

PROGRAMMER TO PROGRAMMER™



Professional VB 2005 with .NET 3.0

VB 2005 & .NET 3.0 高级编程 (第5版)

Bill Evjen
(美) Billy Hollis 等著
Bill Sheldon
杨 浩 译



清华大学出版社

TP312/2722

2008

VB 2005 & .NET 3.0

高级编程

(第 5 版)

Bill Evjen
(美) Billy Hollis 等著
Bill Sheldon
杨 浩 译

清华大学出版社

北 京

Bill Evjen, Billy Hollis, Bill Sheldon, Kent Sharkey, Tim McCarthy

Professional VB 2005 with .NET 3.0

EISBN: 978-0-470-12470-3

Copyright © 2007 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing Inc. 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。

北京市版权局著作权合同登记号 图字: 01-2007-5127

本书封面贴有 John Wiley & Sons 防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

VB 2005 & .NET 3.0 高级编程(第5版)/(美)依夫杰(Evjen, B.)等著; 杨浩译. —北京: 清华大学出版社, 2008.2

书名原文: Professional VB 2005 With .NET 3.0

ISBN 978-7-302-16779-2

I. V… II. ①依… ②杨… III. ①BASIC 语言—程序设计 ②计算机网络—程序设计 IV. TP312 TP393

中国版本图书馆 CIP 数据核字(2008)第 004155 号

责任编辑: 王 军 郑雪梅

装帧设计: 康 博

责任校对: 胡雁翎

责任印制: 孟凡玉

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 70 字 数: 1704 千字

版 次: 2008 年 2 月第 1 版 印 次: 2008 年 2 月第 1 次印刷

印 数: 1~4000

定 价: 128.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系
调换。联系电话: (010)62770177 转 3103 产品编号: 025900-01

前 言

2002 年,伴随着 VB.NET 的引入,Visual Basic(VB)迎来了自发布以来最大的变革。十几年来,VB 一直没有进行大的修订。但.NET 已经远远超出了这种修订,它改变了软件开发的几乎所有方面。从集成 Internet 功能到创建面向对象的架构,VB.NET 向传统的 VB 开发人员发出了挑战,要求他们学习全新的概念和技术。

2005 年,Microsoft 对 Visual Basic 语言做了进一步的改进(重命名为 VB 2005)。作为一种真正的面向对象语言,VB 2005 引入了很多新增功能。传统的 VB 6 开发人员必须从头学习这些内容,但对于熟悉 VB 以前版本的开发人员来说,VB 2005 是比较容易学习的,本书能提供这方面的帮助。

本书所介绍的 .NET Framework 3.0 版本不仅详细介绍了 VB 的最新版本(即 VB 2005),还探讨了新的架构,通过这个架构,VB 开发人员能利用 Windows Presentation Foundation(WPF)建立新的应用程序类型,创建基于 Windows Communication Foundation(WCF)、Windows Workflow Foundation(WF)和 Windows CardSpace 的库。

这些新增功能构成了 .NET Framework 3.0,这个版本安装在 .NET Framework 2.0 上,并以 .NET Framework 2.0 为基础。本书的一些章节将重点讨论 .NET Framework 3.0 引入的新技术。

首先,需要学习 VB 2005 和旧版本之间的区别。在一些情况下,相同的功能要采用不同的方式来实现。这些改变不是任意而为的,而是有很充足的理由。我们必须抛弃一些旧习惯,培养新习惯。

其次,我们必须接受新概念。完全面向对象的概念、新的组件技术、新的用于本地和 Internet 界面的可视化工具,所有这些都是在 VB 2005 中高效开发应用程序的必备技能。

本书将全面阐述 VB 2005。首先讨论 .NET Framework,最后讨论部署 .NET 应用程序的最佳方式。中间部分讨论从数据库访问到与诸如 XML 等其他技术集成的一切内容,并详细地介绍新增功能。您将会看到,VB 2005 已经成为一种强大且易于使用的语言。在编写网络程序时,就如同编写桌面程序一样得心应手。本书将介绍 .NET Framework 2.0 和新发布的 .NET Framework 3.0。如果读者使用 1.0、1.1 或 2.0 编程,本书的一些章节就可能不太合适。要特别注意 .NET Framework 3.0 的新增功能。

VB 的重要性

在 .NET 的早期应用中,Microsoft 的新语言 C#得到了广泛的关注。但随着 .NET 应用的增多,VB 的重要性也逐渐显现出来。Microsoft 对外宣称,当生产率是最重要因素时,

VB 是开发应用程序的最佳语言。

VB 的未来开发将强调以最高效的方式访问整个 .NET Framework 的能力,而 C# 开发强调的是编写代码的经验,这非常切合 VB 的传统作用:语言开发人员在现实世界中使用它能尽快创建商务应用程序。

区别不仅仅在学术方面。.NET Framework 的一个最重要的优点是,它允许用非常少的代码编写应用程序。在商务应用程序中,主要目标是编写业务逻辑,尽可能减少例程编码任务。换言之,其价值在新的 .NET Framework 中不是编写出许多代码,而是用尽可能少的代码编写出更健壮、有效的应用程序。

