 Visual Basic .NET, 所以吸引了许多读者。

本书内容

本书第 1 章主要对 Visual Basic .NET 进行简单的介绍——安装所需的软件、编码以及运行



第一个“Hello World”程序范例。

所有在前言中提到的新特征都是基于 Microsoft 公司的 .NET Framework。我们将在第 2 章中概述 .NET Framework 的知识，并着重强调其对 VB 开发人员的重要性。

尽管 .NET Framework 为我们提供了上面所列出的期盼已久的新功能，不过它还产生了从旧的 VB 版本到 Visual Basic .NET 转变的一些障碍。因此，这里又产生了一种新的 IDE——Visual Studio .NET。这里还存在大量具有重要意义的语法差别；安装 .NET Framework 还需要对 VB 语言的结构做出一些重要的修改——包括数据类型的合理性以及使 VB 符合其他语言的语法。这些修改提出了一些我们必须明白的困难，使开发人员能够快速、高效并且顺利向 Visual Basic .NET 过渡。这些都是第 3 章和第 4 章的核心内容，为我们提供了新的 Visual Studio .NET IDE 的概述，以及探究对语言的语法和编码的修改。

多年来，VB 开发人员使用内置在 Visual Basic IDE 中的窗体引擎。该窗体环境允许 VB 开发人员快速简便地使用 Windows 用户界面来构建应用程序。.NET Framework 提供了一种众所周知的公共窗体环境 Windows Forms——Visual Basic .NET 使用的就是它而不是 VB6 的窗体引擎，Ruby。Windows Forms 提供了与原来的 VB 窗体引擎具有同样强大功能的 Windows GUI (Windows 图形用户界面)，它还是一种全新的环境，提供了新的性能和新的处理事情的方法。第 5 章紧抓这个新的环境，来说明如何构建 GUI 接口的常用类型以及它的一些现有的新功能。

根据 VB4.0 版本的介绍，VB 开发人员获得了使用类模块来编写面向对象(OO)的程序以及创建可重用的 COM 组件的能力。从那时起，缺少一个完全的面向对象环境就变得很明显，尤其是缺少既可见又完整的类继承，当使用 VB 来构建复杂的系统时，这就成为一个限制因素。可视化继承具有创建一个新的 Windows Form 的能力，该窗体能够继承特性、控件以及现有窗体中的行为。类(或者“实现”)继承具有创建一个新类的能力，该类可以“继承”属性、方法以及现有类中的代码。

Visual Basic .NET 通过提供完整的面向对象性能来正面提出该问题，包括既可见又基于代码的继承、方法重载、方法覆盖等更多的内容。一般来说这些性能对 .NET 环境是完整的，对于 VB 开发人员来说也是很有益处的。我们将在第 6 章中探索这些新的性能，向 VB 开发人员举例说明现在所有可以利用的面向对象程序设计的好处。

除了修改传统的 VB 开发之外，Visual Basic .NET 还将对成千上万的还没有接受动态服务器页(Active Server Pages，ASP)以及与之相关技术的开发人员展现所有 Web 程序设计的可能性。我们在第 7 章中可以快速学习 ASP.NET 环境——粗略掌握如何使用 Visual Basic .NET，基于 Web 的屏幕和软件的开发与使用早期版本的 VB 所进行的 Windows 屏幕和软件的开发十分相似。ASP.NET 使 VB 开发人员开发 Web 系统变得更加容易。

在 Visual Basic .NET 中进行简化转换的关键技术是 Web Forms 和 XML Web 服务。由于这些技术的许多特征对 VB 开发人员来说都很熟悉，所以要避免一些显著的差别，例如与服务器进行通信时扩展使用事件的问题。ASP.NET 本身是一个非常大的主题，但是第 7 章将从 VB 开发人员的角度为我们提供一个很好的概念。

多年来，Microsoft 公司已经开发出一系列 VB 开发人员可以利用的数据存取技术。我们已经使用过了 DAO，然后是 RDO，最近是各种版本的 ADO。毫无疑问 .NET 产生了一个全新的称为 ADO.NET 的数据存取技术。对 Visual Basic .NET 程序设计来说，虽然可以通过互操作机制来

