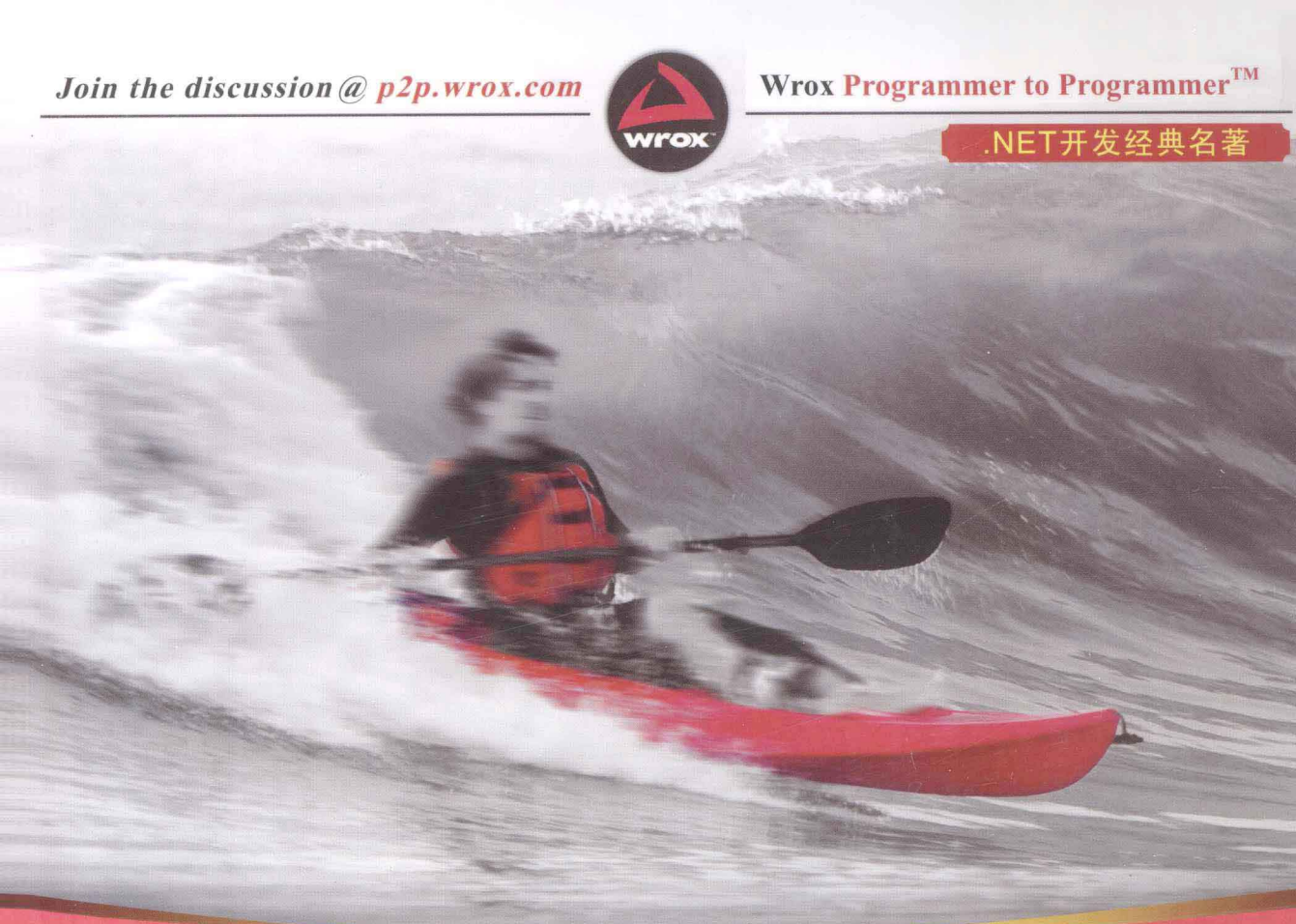


Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™

.NET开发经典名著



Professional Visual Studio 2012

Visual Studio 2012高级编程 (第4版)

[美] Bruce Johnson 著
张卫华 裴洪文 译



清华大学出版社

.NET 开发经典名著

Visual Studio 2012

高级编程

(第 4 版)

[美] Bruce Johnson 著

张卫华 裴洪文 译

清华大学出版社

北 京

Bruce Johnson

Professional Visual Studio 2012

EISBN: 978-1-118-33770-7

Copyright © 2013 by John Wiley & Sons, 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: 01-2013-4899