VB 非常适合于这种类型的开发,因此成为目前既经济又实惠的软件开发工具,将来它还会为此目的做进一步的细化和改进。

本书读者对象

本书主要针对有经验的、希望深入学习 VB 2005 的开发人员。本书讨论了最常见的编程任务和概念,既适合于从早期版本过渡到 VB 2005 版本的开发人员,也适合于以前使用过 VB,目前需要更深入理解的开发人员。

本书对 VB 概念进行了全面的介绍,但 .NET Framework 的内容广博而丰富,不可能在一本书里讲清楚,本书没有把 Web 开发作为最重要的内容,但包含了 VB 中基于浏览器的编程内容,专业的 Web 开发人员应该参考清华大学出版社引进并出版的《ASP.NET 2.0 高级编程(特别版)》。

使用本书需要的条件

尽管可以使用 .NET Framework SDK 中的命令行工具来创建 VB 应用程序,但是本书的大部分内容仍将使用 Visual Studio 2005(专业版或更高,其中包含 .NET Framework SDK)进行举例。如果您使用的是 Visual Studio .NET 2002 或者 Visual Studio .NET 2003,那么本书的很多例子将无法运行,因为代码中的很多功能都是旧版本中没有的。

此外:

- 需要 .NET Framework 3.0(如果机器上没有 .NET Framework 2.0, .NET Framework 3.0 会自动安装 .NET Framework 2.0)。
- 某些章节使用了 SQL Server 2005。但也可以使用 VS 2005 自带的 SQL Express 来运行样例代码。
- 某些章节使用了 Internet Information Services(IIS)。IIS 包含在 Windows 2003 Server、Windows 2000 Server、Windows 2000 Professional 和 Windows XP 中,但它不会默认安装。
- 第 28 章在介绍排队事务时使用了 MSMQ。MSMQ 包含在 Windows 2003 Server、Windows 2000 Server、Windows 2000 Professional 和 Windows XP 中,但它也不会默认安装。

本书主要内容

第1章解释了.NET的重要性的它对应用程序开发的影响。为了理解.NET的重要性以及当前开发技术存在的缺陷,我们首先讨论了包括COM和DNA在内的体系结构模型,然后讲述.NET如何使用公共语言运行库(CLR)来纠正这些技术中的缺点。本章还回顾了.NET Framework 2.0和.NET Framework 3.0的内容,以及如何安装这两个架构。

第2章将讨论VB 2005中的许多常用类型。本章的主题包括类型转换、引用类型、数组和其他集合、参数传送和装箱,还介绍了VB的基本语法。

第3章论述了VB中面向对象编程的特性。本章将定义对象、类、实例、封装、抽象、多态性和继承性。

第4章讲述了继承的概念及在VB中如何利用继承。我们将解释如何创建简单和抽象的基类,如何创建可以派生其他类的基类。本章把前面几章的理论应用于实践。还将讨论4个主要的面向对象概念(抽象性、封装性、多态性、继承),以及如何将这些概念应用到设计和开发中,以创建高效的面向对象的应用程序。

第5章探讨了.NET平台的核心——公共语言运行库(CLR)。CLR负责管理.NET平台编译的代码的执行。我们将论述版本控制和部署、内存管理、跨语言集成、元数据和IL Disassembler。

开发人员通常用英语建立应用程序。但随着应用程序的使用者在不断增长,开发人员意识到需要对应用程序进行全球化处理。当然,最理想的情况是从一开始就使建立的应用程序能正确应对全球的用户。但在许多情况下,这是不太可能的,因为应用程序需要完成额外的工作。.NET Framework为解决应用程序的全球化做了许多努力。我们会很快认识到,API的变化、给服务器控件增加的功能、甚至VS本身都需要我们做出额外的努力,才能更方便地使应用程序全球化。第6章将介绍建立全球化应用程序时要考虑的一些重要事项,详细探讨System.Globalization命名空间和它为应用程序提供的所有功能。

第7章介绍了新版本最大的改进之一——泛型。泛型是一个通用的强类型集合,它可以减少出错几率,提高性能,并为集合提供Intellisense功能。

第8章介绍了命名空间及其层次结构。解释了命名空间的概念和某些常用的命名空间。此外,还介绍了如何创建新的命名空间,以及在项目中如何导入命名空间,如何给命名空间起别名。本章还介绍了VB 2005新引入的My命名空间。

第9章通过详细讨论CLR异常处理程序和新的Try...Catch...Finally结构,描述了VB 2005中错误处理和调试工作的原理。除此之外,还描述了错误和跟踪的记录,以及如何利用这些方法获取程序工作方式的反馈。

第10章集中讨论了需要了解的ADO.NET对象模型,以便能够建立灵活、快速和可伸缩的数据访问对象和应用程序。该章还讲述了ADO向ADO.NET的转化,讨论了为了建立对.NET应用程序的数据访问而需要理解的ADO.NET主要对象。

