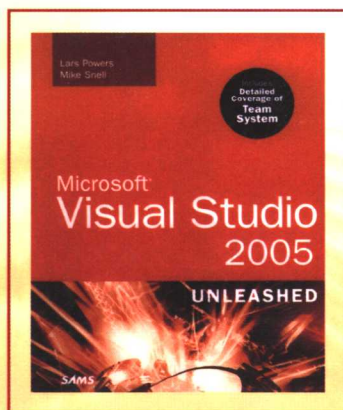


Microsoft Visual Studio 2005 Unleashed

Visual Studio 2005 技术大全

[美] Lars Powers Mike Snell 著
刘彦博 肖鹏 贾菡 译

- 微软工程师力作
- 全面、实用、详尽
- 使 .NET 程序员事半功倍的利器



人民邮电出版社
POSTS & TELECOM PRESS

TP393.09/152

2008

TURING

图灵程序设计丛书

.NET系列

Visual Studio 2005技术大全

Microsoft Visual Studio 2005 Unleashed

[美] Lars Powers Mike Snell 著

刘彦博 肖鹏 贾菡 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

Visual Studio 2005 技术大全 / (美) 鲍尔 (Powers, L.), (美) 斯内尔 (Snell, M.) 著; 刘彦博, 肖鹏, 贾菡译. —北京: 人民邮电出版社, 2008.1
(图灵程序设计丛书)
ISBN 978-7-115-16941-9

I. V… II. ①鲍…②斯…③刘…④肖…⑤贾… III. 计
算机网络—程序设计 IV.TP393

中国版本图书馆 CIP 数据核字 (2007) 第 152718 号

内 容 提 要

本书全面涵盖了 Visual Studio 2005 的所有常用特性。书中从 IDE 基础开始, 介绍了 Visual Studio 2005 中的各种工具和功能, 包括编辑器、各种设计器、重构、调试、共享代码等; 涉及了 Visual Studio 所支持的各种应用, 如创建用户界面、窗体, 使用数据库和 Web 服务等; 引入了 Visual Studio Team System 工具以及 Visual Studio 2005 对于团队开发给予的支持。

本书内容详尽实用, 适合初级到中级的 Visual Studio 2005 开发人员和系统管理人员阅读。

图灵程序设计丛书

Visual Studio 2005 技术大全

-
- ◆ 著 [美] Lars Powers Mike Snell
译 刘彦博 肖 鹏 贾 菡
责任编辑 傅志红 张继发
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 40.25
字数: 1 253 千字 2008 年 1 月第 1 版
印数: 1—5 000 册 2008 年 1 月北京第 1 次印刷
著作权合同登记号 图字: 01-2007-1959 号

ISBN 978-7-115-16941-9/TP

定价: 89.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

译者序

现在, Visual Studio 2005每天都伴我享受着工作和业余生活。当我在头脑中考虑好或在白纸上完成了一个模块的设计后,它总是能让我用最短的时间将其变成一行行代码。编写程序,既是我的工作,也是我的爱好。无论你是和我一样,还是只把它单纯地当作工作用具,相信Visual Studio 2005都能让兴奋与快乐时时萦绕着你。

我是从Visual Basic 4.0开始使用微软Visual系列开发工具的。后来在Visual Studio 6.0时代,我同时使用Visual Basic和Visual C++。第一次接触到.NET时,我为这项伟大的技术所折服,以最快的速度将所有工作转向.NET,开发工具也换成了Visual Studio .NET 2002/2003。任何对某种工具相当熟悉的人,都会遇到这样的情况——当拿到一个工具的新版本时,如果它的向下兼容做得非常棒的话,你就很容易忽略它的新特性,而继续沿用老的、有限的功能,尤其是当工作压力比较大的时候更是如此。新版本的新特性往往很多,常常让人无暇顾及。只有在使用中偶然发现了一个新特性,或者用到了不再支持的老功能,以至于发生错误时,我们才真正开始了解这个工具的新特性。如果你熟悉Visual Studio之前的版本,在初次使用Visual Studio 2005时,尤其可能遇到上述情况。

Visual Studio 2005中各种丰富的特性,会让初次进入程序员行列的你目不暇接、眼花缭乱,以至于学习这款工具所需投入的精力,几乎和学习一门新的语言一样多,甚至在某种情况下还要更多一些。整个2006年我都在和初学者打交道,我至少向200名毫无编程经验的朋友传授过基于.NET和C#的编程技术,其中绝大部分的时间花在了向他们介绍Visual Studio的功能、使用方法和技巧上。

