



Professional Visual Studio 2010

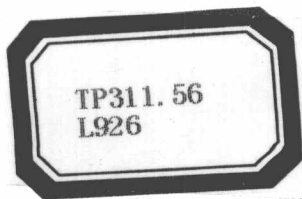
# Visual Studio 2010

# 高级编程

(美) Nick Randolph    Michael Minutillo    著  
David Gardner    Chris Anderson  
任鸿    普杰    高宇辉    译



清华大学出版社



郑州大学 \*04010745128W\*

# Visual Studio 2010 高级编程

(美) Nick Randolph Michael Minutillo 著  
David Gardner Chris Anderson  
任 鸿 普 杰 高宇辉 译



清华大学出版社

北 京

TP311.56  
L926

Nick Randolph, David Gardner, Michael Minutillo, Chris Anderson

Professional Visual Studio 2010

EISBN: 978-0-470-54865-3

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: 01-2011-0945

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 Studio 2010 高级编程/ (美) 兰多夫(Randolph,N.) 等著; 任鸿, 普杰, 高宇辉 译  
—北京: 清华大学出版社, 2012.1

书名原文: Professional Visual Studio 2010

ISBN 978-7-302-27357-8

I. ①V… II. ①兰… ②任… ③普… ④高… III. ①计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2011)第 232996 号

责任编辑: 王 军 王滋润

装帧设计: 牛艳敏

责任校对: 成凤进

责任印制: 王秀菊

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: 65.5 字 数: 1757 千字

版 次: 2012 年 1 月第 1 版 印 次: 2012 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 128.00 元

---

产品编号: 041150-01

# 作者简介

**Nick Randolph** 当前经营 Built To Roam 公司，该公司关注的是构建富移动应用程序。Nick 以前与他人共同成立了 nsquared solutions 公司，并且是该公司的开发经理，领导一个开发团队使用新一轮技术构建富有灵感的软件。在成立 nsquared 公司之前，Nick 是 Intilecta 公司的首席开发师，最初他在该公司负责设计和构建应用程序框架。

在获得工程(信息技术)/贸易双学位后，Nick 因为与 Perth .NET 用户组的精诚合作以及在移动设备方面的成就，而被任命为 Microsoft MVP。他仍通过博客 <http://community.softtek.com/blogs/nick/> 和 Professional Visual Studio 网站 [www.professionalvisualstudio.com/](http://www.professionalvisualstudio.com/) 在设备应用程序开发领域做着积极的贡献。

Nick 被邀请参加了各种会议，包括 TechEd Australia、MEDC 和 Code camp。他在 *MSDN Magazine* (ANZ edition) 上发表过文章，清华大学出版社引进并出版了他撰写的两本书《Visual Studio 2005 高级编程》和《Visual Studio 2008 高级编程》，他还是 2004 年、2005 年、2007 年和 2008 年度 Imagine Cup 决赛的评委。

**David Gardner** 是一位资深的 .NET 开发人员和 Intilecta 公司的首席软件架构师。David 致力于开发设计合理的高质量软件产品，并且一直受到用户的好评。自从 20 世纪 90 年代中期开始，他就是一位解决方案架构师、顾问和开发人员，为澳大利亚、新西兰和马来西亚等地的公司提供专家指导。

David 定期在 Perth .NET 用户组发表看法，并出席过许多会议，包括 Microsoft TechEd 和 Microsoft Executive Summit。他获得了科学(计算机科学)学士学位，是一名 Microsoft 认证系统工程师。David 与他人合著了《Visual Studio 2008 高级编程》，他还在 [www.professionalvisualstudio.com](http://www.professionalvisualstudio.com/) 网站上发布关于 Visual Studio 和 .NET 的博客。

**Michael Minutillo** 是一位 .NET 软件工程师，获得了计算机科学学士学位。他自喻为“信息海绵”。从 2000 年早期就开始编写 .NET 软件，为其大学学习奠定了基础，从那以后，他就成为 .NET 社区的一个积极成员。

Michael 定期参加 Perth .NET Community of Practice，并演示 C#、ASP.NET MVC 和 Test-Driven Philosophy 的新功能。2009 年，Michael 启动了 Perth ALT.NET 用户组，每个月他都会在 .NET 开发区讨论软件工程工具和实践情况。

Michael 维护着一个技术博客 <http://wolfbyte-net.blogspot.com>，其联系方式是 <http://twitter.com/wolfbyte>。

