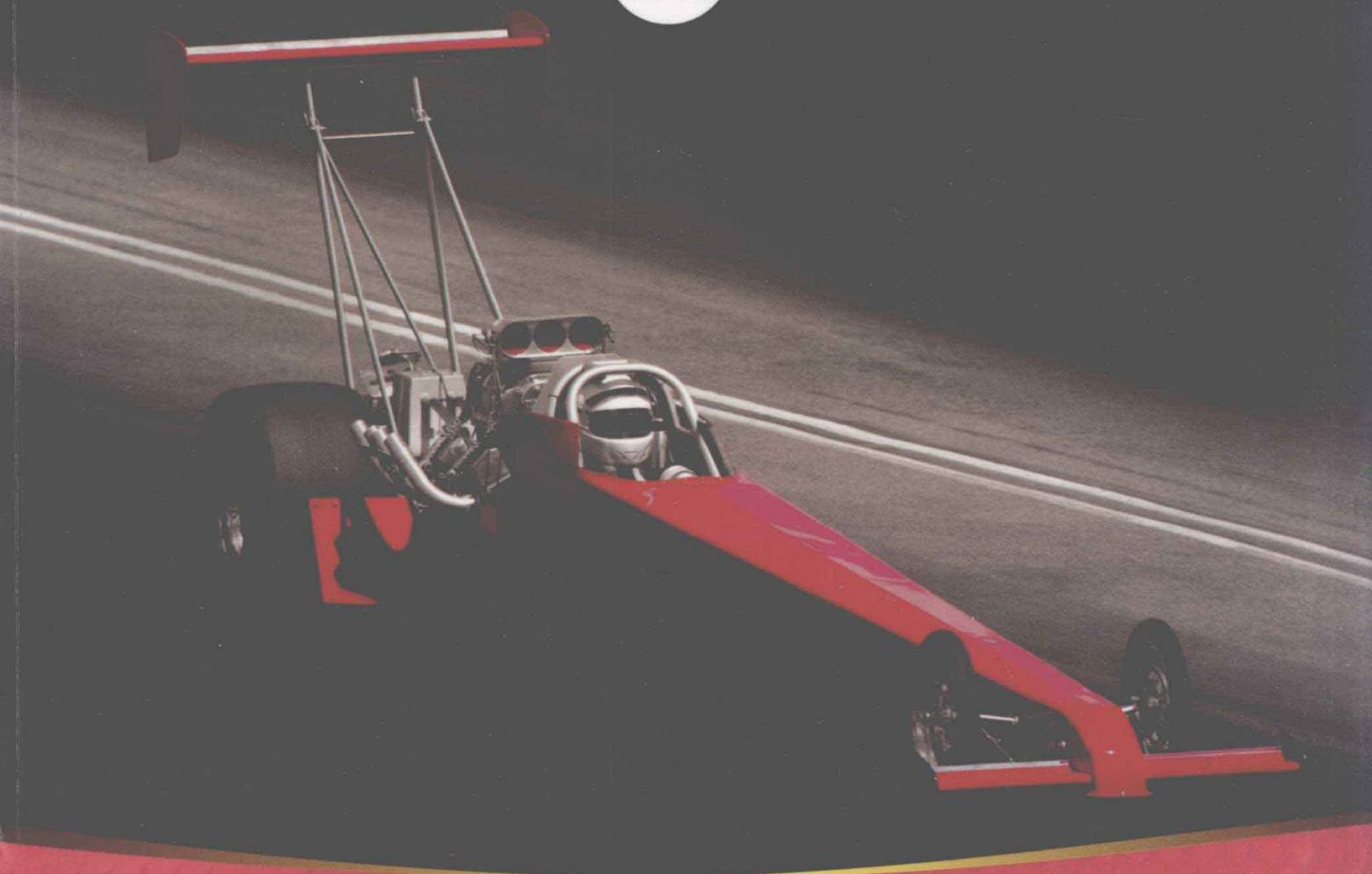


Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Professional Visual Basic 2010 and .NET 4

Visual Basic 2010 & .NET 4

高级编程(第6版)

Bill Sheldon
(美) Billy Hollis 等著
Kent Sharkey
彭琿 余科洋 译

清华大学出版社

Visual Basic 2010 & .NET 4

高级编程(第6版)

Bill Sheldon

(美) Billy Hollis 等著

Kent Sharkey

彭琿 余科洋 译

清华大学出版社

北 京

Bill Sheldon, Billy Hollis, Kent Sharkey, et al.

Professional Visual Basic 2010 and .NET 4

EISBN: 978-0-470-50224-2

Copyright © 2010 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: 01-2010-5680

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual Basic 2010 & .NET 4 高级编程(第6版)/(美) 谢尔登(Sheldon,B.), (美) 霍利斯(Hollis,B.), (美) 夏基(Sharkey,K.) 等著; 彭琨 余科洋 译.