我常常想,如果能有一本书,从基础到高级,将Visual Studio 2005的所有常用特性介绍一下,而脱开具体的语言环境,那该有多好。幸运的是,你手中捧着的就是这样的一本书。本书从IDE的基础开始,介绍了Visual Studio 2005中的各种工具和功能,继而介绍了Visual Studio所支持的各种应用程序,展示了如何对开发环境进行自定义。Visual Studio 2005相对于之前版本最大的改进在于,它引入了Visual Studio Team System工具以及团队开发的概念。本书还为此专门提供了一个部分,使用大量的篇幅介绍了团队协作的基本概念和Visual Studio 2005对于团队开发给予的支持。无论你是熟练掌握Visual Studio之前版本、久经沙场的老黄忠,还是刚刚接触Visual Studio、初出茅庐的小诸葛,本书都能带领你熟练掌握Visual Studio 2005所有常用的功能,为你的开发工作打下坚实的基础。

我非常荣幸能够获得翻译这本书的机会,这要感谢博客园的idior网友,感谢人民邮电出版社图灵公司对我的信任。还要感谢我的合译者肖鹏,他在整个翻译过程中都保持着谦虚、认真和负责的态度,他的技术和翻译功底也非常令人钦佩。感谢我的妻子贾茵,感激上苍赐给我一位志同道合的爱人。感谢在本书翻译过程中帮助和支持我的所有朋友。

本书第1章~第3章、第13章~第16章由贾茵翻译,第4章、第6章~第12章由肖鹏翻译,其余部分由我翻译。由于水平有限,书中难免有翻译错误和不当之处。在这里,我谨代表本书的译者恳请你的批评与指正!

刘彦博

2007年6月于北京

译者简介



刘彦博 高级.NET开发工程师，微软最有价值专家（MVP）。他有超过7年的Visual Studio应用开发经验和超过4年的C#/.NET开发经验，熟练掌握Visual Studio的各种功能，精通C#语言和.NET运行时环境。业余时间，他喜欢翻译技术文档和撰写技术博客，热衷于向初学者介绍C#语言和.NET技术的相关知识。他的博客网址是<http://AndersLiu.cnblogs.com>，可以通过lyb.net@gmail.com与他联系。



肖鹏 电子工程学士、信息安全专业硕士。他自命为“代码打字员”、“网络群居动物”，偏爱计算机，善良热情，以助人为快乐之本，并希望能与志同道合的朋友多多交流。他的博客网址是<http://designpatterns.cnblogs.com>，可以通过eagle.xiao@gmail.com与他联系。



贾茜 微软产品解决方案专家，微软认证系统工程师（MCSE）。她熟悉微软主流产品及技术，有超过6年的企业信息规划管理、知识工作者管理及CRM管理经验，曾多次为国内微软产品技术相关的大型杂志翻译技术文章，如发表于《msdn开发精选》双月刊的《商务智能最佳解决方案——微软数据仓库》、《武装到牙齿——用Visual Studio 2005 Team System组织所有的装备》等。她的博客网址是<http://kitty74jh.cnblogs.com>，可以通过kitty74jh@gmail.com与她联系。

前言

Visual Studio 2005和Visual Studio Team Systems的发布标志着.NET开发体验的巨大改变。编程语言得到了很大改进，框架也添加了大量内容，但最重要的是，开发工具得到了巨大增强。现在的Visual Studio，包含了代码段、自定义项目模板、重构、数据绑定向导、智能标签、建模工具、自动测试工具、项目和任务管理，还有更多未列出的内容。这些工具旨在提高工作效率和成功率。本书将深入介绍这些工具及其功能。

读者对象

本书适用于一直使用Visual Studio进行工作的开发人员。书中详细介绍了最新版本IDE中的大量特性，包含所有下列关键主题：

- 理解解决方案、项目、编辑器和设计器的基本知识；
- 编写宏、外接程序和向导；
- 使用IDE进行调试；
- 重构代码；
- 和团队成员以及范围更广的社区共享代码；
- 编写ASP.NET应用程序；
- 编写和使用Web服务；
- 使用Windows表单进行编码；
- 使用数据和数据库；
- 使用团队协作和团队系统；
- 使用Team Architect进行建模；
- 使用Team Test进行自动测试；
- 管理源代码的变更和生成。