第11章展示了.NET Framework提供的用于生成和处理XML的功能。我们介绍了.NET Framework的XML相关命名空间,并详细讨论了一部分命名空间中的类。本章还涉及到一些利用XML的技术,比如ADO.NET和SQL Server。

第 12 章介绍了 .NET 提供的安全功能和相关工具。Caspol.exe 和 Permview.exe 可以帮助建立和维护各种安全策略。System.Security.Permissions 命名空间可用于管理权限。我们还通过代码演示了 System.Security.Cryptography 命名空间的功能。

第 13 章将介绍开发 .NET 应用程序的下一代 IDE: VS 2005。这个 IDE 的以前版本是 Visual Studio .NET 2003 和 Visual Studio .NET 2002。本章主要讨论 VS 2005 版本, 以及如何使用它更快地建立更好的应用程序。

第 14 章描述了如何联合使用新的 SQL Server 2005 和 .NET 应用程序。SQL Server 提供了与应用程序的强大连接, 本章将解释如何高效地使用这个功能强大的数据库。

第 15 章讲述了 Windows 窗体, 主要讨论了窗体和内置控件。还讨论了哪些是新的, 哪些是对 VB 以前版本的修改, 还讨论了 System.Windows.Forms 命名空间。

第 16 章讨论了在构建 Windows 窗体应用程序时会用到的高级功能。

第 17 章将介绍 Windows Presentation Foundation(WPF), 它是 .NET 3.0 的一个新组件, 提供了建立应用程序的一种新方式。该章将介绍 Microsoft 提升 GUI 显示功能的这种新方式, WPF 所提供的显示层相当流畅和丰富。该章还将介绍 WPF 基础, 以及如何建立简单的基于 XAML 的应用程序。

许多公司都在 Windows 窗体上投入了巨资, 他们也不打算将其应用程序完全转换到这个新技术上。因此, Microsoft 提供了将 WPF 集成到 Windows 窗体应用程序的重要方式, 还允许将 Windows 窗体组件引入 WPF 应用程序。第 18 章将介绍这些功能。

第 19 章介绍了 ASP.NET 2.0 的基础知识, 包括使用 VS 进行在线编码的新功能、新的后台编码模型, 其中包括应用程序和页面框架, 以及 ASP.NET 2.0 的新特性。

第 20 章介绍了很多在 ASP.NET 2.0 中新引入的功能。我们举例说明了跨页面传送、Master 页面、站点导航以及个性化等高级功能。

第 21 章讨论了程序集的概念及其在 CLR 中的作用。我们还学习了程序集的结构、内容以及包含的信息。另外介绍了程序集清单以及它在部署中的作用。

第 22 章介绍了 Windows 窗体和 Web 窗体的部署选项, 包括新的 ClickOnce 部署特性和.msi 文件的创建。

第 23 章介绍了 COM 和 .NET 组件的互操作性, 以及 Microsoft 提供了哪些工具将这两项技术链接起来。

第 24 章将学习线程, 理解 .NET Framework 中的各种对象如何让用户开发出多线程的应用程序。我们将讨论如何创建线程, 如何将它们与进程关联在一起, 以及多任务和多线程之间的差异。

第 25 章将介绍一个新的 .NET 3.0 功能——Windows Workflow Foundation(WF), 它便于将工作流集成到应用程序中。

第 26 章探讨了如何利用 VB 2005 创建和使用 Web 服务。该章还讨论了 CLR 提供的用于建立和利用 Web 服务的抽象类, 论述了一些支持 Web 服务的技术。另外阐述了使用分布式体系结构的缺点以及 Web 服务的未来。

第 27 章讨论了如何在传统的 3 层应用程序设计中使用远程服务。本章将论述远程服务的基本体系结构, 并建立一个基本服务器和客户程序, 使用一个 singleton 对象将客户的请求传递到业务层上。接着介绍如何使用串行化把服务器上更复杂的对象传递给客户, 以

及如何使用调用环境(call context), 在每个调用中, 把额外的数据从客户端传递给服务器, 但无须改变对象模型。

第 28 章探讨了 .NET 组件服务, 特别讨论了事务处理和排队组件。

第 29 章介绍了开发网络应用程序时需要使用的各种协议。我们还讨论了如何把庞大的网络融合到自己的应用程序中。

第 30 章介绍建立基于服务的组件的新方式——Windows Communication Foundation (WCF), 它可以通过许多协议进行基于标准的通信。WCF 是 Microsoft 为企业内部和外部的组件通信提供的一种新方式。

第 31 章讨论了在 Windows 服务的生成过程中, 如何使用 VB 2005。该章将讨论 Windows 服务的创建、安装、运行和调试。

第 32 章介绍了如何从 Web 上下载资源, 如何设计自己的通信协议, 以及如何在应用程序中重用 Web 浏览器控件。

附录 A 介绍了 VB 编译器 vbc.exe 所提供的功能。

附录 B 列出了一些可用的 VB 资源。

用户支持

我们总是想知道您对本书的看法, 您喜欢哪些内容, 您不喜欢哪些内容, 这些信息将有助于我们下一次做得更好。如果您有什么意见和建议, 请向 wkservice@tup.tsinghua.edu.cn 发邮件。但是, 请您一定要在您的信中注明本书的书名。