**Chris Anderson** 是一位具备 10 余年经验的专业开发人员，擅长使用 Microsoft 技术构建桌面、Web 和移动业务应用程序，涉及会计、财产评估、矿业、淡水产品、宠物火葬、后勤、牧场服务、销售和建筑。他获得了计算机系统工程和工程实践学士学位。Chris 与他人合作成立了 Peer Placements(软件开发人员的专家招聘)公司，还是一名顾问和作者。目前他主攻 Silverlight(特别是用 Silverlight 构建业务应用程序)，在 Code Camp Australia 2009、TechEd Australia 2009、Silverlight Code

Camp Australia 2010 和许多 Sydney Silverlight Designer and Developer Network (SDDN, Chris 是组织者之一)会议上就这个主题发表过演讲。Chris 的博客地址是 <http://chrisa.wordpress.com>。

## 技术编辑简介

**Joe Bennett** 是具备 20 余年经验的开发人员和软件解决方案架构师, 自从 .NET 1.0 推出以来就一直使用 .NET。他喜欢为各种规模的公司构建结构良好、健壮的软件, 并且愿意与其他开发人员分享他的知识和经验。

Joe 是 Carolina Software Consultants, LLC 的首席软件架构师, 过去是北卡罗来纳州 Triangle 区的 Triangle .NET 用户组的组长。他获得了休斯敦大学的计算机科学的学士学位, 并且经常在各种会议和活动中发表演讲。

# 前 言

无论从哪个方面来看, Visual Studio 2010 都是一款内容丰富的产品, 它结合了 Microsoft 以往编程语言(Visual Basic 和 C#语言)的优势, 在用户界面上承载了许多改进功能和新功能, 为初学者和经验丰富的.NET 开发人员提供了很多全新的、强大的功能。

《Visual Studio 2010 高级编程》介绍了这个开发工具的所有主要方面, 阐述了如何使用每个功能, 给出了如何高效使用各种组件的建议, 还说明了 Visual Studio 2010 的组成部分, 把用户界面分解为容易管理的块以便于理解。

《Visual Studio 2010 高级编程》还将详细地描述这些组件, 包括它们各自的作用以及相互之间如何协调工作, 使开发工作更高效。

## 本书读者对象

《Visual Studio 2010 高级编程》适合所有 Visual Studio 开发新手以及想学习一些新特性的有经验的编程人员。

熟悉 Visual Studio 编程环境的读者可以跳过本书的第 I 部分(该部分介绍了用户界面的基本构造), 直接学习本书中讨论 Visual Studio 2010 新特性的部分。即使熟悉第 I 部分的大多数内容, 也应阅读该部分, 其中可能介绍了读者以前没有见过或使用过的 Visual Studio 2010 特性。

初次使用 Visual Studio 的读者, 应该先阅读本书的第 I 部分, 该部分介绍了一些最基本的概念, 为读者展示了用户界面并讲解了如何定制自己的编程环境。

## 本书主要内容

Microsoft Visual Studio 2010 无疑是目前开发人员最好的集成开发环境(IDE)。它基于有很长历史的编程语言和接口, 受到开发环境许多不同方面的影响。

下面将简单地介绍有关 Microsoft Visual Studio 2010 的历史和它为开发人员提供的功能。已了解这方面内容的读者可以忽略该部分的内容, 直接跳到下一部分, 深入了解集成开发环境的各个方面。

### Visual Studio 简史

Microsoft 公司致力于工具的开发已经有很长的历史了, 其第一个软件产品是 1975 年的 Basic 版本。那时, 编程语言主要是解释性语言, 计算机是逐行处理代码。在过去的 30 年里, 编程有了许多改进, 最大的改进是开发环境旨在帮助开发人员用他们选择的语言和平台高效地构建应用程序。

为了顺应 32 位计算机的潮流, Microsoft 公司发布了全面的开发工具, 通常称为集成开发环境

(Integrated Development Environment, IDE)。其中包括编译器和一些补充特性，如区分上下文的编辑器和 IntelliSense，以帮助编程人员决定在给定的情况下能做什么和不能做什么。还提供了带拖放功能的直观可视化用户界面设计器，以及一系列工具窗口，可供开发人员查看特定窗口或用户控件上的各种组件的不同属性。

最初，这些 IDE 因编程语言的差异而各不相同——Visual Basic 语言因其图形化的设计而方便使用，Visual C++语言因其功能强大和灵活性较好而著名。后来在 Visual Studio 6 的旗帜下，这些语言的最新版本连同其他的 Visual 工具(如 FoxPro 和 InterDev)一起，在一个大型开发套件中发布。但显然，每种语言的工作环境还是有差别，结果就导致开发的每个解决方案都必须针对一种特定的语言。