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

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 2012 高级编程: 第4版/(美)约翰逊(Johnson, B.)著; 张卫华, 裴洪文译. —北京: 清华大学出版社, 2014

(.NET 开发经典名著)

书名原文: Professional Visual Studio 2012

ISBN 978-7-302-34540-4

I. ①V… II. ①约… ②张… ③裴… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 277163 号

责任编辑: 王 军 于 平

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 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: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 68 字 数: 1825 千字

版 次: 2014 年 1 月第 1 版 印 次: 2014 年 1 月第 1 次印刷

印 数: 1~3000

定 价: 138.00 元

产品编号: 053103-01

译者序

Visual Studio 2012 作为一个集成解决方案，适用于个人和各种规模的开发团队。Visual Studio 2012 实现了同事间的无缝协作，提高了生产效率与专注度，最终好的点子变成了优秀的应用。同时其还是一个综合性产品系列，适用于每一位希望升级和创建精彩应用程序的组织、团队和个人开发人员。Visual Studio 2012 通过整合先前单个产品的功能并为每个版本加入新的强大功能，简化了产品系列，并且，它更适合于开发 Windows 8 应用程序。

Visual Studio 2012 具有 6 大新特性：1、支持 Windows 8 Metro 开发；2、界面更加易用，其彩色的图标和按照开发、运行、调试等环境分区的颜色方案让人爱不释手；3、集成了 ASP.NET MVC 4、全面支持移动和 HTML 5，其设计器也已经支持 C# 表达式；4、支持 .NET 4.5 (Windows RT 被提出后的首个框架库)；5、Visual Studio 2012 和 Team FILE SYSTEM 2012 一起实现了更好的生命周期管理；6、Visual Studio 2012 对系统的资源占用并不大。

非常荣幸能参与本书的翻译工作，在整个翻译过程中，自身也学到更多有关 Visual Studio 深层次的知识。尽管自己是程序员出身，在 IT 界也打拼了多年，自认为对 Visual Studio 的使用已经轻车熟路了，但参与翻译了本书之后，才发现自己的才疏学浅！

首先，本书的语言简单易懂，这对于初学者而言，是莫大的利好！本书用通俗易懂的语言向读者介绍 Visual Studio 的功能。书中所涉及的代码以及用例都是作者精心策划的。每段代码既保证了良好的可读性，同时也能很好地传达作者的意图，从而使读者也能更加轻松地理解每一个功能点！我相信读者通过逐步的学习，一定能够掌握 Visual Studio 的使用和开发秘诀！

其次，本书也介绍了一些使用 Visual Studio 编程的高级技巧，比如自定义调试窗口，读者通过掌握这些技巧，在工作中恰当的运用，便能如鱼得水般地编程。

最后，本书涵盖的内容很丰富。不仅仅介绍了 Visual Studio 工具的使用技巧，同时也包括了使用 Visual Studio 所能开发的软件类型，并介绍了每种软件类型的开发过程，读者也可以根据自己的需要重点学习部分章节！

整体说来，这是一部不可多得的介绍开发工具的经典书籍！读者从这本书中不仅仅可以学习到如何利用 Visual Studio 开发工具，也能掌握一些利用该开发工具开发特定软件类型的基本知识！相信读者在认真详细地阅读完本书后，Visual Studio 方面的技能一定会有很大的提升！

对于这本厚重之作，我本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。感激不尽！本书全部章

节由张卫华、裴洪文翻译，参与本次翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。在这里要感谢清华大学出版社的李阳和于平编辑，她们为本书的翻译投入了巨大的热情并付出了很多心血。没有她们的帮助和鼓励，本书不可能顺利付梓。

最后，祝各位开发者在学习过程中一帆风顺，能熟练掌握使用 Visual Studio 开发的技巧，完成自己满意的作品，从而实现自己的梦想。

译者

作者简介

Bruce Johnson 是 ObjectSharp Consulting 的一位合作伙伴，在计算机界具有 30 年的工作经验。他的前三个职业是做“具体的工作”，即在 UNIX 上编程。但他在将近 20 年的时间内处理的项目都使用 Windows 前沿技术，从 C++ 到 Visual Basic，再到 C#，从胖客户端应用程序到网站，再到服务。