如何下载本书的示例代码

在您登录到 Wrox 站点 <http://www.wrox.com/> 时, 只需使用 Search 工具或使用书名列表就可以找到本书。接着在 Code 栏中单击 Download 链接, 或单击本书信息页面上的 Download Code 链接, 就可以获得本书所有的源代码。另外, 您也可以从本书的合作站点 www.tupwk.com.cn/download 上下载本书的所有源代码。

从以上站点上下载的文件使用 WinZip 进行了压缩。在把文件保存到硬盘的一个文件夹中时, 需要使用解压缩软件(如 WinZip 或 PKUnzip)对该文件解压缩。在解压缩时, 代码常常放在各自的章节文件夹中。在开始解压缩过程时, 一定要将解压缩软件 WinZip 或 PKUnzip 设置为使用文件夹名。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误, 但是错误总是难免的, 如果您在本书中找到了错误, 例如拼写错误或代码错误, 请告诉我们, 我们将非常感激。通过勘误表, 可以让其他读者避免受挫, 当然, 这还有助于提供更高质量的信息。请给 wkservice@tup.tsinghua.edu.cn 发电子邮件, 我们会检查您的信息, 如果是正确的, 就把它发送到本书的勘误表页面上, 或在本书的后续版本中采用。

要在网站上找到勘误表, 可以登录 <http://www.wrox.com>, 通过 Advanced Search 工具

或书名列表查找本书,然后在本书的信息页面上,单击 Book Errata 链接。

E-Mail 支持

如果您希望直接就本书的问题向对本书知之甚多的专家咨询,那么,就向 support@wrox.com 发电子邮件,在电子邮件的“主题”(Subject)栏中,加上本书的名称和 ISBN 的最后 4 位号码。典型的电子邮件应该包括以下内容:

- 在“主题”栏加上书的名称、ISBN 的最后 4 位数字(4703)和问题所在的页码。
- 在邮件的正文中加上您的姓名、联系信息和问题。

我们不会给您发送垃圾邮件。我们只需要详细的情况以节省您的宝贵时间和我们的时间。当您发送电子邮件时,它会直接链接到以下支持链:

- 客户支持——您的消息会传送到我们的客户支持人员,他们是阅读信息的第一人。他们有常见问题的文件,会迅速回答一般性的问题。他们回答关于本书和网站的一般性问题。
- 编辑支持——更深的问题会转发到负责本书的技术编辑处。他(或)她具有编程或特殊产品的经验,能够回答某个主题的详细技术问题。
- 作者支持——最后,在编辑都不能回答问题的情况下(这种情况很少出现),这些问题将转发到作者。我们试图保护作者不要从写作中分心,但是,我们也很愿意将特殊的问题转发给他们。所有的 Wrox 作者帮助支持他们的书籍。他们向客户和编辑回复电子邮件,所有的读者都会从中受益。

Wrox 支持过程只能提供直接与已出版的图书相关的问题。对于超出此范围的问题可以通过 <http://p2p.wrox.com>/论坛的团体列表来提供支持。

p2p.wrox.com

P2P 邮件列表是为作者和同行的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“程序员到程序员的支持”(programmer to programmer support),还包括一对一的电子邮件支持系统。如果把问题发送给 P2P,就可以相信,您的问题不仅是由支持专家解答,而且还要提供给我们邮件列表中的许多 Wrox 作者和其他业界专家。在 p2p.wrox.com 上,可以从许多不同的列表中获得帮助,不仅在阅读本书时获得帮助,还可以在开发应用程序时获得帮助。在网站的.NET 类别中,最适合本书的是 `beginning_vb` 和 `vb_dotnet` 列表。

要订阅一个邮件列表,可以遵循下面的步骤:

- (1) 进入 <http://p2p.wrox.com>。
- (2) 从左侧的菜单栏中选择合适的列表。
- (3) 单击想加入的邮件列表。
- (4) 按照指示订阅和填写电子邮件地址和密码。
- (5) 回复接收到的确认电子邮件。
- (6) 使用订阅管理器加入更多的列表,设置自己的邮件设置。

