

PROGRAMMER TO PROGRAMMER™



Professional Visual Studio 2008

# Visual Studio 2008

## 高级编程

(美) Nick Randolph 著  
David Gardner 译  
李梦汶 任鸿



清华大学出版社

# Visual Studio 2008

## 高级编程

(美) Nick Randolph      著  
David Gardner  
李梦汶 任鸿      译

清华大学出版社

北 京

Nick Randolph, David Gardner  
Professional Visual Studio 2008

EISBN: 978-0-470-22988-0

Copyright © 2008 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: 01-2009-2538

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, 翻印必究。举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual Studio 2008 高级编程/ (美) 兰多夫(Randolph, N.), (美) 加德纳(Gardner, D.) 著;  
李梦汶, 任鸿 译. —北京: 清华大学出版社, 2009.12

书名原文: Professional Visual Studio 2008

ISBN 978-7-302-21231-7

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

中国版本图书馆 CIP 数据核字(2009)第 174663 号

责任编辑: 王 军 王滋润

装帧设计: 孔祥丰

责任校对: 胡雁翎

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: 北京密云胶印厂

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: 50 字 数: 1345 千字

版 次: 2009 年 12 月第 1 版 印 次: 2009 年 12 月第 1 次印刷

印 数: 1~3000

定 价: 98.00 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: 030090-01

# 前 言

无论从哪个方面来看, Visual Studio 2008 都是一个内容丰富的产品, 它结合了 Microsoft 以往编程语言(VB、C#语言)的优势, 在用户界面上包含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。

本书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特性, 提供了高效使用各种组件的建议, 还说明了 Visual Studio 2008 的组成部分, 把用户界面分解为容易管理的块, 以便于理解。

本书还将对这些组件进行细致的描述, 包括它们各自的作用以及相互之间如何协调工作, 使我们的开发工作更高效。

## 本书读者对象

本书适合所有 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。初次使用 Visual Studio 的读者, 则应先阅读本书的第 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。另外, 也为下面具体介绍 Visual Studio 2008 的特性做好了铺垫。

本书假定读者已熟悉传统的编程模式, 会使用 VB 或 C#语言进行简单的编程, 并能阅读用 VB 或 C#语言编写的源代码。如果读者刚开始学编程, 使用 VB 语言的读者可以参考由 Thearon Willis 和 Bryan Newsome 合著的《Visual Basic 2008 入门经典(第 5 版)》一书, 使用 C#语言的读者可以参考已由清华大学出版社引进并出版的《C#入门经典(第 4 版)》一书。

## 本书内容

Microsoft Visual Studio 2008 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, 受到开发环境的许多不同方面的影响。

下面将简单地介绍一下有关 Microsoft Visual Studio 2008 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本部分内容, 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 致力于工具的开发已有很长的历史,其第一个产品是 1975 年的 BASIC。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 发布了全面的开发工具,通常称为 IDE(Integrated Development Environment, 集成开发环境)。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智能感知功能,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互不相同——Visual Basic 因其图形化的设计而方便使用,而 Visual C++ 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 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的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,这是一个全新的版本,其中新的基础架构类远远超过了以往 Microsoft 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 2008 和 .NET Framework 3.5 则建立在这个强大的基础之上。LINQ 大肆改革了访问数据的方式,以前需要单独下载的功能,如 ASP.NET AJAX 和 Visual Studio Tools for Office,现在默认为包含在这个新版本中。

Visual Studio 2008 开发环境如图 0-1 所示,表明 Microsoft IDE 向前迈进了一大步,并朝着一套全面的工具迈进。图 0-1 展示了如何以各种方式将不同组件组装在一起,提供一个有效率的工具,并使所有元素更易于访问。

Visual Studio 2008 有几个主要版本:Express、Standard、Professional 以及 Team System(准确地说,针对不同的人员,有 4 种不同的 Team System,但它们的核心 Visual Studio 功能都是一样的)。本书主要介绍 Visual Studio 2008 的专业版本,但有些实用特性只在 Team System