除了喜欢建立系统之外，Bruce 还在北美会议和用户组中发言数百次。他在过去的三年里是 Microsoft Certified Trainer (MCT)，是 .NET User Group Metro Toronto 的副组长。他还为许多杂志撰写专栏和文章。Twitter 账户(<http://www.twitter.com/lacanuck>)上的活动在增多。Bruce 在过去 6 年中一直是 Microsoft MVP。

致 谢

在外人看来，写书似乎是个人的事。但我写了这么多书，觉得这并不是事实。作者的确要付出许多努力(最后拿出终稿时有很强的成就感)，但没有其他人参与编辑，书是不可能完成的。读者若要从书中获益，编辑、技术审核和责任编辑都必须付出努力。

我由衷地感谢 Wrox 中帮助我完成本书的所有人，特别是 Kelly Talbot，他的耐心和对细节的关注令人难忘，也要感谢 Andrew Moor 和 Joe Bennett，他们做了许多工作，以保证本书技术细节的准确性，最后要感谢 San Dee Phillips，他完成了许多杂项工作，确保我不至分心，并在我停止写作时，帮助我修订本书。所有这些人的努力终于促成了本书成功出版。还要感谢 Mary James，他给我撰写本书的机会。

最后要感谢 ObjectSharp 的合作人和 Microsoft 的人，尽管他们并没有意识到，但他们回答了我的问题，使写作过程能继续下去。

— Bruce Johnson

前 言

Visual Studio 在其整个历史上,都融合了 Microsoft 主要编程语言(Visual Basic 和 C#)的最新改进,这个版本也不例外。它除了支持语言特性之外,还继续支持开发人员。在理论上,使用记事本和命令窗口这样的简单工具也可以创建任意.NET 应用程序,但开发人员一般不会这么做。Visual Studio 拥有的工具和实用程序对提高开发效率的贡献高于任何一款目前的软件,在这方面,Visual Studio 2012 与以前的版本没有区别,它包含了许多改进功能和新功能,以简化开发工作。

无论从哪个方面来看,Visual Studio 2012 都是一款内容丰富的产品,所以初学者和经验丰富的.NET 开发人员要找到需要的功能比较困难。《Visual Studio 2012 高级编程(第 4 版)》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 2012 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 2012 高级编程(第 4 版)》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,以及不包含在该产品中的一些工具,使开发工作更高效。

本书读者对象

《Visual Studio 2012 高级编程(第 4 版)》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,过去各个版本的基本界面没有大的变化,但第一次启动 Visual Studio 2012 时,会发现用户界面有一些重要的变化。可以不熟悉第 I 部分,但 Visual Studio 2012 中的一些变化可以使开发更高效。毕竟,这是读者阅读本书的目的。

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 2012 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。

在 Visual Studio 2012 中,Microsoft 决定修改用户界面(且有一些延迟)。已有的开发人员可能刚开始会觉得有些不舒服(但会很快习惯,且很少遗漏默认情况下不再可见的内容)。但 Visual Studio 新手更容易上手。本书介绍 Visual Studio 2012 的功能,使读者熟悉并习惯这个新界面。

Visual Studio 2012 有几个版本: Express、Professional、Premium 以及 Ultimate。本书主要介绍

Visual Studio 2012 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 54~57 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 2012 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 11 个部分:

- **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。.NET Framework 支持动态语言,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,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 2012 支持的应用程序平台,包括 ASP.NET、Office、WPF、Silverlight 5 和 ASP.NET MVC。第 20 章介绍对新 Windows Store 应用程序的支持。
- **数据:** 大多数应用程序都使用某种形式的数据存储。Visual Studio 2012 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 2012 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,就可以使用扩展 Visual Studio 2012 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,以及如何使用 MEF 来扩展 Visual Studio 2012。

- **Visual Studio Ultimate:** 本书的最后一部分介绍只能在 Visual Studio 2012 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。

本书使用的前提

为了高效地使用本书,需要安装 Microsoft Visual Studio 2012 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。如果读者刚开始学习编程,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 2012 入门经典(第7版)》和《Visual C# 2012 入门经典(第6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 2012:** Visual Studio 2012 的安装包包含 SQL Server 2012 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 2012。有关数据库连接的内容请参见第 27 章。
- **Visual Studio 2012 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 54~57 章。
- **Team Foundation Server:** 这个服务器产品提供了应用程序在 Visual Studio 2012 中的生命周期管理功能,参见第 57 章。
- **Windows 7 或 Windows 8:** Visual Studio 2012 与 Windows 7 或 Windows 8 兼容,可以生成在 Windows XP、Windows Vista、Windows 7 和 Windows 8 上运行的应用程序。

