

PROGRAMMER TO PROGRAMMER™



Professional Visual Studio 2005

Visual Studio 2005 高级编程

Andrew Parsons 著
(美) Nick Randolph 译
吴雷



清华大学出版社

Visual Studio 2005 高级编程

Andrew Parsons
(美) Nick Randolph 著
吴 雷 译

清华大学出版社

北 京

Andrew Parsons, Nick Randolph

Professional Visual Studio 2005

EISBN: 0-7645-9846-5

Copyright©2006 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: 01-2006-1587

本书封面贴有清华大学出版社防伪标签, 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 2005 高级编程/(美)帕森斯(Parsons, A.), (美)伦道夫(Randolph, N.)著; 吴雷译.

—北京: 清华大学出版社, 2008.6

书名原文: Professional Visual Studio 2005

ISBN 978-7-302-17552-0

I.V… II.①帕… ②伦… ③吴… III.计算机网络—程序设计 IV.TP393

中国版本图书馆 CIP 数据核字(2008)第 064370 号

责任编辑: 王 军 王 婷

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 孟凡玉

出版发行: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

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

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 北京密云胶印厂

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: 52.5 字 数: 1278 千字

版 次: 2008 年 6 月第 1 版 印 次: 2008 年 6 月第 1 次印刷

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: 021159-01

作者简介

Andrew Parsons

Andrew Parsons 是一名多才多艺的程序员、记者和作者。他创办、发行了 *Australian Developer* 杂志，并担任该杂志的总编。该杂志获得了巨大的成功，以至于它在全球范围内扩展，最终发展成为现在的 *International Developer*。随后，Parsons 又推出了澳大利亚和新西兰本地版的 MSDN 杂志。此外，他还编写了各种各样的技术书籍，内容涉及 HTML、CSS、Photoshop 和 Visual Basic Express 等。在写作之外的时间里，Parsons 为许多用户提供了 .NET 编程实现方面的咨询，他目前担任 Readify Pty, Ltd(www.readify.net)的高级顾问。与此同时，他还努力地经营自己的业务——Parsons Designs(www.parsonsdesigns.com)和 GAMEparents (www.gameparents.com)，并通过这些网站来帮助家长了解和享受电脑和电子游戏。

Nick Randolph

Nick Randolph 是一位经验丰富的 .NET 开发员和解决方案架构师。他在澳大利亚一家非营利机构担任软件工程师时，创办了 Perth .NET Community of Practice 并全面参与当地的 .NET 社区至今。后来，Nick 加入 AutumnCare(www.autumncare.com.au)并担任开发部经理，他负责产品的结构，该结构整合了一些有关使用 .NET Framework 构建智能客户端应用程序的最佳实践。Nick 目前担任 SoftTeq (<http://softteq.com>)的解决方案架构师，该公司提供咨询、培训和指导服务。作为 Microsoft MVP 的一员，Nick 已多次被应邀出席各种 IT 会议，如 TechEd、MEDC 和 Code Camp，并在过去的两年里一直担任微软创新杯决赛的评审。

前 言

Visual Studio 2005 包含了一系列高效的、智能的开发工具，它结合了 Microsoft 以往编程语言(VB、C# 语言)的优势，不论是为开发初学者，还是为经验丰富的企业开发团队都提供了很多全新的、强大的功能。

Visual Studio 2005 专业版是一个为专业开发人员设计的完善而全面的开发环境，比以往任何一个版本的 Visual Studio 都更加高效地适用于开发高性能的应用程序。本书将向您展示构成 Visual Studio 2005 的模块构建，并分别介绍用户的各个界面。

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以使得你的部署能够更加有效地实现。

本书读者对象

Visual Studio 2005 专业版适合所有新接触 Visual Studio 的开发人员以及想学习一些未曾使用过的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。而初次使用 Visual Studio 的读者，则应先阅读本书的第 I 部分，它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 2005 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 Thearon Willis 和 Bryan Newsome 著的《Visual Basic 2005 入门经典》，使用 C# 语言的读者可以参考多个人合著的《Visual Basic 2005 入门经典》。

本书内容

Microsoft Visual Studio 2005 无疑是目前开发人员最好的集成开发环境(IDE)。它混合了多种用不同语言编写的组件和使用了多种不同的部署结构。下面将简单地介绍一下有关 Microsoft Visual Studio 2005 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不看。