中才有。如果之前没使用过 Team System，请参阅第 54~58 章的有关内容。

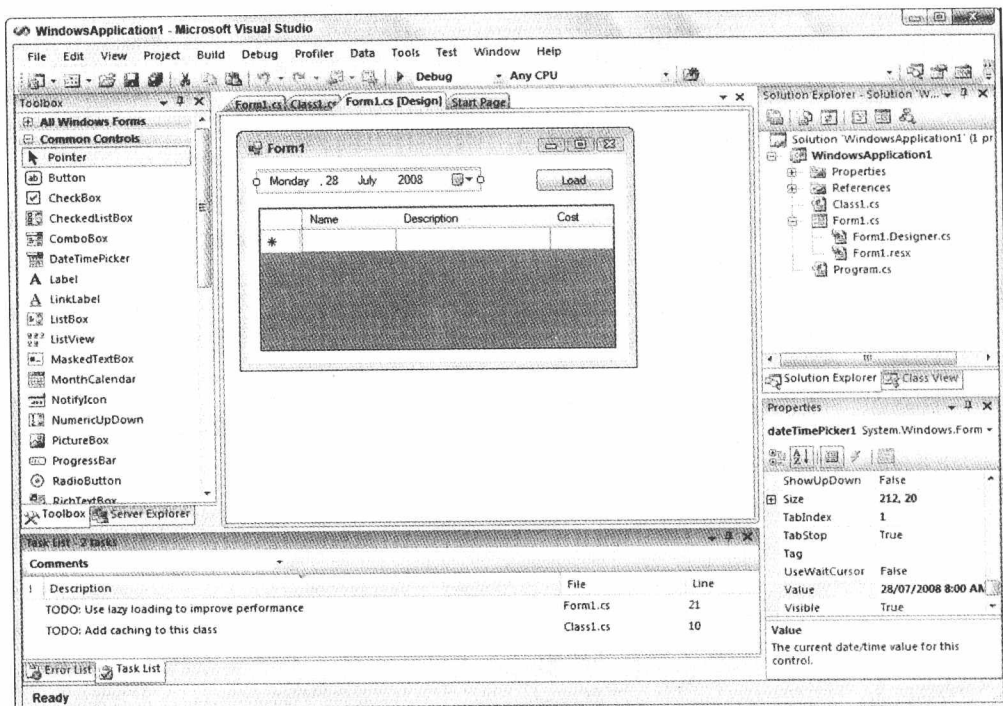


图 0-1

## 本书结构

本书的第 I 部分主要是熟悉 Visual Studio 2008 的核心部分。从集成开发环境的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 11 个部分：

- **入门：**本部分介绍如何控制项目，如何组织它们以符合自己的风格，以及如何编辑应用配置和 XML 资源文件。
- **语言：**.NET 语言继续在演变，以支持在 Framework 中添加的新功能。在 Framework 的最新版本中，添加了对 LINQ 的支持，即隐式类型化、对象初始化和  $\lambda$  表达式。把这些功能添加到以前版本引入的功能中，如泛型和部分类型，Framework 就变成了一个极其强大的应用程序构建工具。这部分将介绍所有这些功能。
- **编码：**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，如智能感知、代码重构以及单元测试的创建和运行。
- **数据：**大多数应用程序都使用某种形式的数据存储。Visual Studio 2008 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数据集、可视化数据库工具、LINQ 和同步服务，构建处理数据的应用程序。

- **安全性:** 应用程序的安全性是常被忽视或直到最后一刻才被重视的功能。本书没有将它们放到最后, 而是将它们安排在一个适当的位置。
- **平台:** 为了支持构建从 Office 插件到移动应用程序等所有产品, Visual Studio 可以为许多平台开发应用程序。本部分介绍已得到支持的应用程序平台, 包括 ASP.NET、Office 和 Mobile, 以及 .NET 3.0 引入的应用程序类型(WPF、WCF 和 WF)。本部分的最后一章介绍下一代 Web 工具 Silverlight 2 和 ASP.NET MVC。
- **配置和国际化:** 对配置文件的内置支持允许调整应用程序执行的方式, 而无需重建应用程序。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文化。本部分介绍如何使用 .NET 配置和资源文件。
- **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 2008 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。本部分介绍了 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, 这部分还涉及如何升级以前版本的项目。
- **自动化:** 如果之前编写的代码不足以减少以后的编码量, 就可以使用微软提供的许多与自动化编程概念密切相关的特性。这部分从自动化模型开始介绍, 讨论了插件和宏等内容。
- **Visual Studio Team System:** Visual Studio Team System 为公司提供了一个工具, 来支持整个软件的生命周期。本书的最后一部分介绍只能在 Visual Studio 2008 的 Team System 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。