本书不介绍编程语言，但会涉及C#和Visual Basic中一些新的语言特性（如泛型）。本书尽量使用简单的示例，以便使C#和Visual Basic的开发人员都能够阅读。本书着重介绍构成Visual Studio 2005的各种工具。

本书结构

本书既可以逐页阅读，也可以选择最需要的章节阅读。有时在讲解某一章时可能会引用另一章的内容，但总体而言，各章是相互独立的。读者可以根据时间安排（或兴趣）进行跳跃式阅读。本书包含4个部分，下面分别进行介绍。

第一部分：Visual Studio 2005 / .NET 简介

这一部分大概要介绍了Visual Studio 2005。只熟悉.NET之前版本的读者应该阅读这一部分。另外，

它还包括了C#和VB2005版本中新的语言特性。

第二部分：深入 Visual Studio 2005 开发环境

这一部分涵盖了Visual Studio相关的核心开发体验。它介绍了一些基本知识，帮助开发人员了解其主要工具的一些丰富特性。它也介绍了定义各种工具的菜单和窗口。其内容涵盖项目、解决方案、编辑器、设计器和其他提高工作效率的工具以及辅助工具。这一部分还深入挖掘了重构代码、使用调试工具集、在Visual Studio中使用自动化模型以及如何准备共享代码等内容。

第三部分：Visual Studio 2005 应用

第三部分侧重于如何使用这些工具编写应用程序。每一章都深入介绍了如何使用IDE来帮助应用程序投入生产。这一部分涵盖如何使用ASP.NET、Web服务、Windows窗体来编写应用程序，以及如何使用数据和数据库。

第四部分：Visual Studio 2005 Team System

第四部分介绍了如何使用Visual Studio Team System来增强项目的可见性并加强团队协作。这些工具都是Visual Studio 2005新提出的。这一部分对团队系统项目、协作和工作项跟踪进行了深度探索。它还涵盖了Team System提供的相关特性和工具，如源代码控制、自动测试和测试用例管理、可视化建模以及Team Foundation Build等。

编排体例

本书使用下列排版约定：

- 代码行、命令、语句、变量和任何出现在屏幕上的文本，使用等宽字体。
- 语法描述中的占位符使用斜等宽字体。该占位符可能是它们所表示的实际文件名、参数或其他任何元素。
- 楷体表示正在定义的技术术语。
- 代码续行符会出现某一行代码之前，表示这一行是接着上一行的。有些代码太长了，很难在书中的一行里印刷出来。如果某一行代码之前出现了☛符号，表示这一行代码只是上一行代码的一部分。
- 书中还包括**注解**、**提示**和**警告**，用于帮助读者快速看到重要的或有用的信息。

致谢

感谢Sams出版社的伟大团队，感谢他们为本书的出版给予的大力协助。这包括我们的开发编辑Mark Renfrow、技术编辑J. Boyd Nolan、文字编辑Chuck Hutchinson、两位项目编辑Seth Kerney和Matt Purcell，以及组稿编辑Neil Rowe。

Mike Snell：感谢我的团队中的架构师们，谢谢他们对本书所包含的大量主题的建议。这个团队包括Jason Agostoni、Geoff Tewksbury、Matt Trevors和Stephen Johnston。还要感谢CEI中Microsoft Practice团队的另外一名关键成员Ryan Baker，谢谢他的鼓励和洞察力。最后，感谢我的好朋友和合著者Lars Powers。这又是一次精彩的合作经历，谢谢！

Lars Powers：感谢GISV技术团队中的各位成员，他们在很多时候都是我坚强的支持者和取之不尽、用之不竭的知识库。感谢我的合著者Mike，他先是说服我加入到这个项目中来，然后在撰写和编辑过程中，一直是我可靠的同事和伙伴。最后，感谢微软公司的各个团队，是他们为我们带来了Visual Studio和Visual Studio Team System。