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 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 978-1-118-33770-7。

在下载了代码后，只需要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 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。

p2p.wrox.com

要与作者和同行讨论，请加入 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，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 2012	3
1.1.2 运行 Visual Studio 2012	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
2.1.1 预览文件	15
2.1.2 常见任务	16
2.2 Toolbox 窗口	21
2.2.1 排列组件	23
2.2.2 添加组件	24
2.3 Properties 窗口	25
2.4 小结	31
第 3 章 选项和定制	33
3.1 Start Page	33
3.2 窗口布局	34
3.2.1 查看窗口和工具栏	35
3.2.2 浏览打开的项	36
3.2.3 停靠	37
3.3 编辑区域	39
3.3.1 字体和颜色	40
3.3.2 可视化指南	40
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 快速启动	44
3.4.3 项目和解决方案	45
3.4.4 Build and Run 界面	46
3.4.5 VB 选项	47
3.5 导入和导出设置	48
3.6 小结	49
第 4 章 Visual Studio 工作区	51
4.1 代码编辑器	51
4.1.1 代码编辑器窗口的布局	51
4.1.2 区域	52
4.1.3 大纲	53
4.1.4 代码的格式化	53
4.1.5 向前/向后	54
4.1.6 其他代码编辑器功能	54
4.1.7 分割视图	55
4.1.8 代码窗口的分离(浮动)	56
4.1.9 复制 Solution Explorer	57
4.1.10 创建选项卡组	58
4.1.11 高级功能	58
4.2 命令窗口	60
4.3 Immediate 窗口	61
4.4 Class View 工具窗口	62
4.5 Error List 窗口	63
4.6 Object Browser 窗口	63
4.7 Code Definition 窗口	64
4.8 Call Hierarchy 窗口	65
4.9 Document Outline 工具窗口	66
4.9.1 HTML 大纲	66
4.9.2 控制大纲	67
4.10 重新安排工具窗口	69
4.11 小结	70