通过以上这些介绍, 读者可以方便地找到自己所需的内容, 并深入自己感兴趣的章节进一步学习。

## 使用本书的要求

在使用本书时, 最好安装 Microsoft Visual Studio 2008 专业版, 结合本书的内容并安装软件来实际操作, 会使您的学习更高效。

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本或试用版本。

- **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 9 章。
- **F#:** F#是一种多范例的功能语言, 诞生于英国剑桥的 Microsoft 研究所。第 15 章介绍了 F#编程语言。
- **代码片段编辑器:** 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 17 章。
- **SQL Server 2005:** Visual Studio 2008 的安装软件包包含 SQL Server 2005 Express, 可以建立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 2005。有关数据库连接的内容请参见第 22 章。

- **Silverlight 2:** Silverlight 2 是一个跨平台、跨浏览器的运行库, 包含 .NET Framework 的一个轻型版本, 拥有一些高级功能, 如矢量图、动画和流媒体。有关 Silverlight 2 的内容请参见第 37 章。
- **ASP.NET MVC:** ASP.NET MVC 架构提供了一种方式, 可以清晰地把应用程序与模型、视图和控制器部分分开, 具有更高的可测试性, 能更多地控制 Web 应用程序的行为和输出。第 37 章介绍如何用 ASP.NET MVC 架构建立应用程序。
- **Web 部署项目:** 使用 Web 部署项目, 可以高效地定制应用程序, 使之可以用数量最少的一组文件来部署。有关 Web 部署项目的内容请参见第 50 章。
- **Visual Studio 2008 Team System:** 一个更强大的 Visual Studio 版本, Team System 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 Team System 的内容请参见第 54~58 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 978-0-470-22988-0。

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**。
- (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**。
- (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。如果要想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 IDE 快速入门	3
1.1 入门	3
1.2 Visual Studio IDE	4
1.3 小结	10
第 2 章 Solution Explorer、工具箱和 Properties 窗口	11
2.1 Solution Explorer	11
2.1.1 常见任务	12
2.1.2 添加项目和项	13
2.1.3 添加引用	15
2.1.4 添加服务引用	16
2.2 工具箱	17
2.2.1 安排组件	18
2.2.2 添加组件	19
2.3 Properties 窗口	20
2.4 小结	24
第 3 章 选项和定制	25
3.1 窗口布局	25
3.1.1 查看窗口和工具栏	25
3.1.2 浏览打开的项	26
3.1.3 停靠	27
3.2 编辑区域	29
3.2.1 字体和颜色	29
3.2.2 可视化指南	30
3.2.3 全屏模式	30
3.2.4 跟踪变化	31
3.3 其他选项	31
3.3.1 快捷键	32
3.3.2 项目和解决方案	33
3.3.3 Build and Run 界面	34
3.3.4 VB.NET 选项	35

3.4 导入导出设置	35
3.5 小结	36
第 4 章 工作区控制	37
4.1 命令窗口	37
4.2 Immediate 窗口	38
4.3 Class View 工具窗口	39
4.4 Object Browser	40
4.5 对象测试工作台	40
4.5.1 调用静态方法	41
4.5.2 实例化对象	42
4.5.3 访问字段和属性	42
4.5.4 调用实例方法	43
4.6 代码视图	43
4.6.1 向前/向后	44
4.6.2 区域	44
4.6.3 大纲	44
4.6.4 代码的格式化	45
4.7 Document Outline 工具窗口	45
4.8 Control Outline	47
4.9 小结	48
第 5 章 查找和替换以及帮助	49
5.1 查找与替换	49
5.1.1 快速查找	50
5.1.2 快速查找和替换对话框选项	51
5.2 文件中查找	53
5.2.1 查找对话框选项	54
5.2.2 结果窗口	54
5.3 文件中替换	55
5.4 增量搜索	56
5.5 符号查找	56
5.6 查找和替换选项	57
5.7 访问帮助	57
5.7.1 文档浏览器	58
5.7.2 动态帮助	59