## 一个理想的开发环境

Microsoft 公司最初于 2002 年发布了 Visual Studio .NET，它继承了该公司以前发布的各种不同开发工具的许多功能和特性。Visual Basic 6、Visual InterDev、Visual C++和其他工具(如 FoxPro)都是 Microsoft 开发团队开发的成果。该团队从外部组中导入了一些内容，但 Visual Studio .NET 2002 和 .NET 1.0 主要以 Microsoft 自己的规则和目标为基础。

Visual Studio .NET 2003 是下一个版本，它只有少量的改进，侧重于修复错误。两年后，Visual Studio 2005 和 .NET Framework 2.0 面世，接着发布了 Visual Studio 2008 和 .NET Framework 3.0、3.5，这是两个主要的版本，其中新的基础结构框架类远远超过了以往 Microsoft 公司发布的所有类，但这个版本最重要的部分是 IDE。在 IDE 中，各种组件以相互关联的方式组合在一起，提供了一个高效的工具集，便于访问所有的元素。

最新版本 Visual Studio 2010 和 .NET Framework 4.0 构建在这个强大的基础上。Windows Presentation Foundation 的代码被完全重写了，许多目前已过时的设计器又焕发出了生机。另外，还有许多新设计器可以帮助构建 Silverlight 应用程序，并且内置了构建 Office 应用程序的支持。

Visual Studio 2010 有几个主要版本：Express、Professional、Premium 以及 Ultimate。本书主要介绍 Visual Studio 2010 的专业版本，但有些实用特性只在 Premium 版本和 Ultimate 版本中才有。如果之前没使用过这些版本，请参阅第 54 章~第 57 章的有关内容。

## 本书组织结构

本书的第 I 部分主要是熟悉 Visual Studio 2010 的核心部分。从 IDE 结构和布局到各种选项和设置，前 5 章包含了使用用户界面匹配自己的工作方式所需的所有内容。

本书的其余内容分为以下部分：

- **入门：**该部分介绍如何控制项目，以及如何组织它们以符合自己的风格。
- **深入：**虽然 Visual Studio 的许多图形组件使程序员的工作更容易完成，但程序员在编码时经常需要一些其他帮助。因此，本部分介绍支持应用程序编码的功能，如 IntelliSense、代码重构以及单元测试的创建和运行。在 .NET Framework 的最新版本中，还添加了支持动态环境的改进功能，使两种主要的 .NET 语言(C#和 VB)的功能更趋于相同。该部分介绍这些语言的变化，并讨论能帮助编写出更好、更一致的代码的许多特性。

- **富客户端应用程序和 Web 应用程序:** 为了支持构建从 Office 插件到云应用程序等所有产品, Visual Studio 可以为许多平台开发应用程序。这两个部分介绍 Visual Studio 2010 支持的应用程序平台, 包括 ASP.NET、Office、WPF、Silverlight 2 和 ASP.NET MVC。
- **数据:** 大多数应用程序都使用某种形式的数据存储。Visual Studio 2010 和 .NET Framework 都包含处理数据库和其他数据源的强大支持。本部分讲述如何使用 DataSets、Visual Database Tools、LINQ、Synchronization Services 和 ADO.NET Entity Framework 构建处理数据的应用程序, 还讨论了如何使用 Reporting 展示这些数据。
- **应用程序服务:** 在构建应用程序的过程中, 可能需要访问组织内部或外部的服务。该部分介绍可用于连接这些服务的核心技术, 如 WCF、WF、Synchronization Services 和 WCF RIA Services。
- **配置和资源:** 对配置文件的内置支持允许调整应用程序执行的方式, 而不必重建应用程序。而且, 资源文件还可以访问静态数据, 把应用程序方便地本地化为另一种语言和区域性。该部分介绍如何使用 .NET 配置文件和资源文件。
- **调试:** 应用程序调试是开发人员必须完成的一项较难的任务, 但正确使用 Visual Studio 2010 的调试功能有助于分析应用程序的状态, 并确定出错的原因。该部分介绍了 IDE 提供的丰富的调试支持功能。
- **构建和部署:** 除了讨论如何构建有效的解决方案和向最终用户交付应用程序外, 该部分还涉及如何升级以前版本的项目。
- **定制和扩展 Visual Studio:** 如果之前编写的代码不足以减少以后的编码量, 就可以使用 Microsoft 提供的扩展 Visual Studio 2010 的功能。该部分从自动化模型开始介绍, 讨论了如何编写插件和宏, 以及如何使用新的扩展框架 MEF 来扩展 Visual Studio 2010。
- **Visual Studio Ultimate:** 本书的最后一部分介绍只能在 Visual Studio 2010 的 Premium 版本和 Ultimate 版本中使用的功能。另外, 还探讨了 Team Foundation Server 为管理软件项目提供的一个基本工具。