目 录

第一部分 Visual Studio 2005 / .NET 简介

第 1 章 Visual Studio 2005 快速入门.....2

1.1 基本的增强特性.....2

1.1.1 设计、编写和发现代码.....2

1.1.2 编辑和调试代码.....9

1.1.3 在社区中共享（和使用）代码.....12

1.1.4 应对不同的客户体验.....12

1.1.5 连接数据源.....16

1.1.6 应用程序的自动测试.....18

1.2 产品家族.....18

1.2.1 速成版.....19

1.2.2 标准版.....19

1.2.3 Visual Studio 专业版.....20

1.2.4 Visual Studio Team System.....20

1.3 小结.....23

第 2 章 IDE 快速入门.....24

2.1 安装.....24

2.1.1 选择语言.....24

2.1.2 安装源代码控制.....25

2.1.3 配置开发环境.....26

2.2 起始页.....27

2.3 第 1 个项目.....28

2.4 菜单栏.....28

2.5 各种工具栏.....32

2.6 工具箱.....33

2.7 可视化设计器.....34

2.8 文本编辑器.....35

2.8.1 代码编辑器.....35

2.8.2 编辑器的自定义.....36

2.9 解决方案资源管理器.....38

2.10 属性窗口.....38

2.11 管理 IDE 中的各种窗口.....39

2.11.1 钉住.....39

2.11.2 停靠.....39

2.12 小结.....40

第 3 章 Visual Studio 2005 中的.NET Framework 和语言增强.....41

3.1 新增的共享.NET 语言特性.....41

3.1.1 泛型.....41

3.1.2 可空类型.....46

3.1.3 分部类型（类）.....47

3.1.4 具有混合访问级别的属性.....48

3.1.5 命名空间的二义性.....49

3.2 VB 语言增强.....49

3.2.1 Continue 语句.....50

3.2.2 无符号类型.....50

3.2.3 IsNot 运算符.....50

3.2.4 Using 块.....50

3.2.5 类似 VB 6 的窗体访问.....51

3.2.6 数组的显式零下界.....51

3.2.7 运算符重载.....51

3.2.8 自定义事件.....51

3.3 C#语言增强.....51

3.3.1 匿名方法.....52

3.3.2 静态类.....53

3.3.3 引用同一个程序集的两个版本.....54

3.3.4 友元程序集	54	5.5.5 理解性能报表	97
3.4 .NET Framework 2.0 增强	55	5.6 宏资源管理器	102
3.5 小结	56	5.6.1 Macros 根节点	102
第二部分 深入 Visual Studio 2005		5.6.2 宏项目	102
开发环境		5.6.3 宏模块	103
		5.6.4 宏	103
第 4 章 解决方案和项目	58	5.7 文档大纲	103
4.1 理解解决方案	58	5.8 小结	105
4.1.1 创建解决方案	59	第 6 章 编辑器和设计器简介	106
4.1.2 使用解决方案	63	6.1 基础知识	106
4.2 理解项目	67	6.1.1 文本编辑器	106
4.2.1 创建项目	67	6.1.2 Visual Studio 设计器	108
4.2.2 使用项目定义文件	69	6.2 使用代码编辑器编写代码	108
4.2.3 使用项目	75	6.2.1 打开编辑器	108
4.3 小结	78	6.2.2 在代码编辑器中编写代码	109
第 5 章 浏览器和资源管理器	80	6.2.3 剖析代码编辑器窗口	110
5.1 解决方案资源管理器	80	6.2.4 代码导航工具	112
5.1.1 提示文字和图标	81	6.2.5 搜索文档	113
5.1.2 管理解决方案	84	6.2.6 在文本编辑器中调试	118
5.1.3 管理项目	84	6.2.7 打印代码	119
5.2 类视图	84	6.2.8 使用代码定义窗口	120
5.2.1 工具栏	85	6.3 创建和编辑 XML 文档和架构	121
5.2.2 搜索栏	86	6.4 开发 Windows Forms 应用程序	123
5.2.3 对象窗格	86	6.4.1 自定义窗体的外观	123
5.2.4 成员窗格	87	6.4.2 向窗体中添加控件	124
5.3 服务器资源管理器	87	6.4.3 编写代码	125
5.3.1 数据连接	88	6.5 开发 Web 窗体	128
5.3.2 服务器组件	88	6.6 创建组件和控件	131
5.4 对象浏览器	91	6.6.1 创建新的组件或控件	131
5.4.1 改变范围	91	6.6.2 编写组件代码的更多说明	132
5.4.2 浏览对象	92	6.7 小结	133
5.5 性能资源管理器	93	第 7 章 使用 Visual Studio 2005 中的	
5.5.1 新建性能会话	93	工作效率提升工具	134
5.5.2 配置会话	94	7.1 文本编辑器的基本辅助功能	135
5.5.3 会话目标	97	7.1.1 变更跟踪	136
5.5.4 报表	97	7.1.2 代码问题指示器	136