5.8	Search 窗口	60	7.2.1	创建存储库	86
5.9	保存常用主题	60	7.2.2	添加解决方案	87
5.10	定制帮助功能	61	7.2.3	Solution Explorer 窗口	87
5.11	小结	62	7.2.4	登入登出	88
<b>第 II 部分 入 门</b>					
<b>第 6 章</b>	<b>解决方案、项目和项</b>	<b>65</b>	7.2.5	挂起更改	88
6.1	解决方案的结构	65	7.2.6	合并更改	89
6.2	解决方案文件的格式	66	7.2.7	历史记录	90
6.3	解决方案的属性	67	7.2.8	固定	90
6.3.1	常规属性	68	7.3	源控制的脱机支持	91
6.3.2	配置属性	68	7.4	小结	92
6.4	项目类型	69	<b>第 8 章</b>	<b>窗体和控件</b>	<b>93</b>
6.5	项目文件格式	70	8.1	Windows 窗体	93
6.6	项目属性	71	8.1.1	Appearance 属性	94
6.6.1	Application 标签	71	8.1.2	Layout 属性	95
6.6.2	Compile 标签 (仅用于 Visual Basic)	74	8.1.3	Window Style 属性	95
6.6.3	Build 标签(仅用于 C#)	75	8.2	窗体设计首选项	95
6.6.4	Debug 标签	75	8.3	添加和定位控件	96
6.6.5	References 标签 (仅用于 Visual Basic)	77	8.3.1	垂直对齐文本控件	97
6.6.6	Resources 标签	78	8.3.2	自动定位多个控件	97
6.6.7	Services 标签	78	8.3.3	锁定控件设计	98
6.6.8	Settings 标签	79	8.3.4	设置控件属性	99
6.6.9	Signing 标签	79	8.3.5	基于服务的组件	100
6.6.10	My Extensions 标签 (仅用于 Visual Basic)	80	8.3.6	智能标签任务	100
6.6.11	Security 标签	81	8.4	容器控件	101
6.6.12	Publish 标签	81	8.4.1	Panel 和 SplitContainer 控件	101
6.6.13	Web 标签(仅用于 Web 应用 程序项目)	82	8.4.2	FlowLayoutPanel 控件	102
6.7	Web Site 项目	83	8.4.3	TableLayoutPanel 控件	102
6.8	小结	83	8.5	停靠和锚定控件	103
<b>第 7 章</b>	<b>源控制</b>	<b>84</b>	8.6	小结	104
7.1	选择源控制存储库	84	<b>第 9 章</b>	<b>文档使用注释和 Sandcastle</b>	<b>105</b>
7.1.1	Environment 设置	85	9.1	内联注释	105
7.1.2	Plug-In Settings 设置	86	9.2	XML 注释	106
7.2	访问源控制	86	9.2.1	添加 XML 注释	106
			9.2.2	XML 注释标记	107
			9.3	使用 XML 注释	115
			9.4	Sandcastle 文档生成工具	116
			9.5	Task List 注释	118
			9.6	小结	120

第 10 章 项目和项模板	121	13.1 C#	162
10.1 创建模板	121	13.1.1 匿名方法	162
10.1.1 项模板	121	13.1.2 迭代器	163
10.1.2 项目模板	123	13.1.3 静态类	165
10.1.3 模板结构	124	13.1.4 命名冲突	165
10.1.4 模板参数	126	13.1.5 Pragma 语句	168
10.2 扩展模板	127	13.1.6 自动属性	169
10.2.1 模板项目的安装	127	13.2 VB.NET	169
10.2.2 IWizard	129	13.2.1 IsNot 语句	169
10.2.3 Starter 模板	132	13.2.2 Global 语句	170
10.3 小结	134	13.2.3 TryCast 语句	170
		13.2.4 三重 If 运算符	171
		13.2.5 非严格委托	171
		13.3 小结	172
		第 14 章 My 名称空间	173
		14.1 什么是 My 名称空间	173
		14.2 在代码中使用 My	175
		14.2.1 在 C# 中使用 My	175
		14.2.2 My 的环境	176
		14.2.3 默认实例	178
		14.3 名称空间概述	178
		14.3.1 My.Application	179
		14.3.2 My.Computer	180
		14.3.3 My.Forms 和 My.WebServices	183
		14.3.4 My 在 Web 中的应用	183
		14.3.5 My.Resources	183
		14.3.6 其他的 My 类	184
		14.4 My 的定制使用	184
		14.4.1 方法和属性	184
		14.4.2 扩展层次结构	185
		14.4.3 封装和部署	186
		14.5 小结	188
		第 15 章 语言生态系统	189
		15.1 用正确的锤子钉钉子	189
		15.1.1 命令式语言	189
		15.1.2 声明式语言	190
		15.1.3 动态式语言	190
		15.1.4 函数式语言	191
		15.1.5 这些类别的含义	191
第 11 章 泛型、可空类型、部分类型 和方法	137		
11.1 泛型	137		
11.1.1 使用	138		
11.1.2 创建	138		
11.1.3 约束	139		
11.2 可空类型	141		
11.3 部分类型	143		
11.3.1 窗体设计器	144		
11.3.2 部分方法	144		
11.4 运算符重载	145		
11.4.1 运算符	146		
11.4.2 类型转换	146		
11.4.3 为什么静态方法不好	147		
11.5 属性的可访问性	149		
11.6 自定义事件	149		
11.7 小结	150		
第 12 章 匿名类型、扩展方法和 λ 表达式	151		
12.1 对象和数组的初始化	151		
12.2 隐式类型化	153		
12.3 匿名类型	154		
12.4 扩展方法	156		
12.5 λ 表达式	158		
12.6 小结	160		
第 13 章 语言相关特性	162		