继续使用原来的数据存取技术,但是大多数的 Visual Basic .NET 开发人员还可能使用 ADO.NET,因为它为我们提供了最简单和最直接的方法。第 8 章将讨论 ADO.NET,说明如何使用新的技术来读取、创建、更新以及删除数据。

整个 .NET 平台,包括 Visual Basic .NET 在内,都是主要以 XML 格式来处理数据。因此,在第 8 章中还将讨论开发人员如何在 Visual Basic .NET 中通过 ADO.NET 和直接从 VB 本身来使用 XML。

我们将在第 9 章中讨论一些更高级的论题,比如中间层和事务处理组件的创建、控制台应用程序的创建(Visual Basic .NET 很容易与标准的控制台输入/输出机制,stdin 和 stdout 交互作用),以及伴随 .NET Framework 所产生的一组功能很强大的外部工具和实用程序,包括 VB 编译器本身,如果需要的话,我们可以直接从命令行来运行它。我们还将简单地了解使用 C# 进行的跨语言开发,如何编写一个程序来监视文件系统的修改,以及如何使用 VB 来创建 Windows NT 服务。

.NET Framework 的一个主要目标就是简化开发和消除 DLL Hell,由于它已成为 VB 开发人员最为头痛的一个问题,所以我们将在第 10 章中详细介绍它。Visual Basic .NET 不仅修复了大多数的 DLL Hell 问题,而且还为我们提供了过去包含在 Visual Basic 中的功能最强大并且完全部署的技术——这与以前的安装向导或者包和部署向导技术相比是一个巨大的进步。通过从 URL 中运行应用程序或者部分应用程序把这种能力添加到“自动部署”组件中,将会更加令人兴奋。

很明显,目前大多数的开发人员都实际开发过基于 COM 以及其他现有的比如 ADO 这样的技术的应用程序。第 11 章将介绍互操作性和迁移,将为我们展示如何使 Visual Basic .NET 程序能够利用现有的 COM 组件,或者如果需要的话,从 COM 组件中进行使用。我们还将快速了解 Microsoft 公司的迁移向导。该向导可以帮助我们吧现有的 VB6 应用程序更新为 Visual Basic .NET。

本书涉及到的 80:20 原则

很明显,对 Microsoft 公司的开发人员来说,.NET 表现出了非常具有实质性的平台修改。虽然 COM 中大多数面向组件的概念推进了 .NET 的发展,但是在这个新的环境中还需要一些较大的修改和提高。

尽管今天的开发人员可以与 Win32 API 交互作用,但是在 .NET 中,大多数都是与 .NET 系统类库之间交互作用。实际上,任何 .NET 程序都将大量利用该系统类库来操作,要求每一个希望在 .NET 环境下工作的开发人员都要在很大程度上学习使用类库。

在所有这些平台修改的最上面还有对 VB 本身的修改。从一个特征的角度上来看,我们现在的继承和自由线程只可以命名一对特征。除了特征的修改之外,还有许多语法和编码的修改,这些都将影响到语言日复一日的使用。

现在 VB IDE 已经连同其他的 .NET 开发工具合并到了 Visual Studio .NET IDE 当中。这个集成的 IDE 与 VB6 IDE 的许多方面都很相似,但是还存在一些根本的差别。

实际上,这里还有很多修改,用一本书是根本讲不完的。与其尽量提供有关每一处修改的肤浅内容,不如我们选择应用 80:20 原则——如果我们介绍 80% 的特征,将得到在大多数情况



下都可以利用的资源。在本书中，我们将提供开发人员所面临的大多数公共修改所完全涉及到的内容。我们将把焦点集中放在 IDE 修改、语法修改和大多数新的特征上面。从某种程度上来说，我们不仅涉及到了 .NET 系统类库，而且它还适合于多数 VB 开发人员所面临的各种通用情况。

使用本书所需的条件

为了运行本书中的范例，需要具备下面的条件：

- Windows 2000 或者 Windows XP。
- Visual Basic .NET，可以重新分布的版本(包含在 .NET SDK 中)或者任何一个版本的 Visual Studio .NET 开发环境。