—北京: 清华大学出版社, 2011.6

书名原文: Professional Visual Basic 2010 and .NET 4

ISBN 978-7-302-25450-8

I. V… II. ①谢… ②霍… ③夏… ④彭… ⑤余… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 079011 号

责任编辑: 王 军 吴 乐

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 何 芊

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

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 79.75 字 数: 2140 千字

版 次: 2011 年 6 月第 1 版 印 次: 2011 年 6 月第 1 次印刷

印 数: 1~3000

定 价: 139.00 元

产品编号: 038829-01

作者简介

Bill Sheldon 是一位软件架构师和工程师。他在伊利诺伊理工学院(Illinois Institute of Technology)获得了计算机科学的学位,从美国海军辞职之后,就积极投身到了软件工程的行列中。他是一位 VB 方面的 Microsoft MVP。Bill 还在加州圣地亚哥大学教授.NET 课程。除了写书之外, Bill 还发表了数十篇文章,包括“Developer Update Newsletter”、“SQL Server Magazine”特色文章和其他 Penton 出版物。他是 MSDN 的杰出在线主持人,讨论实时事件,例如 VSLive、DevConnections、Office Developers Conference 和社区事件(例如用户组和代码阵营)。可以通过其博客 www.nerdnotes.net/blog 来联系 Bill, 或通过 Twitter: NerdNotes 来联系他。

Billy Hollis 是一位作家和软件顾问。Billy 与他人合著了 VB.NET 的第一本书,还编写了关于软件开发的许多其他图书。他是 Microsoft Regional Director 计划的成员之一,也是一位 Microsoft MVP。2002 年, Billy 被选为.NET 最初的“软件先驱者”。他是.NET 平台的咨询、培训和开发的主要参加人,主要关注的是体系结构、智能客户端开发、商用软件包和用户界面技术。他会定期在全世界的主要会议上发表软件开发方面的演讲,包括 Microsoft 的 PDC 和 TechEd 事件、DevConnections、VSLive 和体系结构事件,例如 Patterns and Practices Architect Summit。

Keny Sharkey 是一位独立的顾问。在选择 Solo 之前, Keny 在 Microsoft 工作,是一位技术支持人员和内容战略家,他提倡使用.NET 技术。

Jonathan Marbutt 是英国伯明翰市 WayCool 软件开发公司的副总裁,他自从 1996 年就开始从事软件开发,掌握了从 VB 6 到.NET 的各种 Microsoft 技术。最近几年, Jonathan 一直使用 Silverlight 进行开发,为非盈利部门建立了 Rich Internet Line of Business 应用程序。在开发过程中,他开始关注如何利用 Microsoft 产品(如 Expression blend)和技术(如 Silverlight)提升用户体验。更多信息可联系 Jonathan: www.jmtechware.com。

Rob Windsor 是一位开发人员、培训师、作家、ObjectSharp 咨询公司(加拿大多伦多的一家 Microsoft 金牌合作伙伴)的资深顾问。他在使用 Delphi、VB、C#和 VB.NET 开发胖客户和 Web 应用程序方面有超过 15 年的经验,目前正在研究 SharePoint。Rob 是 INETA Speakers Bureau 和 MSDN Canada Speakers Bureau 的成员,会定期在北美和欧洲的会议、代码阵营和用户组上发表演讲。他是多伦多 VB 用户组的主席,在他参与的开发社区中被评为 Microsoft 最有价值的专家。

Gastón C. Hillar 自从 8 岁开始就在使用计算机,在 20 世纪 80 年代早期开始编程时,使用的是古老的 Texas TI-99/4A 和 Commodore 64 家用计算机。他是阿根廷布宜诺斯艾利斯的许多公司的开发人员、架构师和项目经理。现在他是一位独立的 IT 顾问,为西班牙、德国和拉丁美洲的几家公司工作,他还是一位自由作家。

Gastón 是 40 余本西班牙语图书的作者,还编写过两本英语图书。他供职于 Dobb 博士的 Go

Parallel 编程门户网站 www.parallel.com/go-parallel/, Dobb 博士的地址是 <http://drdobbs.com>, Gastón 也是 Intel Software Network 的客座博客作者。

Gastón 的联系方式是: gastonhillar@hotmail.com, 其 Twitter 地址为 <http://twitter.com/gastonhillar>, 他的博客地址为 <http://csharpmulticore.blogspot.com>。

技术编辑简介

Dianne Siebold 是一位软件开发人员和作家，擅长 VB、C#、.NET Framework、WCF、ADO 和 SQL Server。她为许多 Microsoft 合作伙伴工作，编写服务和数据访问方面的企业应用程序。目前她供职于 Microsoft 编写开发人员文档的动态产品组。她的电子邮件是 dsiebold@earthlink.net。