15.2	F#简介	192	17.3.4	Reorder Parameters (重新排列参数)	222
15.2.1	下载和安装 F#	192	17.3.5	Remove Parameters (删除参数)	223
15.2.2	第一个 F#程序	192	17.3.6	Rename (重命名)	224
15.2.3	研究 F#语言特性	195	17.3.7	Promote Variable to Parameter (把变量转换为参数)	225
15.3	小结	196	17.3.8	Generate Method Stub (生成方法存根)	225
<b>第IV部分 编 码</b>					
<b>第 16 章</b>	<b>IntelliSense 和书签</b>	<b>199</b>	17.3.9	Organize Usings (组织 Using 语句)	226
16.1	IntelliSense 的解释	199	17.4	小结	226
16.1.1	通用的 IntelliSense	199	<b>第 18 章</b>	<b>用类设计器建模</b>	<b>227</b>
16.1.2	单词和短语的自动完成	200	18.1	创建类图	227
16.1.3	参数信息	203	18.2	设计界面	228
16.1.4	快速信息	203	18.3	工具箱	229
16.2	IntelliSense 选项	203	18.3.1	实体	229
16.2.1	通用选项	203	18.3.2	连接器	230
16.2.2	C#的特定选项	205	18.4	类的细节	230
16.3	扩展 IntelliSense	206	18.5	Properties 窗口	231
16.3.1	代码片段	206	18.6	布局	232
16.3.2	XML 注释	206	18.7	导出类图	232
16.3.3	添加自己的 IntelliSense	206	18.8	代码的生成和重构	232
16.4	书签和 Bookmarks 窗口	207	18.8.1	拖放式代码生成	233
16.5	小结	208	18.8.2	IntelliSense 代码生成	234
<b>第 17 章</b>	<b>代码片段和重构</b>	<b>209</b>	18.8.3	用类设计器进行重构	235
17.1	代码片段概述	209	18.9	PowerToys for the Class Designer	235
17.1.1	基本代码片段	209	18.9.1	可视化方面的改进	235
17.1.2	“真正的”代码片段	210	18.9.2	功能方面的改进	237
17.1.3	Visual Basic 代码片段	210	18.10	小结	237
17.1.4	C#和 J#代码片段	212	<b>第 19 章</b>	<b>服务管理</b>	<b>238</b>
17.1.5	用代码段进行封装	213	19.1	Servers 节点	238
17.1.6	代码片段管理器	213	19.1.1	Event Logs 节点	239
17.1.7	创建代码片段	214	19.1.2	Management Classes 节点	241
17.1.8	查看已有的代码片段	215	19.1.3	Management Events 节点	243
17.2	访问重构支持	217	19.1.4	Message Queues 节点	245
17.3	重构操作	218	19.1.5	Performance Counters 节点	247
17.3.1	Extract Method(抽取方法)	218	19.1.6	Services 节点	249
17.3.2	Encapsulate Field (封装字段)	220	19.2	小结	250
17.3.3	Extract Interface (抽取接口)	221			