Visual Studio 简史

Microsoft 致力于开发工具的开发已有 25 年的历史，其第一个产品是 Microsoft Basic，于 1975 发行。在过去的四分之一世纪里，开发工具的功能得到了令人难以置信的增强，而 Microsoft 则一直没有忘记其最高目标——帮助开发人员以更高的效率构建运行在 Microsoft 平台上的功能更强大的软件。

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，如一个敏感的与上下文相关的(context-sensitive)帮助功能和智能感知功能以帮助编程人员决定输入。还提供了一些实现可视化用户界面设计的功能，如可拖拽放置的元素，一系列可供用户查看特定窗口或控件上的组件元素的属性的窗格。

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++ 则因其功能强大和灵活性较好而著名。但在 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被一同发布。但明显的是，每种语言的工作环境还是有差别，结果就导致开发的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

2002 年 2 月发布的 Visual Studio .NET 和 Windows Windows .NET Framework 是软件和开发工具历史上的一个重要里程碑。在热情支持发布的同时，软件开发人员对 Visual Studio .NET 赞不绝口，因为它使他们能够减少复杂项目开发中的周转时间，因而大大节约了时间和金钱。从《PC 杂志》到《软件开发杂志》，五种著名的出版物在 2002 年度将它们的最高奖项授予 Visual Studio .NET 和 Windows Windows .NET Framework，进一步肯定了这两种产品的领导地位及其在解决如今的编程难题方面做出的创新。

Visual Studio 的下一个版本被称为 Visual Studio .NET 2003。该版本侧重于修复错误和各种问题，例如由一些新技术，如 .NET Framework 引入的问题。对此，Framework 本身已经升级到了 1.1 版，也一并对 IDE 做了很小的改动。

在这一点上，Visual Studio 的最终用户和 Microsoft 旗下的各种语言并没有真正地对应程序架构产生什么帮助，而只是提供一些有关错误的反馈。并且 Microsoft 仅针对使用 .NET 第一个版本的程序员所发现的问题进行了解决。

然而，Microsoft 同时还宣布 .NET 的下一个版本，即所谓的 Whidbey，将会是一个“user's .NET”，它允许所有使用 .NET 的开发人员提交所需的特性需求和跟踪他们在开发周期中遇到的问题。微软将与众多合作伙伴密切合作，致力于为广大用户提供更有针对性的、更有益的 .NET 和 Visual Studio 工具。

其结果是，.NET Framework 2.0 和 Visual Studio 2005 这一整套开发工具、编辑器、编程语言和基础架构类，其功能已远远超过了 Microsoft 所提供的任何产品。Visual Studio 2005 development environment (如图 0-1 所示)的出现表明 Microsoft IDE 向前迈进了一大步，并朝着是一套全面的工具迈进。图 0-1 展示了不同部件是如何以各种方式组装在一起以向您提