Doug Parsons 是一位 .NET 软件架构师和专业技术编辑，擅长 C#、SQL Server 和许多体系结构规范。在其职业生涯中，他处理过各种项目，其中最著名的是约翰·麦凯恩的美国 2008 年总统竞选网站。他目前在 NJI New Media 工作，为有显著政治倾向的客户编写软件。他的电子邮件是 douglas.c.parsons@gmail.com。

Doug WaterField 是一位软件架构师和工程师。从 Rose-Hulman 技术大学获得计算机科学的学位后，Doug 就开始使用各种技术设计和开发商用和企业应用程序。Doug 在成为主要关注 .NET 技术的独立顾问之前，领导着几家公司的开发团队和开发部门。

致 谢

与所有重要的项目一样，这么厚的一本书是一个团队的心血，作为作者，我们都非常幸运，遇到这样一个大力支持我们的团队，帮助我们跟踪和关注着本书。我要公开致谢编辑 Adaobi，他在相当短的时间内审阅了后面几章(我编写的)，还要感谢技术编辑，尤其是 Dianne，他完成了卓越的工作。他们找出了数百个问题，帮助确保本书的内容在逻辑上非常流畅。

—— Bill Sheldon

尽管写作是一项孤独的活动，但编写图书肯定不孤独，有许多人和我一起编写。

感谢所有的合著者，让我在编写本书时不会松懈。感谢合著者和编辑 Adaobi Obi Tulton、Paul Reese 和 Dianne Siebold。如果本书的质量尚可，那就是因为他们辛苦的工作。

我还要感谢所有常常要忍耐我的消极情绪，并帮助我重新振作起来的人：Eileen、Chris、Tharun、Kraig 和 Duncan。谢谢你们！

最后感谢选择购买本书的所有人，希望本书对你们能有所帮助。

—— Keny Sharkey

Beth Massi 因为太忙而无暇顾及这个项目，所以推荐我负责这个项目，感谢 Beth Massi，也感谢 Wrox 的人员接受了 Beth 的建议。

我是第一次编写这样的图书，要特别感谢 Adaobi Obi Tulton、Dianne Siebold、Doug Parsons、Doug Waterfield 和 Luann Rouff，他们帮助我把工作做得像是一位专业作者编写的那样，我知道这很不容易。

最后感谢给我提出建议，认为这个机会非常难得的人：Craig Flanagan、Sasha Krsmanovic、Jean-Rene Roy、Mark Dunn、Carl Franklin、Richard Campbell，以及所有加拿大的同事和 ObjectSharp 的所有人。

—— Rob Windsor

我要感谢 Paul Reese 和 Adaobi Obi Tulton，他们非常有耐心，允许我对自己编写的章节做出必要的修改，以包含 .NET Framework 4 和 VB 2010 引入的新并行编程特性的相关信息。

特别感谢我的妻子 Vanesa S. Olsen，因为她理解我需要同时使用许多计算机来测试每个代码段。

—— Gastón C. Hillar

前言

2002 年, 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 VB 开发的几乎所有方面。整个运行库模型变成了一个新的公共语言运行库(CLR)环境, 语言也从基于对象变成面向对象。从集成 Internet 功能到创建面向对象的框架, 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。

之后的每个版本都对 Visual Basic 语言做了进一步的改进。作为一种真正的面向对象语言, VB 引入了很多新增功能, 为底层数据提供了更好的访问特性。VB 2010 继续进行这种改进。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 是比较容易学习的, 本书就能提供这方面的帮助。

Visual Studio(VS) 2010 与 .NET Framework 4 一起使用。本书不仅详细介绍了 VB 的最新版本(即 VB 10), 还探讨了新的 .NET Framework 4, 通过这个框架, VB 开发人员能利用 Windows Presentation Foundation(WPF)、Windows 窗体、VS Tools for Office 和基于 Windows Communication Foundation(WCF)、Windows Workflow Foundation(WF)和 SharePoint 的库构建应用程序。

.NET Framework 4 自 .NET Framework 2.0 以来对核心框架进行了重大的修改。幸好与 VS 2008 一样, VS 2010 允许开发人员继续构建和部署面向 .NET 新发布版本和 .NET Framework 老版本的应用程序。

从 VB 6 开发模型迁移过来的用户会发现, 这个版本的 VB Professional 与传统的 VB 6 代码之间的区别不大。在 .NET 的 4 个版本中, 它的核心语言一直在演变, 以便与十余年前的产品区分开来。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。

同时, 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 中高效开发应用程序的必备技能。

本书将全面阐述 VB。首先讨论 VS 2010, 因为它是使用 VB 的工具, 所以理解 VS 的核心功能是成功构建 .NET 应用程序的关键。在这个过程中, 还要学习数据库访问、LINQ、Entity Framework, 以集成其他技术, 如 WPF、WCF 和基于服务的解决方案。在详细研究新特性的过程中, 会发现 VB 10 已经成为一种强大且易于使用的语言。可以像在编写桌面程序那样得心应手地编写网络程序。本书还将介绍 .NET Framework 4。