第 5 章 查找和替换以及帮助 71

5.1 Quick Find 与 Quick Replace..... 71

5.1.1 Quick Find..... 72

5.1.2 Quick Replace 73

5.1.3 查找选项..... 73

5.1.4 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..... 75

5.2.3 正则表达式..... 76

5.2.4 结果窗口..... 77

5.2.5 文件中替换..... 78

5.3 Navigate To 功能..... 79

5.4 访问帮助 80

5.4.1 浏览和搜索帮助系统..... 81

5.4.2 配置帮助系统..... 81

5.5 小结 82

第 II 部分 入 门

第 6 章 解决方案、项目和项 85

6.1 解决方案的结构 85

6.2 解决方案文件的格式..... 87

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..... 89

6.4 项目类型 91

6.5 项目文件格式 92

6.6 项目属性 92

6.6.1 Application 选项卡 93

6.6.2 Compile 选项卡
(仅用于 Visual Basic)..... 96

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..... 99

6.6.6 References 选项卡
(仅用于 Visual Basic)..... 100

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 104

6.6.12 My Extensions 选项卡
(仅用于 Visual Basic)..... 104

6.6.13 Security 选项卡 105

6.6.14 Publish 选项卡..... 106

6.6.15 Code Analysis 选项卡 106

6.7 Web 应用程序项目属性 107

6.7.1 Web 选项卡 107

6.7.2 Silverlight Applications
选项卡..... 108

6.7.3 Package/Publish Web
选项卡..... 109

6.7.4 Package/Publish SQL
选项卡..... 109

6.8 Web Site 项目 110

6.9 小结..... 110

第 7 章 IntelliSense 和书签 113

7.1 IntelliSense 的解释..... 113

7.1.1 通用的 IntelliSense..... 114

7.1.2 IntelliSense 和 C#..... 115

7.1.3 单词和短语的自动完成..... 115

7.1.4 参数信息 121

7.1.5 快速信息 121

7.2 JavaScript Intellisense..... 121

7.2.1 JavaScript IntelliSense
上下文..... 122

7.2.2 引用另一个 JavaScript
文件..... 122

7.3 IntelliSense 选项..... 123

7.3.1 通用选项 123

7.3.2 C#的特定选项..... 125

7.4 扩展 IntelliSense..... 126

7.4.1 代码片段 126

7.4.2 XML 注释..... 126

7.4.3 添加自己的 IntelliSense..... 126

7.5 书签和 Bookmarks 窗口 127

7.6 小结 128

第 8 章 代码片段和重构 129

8.1 代码片段概述 130

8.1.1 在 Toolbox 中存储代码块..... 130

8.1.2 代码片段 130

8.1.3 使用 C#中的代码片段..... 131

8.1.4 Visual Basic 中的
代码片段 132

8.1.5 用代码片段进行封装 133

8.1.6 Code Snippets Manager 134

8.1.7 创建代码片段 134

8.1.8 查看已有的代码片段 135

8.2 访问重构支持 138

8.3 重构操作 139

8.3.1 Extract Method 重构操作 ... 139

8.3.2 Encapsulate Field 重构操作 ... 141

8.3.3 Extract Interface 重构操作... 143

8.3.4 Reorder Parameters
重构操作..... 145

8.3.5 Remove Parameters
重构操作..... 146

8.3.6 Rename 重构操作 146

8.3.7 Generate Method Stub
重构操作..... 147

8.3.8 Organize Usings 重构操作... 148

8.4 小结 149

第 9 章 Server Explorer..... 151

9.1 Servers 连接 151

9.1.1 Event Logs 节点 152

9.1.2 Management Classes 节点... 154

9.1.3 Management Events 节点... 156

9.1.4 Message Queues 节点 159

9.1.5 Performance Counters
节点 162

9.1.6 Services 节点..... 165

9.2 Data Connections 节点 166

9.3 SharePoint Connections 节点 ... 167

9.4 小结..... 167

第 10 章 用 Class Designer 建模 169

10.1 创建类图..... 169

10.2 设计界面..... 170

10.3 工具箱..... 171

10.3.1 实体..... 172

10.3.2 连接器..... 172

10.4 Class Details 窗口..... 173

10.5 Properties 窗口 174

10.6 布局..... 174

10.7 导出类图..... 175

10.8 代码的生成和重构..... 175

10.8.1 拖放式代码生成 175

10.8.2 IntelliSense 代码生成 177

10.8.3 用 Class Designer 进行
重构..... 178

10.9 Modeling Power Toys for
Visual Studio..... 178

10.9.1 可视化方面的改进 179

10.9.2 功能方面的改进..... 180

10.10 小结..... 181

第III部分 深 入

第 11 章 单元测试 185

11.1 第一个测试用例..... 185

11.1.1 使用特性标识测试..... 191

11.1.2 其他测试特性 191

11.2 指定判断条件..... 193

11.2.1 Assert 类..... 193