供一个有效率的工具。

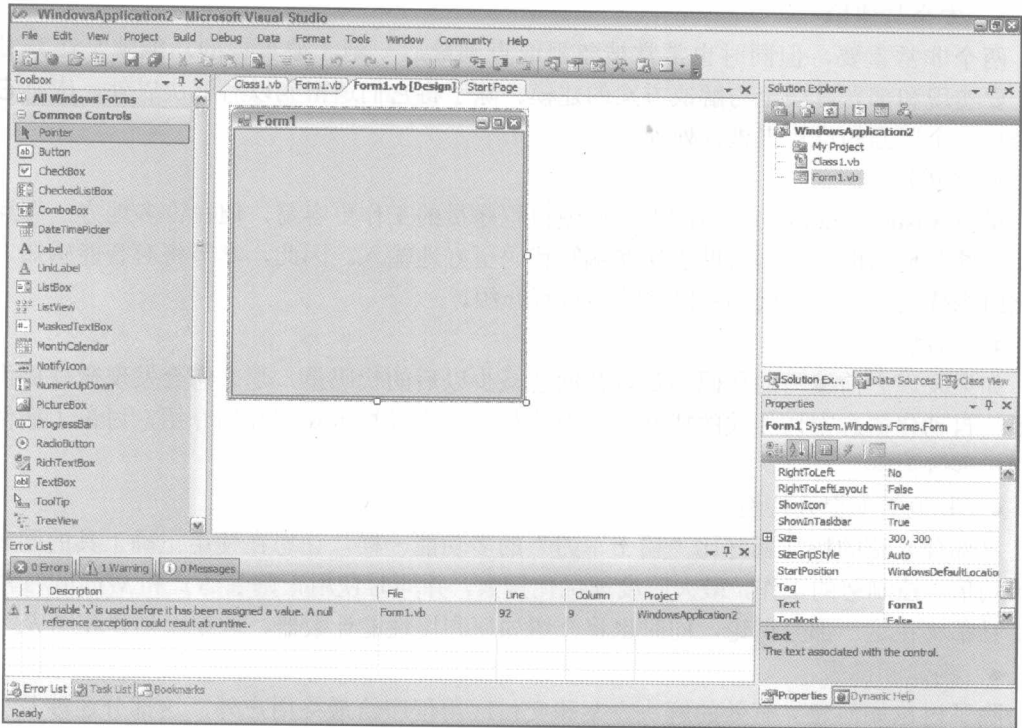


图 0-1 开发界面

Visual Studio 2005 有两个主要版本: Visual Studio 2005 Professional 和 Visual Studio 2005 Team System (准确地说, 针对不同的人员, 有三种不同的 Team System, 但它们的核心理念 Visual Studio 功能都是一样的)。本书主要介绍 Visual Studio 2005 的 Professional 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, 请参照第 56 章的有关内容。

本书结构

本书的第 I 部分主要是让您熟悉 Visual Studio 2005 的核心问题。从集成开发环境的结构和布局到各种选项和设置, 前 6 章包含了您所需的所有内容。

Visual Studio 2005 的功能可细分为以下 9 类:

- 项目与方案设计

本章将向您介绍如何控制您的项目, 如何组织它们以符合您自己的风格, 以及如何编辑应用配置和 XML 资源文件。

- 文档和研究

微软至今仍保持着文档库的不断更新, 并且 MSDN 上的资源随时可用。此外, 微软还致力于帮助您控制您自己的文件, 本节将帮助您一步步完成整个 XML 文档的自动化, 以

及使用纲要模式来检查您的代码。

- 安全与建模

两个非常重要，但同时也常常被忽视或直到最后一刻才被重视的功能就是在开发之前，要实现和部署好安全与解决方案的建模。除了将它们遗留到以后解决以外，应将它们安排在一个适当的位置来进行处理。

- 编码

虽然 Visual Studio 的许多图形组件使得程序员的工作更容易，但在许多地方，您经常需要一些其他帮助，比如有助于您在编写代码更好地输入。因此，本节将对智能感知、区域化组织代码以及标记代码片段等特性进行介绍。

- 自动化

如果发现之前编写的代码不足以帮助您减少以后的代码量，那么您会发现微软提供了许多与自动化概念密切相关的特性。这部分从代码生成技术和宏开始介绍，讨论了插件和功能扩展等内容。

- 其他效率提升特性

这部分介绍的特性就好比“瑞士军刀”的多功能一样，让您在使用正确工具的同时能事半功倍。控制您的工作，最大限度地优化搜索，并同时使用正则表达式和 Windows 注册表工具都将在这一部分讨论，同时也将介绍如何利用性能计数器，数据库创建和图表。

- 构建和部署

除了讨论如何建立有效的解决方案和交付应用程序给最终用户外，这部分还涉及如何将 Visual Basic 6 的项目升级为采用 Visual Basic 2005 格式的应用。

- 调试和测试

无论开发商多么优秀，在将应用交付给用户之前都必须对解决方案进行测试，本部分将教会您使用用来调试代码和数据库的有力工具。

- Visual Studio 2005 的扩展

有许多针对 Visual Studio 2005 的第三方插件和扩展都可用，但有 3 个为核心 IDE 定做的，由微软自己开发的宝贵扩展。最后一部分，这本书将针对这些扩展进行介绍。

通过以上这些介绍，你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行进一步的学习。

使用本书的要求

在使用本书时，最好安装 Microsoft Visual Studio 2005 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。要运行本书的所有示例，还可以安装以下软件，这些软件在 Microsoft 网站上均有提供：