至少需要几个月使用 Visual Basic 的经验，比如熟悉通过 ADO 进行的数据存取。

范例中完整的代码资源可以从我们的网站 <http://www.wrox.com/> 中下载。可以单击页面上方的 VB.NET 链接，找到 Fast Track Visual Basic .NET 链接，然后单击相关的 Code Download 链接就可以下载这些代码。另外一种方法是从本书简介页面上的 http://www.wrox.com/Books/Book_Details.asp?isbn=1861007124 链接上来下载代码。

客户技术支持和反馈

我们一直很重视读者的意见，并且我们想知道您是如何评价本书的，喜欢哪些，不喜欢哪些，以及下次怎样做得更好。您可以把您的意见反馈给我们，也可以通过电子邮件将意见发送到 feedback@wrox.com。请务必在您的消息中写出本书的 ISBN 以及标题。

源代码和更新资料

如果您是通过本书中的样例进行工作的，您可以选择手工输入所有的代码。许多读者都喜欢这样，因为这是一种熟悉使用的编码技术的好方法，不管是否愿意输入代码，我们都将本书中所用到的源代码放在 Wrox.com 站点中。

当您登录 Wrox.com 站点 <http://www.wrox.com/> 时，只需通过 Search 工具或者使用其中一个标题列表就可以查找标题。然后单击本书详细页面上的 Download Code 链接就可以得到所有的源代码。

从我们的站点中下载的文件要使用 WinZip 解压缩后才可以利用。当把附件保存到硬件设备的文件夹中时，需要使用一个解压程序，例如 WinZip 或者 PKUnzip，来解压缩文件。当解压缩该文件时，代码通常被解压缩在各章文件夹中。当开始解压缩过程时，要确保您的软件(WinZip、PKUnzip 等等)在 Extract to: (或者等价物)下选中了 Use folder names。

即使您愿意输入代码，也可以使用源代码来检验您应当得到的结果——如果您认为有一处输入错误就可以第一次停止。如果您不愿意输入，那么就必须从我们的站点中下载源代码，它将

有助于修改并调试。

勘误表

我们已经尽最大的努力来保证文本或者代码的正确性，但是错误在所难免。如果您在本书中发现错误，比如拼写错误或者代码不完善的地方，我们将非常感激您的反馈。通过发送勘误表，可以节省其他读者所耗费的时间，并且当然还将帮助您获得更高质量的信息。您只需简单地发送电子邮件至 support@wrox.com 即可；您的信息将被检验，如果正确的话，发布该标题的勘误表页，并且在本书后面的编辑中使用。

要想在 Web 站点上找到勘误表，只需登录 <http://www.wrox.com/>，并可以通过我们的 Search 工具或者标题列表简便地进行查找。然后在本书的详细页上，单击 Book Errata 链接。在该页面上，便可以看到所有被提交的以及被编辑验证过的勘误表。还可以单击 Submit Errata 链接来为我们指出您所发现的任何一处勘误。

技术支持

如果希望直接询问本书中的某一问题，那么可向我们发送电子邮件至 support@wrox.com。所发送的电子邮件应包括以下内容：

- 在邮件主题中应包括书名，最后 4 位 ISBN 号码（本书为 7124），以及所发现问题的页码号
- 姓名，联系方式，以及在邮件正文中指出所发现的问题

我们不会向您发送垃圾邮件。我们需要详细资料以便为双方都节省时间。当您发送电子邮件信息时，将通过以下的支持环节：

(1) 客户支持——您的信息被传送到我们的一个客户支持工作人员那里，他们将首先阅读您的信息。他们有许多有关常见回答的文件，并将立刻回答任何有关本书或者 Web 站点上的一般问题。

(2) 编辑——更深一层的询问将反馈到本书的技术编辑那里。他们具有丰富的程序设计语言经验或者有关作品的经验，并可以回答有关该主题的详细技术问题。只要该问题被解决了，编辑将可以把勘误表发送到 Web 站点上。