目 录

第 1 章 .NET Framework 体系

结构综述.....1

1.1 什么是.NET Framework.....1

1.1.1 .NET Framework 的版本.....1

1.1.2 .NET Framework 3.0 的新特性.....2

1.1.3 后 Internet 时代的平台.....2

1.1.4 克服 COM 的缺陷.....3

1.2 .NET Framework 概述.....3

1.3 公共语言运行库.....4

1.3.1 主要的设计目标.....4

1.3.2 元数据.....6

1.3.3 多语言集成和支持.....6

1.3.4 公共类型系统.....7

1.3.5 命名空间.....7

1.4 .NET Class Framework.....7

1.5 .NET 的用户和程序接口.....9

1.5.1 Windows 窗体.....9

1.5.2 Web 窗体.....10

1.5.3 Windows Presentation Foundation.....10

1.5.4 控制台应用程序.....10

1.5.5 Web 服务和 Windows Communication Foundation.....11

1.6 作为.NET “元语言”的 XML.....12

1.7 COM 的角色.....12

1.8 .NET 2.0 中的主要区别.....12

1.9 小结.....13

第 2 章 VB 2005 核心元素.....15

2.1 原来的关键字和语法.....15

2.2 值类型和引用类型.....19

2.3 值类型(结构).....22

2.4 引用类型(类).....29

2.4.1 Object 类.....29

2.4.2 String 类.....30

2.4.3 DBNull 类和 IsDBNull() 函数.....33

2.4.4 Nullable 类型.....33

2.4.5 数组.....34

2.4.6 集合.....37

2.5 装箱.....38

2.6 参数传递.....40

2.7 变量的作用域.....41

2.8 数据类型的转换.....42

2.8.1 隐式转换和编译器选项.....43

2.8.2 执行显式转换.....45

2.9 命令.....47

2.9.1 If Then.....48

2.9.2 比较运算符.....48

2.9.3 Select Case.....50

2.9.4 For Each 和 For Next.....51

2.9.5 While、Do While 和 Do Until.....51

2.10 小结.....52

第 3 章 对象语法综述.....55

3.1 面向对象的术语.....56

3.1.1 对象、类和实例.....56

3.1.2 对象的组成.....56

3.2 使用对象.....59

3.2.1 对象的声明和实例化.....59

3.2.2 对象引用.....60

3.2.3 取消对象的引用.....61

3.2.4 前期及后期绑定.....61

3.3 创建类.....65

3.3.1 创建基类	65	5.3.3 对元数据更好的支持	191
3.3.2 构造函数方法	82	5.3.4 属性	192
3.3.3 终止和清除	83	5.3.5 Reflection API	194
3.4 高级概念	84	5.4 IL Disassembler	195
3.4.1 重载方法	84	5.5 内存管理	196
3.4.2 重载构造函数方法	87	5.5.1 传统的垃圾回收机制	196
3.4.3 共享方法、变量和事件	88	5.5.2 更快地为对象分配内存	203
3.4.4 运算符重载	93	5.5.3 垃圾回收器的优化	204
3.4.5 委托	96	5.6 小结	206
3.4.6 类和组件	100		
3.5 小结	101		
第 4 章 面向对象编程	103	第 6 章 本地化	207
4.1 继承	103	6.1 文化和区域	207
4.2 多接口	147	6.1.1 理解文化类型	208
4.2.1 对象接口	147	6.1.2 线程	209
4.2.2 辅助接口	149	6.2 在 ASP.NET 中声明	
4.3 抽象性	154	全局文化	211
4.4 封装性	158	6.3 在 ASP.NET 中使用	
4.5 多态性	160	文化设置	212
4.5.1 方法签名	160	6.4 转换数值和操作	213
4.5.2 实现多态性	161	6.4.1 理解日期之间的区别	214
4.6 继承	170	6.4.2 理解数字和货币的区别	215
4.6.1 使用继承的场合	170	6.4.3 理解排序字符串的区别	218
4.6.2 继承和多接口	174	6.5 使用 ASP.NET 2.0	
4.6.3 多级继承	179	资源文件	220
4.6.4 “脆弱”的基类问题	180	6.5.1 使用本地资源	220
4.7 小结	183	6.5.2 使用全局资源	226
第 5 章 公共语言运行库	185	6.6 Windows 窗体中的	
5.1 .NET 应用程序的组成元素	186	资源文件	227
5.1.1 模块	186	6.7 使用资源编辑器	232
5.1.2 程序集	187	6.8 小结	232
5.1.3 类型	187		
5.2 版本化与部署	188	第 7 章 泛型	233
5.2.1 对版本化更好的支持	188	7.1 泛型的使用	233
5.2.2 更好的部署	189	7.1.1 泛型类型	234
5.3 跨语言集成	190	7.1.2 泛型方法	238
5.3.1 通用类型系统	190	7.2 创建泛型	239
5.3.2 元数据	191	7.2.1 泛型类型	240
		7.2.2 泛型方法	247
		7.2.3 约束	248
		7.2.4 泛型和后期绑定	252