- Microsoft Visual Studio 2005 Team System

一个更强大的 Visual Studio 版本，Team System 引入了针对发展过程中其他阶段的工具，如测试和设计。有关 Team System 的内容将在第 56 章介绍。

- SQL Server 2005

Visual Studio 2005 的安装中包含一个 SQL Server 2005 Express 的安装，从而使你能够构建一个使用数据库文件的应用程序。但若想获得更全面的企业解决方案，可以使用 SQL Server 2005。数据库连接将在第 38 章讨论。

- 针对 Office 2005 的 Visual Studio 工具

简称为 vsto2005，该插件将新的项目模板整合到 Visual Studio 2005 中，以使用户为 Microsoft Word、Excel 和 Outlook 创建托管的解决方案。可参考第 55 章以获取更多信息。

- 针对 Visual Studio 2005 的 InfoPath 2003 Toolkit

另外，除了核心的 Visual Studio 2005 环境中，这个工具包整合了带 InfoPath 表单的托管代码解决方案，且无须使用 JavaScript 以支持 InfoPath 背后的代码并用您所用的语言支持它。第 54 章有关于这方面的介绍。

- Code Snippet Editor

这是一个针对 Visual Basic 中的代码片段创建开发的第三方工具。第 19 章有更详细的介绍。

- 其他第三方工具

第 33 章的内容全是介绍如何使用附加的第三方插件和实用程序以最大化您使用 Visual Studio 2005 的成效。有关这方面的详细信息请参看该章。

源代码

对于本书中的所有示例来说，您可以手动输入所有的代码，也可以使用本书的源代码文件。本书中的所有源代码都放在 www.wrox.com 网站中。登录 Wrox.com 站点 www.wrox.com 时，只需使用 Search 工具框或使用书名列表就可以找到本书。接着单击本书信息页面上的 Download Code 链接，就可以获得所有的源代码。

提示：

因为很多书的标题相类似，所以最简单的方法是通过 ISBN 进行搜索。本书的英文 ISBN 是 0-7645-9846-5。

从该站点上下载的文件是已经使用了 WinZip 进行压缩的文件。在把附件保存到硬盘的一个文件夹中时，需要使用解压缩工具对该文件进行解压缩。或者，可以访问 Wrox 的下载页面：www.wrox.com/dynamic/books/download.aspx，查看本书或其他所有由 Wrox 出版的书籍中的代码。

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错误，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

它发送到该书的勘误表页面上，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。

p2p. wrox.com

P2P 论坛(p2p.wrox.com)是为作者和同行之间的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“由程序员为程序员的支持”(programmer to programmer support)，还包括一对一的电子邮件支持系统。您可以相信，您的问题不仅仅是由支持专家来解答，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，您可以从许多不同的列表中获得帮助，不仅对阅读本书有所帮助，还可以在开发应用程序时获得帮助。

要加入该论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com 并单击 Register 链接。
- (2) 阅读使用条款并单击 Agree 按钮。
- (3) 填写表单中要求填写的信息，以及其他可选的信息，然后单击 Submit 按钮，即可加入论坛。

您将接收到一封确认电子邮件，其中列出如何验证您的账户和加入论坛的完整步骤。

提示：

不加入 P2P 论坛也可浏览其中的信息，但要发布新消息，则必须注册。

在加入该论坛后，您就可发布新消息并回复其他人的文章了。可在任意时刻通过 Web 阅读其中的信息。如果需要，也可通过电子邮件订阅论坛中特定主题的新信息，为此，请在论坛列表中选中特定论坛名称对应的 Subscribe to this Forum 图标。

要更详细地了解如何使用 Wrox P2P 论坛，请阅读 P2P FAQ，其中列出了论坛软件如何工作等众多常见的问题。单击 P2P 论坛中任意页面的 FAQ 链接即可阅读相关内容。