(3) 作者——最后，编辑不能回答的问题将会请教作者来回答。我们尽量保护作者使他们的写作不受任何干扰，我们十分荣幸将一些特殊的请求转发给他们。所有的 Wrox 作者将为他们的著作提供支持。他们将回复客户和编辑所提出的问题，并保证所有的读者都能受益。

说明：

注意 Wrox 公司的支持处理只能为与我们所出版的有关书籍直接相关的内容提供支持。超出本书正常范围的支持问题将通过 <http://p2p.wrox.com/> 论坛的社区列表提供给读者。



p2p.wrox.com

对于作者以及参与讨论的人员来说，我们独特的系统能够通过 p2p 邮件发送清单、论坛以及新闻组来提供 programmer to programmer™ 的相互联系，另外我们还具有一对一的电子邮件支持系统。我们保证您的问题将由多位 Wrox 公司的作者以及其他行业的专家来解答，他们的联系方式都存放在我们的邮件发送清单中。在 p2p.wrox.com 上您还将发现有许多不同的列表可以为您提供帮助，不仅是在阅读本书的情况下，而且在开发自己的应用程序时也可以寻求帮助。

订阅邮件发送清单需要以下几个步骤：

- (1) 登录 <http://p2p.wrox.com/> 并从左边的菜单栏中选择相应的类别。
- (2) 单击希望加入的邮件发送清单。
- (3) 按照提示订阅并填写您的电子邮件地址以及密码。
- (4) 回复您所收到的确认邮件。
- (5) 使用订阅管理器加入更多的清单并设置邮件选项。

目 录

第 1 章 初识 VB.NET	1
1.1 .NET 环境	2
1.1.1 .NET 系统要求	3
1.1.2 其他问题	5
1.1.3 安装和配置示例应用程序	8
1.1.4 配置远程 IIS 服务器	9
1.2 从 Hello World 入手	10
1.3 本章小结	13
第 2 章 .NET Framework	14
2.1 什么是.NET	14
2.1.1 一个涉及面广、功能强大的平台	15
2.1.2 .NET Enterprise Servers 的角色	15
2.1.3 DNA 和 COM 所存在的问题	16
2.1.4 .NET 的起源	18
2.1.5 .NET Framework 概述	18
2.2 公共语言运行库	19
2.2.1 设计的主要目标	20
2.2.2 元数据	22
2.2.3 多语言集成和支持	23
2.2.4 通用类型系统	24
2.2.5 命名空间	26
2.2.6 .NET 应用程序的结构	27
2.2.7 部署和执行	28
2.3 下一层——.NET Class Framework	30
2.4 用户界面和程序接口	31
2.4.1 Windows Forms	32
2.4.2 Web Forms	32
2.4.3 控制台应用程序	33
2.4.4 Web 服务	33
2.5 作为.NET “元语言”的 XML	34
2.6 COM 的角色	34
2.7 DNA 的角色	35



2.8 其他优点	35
2.9 第一步——Visual Studio .NET	36
2.10 .NET 对 Visual Basic 的改变	36
2.11 .NET 对开发人员的影响	37
2.11.1 编程模型	37
2.11.2 减少了 Internet 开发的障碍	37
2.11.3 库的预编写功能	38
2.11.4 更简单的部署	38
2.12 本章小结	38
第 3 章 新的 IDE 特征	39
3.1 Hello World 程序	39
3.1.1 窗体的声明、初始化以及终止	40
3.1.2 Windows 窗体设计器生成的代码	42
3.1.3 事件处理程序代码	47
3.2 新的 IDE 特征	49
3.2.1 Start Page 选项卡	50
3.2.2 停靠窗口	50
3.2.3 主窗口	52
3.2.4 宏支持	65
3.2.5 集成调试	65
3.3 本章小结	67
第 4 章 语言和语法	68
4.1 命名空间	68
4.1.1 使用命名空间	70
4.1.2 创建命名空间	71
4.1.3 Microsoft.VisualBasic 命名空间	73
4.2 语言和语法的变化	73
4.2.1 Option 语句	73
4.2.2 数据类型的变化	74
4.2.3 变量声明的变化	79
4.2.4 数组的变化	81
4.2.5 用户自定义数据类型的变化	83
4.2.6 集合的变化	84
4.2.7 新的算术运算符	87
4.2.8 中断 If...Then 语句	87
4.2.9 使用保留字作为过程的名称	88
4.2.10 取消 Set 语句	88