尽管对 Visual Studio 功能集进行了上述分解, 并且提供了逻辑性最强、易于理解的主题, 但读者仍可能需要查找特定的功能来帮助自己完成某个活动。为了满足这个需求, 只要在本书的其他地方详细介绍某个功能, 本书就会提供对应章节的参考。

## 本书使用前提

为了高效地使用本书, 需要安装 Microsoft Visual Studio 2010 专业版, 结合本书的内容安装软件并实际操作, 会使您的学习更高效。

本书假定读者已经熟悉传统的编程模式, 可以使用 VB 或 C# 语言进行简单的编程, 并且能够阅读用 VB 或 C# 语言编写的源代码。如果读者刚开始学编程, 使用 VB 和 C# 语言的读者可以分别参考由清华大学出版社引进并出版的《Visual Basic 2010 入门经典(第 6 版)》和《C#入门经典(第 5 版)》。

一些章节讨论了与 Visual Studio 一起使用的其他产品和工具, 它们都可以从网站上下载免费版或试用版本。



- **Code Snippet Editor:** 这是一个第三方工具, 用于在 Visual Basic 中创建代码片段。Snippet Editor 工具参见第 8 章。
- **Sandcastle:** 使用 Sandcastle 可以通过代码中的 XML 注释, 为解决方案中的每个成员和类生成详细的文档。有关 XML 注释和 Sandcastle 的内容请参见第 12 章。
- **SQL Server 2008:** Visual Studio 2010 的安装包包含 SQL Server 2008 Express, 可以构建使用数据库文件的应用程序。但对于比较全面的企业解决方案而言, 可以使用 SQL Server 2008。有关数据库连接的内容请参见第 26 章。
- **Visual Studio 2010 Premium 或 Ultimate 版:** 一个更强大的 Visual Studio 版本, 针对开发过程中的其他阶段(如测试和设计)引入了工具。有关内容请参见第 54 章~第 57 章。
- **Team Foundation Server:** 这个服务器产品提供了应用程序在 Visual Studio 2010 中的生命周期管理功能, 参见第 57 章。

## 源代码

在读者学习本书中的示例时, 可以手动输入所有的代码, 也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/>或 [www.tupwk.com.cn/downpage](http://www.tupwk.com.cn/downpage) 上下载。登录到站点 <http://www.wrox.com/>, 使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接, 就可以获得所有的源代码。



**注释:** 由于许多图书的标题都很类似, 因此按 ISBN 搜索是最简单的, 本书英文版的 ISBN 是 978-0-470-54865-3。

在下载了代码后, 只需用自己喜欢的解压缩软件对它进行解压缩即可。另外, 也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页, 查看本书和其他 Wrox 图书的所有代码。

## 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@vip.163.com](mailto:wkservice@vip.163.com) 发电子邮件, 我们就会检查您的信息, 如果是正确的, 我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表, 可以登录 <http://www.wrox.com>, 通过 Search 工具或书名列表查找本书, 然后在本书的细目页面上, 单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表, 网址是 [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml)。

## P2P.WROX.COM

要与作者和同行讨论，请加入 [p2p.wrox.com](http://p2p.wrox.com) 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 [p2p.wrox.com](http://p2p.wrox.com)，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



**注释：**不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

# 目 录

第 I 部分 集成开发环境	
第 1 章 快速入门	3
1.1 入门	3
1.1.1 安装 Visual Studio 2010	3
1.1.2 运行 Visual Studio 2010	5
1.2 Visual Studio IDE	6
1.3 小结	12
第 2 章 Solution Explorer、Toolbox 和 Properties 窗口	13
2.1 Solution Explorer 窗口	13
2.2 Toolbox 窗口	20
2.2.1 排列组件	22
2.2.2 添加组件	23
2.3 Properties 窗口	24
2.4 小结	29
第 3 章 选项和定制	31
3.1 Start Page	31
3.1.1 定制 Start Page	32
3.1.2 带用户控件的代码隐藏	34
3.2 窗口布局	36
3.2.1 查看窗口和工具栏	36
3.2.2 浏览打开的项	37
3.2.3 停靠	38
3.3 编辑区域	40
3.3.1 字体和颜色	40
3.3.2 可视化指南	41
3.3.3 全屏模式	42
3.3.4 跟踪变化	42
3.4 其他选项	43
3.4.1 快捷键	43
3.4.2 项目和解决方案	45
3.4.3 Build and Run 界面	46
3.4.4 VB 选项	47
3.5 导入和导出设置	47
3.6 小结	48
第 4 章 Visual Studio 工作区	49
4.1 代码编辑器	49
4.1.1 代码编辑器窗口的布局	49
4.1.2 区域	50
4.1.3 大纲	51
4.1.4 代码的格式化	51
4.1.5 向前/向后	53
4.1.6 高级代码编辑器功能	53
4.1.7 分割视图	54
4.1.8 代码窗口的分离(浮动)	54
4.1.9 创建选项卡组	55
4.1.10 高级功能	56
4.2 命令窗口	58
4.3 Immediate 窗口	59
4.4 Class View 工具窗口	60
4.5 Error List 窗口	61
4.6 Object Browser 窗口	61
4.7 Code Definition 窗口	62
4.8 Call Hierarchy 窗口	63
4.9 Document Outline 工具窗口	64
4.9.1 HTML 大纲	64
4.9.2 控制大纲	65
4.10 重新安排工具窗口	67
4.11 小结	68
第 5 章 查找和替换以及帮助	69
5.1 Quick Find 与 Quick Replace	69