7.3 小结	252	9.4 与 VB 6 样式的错误处理 交互操作	292
第 8 章 命名空间	253	9.5 错误记录	292
8.1 命名空间的概念	253	9.5.1 事件日志	293
8.1.1 命名空间与引用	257	9.5.2 事件、方法和属性	294
8.1.2 常用的命名空间	258	9.5.3 写入追踪文件	295
8.2 导入命名空间并指定别名	261	9.6 使用 Trace 类分析问题 和测试性能	297
8.2.1 导入命名空间	261	9.7 小结	300
8.2.2 为命名空间指定别名	262		
8.2.3 在 ASP.NET 中引用 命名空间	263	第 10 章 用 ADO.NET 2.0	
8.3 创建自己的命名空间	263	访问数据	303
8.4 My 关键字	266	10.1 ADO.NET 2.0 体系结构 的改进	304
8.4.1 My.Application 命名空间	267	10.2 .NET 数据提供程序	306
8.4.2 My.Computer 命名空间	271	10.2.1 Connection 对象	306
8.4.3 My.Forms 命名空间	274	10.2.2 Command 对象	306
8.4.4 My.Resources 命名空间	274	10.2.3 通过 Command 对象使用存储过程	307
8.4.5 My.User 命名空间	274	10.2.4 DataReader 对象	311
8.4.6 My.WebServices 命名空间	275	10.2.5 命令的异步执行	313
8.5 小结	275	10.2.6 DataAdapter 对象	315
第 9 章 异常处理和调试	277	10.2.7 SQL Server .NET 数据提供程序	319
9.1 VB 6 中的错误处理	277	10.2.8 OLE DB .NET 数据提供程序	319
9.2 .NET 中的异常处理	278	10.3 DataSet 组件	320
9.2.1 异常的重要属性和方法	278	10.3.1 DataTableCollection 对象	320
9.2.2 异常与 VB 6 中 Err 对象的区别	279	10.3.2 DataRelationCollection 对象	320
9.3 结构化异常处理的关键字	280	10.3.3 ExtendedProperties 属性	321
9.3.1 Try、Catch 和 Finally 关键字	280	10.3.4 创建和使用 DataSet 对象	321
9.3.2 Throw 关键字	282	10.3.5 ADO.NET 的 DataTable 对象	324
9.3.3 抛出新的异常	283	10.3.6 ADO.NET 2.0 对 DataSet 和 DataTable 的改进	324
9.3.4 Exit Try 语句	284		
9.3.5 嵌套的 Try 结构	285		
9.3.6 Message 属性	287		
9.3.7 InnerException 和 TargetSite 属性	287		

10.4	使用通用提供程序模型	327	11.7.3	Xml 服务器控件	400
10.5	ADO.NET 2.0 对连接池 的改进	329	11.8	小结	402
10.6	创建数据访问组件	329	第 12 章	.NET Framework 中 的安全性	403
10.6.1	构造函数	330	12.1	安全的概念与定义	404
10.6.2	属性	331	12.2	System.Security.Permissions 命名空间中的权限	405
10.6.3	存储过程使用的 XML 结构	332	12.2.1	代码访问权限	407
10.6.4	方法	333	12.2.2	基于角色的权限	408
10.6.5	使用 DataSet 对象绑定 DataGrid	345	12.2.3	身份权限	410
10.7	小结	349	12.3	管理代码访问权限	410
第 11 章	在 VB 2005 中使用 XML	351	12.4	管理安全策略	415
11.1	XML 简介	352	12.4.1	应用程序顺利运行所 需要的最小权限	424
11.2	XML 串行化	353	12.4.2	使用 Visual Studio 判断 最小权限集合	426
11.3	System.Xml 文档支持	360	12.4.3	安全性工具	428
11.4	XML 流样式分析程序	360	12.4.4	使用 SecurityException 类处理异常	429
11.4.1	写入 XML 流	361	12.5	加密基础	430
11.4.2	读取 XML 流	365	12.6	小结	451
11.4.3	使用 MemoryStream 对象	375	第 13 章	Visual Studio 2005	453
11.4.4	文档对象模型(DOM)	378	13.1	Visual Studio 2005 从 Express 到 Team Suite 的各种版本	454
11.5	XSLT 转换	384	13.1.1	从项目模板上 创建项目	456
11.5.1	使用 XSLT 转换不同的 XML 标准	389	13.1.2	Solution Explorer 窗口	460
11.5.2	System.Xml.Xsl 中定义 的其他类和接口	391	13.1.3	My Project 视图	461
11.6	ADO.NET	391	13.1.4	Assembly Information 屏幕	462
11.6.1	ADO.NET 和 SQL Server 2000 内置的 XML 功能	393	13.1.5	编译设置	463
11.6.2	XML 与 SQL Server 2005	395	13.1.6	调试属性	464
11.7	ASP.NET 2.0 中的 XML	396	13.1.7	引用	466
11.7.1	XmlDataSource 服务器控件	396	13.1.8	资源	467
11.7.2	XmlDataSource 控件的 命名空间问题	400	13.1.9	设置	468
			13.2	Visual Studio 中的 ProVB.NET 项目	471