4.2.11 属性例程的变化	89
4.2.12 结构化错误处理	91
4.2.13 过程的语法变化	96
4.2.14 事件处理的变化	99
4.2.15 不重要的、废弃的已不受支持的 VB 语法	102
4.3 委托	103
4.4 属性	104
4.5 本章小结	105
第 5 章 Windows 用户界面的性能	106
5.1 Windows Forms 的重要性	106
5.2 Windows Forms 的基础知识	107
5.3 Windows Forms 的体系结构	108
5.4 比较 Windows Forms 和 VB6 Forms	110
5.4.1 对 Windows Forms 使用可视化设计器	111
5.4.2 取消和默认按钮的窗体属性	112
5.4.3 定位和布局	112
5.4.4 对话框的区别	113
5.5 Windows Forms 的新性能	117
5.5.1 改变窗体的形状	117
5.5.2 改变窗体外观的新属性	118
5.5.3 专用窗体	119
5.6 Windows Forms 中的控件	120
5.6.1 控件的新属性	120
5.6.2 在运行时添加新控件	122
5.7 重要控件的总结	123
5.7.1 新的控件	124
5.7.2 对现有控件的修改	133
5.8 多文档界面(MDI) 窗体	134
5.8.1 创建一个 MDI 父窗体	134
5.8.2 MDI 父窗体的不同	134
5.8.3 MDI 子窗体的不同	135
5.8.4 在 MDI 父窗体中安排子窗口	135
5.8.5 Visual Basic .NET 中的一个 MDI 范例	135
5.9 Windows Forms 中的继承	137
5.10 GDI+	140
5.10.1 System.Drawing 命名空间	140
5.10.2 在 Windows Form 中使用 GDI+	141



5.10.3 与 VB6 及其早期版本相比的变化	145
5.11 创建定制的 Windows Forms 控件	146
5.11.1 继承自现有的控件	146
5.11.2 复合控件	148
5.11.3 从 Control 类中继承新控件	148
5.12 本章小结	148
第 6 章 面向对象的新功能	149
6.1 什么是面向对象	149
6.2 对象与组件的关系	150
6.3 Visual Basic .NET 的 OO 实现	150
6.3.1 创建类	151
6.3.2 对象的生命周期	157
6.3.3 与对象交互	163
6.3.4 共享的类成员	168
6.3.5 继承	171
6.3.6 接口	189
6.4 跨语言继承	193
6.4.1 创建 Visual Basic .NET 基类	193
6.4.2 创建 C# 子类	193
6.4.3 创建客户应用程序	194
6.5 可视化继承	195
6.6 本章小结	195
第 7 章 Web 功能	197
7.1 ASP 存在的问题	198
7.2 Web Forms 概述	198
7.2.1 运行的 Web Form	199
7.2.2 后台代码	202
7.2.3 作为运行库引擎的 ASP.NET	202
7.2.4 分析 Web Form	202
7.2.5 Web Forms 事件模型	205
7.3 Web Forms 的布局	208
7.4 服务器控件	209
7.4.1 需要服务器控件的原因	209
7.4.2 ASP.NET 服务器控件	210
7.4.3 验证控件	211
7.4.4 定制 Web Form 服务器控件	213
7.5 Web Forms 中的数据绑定	215