5.1.1	Quick Find	70
5.1.2	Quick Replace	70
5.1.3	查找选项	71
5.1.4	通配符	72
5.1.5	正则表达式	72
5.1.6	Find and Replace 选项	74
5.2	文件中查找/替换	74
5.2.1	在文件中查找	74
5.2.2	查找对话框选项	76
5.2.3	结果窗口	76
5.2.4	文件中替换	77
5.3	符号查找	77
5.4	Navigate To 功能	78
5.5	增量搜索	79
5.6	访问帮助	80
5.6.1	浏览和搜索帮助系统	81
5.6.2	配置帮助系统	81
5.7	小结	82
<b>第 II 部分 入 门</b>		
<b>第 6 章</b>	<b>解决方案、项目和项</b>	<b>85</b>
6.1	解决方案的结构	85
6.2	解决方案文件的格式	86
6.3	解决方案的属性	88
6.3.1	常规属性	88
6.3.2	配置属性	88
6.4	项目类型	90
6.5	项目文件格式	91
6.6	项目属性	92
6.6.1	Application 选项卡	92
6.6.2	Compile 选项卡(仅用于 Visual Basic)	95
6.6.3	Build 选项卡(仅用于 C#和 F#)	97
6.6.4	Build Events 选项卡(仅用于 C#和 F#)	98
6.6.5	Debug 选项卡	98
6.6.6	References 选项卡(仅用于 Visual Basic)	99
6.6.7	Resources 选项卡	101
6.6.8	Services 选项卡	101
6.6.9	Settings 选项卡	102
6.6.10	Reference Paths 选项卡(仅用于 C#和 F#)	103
6.6.11	Signing 选项卡	103
6.6.12	My Extensions 选项卡(仅用于 Visual Basic)	104
6.6.13	Security 选项卡	104
6.6.14	Publish 选项卡	105
6.6.15	Code Analysis 选项卡(仅用于 VSTS Premium 和 Ultimate 版)	106
6.7	Web 应用程序项目属性	106
6.7.1	Web 选项卡	107
6.7.2	Silverlight Applications 选项卡	107
6.7.3	Package/Publish Web 选项卡	108
6.7.4	Package/Publish SQL 选项卡	109
6.8	Web Site 项目	109
6.9	小结	110
<b>第 7 章</b>	<b>IntelliSense 和书签</b>	<b>111</b>
7.1	IntelliSense 的解释	111
7.1.1	通用的 IntelliSense	112
7.1.2	单词和短语的自动完成	113
7.1.3	参数信息	118
7.1.4	快速信息	119
7.2	JavaScript Intellisense	119
7.2.1	JavaScript IntelliSense 上下文	120
7.2.2	引用另一个 JavaScript 文件	120
7.3	IntelliSense 选项	121
7.3.1	通用选项	121
7.3.2	C#的特定选项	123
7.4	扩展 IntelliSense	123
7.4.1	代码片段	124
7.4.2	XML 注释	124
7.4.3	添加自己的 IntelliSense	124
7.5	书签和 Bookmarks 窗口	124