11.2.2 StringAssert 类..... 194

11.2.3 CollectionAssert 类..... 194

11.2.4 ExpectedException 特性 ... 194

11.3 初始化和清理..... 196

11.3.1 TestInitialize 和 TestCleanup
特性 196

11.3.2 ClassInitialize 和 ClassCleanup
特性 197

11.3.3 AssemblyInitialize 和
AssemblyCleanup 特性 ... 197

11.4	测试环境	197	14.4.3	assembly 指令	256
11.4.1	数据	197	14.4.4	import 指令	256
11.4.2	编写测试输出	200	14.4.5	include 指令	257
11.5	高级单元测试	201	14.5	疑难解答	258
11.5.1	定制属性	201	14.5.1	设计时错误	258
11.5.2	测试私有成员	202	14.5.2	编译转换错误	258
11.6	测试 Code Contracts	203	14.5.3	执行转换错误	259
11.7	小结	205	14.5.4	生成的代码错误	259
第 12 章	使用 XML 注释生成文档	207	14.6	生成代码资源	259
12.1	内联注释	207	14.7	Runtime Text Template	263
12.2	XML 注释	208	14.7.1	使用 Runtime Text Template	265
12.2.1	添加 XML 注释	208	14.7.2	Runtime Text Template 与 标准 T4 模板的区别	267
12.2.2	XML 注释标记	209	14.8	提示和技巧	268
12.3	使用 XML 注释	222	14.9	小结	268
12.4	用 GhostDoc 生成文档	223	第 15 章	项目模板和项模板	269
12.5	用 Sandcastle 编译文档	224	15.1	创建模板	269
12.6	Task List 注释	227	15.1.1	项模板	269
12.7	小结	229	15.1.2	项目模板	273
第 13 章	确保编码一致性的工具	231	15.1.3	模板结构	274
13.1	源控制	231	15.1.4	模板参数	276
13.1.1	选择源控制存储库	232	15.1.5	模板位置	277
13.1.2	访问源控制	234	15.2	扩展模板	277
13.2	编码标准	238	15.2.1	模板项目的安装	277
13.2.1	用 FxCop 分析代码	238	15.2.2	IWizard	279
13.2.2	用 StyleCop 指定样式	241	15.2.3	生成扩展项目模板	284
13.2.3	Code Contracts	242	15.3	Starter Kit	286
13.3	小结	244	15.4	联机模板	286
第 14 章	用 T4 生成代码	245	15.5	小结	287
14.1	创建 T4 模板	245	第 16 章	语言相关特性	289
14.2	T4 构建基块	249	16.1	使用正确的语言	289
14.2.1	Expression 块	249	16.1.1	命令式语言	289
14.2.2	Statement 块	249	16.1.2	声明式语言	290
14.2.3	Class Feature 块	251	16.1.3	动态语言	290
14.3	T4 的工作原理	252	16.1.4	函数式语言	291
14.4	T4 指令	255	16.1.5	这些类别的含义	291
14.4.1	template 指令	255	16.2	两种语言的选择	292
14.4.2	output 指令	256			

16.2.1	async 关键字	292
16.2.2	调用者信息	294
16.3	Visual Basic	294
16.3.1	迭代器	295
16.3.2	Global 关键字	295
16.3.3	Visual Basic PowerPacks	296
16.4	F#	297
16.4.1	第一个 F#程序	297
16.4.2	研究 F#语言特性	299
16.4.3	类型提供程序	301
16.4.4	查询表达式	302
16.4.5	自动实现属性	302
16.5	小结	303
第IV部分 富客户端应用程序		
第 17 章 Windows Forms 应用程序 307		
17.1	入门	307
17.2	Windows 窗体	308
17.2.1	Appearance 属性	309
17.2.2	Layout 属性	310
17.2.3	Window Style 属性	310
17.3	窗体设计首选项	310
17.4	添加和定位控件	312
17.4.1	垂直对齐文本控件	313
17.4.2	自动定位多个控件	313
17.4.3	控件的 Tab 键顺序和 分层	314
17.4.4	锁定控件设计	315
17.4.5	设置控件属性	316
17.4.6	基于服务的组件	316
17.4.7	智能标记任务	317
17.5	容器控件	318
17.5.1	Panel 和 SplitContainer 控件	318
17.5.2	FlowLayoutPanel 控件	318
17.5.3	TableLayoutPanel 控件	319
17.6	停靠和锚定控件	320
17.7	小结	320
第 18 章 Windows Presentation Foundation(WPF) 321		
18.1	WPF 的概念	321
18.2	开始使用 WPF	323
18.2.1	XAML 基础	324
18.2.2	WPF 控件	325
18.2.3	WPF 布局控件	326
18.3	WPF 设计器和 XAML 编辑器	328
18.3.1	使用 XAML 编辑器	330
18.3.2	使用 WPF 设计器	330
18.3.3	Properties 工具窗口	333
18.3.4	数据绑定功能	337
18.4	设置应用程序的样式	341
18.5	Windows Forms 的 交互操作性	343
18.5.1	在 Windows Forms 中 驻留 WPF 控件	343
18.5.2	在 WPF 中驻留 Windows Forms 控件	345
18.6	用 WPF Visualizer 调试	347
18.7	小结	348
第 19 章 Office 业务应用程序 349		
19.1	选择 Office 项目类型	349
19.1.1	文档级定制	350
19.1.2	应用程序级插件	351
19.2	创建文档级定制	351
19.2.1	第一个 VSTO 项目	351
19.2.2	保护文档设计	354
19.2.3	添加 Actions Pane	355
19.3	创建应用程序级插件	356
19.3.1	一些 Outlook 概念	356
19.3.2	创建 Outlook 窗体区域	357
19.4	调试 Office 应用程序	361
19.4.1	注销插件	361
19.4.2	禁用的插件	362
19.5	部署 Office 应用程序	363
19.6	小结	364