目 录

第 I 部分 集成开发环境

第 1 章 IDE 快速入门	3
1.1 配置 VS 2005	3
1.2 IDE 结构	5
1.2.1 熟悉 IDE 的结构	6
1.2.2 基本布局	7
1.3 其他窗口	12
1.4 小结	13
第 2 章 Options 窗口	15
2.1 Options 窗口	15
2.2 环境选项	16
2.2.1 文档设置	17
2.2.2 字体和颜色	17
2.2.3 快捷键	19
2.2.4 任务列表	20
2.3 项目和解决方案	21
2.3.1 Build and Run 页面	22
2.3.2 Option Strict	23
2.4 Text Editor 页面	23
2.5 Debugging 页面	24
2.6 小结	27
第 3 章 工具箱	29
3.1 Toolbox 窗口	29
3.2 安排组件	31
3.3 添加组件	33
3.4 常用组件	34
3.5 小结	37
第 4 章 Solution Explorer 窗口	39
4.1 Solution Explorer 窗口的结构	39
4.2 显示隐藏元素	40
4.3 临时解决方案	41

4.4 Web 解决方案	42
4.5 项目和解决方案的常规操作	43
4.5.1 添加 Windows 引用	44
4.5.2 添加 Web 引用	45
4.5.3 设置解决方案属性	46
4.6 小结	47
第 5 章 定制 IDE	49
5.1 定制工具窗口	49
5.1.1 工具窗口概述	49
5.1.2 移动工具窗口	51
5.2 导入导出 IDE 设置	54
5.3 分割工作区	56
5.4 小结	57
第 6 章 窗体设计	59
6.1 窗体	59
6.2 窗体设计偏好	62
6.3 为窗体添加控件	63
6.3.1 控件引导线	64
6.3.2 垂直对齐文本控件	65
6.3.3 自动格式化多个控件	66
6.3.4 锁定控件设计	68
6.3.5 设置控件属性	68
6.3.6 基于服务的控件	70
6.3.7 智能标签技术	70
6.3.8 其他命令	71
6.4 容器控件	72
6.4.1 Panel 控件和 SplitContainer 控件	72
6.4.2 FlowLayoutPanel 控件	73
6.4.3 TableLayoutPanel 控件	73
6.5 小结	74

第 II 部分 项目和解决方案设计

第 7 章 项目和解决方案	77
7.1 解决方案的结构	77
7.2 解决方案文件的格式	79
7.3 解决方案的属性	79
7.4 配置属性	81
7.5 项目类型	82
7.6 项目文件的格式	82
7.7 项目的属性	82
7.7.1 Application 标签	83
7.7.2 Compile 标签	86
7.7.3 Debug 标签	87
7.8 自定义设置提供程序	93
7.9 小结	95
第 8 章 源控制	97
8.1 选择源控制仓库	97
8.1.1 环境设置	98
8.1.2 Plug-In Settings 面板	99
8.2 使用源控制	99
8.2.1 创建仓库	99
8.2.2 添加解决方案	100
8.2.3 Solution Explorer 窗口	101
8.2.4 登入登出	101
8.2.5 挂起更改	102
8.2.6 合并更改	103
8.2.7 历史记录	104
8.2.8 固定	104
8.3 Team Foundation 源控制	104
8.3.1 Source Control Explorer	105
8.3.2 挂起更改	105
8.3.3 Shelving	107
8.4 小结	108
第 9 章 应用程序配置文件	109
9.1 配置文件	109
9.1.1 machine.config	109
9.1.2 web.config	109
9.1.3 app.config	110
9.1.4 security.config	110
9.2 配置架构	110
9.2.1 配置属性	110
9.2.2 startup 区域	111
9.2.3 runtime 区域	112
9.2.4 system.runtime.remoting 区域	112
9.2.5 system.net 区域	113
9.2.6 cryptographySettings 区域	113
9.2.7 configurationSections 区域	113
9.2.8 system.diagnostics 区域	114
9.2.9 system.web 区域	115
9.2.10 webservice 区域	115
9.2.11 compiler 区域	115
9.3 应用程序设置	116
9.3.1 appSettings 区域	116
9.3.2 动态属性	116
9.3.3 自定义配置区域	117
9.4 使用 SCDL 实现自动化	119
9.5 小结	121
第 10 章 XML 资源文件	123
10.1 为应用程序提供资源	123
10.2 什么是资源?	125
10.2.1 文本文件	125
10.2.2 XML 资源文件	126
10.2.3 添加资源	127
10.2.4 嵌入文件作为资源	128
10.2.5 访问资源	128
10.2.6 资源命名	128
10.3 卫星资源	128
10.3.1 文化	129
10.3.2 创建文化资源	129
10.3.3 加载文化资源文件	130
10.3.4 卫星文化资源	130
10.4 访问细节	131
10.4.1 My 命名空间	131
10.4.2 加载位图和图标	131