VB 的特性

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, 当生产率是最重要的因素时, 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。过去, Microsoft 和其他人常常面向不同的开发样式, 而在 VS 2010 中, Microsoft 宣布 VB 和 C# 会共同进化。在开发新语言特性时, 会同时把它们引入 VB 和 C#。这个版本是这个过程的第一步, 但目前还未完成。

共同进化并不意味着这两种语言看起来一样, 而是指它们会支持相同的功能。例如, VB 有 XML 字面量, 但并不意味着 C# 也有完全相同的功能, 因为 C# 可以通过已有的框架类处理 XML。Microsoft 不会在 VB 中先引入一个特性, 再在 C# 的下一版中引入它, 或者在 C# 中先引入一个特性, 再在 VB 的下一版中引入它。引入新功能和特性时, 它们会同时引入 VB 和 C# 中。

如前所述, 尽管还未完成修改, 但 VB 的下一个版本会与 VS 的新版本一起使用, C# 和 VB 的功能应完全相同, 因为它们都是第一流的 .NET 开发语言。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业务应用程序。

.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。在业务应用程序中, 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、更有效的应用程序。

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 2010 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 2010 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, 本书的重点是对 VB 开发人员非常重要的专业知识, 为工作在这些领域的开发人员提供了足够的内容, 开发人员也可以利用深入介绍某个技术领域的图书, 来增加自己的知识量。

使用本书需要的条件

尽管可以使用 .NET Framework 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 2010(标准版或更高, 其中包含 .NET Framework 4)。可以使用 VB Express Edition, 但一些例子将无法运行, 因为代码中的很多功能都是该版本中所没有的。此外:

- 需要 .NET Framework 4, 它会与所选的 VS 2010 版一起安装。
- 某些章节使用了 SQL Server 2008。也可以使用 VS 2010 自带的 SQL Express 来运行样例代码。
- 第 7 章使用了单元测试工具, 它包含在 VS 专业版或更高版本中。
- 某些章节使用了 Internet Information Services(IIS)。IIS 包含在 Microsoft 发布的 Windows XP 以后的每个操作系统中, 但在新操作系统中需要以管理员身份运行, 才能使用 IIS 进行开发。另外还可以利用 VS 2010 附带的开发服务器。

- 第 18 章使用了 Expression Blend 3.0。Blend 可以通过上层的 MSDN 订阅来获得，但应使用试用版研究该章介绍的功能。
- 第 24 章介绍了 SharePoint 开发。Windows Server 版本附带了 SharePoint 服务。完整的 Microsoft Office SharePoint Server 产品需要注册，但 MSDN 拥有者可以访问开发服务器。
- 第 25 章介绍了 VS Tools for Office，为了利用在 Office 套件上构建的产品，需要有指定产品的副本。
- 附录 D 在介绍排队事务时使用了 MSMQ。MSMQ 包含在 Windows 中，但它不会默认安装。
- 附录 E 介绍了 Microsoft 开发人员可以访问的联机服务。Azure 有试用期，可使用它运行该附录中的示例。

本书主要内容

第 I 部分是语言结构和环境，本书的前 7 章重点介绍 VB 开发人员使用的核心语言元素和开发工具。之后部分介绍 VS 2010、对象、语法和调试。

第 1 章首先介绍使用 VB10 的环境。该章讨论新设计的、基于 WPF 的 VS 开发环境，介绍一个简单的 Windows 窗体项目，复习调试器等重要功能，帮助读者熟悉这个强大的环境。

第 2 章是研究面向对象编程和 .NET Framework 在 VB 中的作用的三章中的第一章。该章介绍对象、类型、类型转换、引用类型和构成 VB 核心的重要语法。

第 3 章研究对象的创建，论述对象如何在 VB 中使用。首先介绍继承，然后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。该章把面向对象开发的理论应用于实践。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 4 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。该章还介绍了名称空间及其层次结构。解释了名称空间的概念和一些常用的名称空间。此外，还介绍了如何创建定制的名称空间，以及在项目中如何导入名称空间，如何给名称空间起别名。该章还介绍了 VB 中的 My 名称空间。

第 5 章中，WPF、WF 和 Silverlight 的引入给 .NET 带来了一种新语法：XAML(XML for Application Markup Language)。XAML 是新的声明式编程模型的核心。开发人员使用这个模型可以描述他们需要的对象，例如窗口。创建该窗口的代码从请求中抽象出去了。如前所述，XAML 支持几种未来的技术集。该章介绍了 XAML 语法的核心常用元素，给其他利用这个语法的章节打下基础。

第 6 章通过详细讨论 CLR 异常处理程序和 Try...Catch...Finally 结构，描述了 VB 2010 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情况的反馈。