7.1.3 活动超链接	137	8.5 重构参数	168
7.1.4 语法着色	137	8.5.1 移除参数	168
7.2 大纲和导航	138	8.5.2 将局部变量提升为参数	169
7.2.1 代码大纲	138	8.5.3 重新排列参数	170
7.2.2 HTML 导航	139	8.6 封装字段	171
7.3 智能标签和智能任务	140	8.6.1 执行封装字段操作	171
7.3.1 HTML 设计器	140	8.6.2 封装字段对话框	172
7.3.2 Windows Forms 设计器	141	8.7 小结	172
7.3.3 代码编辑器	141		
7.4 智能感知	142	第 9 章 在 Visual Studio 2005 中进行调试	173
7.4.1 完成单词	142	9.1 调试基础	173
7.4.2 快速信息	143	9.1.1 场景	173
7.4.3 列表成员	143	9.1.2 调试的多个阶段	174
7.4.4 参数信息	143	9.1.3 调试应用程序(自检查)	174
7.4.5 代码段和模板代码	144	9.1.4 小结	180
7.4.6 花括号匹配	150	9.2 Visual Studio 调试器	181
7.4.7 自定义智能感知	151	9.2.1 调试菜单和工具栏	181
7.5 任务列表	152	9.2.2 调试选项	184
7.5.1 注释任务	153	9.2.3 逐语句、逐过程和执行到光标处	185
7.5.2 快捷任务	154	9.2.4 在代码中指定断点	189
7.5.3 用户任务	154	9.2.5 使用跟踪点	194
7.6 小结	154	9.2.6 在调试器中查看数据	196
第 8 章 重构代码	155	9.2.7 使用编辑并继续特性	200
8.1 Visual Studio 重构基础	156	9.2.8 远程调试	200
8.1.1 调用重构工具	156	9.3 小结	201
8.1.2 预览更改	159		
8.2 重命名	159	第 10 章 Visual Studio 自动化对象模型	202
8.2.1 执行重命名操作	160	10.1 自动化对象模型概述	202
8.2.2 使用重命名对话框	160	10.1.1 对象模型版本	203
8.3 提取方法	161	10.1.2 自动化的种类	204
8.3.1 执行提取方法操作	162	10.1.3 DTE/DTE2 根对象	204
8.3.2 提取方法	162	10.2 Solution 和 Project 对象	205
8.3.3 生成方法存根	166	10.2.1 控制解决方案中的项目	207
8.4 提取接口	166	10.2.2 访问项目中的代码	208
8.4.1 执行提取接口操作	166	10.3 窗口	210
8.4.2 提取接口	167		

10.3.1	引用窗口	210
10.3.2	与窗口交互	211
10.3.3	文本窗口和窗格	213
10.3.4	工具窗口的类型	215
10.3.5	链接窗口	222
10.4	命令栏	224
10.5	文档	227
10.6	命令对象	237
10.6.1	执行命令	238
10.6.2	映射快捷键	239
10.7	调试器对象	239
10.8	自动化事件	240
10.9	小结	240
第 11 章	编写宏、外接程序和向导	241
11.1	编写宏	241
11.1.1	记录宏	242
11.1.2	使用宏资源管理器	243
11.1.3	使用宏 IDE	243
11.1.4	处理事件	250
11.1.5	调用宏	254
11.2	编写 Visual Studio 外接程序	256
11.2.1	管理外接程序	256
11.2.2	运行外接程序向导	257
11.2.3	外接程序的结构	264
11.2.4	示例外接程序——调色板	270
11.3	创建 Visual Studio 向导	291
11.3.1	向导的结构	291
11.3.2	创建 Add New Item 向导	294
11.4	小结	298
第 12 章	.NET 社区——使用 and 创建共享代码	299
12.1	Visual Studio 的社区功能	299
12.1.1	Visual Studio 起始页	299
12.1.2	社区菜单	303
12.2	发现和使用共享内容	312
12.2.1	共享内容的类型	312