10.4.3	ComponentResourceManager 对象	131
10.5	编码资源文件	132
10.6	自定义资源	134
10.7	小结	139
第III部分 归档和搜索		
第 11 章	帮助	143
11.1	访问帮助	143
11.1.1	文档浏览器	145
11.1.2	动态帮助	146
11.2	搜索窗口(Search)	148
11.2.1	排序结果	149
11.2.2	过滤结果	149
11.3	保存常用主题	150
11.4	定制帮助功能	151
11.5	Ask a Question 命令	151
11.6	小结	152
第 12 章	XML 注释	153
12.1	XML 注释	153
12.2	添加 XML 注释	154
12.3	XML 注释标签	154
12.3.1	<c> 标签	155
12.3.2	<code> 标签	155
12.3.3	<example>标签	156
12.3.4	<exception>标签	157
12.3.5	<include> 标签	158
12.3.6	<list>标签	160
12.3.7	<para>标签	161
12.3.8	<param>标签	161
12.3.9	<paramref>标签	162
12.3.10	<permission>标签	163
12.3.11	<remarks>标签	163
12.3.12	<returns>标签	163
12.3.13	<see>标签	164
12.3.14	<seealso>标签	164
12.3.15	<summary>标签	166
12.3.16	<typeparam>标签	166
12.3.17	<value>标签	166
12.4	使用 XML 注释	166
12.5	小结	168
第 13 章	控件与文档大纲	169
13.1	文档大纲	169
13.2	控件大纲	171
13.3	控件大纲模式下的其他 命令	172
13.4	小结	173
第IV部分 安全和建模		
第 14 章	代码生成技术	177
14.1	类设计器	177
14.1.1	设计界面	178
14.1.2	工具箱	178
14.1.3	类的细节	180
14.1.4	属性窗口	181
14.1.5	布局	182
14.1.6	导出类图	182
14.2	其他的代码生成技巧	182
14.2.1	代码片断	183
14.2.2	重组	183
14.2.3	项目和元素模板	184
14.2.4	强类型数据集	184
14.2.5	窗体	185
14.2.6	My 命名空间	186
14.3	类设计器的使用	187
14.3.1	类图架构	188
14.3.2	IntelliSense 对代码生成的 支持	189
14.4	对象测试工具	189
14.4.1	调用静态方法	189
14.4.2	实例化实体	190
14.4.3	访问字段和属性	191
14.4.4	调用实例方法	191
14.5	小结	192

第 15 章	安全相关的概念	193
15.1	应用程序安全	193
15.1.1	基于代码的安全	193
15.1.2	基于角色的安全	195
15.2	小结	197
第 16 章	密码技术	199
16.1	基本原则	199
16.2	技术	199
16.2.1	哈希	199
16.2.2	对称(秘密)密钥.....	200
16.2.3	非对称(公/私)密钥	200
16.2.4	签名	201
16.2.5	目标的总结	201
16.3	应用密码技术	202
16.3.1	创建非对称密钥对	202
16.3.2	创建对称密钥	204
16.3.3	加密和给密钥签名	205
16.3.4	验证密钥和签名	207
16.3.5	解密对称密钥	209
16.3.6	发送消息	210
16.3.7	接收消息	212
16.4	其他	214
16.4.1	SecureString 类.....	214
16.4.2	密钥容器	216
16.5	小结	217
第 17 章	混淆	219
17.1	MSIL 反汇编器	219
17.2	反编译器	220
17.3	混淆您的代码	221
17.3.1	Dotfuscator	222
17.3.2	警告的话	224
17.4	属性	227
17.4.1	ObfuscationAssembly 属性	227
17.4.2	Obfuscation 属性	227
17.5	小结	229