13.2.1 在代码中设置窗体属性	474	15.2.3 对老版本 Windows 窗体控件的替换	540
13.2.2 Visual Studio 2005 的其他组件	481	15.2.4 默认的窗体实例	541
13.3 增强范例应用程序的功能	481	15.3 System.Windows.Forms 命名空间	541
13.3.1 定制代码	483	15.4 窗体的使用	542
13.3.2 构建应用程序	487	15.4.1 通过 Sub Main 显示窗体	542
13.4 Visual Studio 2005 中的 特色功能	492	15.4.2 设置启动窗体	543
13.4.1 构建配置	492	15.4.3 启动位置	543
13.4.2 任务列表	494	15.4.4 窗体边框	544
13.4.3 Command 窗口	495	15.4.5 始终置顶——TopMost 属性	544
13.4.4 Server Explorer	495	15.4.6 附属窗体	544
13.4.5 在 Visual Studio 2005 中 记录和使用宏	496	15.4.7 改变窗体的透明度	545
13.4.6 类图	498	15.4.8 可视化继承	547
13.4.7 Team System	499	15.4.9 滚动窗体	547
13.4.8 Team Editions for Developers	501	15.4.10 MDI 窗体	547
13.4.9 性能工具	503	15.4.11 VB 2005 中的 MDI 样例	549
13.4.10 单元测试	504	15.4.12 对话框窗体	550
13.5 小结	506	15.4.13 运行时的窗体	552
第 14 章 使用 SQL Server	507	15.5 控件	553
14.1 SQL Server 2005 中的 CLR 集成	507	15.5.1 以 Tab 键切换控件的 顺序	553
14.1.1 决定使用 T-SQL 还是 VB	508	15.5.2 自动调整控件的大小 和布局	553
14.1.2 创建用户定义的类型	509	15.5.3 FlowLayoutPanel 控件	556
14.1.3 创建存储过程	523	15.5.4 TableLayoutPanel 控件	558
14.1.4 在 SQL Server 中 使用 Web 服务	530	15.5.5 Panel 和 GroupBox 容器控件	559
14.2 小结	536	15.5.6 扩展的 Provider 控件	560
第 15 章 Windows 窗体	537	15.5.7 数据输入的高级功能	562
15.1 Windows 窗体的重要性	537	15.5.8 验证数据输入	565
15.2 Windows 窗体 2.0 版的 更新内容	538	15.5.9 工具栏与新增的 ToolStrip 控件	566
15.2.1 现有控件的改进	538	15.5.10 菜单	569
15.2.2 新增控件	539	15.5.11 通用对话框	570
		15.5.12 拖放操作	573

15.5.13	标准 Windows 窗体 控件小结	575
15.5.14	已撤销的控件	578
15.5.15	控件数组	578
15.5.16	其他编程技巧	580
15.6	小结	581

第 16 章 Windows 窗体的

	高级功能	583
16.1	在可视化控件中封装逻辑	583
16.2	在 Windows 窗体中开发 自定义的控件	584
16.2.1	继承现有的控件	584
16.2.2	构建复合控件	584
16.2.3	从头编写控件	585
16.3	继承现有的控件	585
16.3.1	基本步骤	585
16.3.2	给派生的控件 添加代码	586
16.3.3	其他有用的特性	589
16.3.4	为派生的控件 自定义一个事件	589
16.3.5	创建一个可以限制选中 项数的 CheckedListBox	590
16.4	Control 与 UserControl 基类	593
16.4.1	Control 类	593
16.4.2	UserControl 类	594
16.5	复合的 UserControl 控件	595
16.5.1	创建一个复合的 UserControl	595
16.5.2	改变控件的大小	596
16.5.3	提供子控件的属性	596
16.5.4	一个具体的例子	597
16.6	从头构建控件	601
16.7	给工具箱中的控件添加 图标	607
16.8	在控件中嵌入其他控件	608
16.9	小结	610

第 17 章 Windows Presentation

	Foundation	611
17.1	内容、位置、原因、方式 ——WPF 策略	612
17.1.1	光栅图形和矢量图形	613
17.1.2	安装和配置 WPF 以 建立应用程序	614
17.1.3	创建 WPF 应用程序	614
17.1.4	XAML	617
17.1.5	Visual Studio 集成	627
17.2	小结	632

第 18 章 集成 WPF 和 Windows

	窗体	635
18.1	窗体集成——Crossbow	635
18.2	在 Windows 窗体中包含 WPF 控件	637
18.3	在 WPF 中包含 Windows 窗体控件	643
18.4	集成的限制	647
18.5	小结	648

第 19 章 使用 ASP.NET 2.0

19.1	ASP.NET 2.0 的目标	649
19.1.1	开发效率	649
19.1.2	管理和维护	649
19.1.3	性能和可伸缩性	650
19.1.4	新的编译系统	650
19.1.5	ASP.NET 应用程序的 健康监控	651
19.1.6	读写配置设置	651
19.1.7	本地化	651
19.1.8	访问数据的新对象	652
19.1.9	新的服务器控件	652
19.2	建立 ASP.NET 2.0 页面的 新 IDE	652
19.3	建立 ASP.NET 应用程序	653
19.3.1	应用程序的定位选项	653
19.3.2	IIS	655
19.3.3	FTP	655