第 7 章介绍了如何使用 VS 2010 和单元测试工具进行测试驱动的开发(TDD)。

第 II 部分接下来的 6 章(第 8~13 章)介绍了用于包含和访问数据的常用结构，包括框架元素，如数组和集合、XML、数据库访问和 WCF 服务。这些章节主要讨论如何收集用于应用程序的数据。

第 8 章介绍数组和集合，作为包含一组相关项的基础，接着扩展了这些基本结构，研究了泛型。

泛型在 .NET Framework 2.0 中引入, 支持强类型集合。 .NET Framework 4 中一个与泛型相关的重要新特性是扩展了对泛型的支持, 包含协变特性。

第 9 章展示了 .NET Framework 中生成和处理 XML 的功能。我们介绍了 .NET Framework 的 XML 相关名称空间, 并详细讨论了这些名称空间中的一部分。该章还涉及了一些使用 XML 的技术, 比如 ADO.NET 和 SQL Server, 接着详细介绍了如何给 XML 使用 LINQ。

第 10 章集中讨论了 ADO.NET 对象模型, 以便建立灵活、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 ADO.NET 的主要对象, 以便建立对 .NET 应用程序的数据访问。该章还深入讨论了 LINQ to SQL。LINQ 可以方便地访问底层数据, 它基本上位于 ADO.NET 的上层。Microsoft 把 LINQ 作为一个轻型版本, 给底层的数据库提供了一个强类型化的接口。

第 11 章中, VS 2010 版本引入的一个主要改进是 Entity Framework(EF)。EF 表示 Microsoft 实现 Entity Relationship Modeling(ERM)工具的方式。开发人员使用 EF 可以生成类, 来表示在 SQL Server 中定义的数据结构, 并在其应用程序中利用这些对象。

第 12 章描述了如何联合使用 SQL Server 2008 和 .NET 应用程序。SQL Server 提供了与应用程序的强大连接, 该章将解释如何高效地使用这个功能强大的数据库。

第 13 章介绍建立面向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相互通信提供的一种新方式。另外, 该章还将讨论 XML Web 服务的创建和使用, 论述 CLR 提供的、建立和使用 Web 服务的抽象类, 论述了一些支持 Web 服务的技术。另外还阐述了使用分布式体系结构的缺点。

第 III 部分中接下来的 6 章(第 14 章~第 19 章)讨论客户端应用程序的创建。首先介绍 .NET 1.0 引入的 Windows 窗体应用程序模型, 之后讨论 WPF、Blend 设计引擎和 Silverlight。

第 14 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, 以及 System.Windows.Forms 名称空间。

第 15 章讨论了在构建 Windows 窗体应用程序时会用到的高级功能。

第 16 章中, 创建 Windows 定制应用程序的一个最佳实践方式是使用用户控件。用户控件可以封装相关的用户界面元素。另外, 这些控件已经成为从 Windows 窗体迁移到 WPF 的关键。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 应用程序中。

第 17 章将介绍 Windows Presentation Foundation(WPF), 它是 .NET 3.0 的一个新组件, 提供了建立桌面应用程序的一种新方式。该章将介绍 WPF 所提供的相当流畅和丰富的表示层。

第 18 章介绍了在发布 WPF 的同时, Microsoft 还引入了一套新工具 Expression Studio。这些工具可以根据 XAML 构建丰富的用户界面。Blend 工具(包含在 Expression Studio 中)在设计 WPF 用户界面时尤其有效。该章介绍 Expression Blend, 它为设计应用程序和使用 XAML 提供了一组强大的工具。

第 19 章介绍了 XAML 在构建用户界面时的最新用法: Silverlight。Silverlight 为基于 .NET 的客户端应用程序开发提供了一个独立于平台的解决方案, 允许开发人员使用 XAML 标记, 给最终用户使用浏览器或桌面程序提供更流畅的体验。

第 IV 部分中接下来的 5 章(第 20 章~第 24 章)讨论 Web 应用程序的创建。这些章节利用具有客

户端特征的 Silverlight 介绍 ASP.NET 和 AJAX、MVC 等功能,包括完整的云解决方案以及 SharePoint。

第 20 章在学习了 Silverlight 及其能对客户端应用程序执行的操作后,介绍如何在网站和 Web 服务中嵌入 Silverlight, 以提供业务数据。

第 21 章介绍了 ASP.NET 的基础知识,包括使用 VS 建立 Web 应用程序,以及应用程序和页面框架。

第 22 章介绍了 ASP.NET 的几个高级功能,特别是 AJAX。其内容包括跨页面传送、母版页、站点导航、个性化等。

第 23 章介绍了 ASP.NET MVC。VS 2010 给 ASP.NET 主流开发引入了 MVC(Model-View-Controller)模式。这个模式为开发 Web 应用程序提供了更结构化的框架。该章讨论了给新 ASP.NET 项目使用这个模式的优点。