12.2.2	查找正确的内容	312
12.2.3	安装和保存共享内容	313
12.3	回馈社区	314
12.3.1	创建共享项 (项目和项模板)	314
12.3.2	创建项目模板	314
12.3.3	创建项模板	319
12.3.4	将作品打包	320
12.3.5	发布作品	326
12.4	小结	326

第三部分 Visual Studio 2005 应用

第 13 章 创建 ASP.NET 用户界面

13.1	ASP.NET 网站基础	328
13.1.1	新建 Web 应用程序项目	328
13.1.2	控制项目属性和选项	335
13.1.3	创建网页	340
13.2	设计用户界面	343
13.2.1	确定页面布局和控件定位	343
13.2.2	创建通用的界面外观	345
13.2.3	创建用户可配置的 UI	353
13.3	使用 ASP.NET 控件	360
13.3.1	ASP.NET 控件增强	360
13.3.2	ASP.NET 中的新控件	360
13.4	小结	365

第 14 章 构建 Windows 窗体

14.1	窗体设计基础	366
14.1.1	考虑最终用户	366
14.1.2	理解 UI 标准的角色	367
14.1.3	规划用户界面	368
14.2	创建窗体	368
14.2.1	Windows 应用程序项目类型	369
14.2.2	窗体的属性和事件	370
14.3	添加控件和组件	371
14.3.1	控件的布局 and 定位	372
14.3.2	使用容器	374
14.3.3	控件的外观 and 行为	377

14.3.4 使用 ToolStrip 控件	378	16.2.1 .NET Web 服务	422
14.3.5 显示数据	384	16.2.2 ASP.NET Web 服务项目	422
14.4 建自定义控件	387	16.2.3 Web 服务文件	423
14.4.1 子类化现有控件	387	16.3 开发 Web 服务	424
14.4.2 设计用户控件	388	16.3.1 创建 Web 服务	424
14.4.3 创建自定义控件	390	16.3.2 访问和调用 Web 服务	426
14.5 小结	390	16.4 使用 Web 服务	433
第 15 章 使用数据库	391	16.4.1 定义 Web 引用	433
15.1 创建表和关系	391	16.4.2 查看 Web 引用	434
15.1.1 新建 SQL Server 数据库	391	16.4.3 调用 Web 服务	435
15.1.2 定义表	392	16.5 管理 Web 服务中的异常	436
15.1.3 使用数据库关系图设计器	394	16.5.1 创建 Web 服务异常	436
15.2 使用 SQL 语句	396	16.5.2 处理 Web 服务异常	437
15.2.1 编写查询	397	16.6 小结	437
15.2.2 创建视图	399		
15.2.3 开发存储过程	399	第四部分 Visual Studio 2005	
15.2.4 创建触发器	402	Team System	
15.2.5 创建用户定义函数	403	第 17 章 团队协作和 Visual Studio Team	
15.3 使用数据库项目	403	System	440
15.3.1 创建数据库项目	403	17.1 软件开发项目概览	440
15.3.2 自动生成脚本	404	17.1.1 MSF Agile	441
15.3.3 执行脚本	405	17.1.2 MSF for CMMI	442
15.4 在托管代码中创建数据库对象	405	17.2 Visual Studio Team System 简介	443
15.4.1 创建 SQL Server 项目	405	17.2.1 Visual Studio Team Architect	444
15.4.2 用 C# 创建存储过程	406	17.2.2 Visual Studio Team Developer	444
15.5 将控件绑定到数据	408	17.2.3 Visual Studio Team Test	446
15.5.1 数据绑定简介	408	17.2.4 Team Foundation Server	446
15.5.2 自动生成 Windows Forms 控件		17.3 小结	448
绑定	409		
15.5.3 手动绑定 Windows Forms		第 18 章 管理和使用团队项目	449
控件	413	18.1 Team Foundation Server 剖析	449
15.5.4 Web 控件上的数据绑定	416	18.1.1 应用程序层	449
15.6 小结	419	18.1.2 数据层	451
		18.1.3 安全	452
第 16 章 Web 服务和 Visual Studio	420	18.2 管理团队项目	454
16.1 Web 服务定义	420	18.2.1 新建团队项目	454
16.2 Web 服务项目的组件	422	18.2.2 向项目团队中添加用户	457