<b>第 20 章 单元测试</b> .....	251	21.4 数据源的使用 .....	299
20.1 第一个测试案例 .....	251	21.4.1 Web 服务数据源 .....	300
20.1.1 测试属性 .....	255	21.4.2 浏览数据 .....	301
20.1.2 常规测试属性 .....	256	21.5 小结 .....	302
20.2 指定判断条件 .....	257	<b>第 22 章 可视化数据库工具</b> .....	303
20.2.1 Assert 类 .....	257	22.1 Visual Studio 2008 中的	
20.2.2 StringAssert 类 .....	258	数据库窗口 .....	303
20.2.3 CollectionAssert 类 .....	258	22.1.1 Server Explorer 窗口 .....	304
20.2.4 ExpectedException 属性 .....	258	22.1.2 表的编辑 .....	305
20.3 初始化和清理 .....	259	22.1.3 关系的编辑 .....	306
20.3.1 TestInitialize 和		22.1.4 视图 .....	307
TestCleanup .....	259	22.1.5 存储过程和函数 .....	307
20.3.2 ClassInitialize 和		22.1.6 数据库图 .....	308
ClassCleanup .....	259	22.1.7 Data Sources 窗口 .....	309
20.3.3 AssemblyInitialize 和		22.2 管理测试数据 .....	313
AssemblyCleanup .....	260	22.3 预览数据 .....	314
20.4 测试环境 .....	260	22.4 小结 .....	315
20.4.1 数据 .....	260	<b>第 23 章 LINQ</b> .....	316
20.4.2 编写测试输出 .....	261	23.1 LINQ 提供程序 .....	316
20.5 高级功能 .....	262	23.2 旧式查询 .....	317
20.5.1 定制属性 .....	262	23.3 查询的各个部分 .....	318
20.5.2 测试私有成员 .....	264	23.3.1 From .....	318
20.6 管理大量的测试案例 .....	264	23.3.2 Select .....	319
20.7 小结 .....	265	23.3.3 Where .....	320
		23.3.4 Group By .....	321
		23.3.5 定制投射 .....	321
		23.3.6 Order By .....	322
		23.4 调试和执行 .....	322
		23.5 小结 .....	323
<b>第 V 部分 数 据</b>		<b>第 24 章 LINQ to XML</b> .....	324
<b>第 21 章 数据集和数据绑定</b> .....	269	24.1 XML 对象模型 .....	324
21.1 DataSet 类概述 .....	269	24.1.1 VB.NET 的 XML 字面量 .....	325
21.1.1 添加数据源 .....	270	24.1.2 把 XML 解析为 XElement .....	325
21.1.2 数据集设计器 .....	271	24.2 用 LINQ 创建 XML .....	327
21.2 数据绑定 .....	274	24.3 查询 XML .....	328
21.2.1 BindingSource 类 .....	275	24.4 模式支持 .....	329
21.2.2 BindingNavigator 类 .....	277	24.5 小结 .....	331
21.2.3 选择数据源 .....	278	<b>第 25 章 LINQ to SQL 和 Entities</b> .....	332
21.2.4 BindingSource 链 .....	280	25.1 LINQ to SQL .....	332
21.2.5 保存更改 .....	284		
21.2.6 插入新记录 .....	286		
21.2.7 验证 .....	287		
21.2.8 DataGridView 控件 .....	294		
21.3 对象数据源 .....	295		