7.6 最终的例子——使用 Web Forms 的小应用程序	215
7.7 用 Web.config 配置应用程序	219
7.7.1 ASP.NET Web 站点的安全性	219
7.7.2 状态管理选项	221
7.8 Web 服务概述	221
7.8.1 Web 服务的用途	222
7.8.2 了解 SOAP 协议	222
7.8.3 在 Visual Basic .NET 中创建 Web 服务	223
7.8.4 在 Visual Basic .NET 中使用 Web 服务	225
7.8.5 创建和使用 Web 服务	226
7.8.6 Web 服务的安全性	227
7.9 本章小结	227
第 8 章 数据访问	228
8.1 ADO.NET 和 ADO	229
8.2 ADO.NET——重要的概念	230
8.2.1 ADO.NET 类的位置	230
8.2.2 替代 Recordset	230
8.2.3 DataSet 的定义	231
8.2.4 DataSet 的数据流概述	231
8.3 用于连接操作的数据提供者	233
8.3.1 .NET 包含的提供者	233
8.3.2 数据提供者实现的类	234
8.4 ADO.NET 类如何协同工作	242
8.5 DataSet 的结构	242
8.5.1 DataTable	243
8.5.2 DataColumn	244
8.5.3 DataRow	245
8.5.4 Constraints	247
8.5.5 DataRelation	248
8.6 使用 DataSet	250
8.6.1 在 DataTable 中添加行	250
8.6.2 在 DataTable 中查找特定行	250
8.6.3 删除 DataTable 中的行	251
8.6.4 手工创建 DataSet	252
8.6.5 使用 DataAdapter 向导创建 DataSet	253
8.7 使用 DataViews	258
8.7.1 使 DataView 获得数据子集	259



8.7.2 排序 DataView	260
8.8 Typed DataSet	260
8.9 在 ADO.NET 中使用存储过程	262
8.9.1 Command 类的存储过程	262
8.9.2 使用 DataAdapters 的存储过程	263
8.10 ADO.NET 的异常类	264
8.11 数据绑定	265
8.11.1 列表框和组合框的数据绑定	265
8.11.2 将数据绑定到对象集合	266
8.11.3 将数据绑定到任何属性上	266
8.12 通过 ADO.NET 访问 XML	267
8.12.1 保存数据的变化	268
8.12.2 在 XML 中维持 DataSet	268
8.13 通过 DOM 访问 XML	269
8.14 使用 ADO.NET 的一些综合指南	269
8.14.1 什么时候使用 DataReader	269
8.14.2 什么时候使用 DataSet	269
8.14.3 用 ADO.NET 改变事务层	270
8.15 什么时候需要传统 ADO	270
8.15.1 保守式并发	270
8.15.2 需要服务器端游标的应用程序	271
8.16 本章小结	271
第 9 章 高级主题	272
9.1 中间层组件	272
9.1.1 .NET Remoting	273
9.1.2 Enterprise Services	286
9.1.3 使用 Microsoft 消息队列(MSMQ)	290
9.2 .NET Threading	293
9.2.1 协同线程工作	294
9.2.2 线程生存期	298
9.2.3 与当前线程交互	299
9.2.4 给线程传递数据	299
9.2.5 异步处理	301
9.3 控制台应用程序	308
9.4 新打印模式	311
9.4.1 PrintDocument 及相关对象	311
9.4.2 实现打印	311

9.5 用 VB 创建一个 Windows NT/2K 服务	317
9.5.1 创建一个 Windows 服务	318
9.5.2 安装服务	321
9.6 命令行选项	325
9.6.1 vbc 命令行参数	325
9.6.2 在 bin 目录中的工具描述	326
9.7 本章小结	330
第 10 章 安装及部署	331
10.1 构造配置	331
10.2 可再分布的 .NET Framework	333
10.3 XCOPY 部署	333
10.3.1 Windows 应用程序	333
10.3.2 Web 应用程序	334
10.4 正式安装	336
10.4.1 Windows 应用程序	337
10.4.2 Web 应用程序	342
10.5 自动部署	345
10.5.1 通过 URL 启动 .NET 程序	346
10.5.2 运用 Shell 程序	349
10.5.3 安全性考虑	353
10.6 本章小结	355
第 11 章 互操作性和迁移	356
11.1 COM/DCOM 互操作性	356
11.1.1 从 .NET 调用 COM 组件	357
11.1.2 从 COM 调用 .NET 组件	368
11.2 调用 Windows API	372
11.3 使用迁移向导	377
11.4 本章小结	380