18.2.3 控制项目结构和迭代	461	20.1.3 理解工作项的通用性	491
18.3 参与项目团队	462	20.2 使用 Team Explorer 管理工作项	498
18.3.1 连接到 Team Foundation Server	462	20.3 理解团队角色透视	502
18.3.2 使用团队资源管理器	463	20.3.1 项目的视角	503
18.3.3 使用项目门户	463	20.3.2 项目经理	503
18.3.4 使用 Microsoft Office	464	20.3.3 业务分析师	508
18.3.5 使用项目警告	465	20.3.4 开发人员	510
18.3.6 使用项目报表	466	20.3.5 测试人员	513
18.4 小结	468	20.4 自定义工作项	514
第 19 章 源代码控制	469	20.4.1 将工作项植入流程	514
19.1 Team Foundation Source Control 基础	469	20.4.2 自定义现有工作项	518
19.1.1 基本架构	470	20.5 小结	520
19.1.2 安全权限和许可	470	第 21 章 建模	521
19.2 开始使用 Team Foundation Source Control	471	21.1 Team Architect 制品	521
19.2.1 配置 Visual Studio	472	21.1.1 项目模板	522
19.2.2 使用源代码控制资源管理器窗口	472	21.1.2 项模板	522
19.2.3 管理工作区	474	21.2 设计应用程序	523
19.2.4 向源代码控制中添加文件	475	21.3 定义系统	529
19.3 在源代码控制中编辑文件	477	21.4 定义基础结构	531
19.3.1 从代码仓库中获取文件	477	21.5 部署应用程序	537
19.3.2 签入变更	477	21.5.1 部署关系图	537
19.3.3 理解变更集的概念	481	21.5.2 验证部署	538
19.3.4 搁置代码	482	21.5.3 部署报表	539
19.3.5 合并更改	483	21.6 实现应用程序	540
19.4 分支和合并	485	21.6.1 设置实现属性	540
19.4.1 分支	486	21.6.2 生成项目	540
19.4.2 合并	487	21.7 可视化代码开发	541
19.5 小结	487	21.7.1 类关系图	541
第 20 章 工作项跟踪	488	21.7.2 添加项	542
20.1 理解工作项	489	21.7.3 定义类之间的关联	543
20.1.1 理解工作项的角色和 SDLC	489	21.7.4 定义方法、属性、字段和事件	545
20.1.2 为项目选择工作项集	489	21.8 小结	545
		第 22 章 测试	546
		22.1 创建、配置和管理测试	546

22.1.1 测试项目	547	22.5 手动测试	573
22.1.2 测试项	548	22.5.1 创建手动测试	573
22.1.3 测试管理器	549	22.5.2 执行手动测试	573
22.1.4 测试的配置	550	22.6 一般测试	574
22.2 开发人员测试	551	22.7 顺序测试	575
22.2.1 样例单元测试	551	22.8 小结	575
22.2.2 编写有效的单元测试	551	第 23 章 Team Foundation Build	576
22.2.3 使用单元测试类和方法	552	23.1 Team Foundation Build 简介	576
22.2.4 创建单元测试	553	23.2 新建生成	579
22.2.5 运行单元测试	554	23.2.1 指定新建生成的信息	579
22.2.6 代码覆盖率分析	555	23.2.2 编辑生成类型	582
22.3 Web 测试	557	23.2.3 MSBuild 的角色	586
22.3.1 记录 Web 测试	557	23.3 启动生成	586
22.3.2 管理 Web 测试请求	559	23.3.1 计划生成	586
22.3.3 运行 Web 测试和查看结果	559	23.3.2 调用生成	587
22.3.4 为 Web 测试设置数据	560	23.4 监视和分析生成	588
22.3.5 从 Web 测试中提取值	564	23.4.1 Team Build 浏览器简介	588
22.3.6 请求验证规则	566	23.4.2 生成报表	589
22.4 负载测试	567	23.5 小结	590
22.4.1 创建负载测试	567	索引	591
22.4.2 复审和编辑负载测试	572		
22.4.3 运行负载测试和检查结果	572		

Part 1

第一部分

Visual Studio 2005 / .NET 简介

本 部 分 内 容

- 第 1 章 Visual Studio 2005 快速入门
- 第 2 章 IDE 快速入门
- 第 3 章 Visual Studio 2005 中的 .NET Framework
和语言增强