25.1.1 创建对象模型.....332

25.1.2 用 LINQ to SQL 查询.....335

25.1.3 插入、更新和删除.....336

25.1.4 存储过程.....337

25.1.5 绑定 LINQ to SQL 对象.....339

25.2 LINQ to Entities.....341

25.3 小结.....343

**第 26 章 同步服务.....344**

26.1 偶尔连接的应用程序.....344

26.1.1 Server Direct.....344

26.1.2 开始使用同步服务.....347

26.1.3 N 层上的同步服务.....351

26.1.4 后台同步.....355

26.1.5 客户端的变化.....356

26.2 小结.....357

**第VI部分 安 全**

**第 27 章 .NET Framework 中的**

**安全性.....361**

27.1 重要的安全概念.....361

27.2 代码访问安全性.....362

27.2.1 权限集.....363

27.2.2 凭证和代码组.....363

27.2.3 安全策略.....364

27.2.4 代码访问安全性的实现.....364

27.3 基于角色的安全性.....366

27.3.1 用户标识.....367

27.3.2 实现基于角色的安全性.....368

27.4 小结.....368

**第 28 章 密码技术.....369**

28.1 基本原则.....369

28.2 技术.....369

28.2.1 散列.....369

28.2.2 对称(秘密)密钥.....371

28.2.3 非对称(公/私)密钥.....371

28.2.4 签名.....371

28.2.5 目标的总结.....372

28.3 应用密码技术.....372

28.3.1 创建非对称密钥对.....372

28.3.2 创建对称密钥.....374

28.3.3 加密和给密钥签名.....375

28.3.4 验证密钥和签名.....377

28.3.5 解密对称密钥.....378

28.3.6 发送消息.....380

28.3.7 接收消息.....381

28.4 其他.....383

28.4.1 SecureString 类.....383

28.4.2 密钥容器.....385

28.5 小结.....386

**第 29 章 混淆.....387**

29.1 MSIL 反汇编器.....387

29.2 反编译器.....388

29.3 混淆代码.....389

29.3.1 Dotfuscator.....389

29.3.2 警告.....392

29.4 属性.....393

29.4.1 ObfuscationAssemblyAttribute  
属性.....393

29.4.2 ObfuscationAttribute 属性.....394

29.5 小结.....395

**第 30 章 客户端应用程序服务.....396**

30.1 客户端服务.....396

30.2 角色授权.....398

30.3 用户验证.....400

30.4 设置.....401

30.5 登录窗体.....405

30.6 脱线支持.....406

30.7 小结.....408

**第 31 章 设备安全管理器.....409**

31.1 安全配置.....409

31.2 设备模拟.....413

31.2.1 设备仿真程序管理器.....413

31.2.2 连接.....414

31.2.3 插入底座.....414

31.3 小结.....415

**第VII部分 平 台**

**第 32 章 ASP.NET Web 应用程序.....419**

32.1	Web 应用程序和 Web Site 项目 .....	419	33.2.3	添加操作窗格 .....	456
32.2	创建 Web 项目 .....	420	33.3	创建应用程序级插件 .....	458
32.2.1	创建 Web Site 项目 .....	420	33.3.1	一些 Outlook 概念 .....	458
32.2.2	创建 Web Application 项目 .....	422	32.3.2	创建 Outlook 窗体区域 .....	459
32.2.3	其他 Web 项目 .....	424	33.4	调试 Office 应用程序 .....	462
32.2.4	入门套件、团队项目和 开放源代码的应用程序 .....	424	33.4.1	注销插件 .....	462
32.3	设计 Web 窗体 .....	425	33.4.2	禁用的插件 .....	463
32.3.1	HTML 设计器 .....	425	33.5	部署 Office 应用程序 .....	464
32.3.2	定位控件和 HTML 元素 .....	427	33.6	小结 .....	464
32.3.3	格式化控件和 HTML 元素 .....	429	<b>第 34 章</b>	<b>移动应用程序 .....</b>	<b>465</b>
32.3.4	CSS 工具 .....	429	34.1	基本知识 .....	465
32.3.5	验证工具 .....	433	34.2	设计皮肤 .....	466
32.4	Web 控件 .....	434	34.2.1	方向 .....	467
32.4.1	导航组件 .....	434	34.2.2	按钮 .....	468
32.4.2	用户验证 .....	435	34.3	Toolbox 窗口 .....	468
32.4.3	数据组件 .....	436	34.3.1	标准控件 .....	469
32.4.4	Web Parts .....	438	34.3.2	移动控件 .....	469
32.5	主/从页面 .....	439	34.4	调试 .....	475
32.6	富客户端开发 .....	440	34.5	项目设置 .....	475
32.6.1	用 JavaScript 开发 .....	441	34.6	数据源 .....	476
32.6.2	使用 ASP.NET AJAX .....	441	34.6.1	数据集 .....	478
32.6.3	使用 AJAX Control Extenders .....	444	34.6.2	ResultSet .....	484
32.7	ASP.NET Web 站点管理 .....	446	34.7	Windows Mobile API .....	484
32.7.1	安全 .....	447	34.7.1	Configuration 名称空间 .....	485
32.7.2	应用程序设置 .....	448	34.7.2	Forms 名称空间 .....	485
32.7.3	IIS 中的 ASP.NET 配置 .....	448	34.7.3	PocketOutlook 名称空间 .....	486
32.8	小结 .....	449	34.7.4	Status 名称空间 .....	486
<b>第 33 章</b>	<b>Office 应用程序 .....</b>	<b>450</b>	34.7.5	Telephony 名称空间 .....	487
33.1	选择 Office 项目类型 .....	450	34.7.6	Notification Broker .....	487
33.1.1	文档级定制 .....	451	34.8	小结 .....	488
33.1.2	应用程序级插件 .....	452	<b>第 35 章</b>	<b>WPF 应用程序 .....</b>	<b>489</b>
33.1.3	SharePoint 工作流 .....	452	35.1	基本知识 .....	489
33.1.4	InfoPath 窗体模板 .....	453	35.2	WPF 设计器 .....	491
33.2	创建文档级定制 .....	453	35.2.1	操作控件 .....	492
33.2.1	第一个 VSTO 项目 .....	453	35.2.2	属性和事件 .....	493
33.2.2	保护文档设计 .....	456	35.3	给应用程序设置样式 .....	494
			35.4	与 Windows 窗体的 交互操作性 .....	496
			35.5	小结 .....	499