19.3.4 Web 站点需要 FrontPage Extensions	656
19.4 ASP.NET 页面结构选项	657
19.4.1 在线编码	658
19.4.2 新的后台编码模型	660
19.5 ASP.NET 2.0 的页面指令	662
19.6 ASP.NET 页面事件	663
19.7 ASP.NET 应用程序文件夹	664
19.7.1 \App_Code 文件夹	664
19.7.2 \App_Data 文件夹	668
19.7.3 \App_Themes 文件夹	668
19.7.4 \App_GlobalResources 文件夹	668
19.7.5 \App_LocalResources 文件夹	668
19.7.6 \App_WebReferences 文件夹	669
19.7.7 \App_Browsers 文件夹	669
19.8 Global.asax	669
19.9 ASP.NET 服务器控件	672
19.9.1 服务器控件的类型	673
19.9.2 用服务器控件构建页面	674
19.9.3 使用服务器控件的事件	676
19.10 用 JavaScript 处理页面和服务器控件	677
19.10.1 使用 Page.ClientScript.RegisterClientScriptBlock	678
19.10.2 使用 Page.ClientScript.RegisterStartupScript	680
19.10.3 使用 Page.ClientScript.RegisterClientScriptInclude	681
19.11 小结	681
第 20 章 ASP.NET 2.0 的高级功能	683
20.1 应用程序和页面	683
20.1.1 跨页面传送	683
20.1.2 ASP.NET 编译功能	687
20.2 Master 页面	690
20.2.1 创建 Master 页面	691
20.2.2 创建内容页面	693
20.2.3 在应用程序范围内声明 Master 页面	696
20.2.4 为 Master 页面提供默认内容	696
20.3 数据驱动的应用程序	697
20.3.1 使用 GridView 和 SqlDataSource 控件	698
20.3.2 允许在 GridView 中编辑和删除记录	702
20.3.3 其他内容	706
20.4 页面导航	707
20.4.1 使用 SiteMapPath 服务器控件	708
20.4.2 Menu 服务器控件	709
20.4.3 TreeView 服务器控件	710
20.5 成员和角色管理	714
20.6 个性化功能	717
20.7 配置 ASP.NET	719
20.8 使用 ASP.NET 的提供程序模型	720
20.9 小结	727
第 21 章 程序集	729
21.1 程序集	729
21.2 清单	730
21.2.1 标识部分	732
21.2.2 引用的程序集	735
21.3 程序集与部署	735
21.3.1 应用程序私有的程序集	735
21.3.2 共享程序集	736
21.4 版本化问题	737

21.4.1	应用程序隔离	737	第 23 章	使用传统的 COM 和接口	785
21.4.2	并行执行	738	23.1	传统的 COM	785
21.4.3	自描述	738	23.2	COM 和 .NET 的交互	786
21.4.4	版本策略	738	23.2.1	传统的组件	787
21.4.5	配置文件	739	23.2.2	.NET 应用程序	789
21.5	程序集的动态加载	742	23.2.3	调试	791
21.5.1	Assembly 类	743	23.2.4	直接使用 TlbImp	792
21.5.2	动态加载示例	744	23.2.5	后期绑定	793
21.5.3	传入程序集	745	23.3	ActiveX 控件	798
21.6	小结	745	23.3.1	传统的 ActiveX 控件	798
第 22 章	部署	747	23.3.2	另一个 .NET 应用程序	800
22.1	应用程序部署	747	23.3.3	再次调试	802
22.1.1	.NET 对应用程序 部署方法的改进	748	23.4	在 COM 应用程序中使用 .NET 组件	802
22.1.2	XCOPY 部署	748	23.4.1	.NET 组件	803
22.1.3	使用 Windows Installer	748	23.4.2	RegAsm	804
22.1.4	ClickOnce 部署	749	23.4.3	TlbExp	807
22.2	Visual Studio 部署项目	749	23.5	小结	808
22.2.1	项目模板	749	第 24 章	线程技术	809
22.2.2	创建部署项目	751	24.1	线程的含义	809
22.3	修改部署项目	756	24.1.1	进程、应用程序域 和线程	810
22.3.1	项目属性	757	24.1.2	线程的调度	812
22.3.2	File System 编辑器	759	24.1.3	线程安全性与线程 亲和性	814
22.3.3	Registry 编辑器	762	24.1.4	何时使用线程	814
22.3.4	File Types 编辑器	764	24.1.5	设计后台任务	815
22.3.5	User Interface 编辑器	765	24.1.6	交互式应用程序	815
22.3.6	Custom Actions 编辑器	768	24.2	线程的实现	816
22.3.7	Launch Conditions 编辑器	769	24.2.1	一个简单的例子	816
22.3.8	构建	772	24.2.2	线程的实现方式	819
22.4	Windows 应用程序的 Internet 部署	772	24.2.3	手动创建线程	824
22.4.1	“无接触”部署	773	24.2.4	共享数据	825
22.4.2	ClickOnce 部署	774	24.2.5	避免共享数据的使用	826
22.4.3	自定义的部署选项	782	24.2.6	通过同步共享数据	828
22.5	小结	783	24.2.7	同步对象	830
			24.3	小结	836