7.6	小结	126	第 10 章	用 Class Designer 建模	165
第 8 章	代码片段和重构	127	10.1	创建类图	165
8.1	代码片段概述	128	10.2	设计界面	166
8.1.1	在 Toolbox 中存储代码块	128	10.3	工具箱	167
8.1.2	代码片段	128	10.3.1	实体	167
8.1.3	C# 中的代码片段	129	10.3.2	连接器	168
8.1.4	Visual Basic 中的代码片段	130	10.4	类的细节	168
8.1.5	用代码段进行封装	131	10.5	Properties 窗口	169
8.1.6	Code Snippets Manager	131	10.6	布局	170
8.1.7	创建代码片段	132	10.7	导出类图	170
8.1.8	查看已有的代码片段	133	10.8	代码的生成和重构	170
8.2	访问重构支持	136	10.8.1	拖放式代码生成	170
8.3	重构操作	137	10.8.2	IntelliSense 代码生成	173
8.3.1	Extract Method 重构操作	137	10.8.3	用类设计器进行重构	173
8.3.2	Encapsulate Field 重构操作	139	10.9	Modeling PowerToys for Visual Studio	174
8.3.3	Extract Interface 重构操作	141	10.9.1	可视化方面的改进	174
8.3.4	Reorder Parameters 重构操作	143	10.9.2	功能方面的改进	175
8.3.5	Remove Parameters 重构操作	144	10.10	小结	176
8.3.6	Rename 重构操作	144	<b>第 III 部分 深 入</b>		
8.3.7	Promote Variable to Parameter 重构操作	145	第 11 章	单元测试	179
8.3.8	Generate Method Stub 重构操作	146	11.1	第一个测试用例	179
8.3.9	Organize Usings 重构操作	147	11.1.1	使用特性标识测试	186
8.4	小结	148	11.1.2	其他测试特性	187
第 9 章	Server Explorer	149	11.2	指定判断条件	189
9.1	Servers 连接	149	11.2.1	Assert 类	189
9.1.1	Event Logs 节点	150	11.2.2	StringAssert 类	190
9.1.2	Management Classes 节点	152	11.2.3	CollectionAssert 类	190
9.1.3	Management Events 节点	154	11.2.4	ExpectedException 特性	190
9.1.4	Message Queues 节点	157	11.3	初始化和清理	192
9.1.5	Performance Counters 节点	160	11.3.1	TestInitialize 和 TestCleanup 特性	192
9.1.6	Services 节点	162	11.3.2	ClassInitialize 和 ClassCleanup 特性	192
9.2	Data Connections 节点	163	11.3.3	AssemblyInitialize 和 AssemblyCleanup 特性	193
9.3	SharePoint Connections 节点	164	11.4	测试环境	193
9.4	小结	164	11.4.1	数据	193

11.4.2	编写测试输出	195	14.4.3	assembly 指令	252
11.5	高级单元测试	196	14.4.4	import 指令	253
11.5.1	定制属性	196	14.4.5	include 指令	254
11.5.2	测试私有成员	198	14.5	疑难解答	255
11.6	测试 Code Contracts	199	14.5.1	设计时错误	255
11.7	管理大量的测试用例	201	14.5.2	编译转换错误	255
11.8	小结	202	14.5.3	执行转换错误	256
<b>第 12 章</b>	<b>XML 文档使用注释</b>	<b>203</b>	14.5.4	生成的代码错误	256
12.1	内联注释	203	14.6	生成代码资源	256
12.2	XML 注释	204	14.7	Preprocessed Text Template	260
12.2.1	添加 XML 注释	204	14.7.1	使用 Preprocessed Text Template	261
12.2.2	XML 注释标记	205	14.7.2	与标准 T4 模板的区别	264
12.3	使用 XML 注释	218	14.8	提示和技巧	265
12.4	用 GhostDoc 生成文档	219	14.9	小结	265
12.5	用 Sandcastle 编译文档	220	<b>第 15 章</b>	<b>项目模板和项模板</b>	<b>267</b>
12.6	Task List 注释	223	15.1	创建模板	267
12.7	小结	225	15.1.1	项模板	267
<b>第 13 章</b>	<b>编码一致性工具</b>	<b>227</b>	15.1.2	项目模板	270
13.1	源控制	227	15.1.3	模板结构	271
13.1.1	选择源控制存储库	228	15.1.4	模板参数	272
13.1.2	访问源控制	230	15.1.5	模板位置	273
13.1.3	源控制的脱机支持	234	15.2	扩展模板	273
13.2	编码标准	235	15.2.1	模板项目的安装	274
13.2.1	用 FxCop 分析代码	235	15.2.2	IWizard	275
13.2.2	用 StyleCop 指定样式	238	15.2.3	生成扩展项目模板	279
13.2.3	Code Contracts	238	15.3	Starter Kits	281
13.3	小结	240	15.4	联机模板	281
<b>第 14 章</b>	<b>用 T4 生成代码</b>	<b>241</b>	15.5	小结	282
14.1	创建 T4 模板	241	<b>第 16 章</b>	<b>语言相关特性</b>	<b>283</b>
14.2	T4 构建块	245	16.1	使用正确语言	283
14.2.1	Expression 块	245	16.1.1	命令式语言	284
14.2.2	Statement 块	246	16.1.2	声明式语言	284
14.2.3	Class Feature 块	248	16.1.3	动态式语言	284
14.3	T4 的工作原理	249	16.1.4	函数式语言	285
14.4	T4 指令	252	16.1.5	这些类别的含义	285
14.4.1	template 指令	252	16.2	两种语言的选择	286
14.4.2	output 指令	252	16.2.1	无 PIA 的编译	286