第 24 章介绍了 SharePoint。SharePoint 包含许多技术和服务,是 Microsoft 增长最快的产品。本章介绍 VB 开发人员如何定制和使用这个多功能工具来开发定制的解决方案。

第 V 部分是最后的 10 章(第 25 章~第 34 章)讨论了各个专业主题。这些主题将引用特定的 .NET 库,用户在创建新解决方案,修改旧解决方案时,读者可能对这些库的使用感兴趣。

第 25 章介绍了如何使用 VB 处理 Microsoft Office 应用程序。

第 26 章将介绍最近更新的 Workflow 实现方式。VS 2010 引入的新功能便于将工作流集成到应用程序中。Windows Workflow Foundation(WF)在 .NET Framework 3.0 中引入,但新版本与最初的逻辑有很大的区别(最初工作流的讨论移动到附录 D)。

第 27 章将介绍构建全球化应用程序时要考虑的一些重要事项,详细探讨了 System.Globalization 名称空间和它为应用程序提供的所有功能。

第 28 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 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 章讨论了在 Windows 服务的生成过程中,如何使用 VB。该章将讨论 Windows 服务的创建、安装、运行和调试。

第 31 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,以及如何使用远程服务。该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端,使用一个 singleton 对象将客户端的请求传递到业务层上。接着介绍如何使用序列化把服务器上更复杂的对象传递给客户端,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送给服务器,但无须改变对象模型。

第 32 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 名称空间可用于管理权限。我们还通过代码演示了 System.Security.Cryptography 名称空间的功能。

第 33 章将学习线程,介绍 .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。另外,VS 2010 还引入了一个全新的并行处理框架。

第 34 章介绍了 Windows 窗体和 Web 窗体的部署选项,包括 ClickOnce 部署特性和.msi 文件的创建。

第VI部分为全书的附录部分，从附录 A 到附录 E，内容如下：

附录 A 介绍了 VB 编译器 vbc.exe 所提供的功能。

附录 B 介绍了 Visual Basic Power Packs 工具，这是一组计划外的软件包，主要用于帮助开发人员维护传统 Visual Basic 6.0 应用程序或在 Visual Basic 6 中查找类似的功能。这些工具集成到 VS 中，帮助开发人员迁移到 VB 的当前版本上。

附录 C 介绍了 .NET 3.0 引入。VS 2008 支持的 Windows Workflow Foundation 在 VS 2010 中进行了全面改写。但我们把最初工作流的讨论移动到本附录中，以便读者继续给已有的解决方案参考这些内容。

附录 D 介绍了随着时间的推移，有了新的事务处理支持和相关的功能，于是本附录的内容(原来独立成章)不再重要。现在将这些内容移动到这个附录中，是为了支持引用企业服务的已有实现方案。该附录探讨了 .NET 组件服务，特别讨论了事务处理和排队组件。

附录 E 介绍了 Microsoft 提供的几个基于云的新环境，以及它们对 VB 开发人员的影响。无论是在云中保存数据，还是开发在云中运行的应用程序，该附录都有助于理解这个新的应用程序环境。

源代码

读者在阅读本书提供的代码时既可以亲自键入所有代码，也可以使用随书提供的代码文件。本书所有代码均可以从 <http://www.wrox.com> 网站下载。进入该网站后，请读者根据本书的书名查找本书(读者既可以使用搜索框进行查找，也可以使用书名列表进行查找)，然后单击本书详细内容页面上提供的 Download Code 链接，就可以下载本书提供的所有代码。



因为许多书籍名称与本书类似，因此读者也可以通过 ISBN 进行查找，本书的 ISBN 为：978-0-470-50224 -2。

下载代码后，读者可以利用一种压缩工具将代码解压。此外，读者还可以通过访问网站 <http://www.wrox.com/dynamic/books/download.aspx> 中提供的 Wrox 代码下载页面来获取本书提供的代码，也可以下载 Wrox 出版的其他书籍提供的代码。

勘误表

为了避免本书文字和代码中存在错误，我们已经竭尽全力。然而，就如世界上不存在完美无缺的事物，本书仍然可能存在错误。如果读者在我们编写的书籍中发现了诸如拼写错误或代码缺陷等问题，那么请告诉我们，我们对此表示感谢。利用勘误表反馈错误信息，可以为其他读者节省大量时间，同时，我们也能够受益于读者的帮助，这样有助于我们编写出质量更高的专业著作。