第 V 部分 编 码

第 18 章	智能感知	233
18.1	智能感知的解释.....	233
18.1.1	通用的智能感知	234
18.1.2	单词和短语的自动完成	235
18.1.3	参数信息	237
18.1.4	快速信息	238
18.2	智能感知选项.....	238
18.2.1	通用选项	238
18.2.2	C#和 J#的特定选项	240
18.3	扩展智能感知.....	240
18.3.1	代码片段	240
18.3.2	XML 注释	241
18.3.3	添加您自己的智能感知	241
18.4	小结	241
第 19 章	代码片段	243
19.1	代码片段概述.....	243
19.1.1	基本代码片段	243
19.1.2	“真正的”代码片段	244
19.1.3	Visual Basic 代码片段	244
19.1.4	C#和 J#代码片段	247
19.1.5	创建代码片段	248
19.1.6	代码片段管理器	254
19.1.7	VB 代码片段编辑器	255
19.2	小结	260
第 20 章	代码区域和书签	261
20.1	代码区域.....	261
20.1.1	创建代码区域	262
20.1.2	使用代码区域	263
20.1.3	自动大纲功能	263
20.2	可视化指示器.....	265
20.2.1	彩色编码	265
20.2.2	边界图标	265
20.3	书签和 Bookmarks 窗口	266
20.4	小结	268

第 21 章 代码重组	269	22.5.2 Comparison 代理	292
21.1 访问重组支持	269	22.5.3 Converter 代理	293
21.1.1 C#——Visual Studio		22.5.4 Predicate 代理	294
2005	270	22.5.5 EventHandler 代理	294
21.1.2 VB.NET——Refactor!	270	22.6 属性访问限制	296
21.2 重组操作	271	22.7 自定义事件	297
21.2.1 抽取方法		22.8 小结	298
(Extract Method)	271	第 23 章 语言相关特性	299
21.2.2 封装字段		23.1 C#	299
(Encapsulate Field)	273	23.1.1 匿名方法	299
21.2.3 抽取接口		23.1.2 迭代器	300
(Extract Interface)	274	23.1.3 静态类	301
21.2.4 重新排列参数(Reorder		23.2 命名冲突	302
Parameters)	276	23.2.1 命名空间别名限定符	303
21.2.5 删除参数(Remove		23.2.2 Global 标识符	304
Parameters)	276	23.2.3 Extern 别名	304
21.2.6 重命名(Rename)	277	23.3 Pragma 语句	305
21.2.7 把变量转换为参数(Promote		23.4 VB.NET	306
to Parameter)	278	23.4.1 Continue 语句	306
21.2.8 产生方法框架(Generate		23.4.2 IsNot 语句	307
Method Stub)	278	23.4.3 Global 语句	307
21.2.9 用代码段进行封装		23.4.4 TryCast 语句	307
(Surround with Snippet)	279	23.5 小结	308
21.3 小结	279	第 24 章 My 命名空间	309
第 22 章 泛型、Nullable 类型和		24.1 什么是 My 命名空间?	309
Partial 类型	281	24.2 My 的组成	310
22.1 泛型	281	24.3 My 的使用	311
22.1.1 使用	282	24.3.1 C#	311
22.1.2 创建	282	24.3.2 My 的环境	312
22.1.3 约束	284	24.3.3 默认实例	314
22.2 Nullable 类型	285	24.4 My.Application	315
22.3 Partial 类型	287	24.5 My.Computer	316
22.4 运算符重载	288	24.4.1 My.Computer.Audio	316
22.4.1 运算符	288	24.4.2 My.Computer.Clipboard	316
22.4.2 类型转换	289	24.4.3 My.Computer.Clock	317
22.4.3 为什么静态方法不好	290	24.4.4 My.Computer.File	
22.5 预定义的代理	292	System	317
22.5.1 Action 代理	292	24.4.5 My.Computer.Info	317

24.4.6	My.Computer.Keyboard 和 My.Computer.Mouse	318
24.4.7	My.Computer.Network	318
24.4.8	My.Computer.Ports	319
24.4.9	My.Computer.Registry	319
24.5	My.Forms 和 My.Web Services	319
24.6	Web 应用	320
24.7	My.Resources	320
24.8	其他的 My 类	322
24.9	小结	322