16.2.2	泛型变体	287	17.6	停靠和锚定控件	316
16.3	Visual Basic	291	17.7	小结	317
16.3.1	λ表达式和匿名方法	291	<b>第 18 章 Windows Presentation Foundation(WPF)</b> 319		
16.3.2	隐式的续行符	292	18.1	WPF 的概念	319
16.3.3	带初始值的自动属性	293	18.2	开始使用 WPF	321
16.3.4	集合初始化和数组 字面量	293	18.2.1	XAML 基础	322
16.3.5	可空的可选参数	294	18.2.2	WPF 控件	323
16.3.6	Visual Basic PowerPacks	295	18.2.3	WPF 布局控件	324
16.4	C#	295	18.3	WPF 设计器和 XAML 编辑器	326
16.4.1	带动态查找功能的 后期绑定	295	18.3.1	使用 XAML 编辑器	328
16.4.2	命名参数和可选参数	296	18.3.2	使用 WPF 设计器	328
16.5	F#	297	18.3.3	Properties 工具窗口	330
16.5.1	第一个 F#程序	297	18.3.4	数据绑定功能	333
16.5.2	研究 F#语言特性	300	18.4	设置应用程序的样式	337
16.6	小结	301	18.5	Windows Forms 的交互 操作性	338
<b>第IV部分 富客户端应用程序</b>			18.5.1	在 Windows Forms 中承载 WPF 控件	338
17	Windows Forms 应用程序	305	18.5.2	在 WPF 中承载 Windows Forms 控件	340
17.1	入门	305	18.6	用 WPF Visualizer 调试	342
17.2	Windows 窗体	306	18.7	小结	343
17.2.1	Appearance 属性	307	<b>第 19 章 Office 业务应用程序</b> 345		
17.2.2	Layout 属性	307	19.1	选择 Office 项目类型	345
17.2.3	Window Style 属性	307	19.1.1	文档级定制	346
17.3	窗体设计首选项	307	19.1.2	应用程序级插件	347
17.4	添加和定位控件	309	19.2	创建文档级定制	347
17.4.1	垂直对齐文本控件	310	19.2.1	第一个 VSTO 项目	347
17.4.2	自动定位多个控件	310	19.2.2	保护文档设计	350
17.4.3	控件的 Tab 键顺序和分层	311	19.2.3	添加操作窗格	351
17.4.4	锁定控件设计	312	19.3	创建应用程序级插件	352
17.4.5	设置控件属性	312	19.3.1	一些 Outlook 概念	352
17.4.6	基于服务的组件	313	19.3.2	创建 Outlook 窗体区域	353
17.4.7	智能标签任务	314	19.4	调试 Office 应用程序	356
17.5	容器控件	314	19.4.1	注销插件	357
17.5.1	Panel 和 SplitContainer 控件	314	19.4.2	禁用的插件	358
17.5.2	FlowLayoutPanel 控件	315			
17.5.3	TableLayoutPanel 控件	316			