如果读者需要参考本书的勘误表，请在网站 <http://www.wrox.com> 中用搜索框或书名列表查找本书书名。然后，在本书的详细内容页面上，单击 Book Errata 链接。在随后显示的页面中，读者可以看到与本书相关的所有勘误信息，这些信息是由读者提交、并由 Wrox 的编辑们加上的。通过访问

www.wrox.com/misc-pages/booklist.shtml, 读者还可以看到 Wrox 出版的所有书籍的勘误表。

如果读者没有在 Book Errata 页面上找到其发现的错误, 那么请读者转到页面 www.wrox.com/contact/techsupport.shtml 上, 针对您所发现的每一项错误填写表格, 并将表格发送给我们, 我们将对表格内容进行认真审查, 如果确实是我们书中的错误, 那么我们将在该书的 Book Errata 页面上标明该错误信息, 并在该书的后续版本中改正相关错误。

关于 p2p.wrox.com

如果读者希望能够与作者进行讨论, 或希望能够参与读者的共同讨论, 那么请加入 p2p.wrox.com 的论坛。这个论坛是一个基于 Web 的系统, 读者可以在论坛上发表与 Wrox 出版的书籍有关的技术信息, 并与其他读者和技术用户进行讨论。论坛提供了订阅功能, 可以将与读者所选主题相关的新帖子定期发送到读者的电子邮箱。Wrox 的作者、编辑、业界专家以及其他读者都会参与论坛中的讨论。

读者可以在 <http://p2p.wrox.com> 上参与多个论坛的讨论, 这些论坛不仅能够帮助读者更好地理解本书, 还有助于读者更好地开发应用程序。如果读者希望加入论坛, 那么请按照以下步骤执行:

- (1) 进入 <http://p2p.wrox.com> 页面, 单击 Register 链接。
- (2) 阅读使用条款, 然后单击 Agree。
- (3) 填写必要的信息(必要时也需要填写可选信息), 然后单击 Submit。
- (4) 随后读者会收到一封电子邮件, 邮件中说明了如何验证帐号并完成整个加入过程。



要阅读论坛信息, 读者无须加入 P2P。但是如果读者需要发表主题或发表回复, 那么读者必须加入论坛。

成功加入论坛后, 读者就可以发表新主题了。此外, 读者还可以回复其他主题。读者在任何时间都可以阅读论坛信息。如果读者需要论坛将新的信息发送到自己的电子邮箱中, 那么可以单击论坛列表中论坛名称旁的 **Subscribe to this Forum** 图标完成该功能的设置。

如果读者需要获得更多与 Wrox P2P 相关的信息, 请阅读 P2P FAQs, 这样可以获得大量与 P2P 和 Wrox 出版的书籍相关的具体信息。阅读 FAQs 时, 请单击 P2P 页面上的 FAQs 链接。

目 录

第 I 部分 语言结构和环境

第 1 章 Visual Studio 2010	3
1.1 Visual Studio 2010: 从 Express 到 Ultimate 的各种版本	4
1.2 Visual Basic 的关键字和语法	7
1.2.1 控制台应用程序	10
1.2.2 从项目模板上创建项目	11
1.2.3 Solution Explorer 窗口	13
1.2.4 项目属性	14
1.2.5 Assembly Information 屏幕	15
1.2.6 编译设置	16
1.2.7 调试属性	20
1.2.8 引用	21
1.2.9 资源	23
1.2.10 设置	24
1.2.11 其他项目属性选项卡	26
1.3 ProVB_VS2010 项目	27
1.3.1 在代码中设置窗体属性	29
1.3.2 Visual Studio 的其他组件	37
1.4 增强示例应用程序的功能	37
1.4.1 定制代码	39
1.4.2 构建应用程序	43
1.4.3 重用第一个 Windows 窗体	50
1.5 Visual Studio 2010 中的特色功能	51
1.5.1 构建配置	51
1.5.2 任务列表	53
1.5.3 Command 窗口	54
1.5.4 Server Explorer	54

1.5.5 在 Visual Studio 2010 中记录和使用宏	55
1.5.6 类图	57
1.5.7 应用程序生命周期管理	58
1.5.8 性能工具	60
1.6 小结	62
第 2 章 对象和 Visual Basic	63
2.1 面向对象的术语	64
2.1.1 对象、类和实例	64
2.1.2 对象的组成	65
2.1.3 System.Object	68
2.2 使用 Visual Basic 类型	68
2.2.1 值类型和引用类型	69
2.2.2 基本类型	71
2.3 命令: 条件语句	72
2.3.1 If Then	73
2.3.2 比较运算符	73
2.3.3 Select Case	75
2.4 值类型(结构)	75
2.4.1 布尔类型	76
2.4.2 整数类型	77
2.4.3 无符号类型	78
2.4.4 小数类型	78
2.4.5 Char 和 Byte 类型	81
2.4.6 DateTime 类型	81
2.5 引用类型(类)	82
2.5.1 Object 类	82
2.5.2 String 类	83
2.5.3 XML 字面量	87