第 36 章 WCF 和 WF 应用程序.....	500		
36.1 Windows Communication Foundation .....	500		
36.2 Windows Workflow Foundation .....	507		
36.3 小结 .....	511		
第 37 章 下一代 Web: Silverlight 和 ASP.NET MVC .....	512		
37.1 Silverlight .....	512		
37.1.1 开始使用 Silverlight 2 .....	513		
37.1.2 与 Web 页面交互操作 .....	517		
37.1.3 承载 Silverlight 应用程序 .....	519		
37.2 ASP.NET MVC .....	520		
37.2.1 Model - View - Controller .....	521		
37.2.2 开始使用 ASP.NET MVC .....	521		
37.2.3 Controller 和 Action 方法 .....	523		
37.2.4 用 Views 显示 UI .....	526		
37.2.5 定制 URL 路由 .....	528		
37.3 小结 .....	529		
<b>第 VIII 部分 配置和国际化</b>			
第 38 章 应用程序配置文件 .....	533		
38.1 配置文件 .....	533		
38.1.1 machine.config .....	533		
38.1.2 web.config .....	533		
38.1.3 app.config .....	534		
38.1.4 security.config .....	534		
38.2 配置架构 .....	534		
38.2.1 configurationSections 区域 .....	534		
38.2.2 startup 区域 .....	536		
38.2.3 runtime 区域 .....	536		
38.2.4 system.runtime.remoting 区域 .....	537		
38.2.5 system.net 区域 .....	537		
38.2.6 cryptographySettings 区域 .....	538		
38.2.7 system.diagnostics 区域 .....	538		
38.2.8 system.web 区域 .....	538		
38.2.9 webserver 区域 .....	539		
38.2.10 compiler 区域 .....	539		
		38.2.11 配置属性 .....	539
	38.3 应用程序设置 .....		541
	38.3.1 使用 appSettings .....		541
	38.3.2 项目设置 .....		541
	38.3.3 动态属性 .....		542
	38.3.4 自定义配置区域 .....		543
	38.4 用设置引用的项目 .....		546
	38.5 小结 .....		547
第 39 章 连接字符串 .....	548		
39.1 连接字符串向导 .....	548		
39.2 SQL Server 格式 .....	552		
39.3 代码构建 .....	553		
39.4 加密连接字符串 .....	554		
39.5 小结 .....	556		
第 40 章 资源文件 .....	557		
40.1 什么是资源 .....	557		
40.1.1 文本文件资源 .....	557		
40.1.2 Resx 资源文件 .....	558		
40.1.3 二进制资源文件 .....	559		
40.1.4 添加资源 .....	559		
40.1.5 嵌入文件作为资源 .....	560		
40.1.6 资源命名 .....	561		
40.1.7 访问资源 .....	561		
40.1.8 设计器文件 .....	562		
40.2 为应用程序提供资源 .....	562		
40.3 辅助资源 .....	565		
40.3.1 文化 .....	565		
40.3.2 创建文化资源 .....	565		
40.3.3 加载文化资源文件 .....	566		
40.3.4 辅助文化资源 .....	566		
40.4 访问细节 .....	567		
40.4.1 加载位图和图标 .....	567		
40.4.2 跨程序集引用 .....	567		
40.4.3 ComponentResourceManager 对象 .....	568		
40.5 编码资源文件 .....	568		
40.5.1 ResourceReader 和 ResourceWriter .....	569		
40.5.2 ResxResourceReader 和 ResxResourceWriter .....	569		