第VI部分 自动化

第 25 章	代码模板	325
25.1	创建模板	325
25.2	元素模板	325
25.2.1	项目模板	328
25.2.2	模板结构	329
25.3	扩展模板	331
25.3.1	模板项目设置	331
25.3.2	IWizard 接口	333
25.3.3	Starter 模板	335
25.4	小结	338
第 26 章	宏	339
26.1	宏管理器(Macro Explorer)	339
26.2	创建宏	341
26.2.1	录制临时宏	341
26.2.2	录制的缺陷	342
26.3	VS 宏编辑器	343
26.4	DTE 对象	344
26.5	构建和部署	347
26.6	小结	348
第 27 章	连接字符串	349
27.1	数据源配置向导	349
27.2	SQL Server 格式	353
27.3	代码构建	354
27.4	加密连接字符串	355
27.5	小结	357

第 28 章	程序集签名	359
28.1	强名程序集	359
28.2	全局程序集缓存	359
28.3	对程序集进行签名	360
28.4	小结	361
第 29 章	预先纠错设置	363
29.1	智能编译自动纠错功能	363
29.2	自定义 Visual Basic 中的警告 信息报告	366
29.2.1	默认设置下不会显示的 警告信息	368
29.2.2	可自定义的其他警告 信息	369
29.3	自定义 C#编译环境下的 警告信息	372
29.4	小结	374
第 30 章	强类型数据集	375
30.1	DataSet 类概述	375
30.2	添加数据源	376
30.3	数据集设计器	379
30.4	数据源的使用	382
30.4.1	Web 服务数据源	383
30.4.2	浏览数据	384
30.5	小结	386
第 31 章	数据绑定和对象数据源	387
31.1	数据绑定	387
31.1.1	BindingSource 类	388
31.1.2	Binding Navigator 类	390
31.1.3	选择数据源	392
31.1.4	BindingSource 链	394
31.1.5	保存更改	399
31.1.6	插入新记录	402
31.1.7	验证	403
31.1.8	DataGridView 控件	411
31.2	对象数据源	412
31.3	应用程序设置	417
31.4	小结	418

第 32 章 插件	419	34.6 小结	446
32.1 插件管理器 (Add-in Manager)	419	第七部分 其他的省时办法	
32.2 插件的类型	420	第 35 章 工作区控件	449
32.3 使用向导创建一个简单的 插件	420	35.1 VS2005 窗口	449
32.4 常用的类、对象和方法	426	35.1.1 启动页面(Start Page)	449
32.4.1 IDTExtensibility2 接口	426	35.1.2 代码/设计器(Code/ Designer)	450
32.4.2 IDTCommandTarget 接口	427	35.1.3 解决方案浏览器(Solution Explorer)	450
32.4.3 AddNamedCommand2 方法	428	35.1.4 属性(Properties)	451
32.4.4 CreateToolWindow2 窗口	430	35.1.5 工具箱(Toolbox)	451
32.4.5 调试	430	35.1.6 服务器浏览器(Server Explorer)	451
32.4.6 注册和部署	430	35.1.7 错误列表(Error List)	452
32.5 小结	430	35.1.8 对象浏览器(Object Browser)	453
第 33 章 第三方扩展	431	35.1.9 任务列表(Task List)	453
33.1 改善开发环境	431	35.1.10 类视图(Class View)	453
33.1.1 CoolCommands for VS2005	431	35.1.11 代码定义(Code Definition)	454
33.1.2 MZ-Tools	432	35.1.12 输出(Output)	454
33.2 代码辅助工具	434	35.1.13 搜索结果(Find Results)	455
33.2.1 Imports Sorter	434	35.1.14 调用管理器(Call Browser)	455
33.2.2 CodeKeep	435	35.1.15 命令(Command)	456
33.3 文档	437	35.1.16 文档大纲(Document Outline)	456
33.4 测试和调试	438	35.1.17 对象测试(Object Test Bench)	456
33.4.1 Regex Visualizer	438	35.1.18 性能管理器(Performance Explorer)	457
33.4.2 TestDriven.Net	439	35.1.19 属性管理器(Property Manager)	457
33.5 小结	439	35.1.20 资源视图(Resource View)	458
第 34 章 入门套件	441	35.1.21 历史(History)	458
34.1 The Card Game 入门套件	441	35.1.22 源控制浏览器(Source Control Explorer)	459
34.2 The Screensaver 入门套件	442		
34.3 The Movie Collection 入门套件	443		
34.4 The Personal Web Site 入门套件	444		
34.5 创建入门套件	445		