2.5.4	DBNull 类和 IsDBNull() 函数	89
2.6	参数传递	89
2.7	变量的作用域	91
2.8	使用对象	92
2.8.1	对象的声明和实例化	92
2.8.2	对象引用	93
2.8.3	取消对象的引用	93
2.8.4	前期绑定与后期绑定	94
2.9	数据类型转换	95
2.10	创建类	100
2.10.1	类	100
2.10.2	事件的处理	110
2.10.3	处理多个事件	111
2.10.4	WithEvents 关键字	111
2.10.5	触发事件	111
2.10.6	声明和触发定制事件	112
2.10.7	用 WithEvents 关键字 接收事件	113
2.10.8	用 AddHandler 接收事件	115
2.10.9	构造函数方法	116
2.10.10	终止和清除	117
2.11	高级概念	118
2.11.1	重载方法	119
2.11.2	重载构造函数方法	121
2.11.3	共享方法、变量和 事件	122
2.11.4	运算符重载	127
2.11.5	委托	129
2.11.6	类和组件	133
2.11.7	Lambda 表达式	134
2.12	小结	135
第 3 章	定制对象	137
3.1	继承	138
3.1.1	继承的实现	139
3.1.2	继承的层次	157
3.1.3	与基类、类及对象交互	159
3.1.4	构造函数	164
3.1.5	Protected 作用域	169
3.1.6	事件与继承	171
3.1.7	共享方法	174
3.1.8	共享事件	176
3.1.9	创建抽象基类	176
3.2	多接口	178
3.2.1	对象接口	178
3.2.2	辅助接口	180
3.3	抽象性	185
3.4	封装性	188
3.5	多态性	190
3.5.1	方法签名	190
3.5.2	实现多态性	191
3.6	进一步讨论继承	200
3.7	小结	211
第 4 章	公共语言运行库	213
4.1	.NET 应用程序的组成元素	214
4.1.1	模块	214
4.1.2	程序集	215
4.1.3	类型	215
4.2	版本化与部署	216
4.2.1	对版本化更好的支持	216
4.2.2	Major.Minor.Build.Revision 版本介绍	217
4.2.3	更好的部署	217
4.3	跨语言集成	218
4.3.1	通用类型系统	218
4.3.2	元数据	219
4.3.3	对元数据更好的支持	219
4.3.4	属性	220
4.3.5	Reflection API	222
4.4	IL 反汇编程序	222
4.5	内存管理	223

4.5.1 传统的垃圾回收机制	223	6.4 结构化异常处理的关键字	269
4.5.2 更快地为对象分配内存	230	6.4.1 Try、Catch 和 Finally 关键字	270
4.5.3 垃圾回收器的优化	231	6.4.2 Throw 关键字	271
4.6 名称空间	232	6.4.3 抛出新的异常	272
4.6.1 名称空间的概念	233	6.4.4 Exit Try 语句	273
4.6.2 名称空间与引用	236	6.4.5 嵌套的 Try 结构	274
4.6.3 常用的名称空间	237	6.4.6 异常属性的使用	275
4.6.4 导入名称空间并指定别名	239	6.4.7 Message 属性	276
4.6.5 为名称空间指定别名	240	6.4.8 InnerException 和 TargetSite 属性	276
4.6.6 在 ASP.NET 中引用名称 空间	241	6.5 与 Visual Basic 6 样式的 错误处理交互操作	280
4.7 创建自己的名称空间	241	6.6 记录错误	281
4.8 My 关键字	244	6.6.1 事件日志	281
4.8.1 My.Application 名称空间	244	6.6.2 事件、方法和属性	282
4.8.2 My.Computer 名称空间	248	6.6.3 写入跟踪文件	284
4.8.3 My.Forms 名称空间	251	6.7 小结	286
4.8.4 My.Resources 名称空间	251	第 7 章 测试驱动的开发	287
4.8.5 My.User 名称空间	251	7.1 测试的内容和方式	288
4.8.6 My.WebServices 名称空间	251	7.2 Visual Studio 中的 TDD 工具	290
4.9 扩展 My 名称空间	252	7.3 单元测试过程	291
4.10 小结	254	7.3.1 创建测试程序	291
第 5 章 用 Visual Basic 进行声明式 编程	255	7.3.2 运行测试程序	294
5.1 声明式编程与 Visual Basic	256	7.3.3 测试数据访问代码	295
5.2 使用 XAML 创建窗口	257	7.3.4 使用 Generate from Usage 特性	302
5.3 XAML 语法	260	7.4 其他 Visual Studio 版本	306
5.3.1 XAML 语言基础	261	7.5 第三方测试框架	306
5.3.2 使用 XAML 声明工作流	264	7.6 小结	307
5.4 小结	265	第 II 部分 业务对象和数据访问	
第 6 章 异常处理和调试	267	第 8 章 数组、集合和泛型	311
6.1 Visual Studio 2010 Team System 的新增内容: 历史调试	267	8.1 数组	312
6.2 与 Visual Basic 6 兼容的 注意事项	268	8.1.1 多维数组	313
6.3 .NET 中的异常处理	268		