19.5	部署 Office 应用程序	359	21.3	选择 Model	400
19.6	小结	360	21.4	Controller 和 Action 方法	400
<b>第 V 部分 Web 应用程序</b>					
<b>第 20 章</b>	<b>ASP.NET Web 窗体</b>	<b>363</b>	21.5	用 View 显示 UI	403
20.1	Web Application 项目和 Web Site 项目	363	21.6	高级 MVC	410
20.2	创建 Web 项目	364	21.6.1	路由	410
20.2.1	创建 Web Site 项目	364	21.6.2	Action 方法参数	414
20.2.2	创建 Web Application 项目	367	21.6.3	区域	417
20.2.3	其他 Web 项目	369	21.6.4	验证	420
20.2.4	入门套件、社区项目和 开放源代码的应用程序	369	21.6.5	部分视图	421
20.3	设计 Web 窗体	370	21.6.6	定制 View 模板	422
20.3.1	HTML 设计器	370	21.6.7	Dynamic Data 模板	422
20.3.2	定位控件和 HTML 元素	372	21.6.8	jQuery	426
20.3.3	格式化控件和 HTML 元素	374	21.7	小结	428
20.3.4	CSS 工具	374	<b>第 22 章</b>	<b>Silverlight</b>	<b>429</b>
20.3.5	验证工具	377	22.1	Silverlight 简介	429
20.4	Web 控件	379	22.2	开始使用 Silverlight	430
20.4.1	导航组件	379	22.3	Navigation Framework	435
20.4.2	用户验证	380	22.4	设置主题	436
20.4.3	数据组件	381	22.5	在浏览器外部运行	438
20.4.4	Web Parts	384	22.6	小结	440
20.5	主/从页面	384	<b>第 23 章</b>	<b>Dynamic Data</b>	<b>441</b>
20.6	富客户端开发	386	23.1	创建动态数据 Web 应用程序	442
20.6.1	用 JavaScript 开发	386	23.1.1	添加数据模型	443
20.6.2	使用 ASP.NET AJAX	387	23.1.2	研究 Dynamic Data 应用程序	444
20.6.3	使用 AJAX Control Extenders	390	23.2	定制数据模型	446
20.7	ASP.NET Web 站点管理	391	23.2.1	基架各个表	446
20.7.1	安全	392	23.2.2	定制各个数据字段	446
20.7.2	应用程序设置	394	23.2.3	添加定制的验证规则	448
20.7.3	IIS 中的 ASP.NET 配置	394	23.2.4	定制显示格式	450
20.8	小结	395	23.3	定制显示	452
<b>第 21 章</b>	<b>ASP.NET MVC</b>	<b>397</b>	23.3.1	页面模板	452
21.1	Model-View-Controller	398	23.3.2	字段模板	455
21.2	开始使用 ASP.NET MVC	398	23.3.3	实体模板	459
			23.3.4	过滤器模板	461
			23.4	为已有的项目启用动态数据	462
			23.5	小结	463



第 24 章 SharePoint	465	27.1.2 DataSet 设计器	510
24.1 准备开发环境	466	27.2 绑定数据	513
24.1.1 安装预备软件	466	27.2.1 BindingSource 类	514
24.1.2 安装 SharePoint 2010	468	27.2.2 BindingNavigator 类	516
24.2 研究 SharePoint 2010	469	27.2.3 选择数据源	517
24.3 创建 SharePoint 项目	471	27.2.4 保存更改	520
24.4 构建定制的 SharePoint 组件	474	27.2.5 插入新记录	521
24.4.1 开发 Web 部件	474	27.2.6 验证	522
24.4.2 创建内容类型和列表	475	27.2.7 定制的 DataSet	524
24.4.3 添加事件接收器	477	27.2.8 BindingSource 链和 DataGridView 控件	526
24.4.4 创建 SharePoint 工作流	478	27.3 数据源的使用	527
24.5 使用 Feature	479	27.3.1 Web 服务数据源	529
24.6 打包和部署	480	27.3.2 浏览数据	529
24.7 小结	482	27.4 小结	531
第 25 章 Windows Azure	483	第 28 章 LINQ	533
25.1 Windows Azure 平台	483	28.1 LINQ 提供程序	533
25.1.1 Development Fabric	485	28.2 旧式查询	534
25.1.2 表、博客和查询存储器	486	28.3 查询的各个部分	536
25.1.3 应用程序部署	489	28.3.1 From 语句	536
25.1.4 调整应用程序	492	28.3.2 Select 语句	537
25.2 SQL Azure	492	28.3.3 Where 语句	538
25.3 AppFabric	494	28.3.4 Group By 语句	538
25.3.1 Service Bus	494	28.3.5 定制投射	539
25.3.3 访问控制服务	494	28.3.6 Order By 语句	540
25.4 小结	494	28.4 调试和执行	541
第 VI 部分 数 据		28.5 LINQ to XML 对象模型	542
第 26 章 可视化数据库工具	497	28.5.1 VB 的 XML 字面量	543
26.1 Visual Studio 2010 中的 数据库窗口	497	28.5.2 把 XML 分析为 XElement	543
26.1.1 Server Explorer 窗口	498	28.5.2 用 LINQ 创建 XML	545
26.1.2 Data Sources 窗口	503	28.6 查询 XML	547
26.2 编辑数据	504	28.7 架构支持	548
26.3 预览数据	504	28.8 LINQ to SQL	549
26.4 小结	505	28.8.1 创建对象模型	549
第 27 章 数据集和数据绑定	507	28.8.2 用 LINQ to SQL 查询	551
27.1 DataSet 概述	507	28.8.3 绑定 LINQ to SQL 对象	556
27.1.1 添加数据源	508	28.9 LINQ Pad	560
		28.10 小结	560