本章内容

- 基本的增强特性
- 产品家族

第一次使用新的编程工具往往让人手足无措。同原本熟悉的环境相疏远会经常感觉不那么自在。我们大多数人面对的情况是，刚对现有的工具和语言有所感觉的时候，新一波的更新狂潮又到来了。它向我们承诺新的工作效率水平，用光鲜的交互式开发环境（Interactive Development Environment, IDE）和一套新的炫酷特色来诱惑我们……当这些更新由星星之火壮大成为燎原之势，当这些新鲜刻录的DVD出现在面前，我们在开始新的探索之前，已经感受到因落伍带来的失落。

长此以往，你会期待发布的新版本Visual Studio 2005拥有新的编程模型、陌生的术语、新的对话框、新的查看代码的方式……这些不熟悉的事物也许会让你觉得无所适从。正如身处不熟悉的环境一样，你所需要的是一张地图或者一个指南针来找到正确的方向。我们希望通过本章的介绍，能够帮助你判明方位。我们将在这里向你介绍这个新工具的特色以及各种新的Visual Studio家族产品。

1.1 基本的增强特性

Visual Studio 2005和.NET Framework 2.0向现有的完整功能的工具集中引入了数百项的新特性。难以相信Visual Studio 2003竟然需要如此大量的改进！实际上语言方面的改进只占相当小的比例。最新版本提高了开发人员工作效率、减少了寻常的任务，涉及软件工程过程中的全部项目团队。接下来的小节将着重介绍这些能使工作更加简单高效的增强特性。

当然，我们将在整本书中详细地介绍每个增强项，对于忙碌的开发人员来说，本节可以作为“整体概述”来看。

注解 我们在这里将专门介绍Visual Studio的增强特性。.NET Framework和.NET语言（包括增强特性）将在以后的章节中进行介绍。

1.1.1 设计、编写和发现代码

每一个开发人员都希望编写出优秀的代码。没有人会故意写出一大堆杂乱无章的、难以理解的、不易维护的或无法扩展的代码。既然如此，为什么仍然有那么多这样的代码存在呢？开发人员通常缺少正规的培训（不允许脱产进行培训），并且迫于交付产品的时间压力（或者着急下班回家）。复杂性并没有降低，而问题却越积越多——最后期限、时间计划表、项目进度表，还有项目经理都会不停地催促你尽快完成工作。此时Visual Studio 2005能提供怎样的帮助？它能在编辑器中提供指导和“培训”；还能用正确的方式为你处理常规和寻常的任务，将你从中解放出来，使你能集中处理业务问题，用更

短的时间生产出更高质量的产品。接下来的几节将重点介绍这方面的新特性。

1. 首先确定代码的架构和设计

大多数开发人员都同意,拥有牢固的架构和可视化的代码将带来更好的软件产品和开发体验。但是架构文档通常是在项目开始时使用像Visio这类工具创建的,而在生成过程中会很快过时。然而再也没有足够的时间回去更新那些图表了,只能眼睁睁地看着它们很快恶化成为“原始设计”或者“系统愿景”文档——这不过是新瓶装旧酒,给那些以前表示应用程序的实际物理结构的东西贴上了新标签。建模工具和开发项目中往复地进行同步的方法本应该解决这个问题。然而,这个解决方法自身却充满着问题,其中主要的问题就是开发人员希望在他们的编码窗口或者IDE中(而不是在其他工具中)看到自己的代码。

Visual Studio 2005使得架构师和开发人员可以解决这些问题。架构师能够在编裁代码之前就创建完整的应用程序设计。建模工具在IDE中正常运行。代码和模型能够保持同步。例如类设计器就像是代码的图像化视图。使用设计器进行的编辑工作将立即在代码中反映出来,反之亦然。接下来介绍构建在Visual Studio 2005中的可视化设计器。

● 应用程序设计器

应用程序设计器(application designer)允许架构师创建一个系统定义模型(System Definition Model, SDM),用来详细说明如何为给定的解决方案组合不同的“应用程序”。这里所提到的应用程序指的是网站、Web服务、数据库等。例如,一个解决方案中的Web站点可能要与多个Web服务进行对话。这些Web服务也许需要与一个数据库或者消息队列依次进行通信。应用程序设计器支持以上这种模型。架构师可以指定哪些应用程序彼此进行通信,并且能够定义这些关系间的约束条件。图1-1展示了应用程序设计器的外观。

8

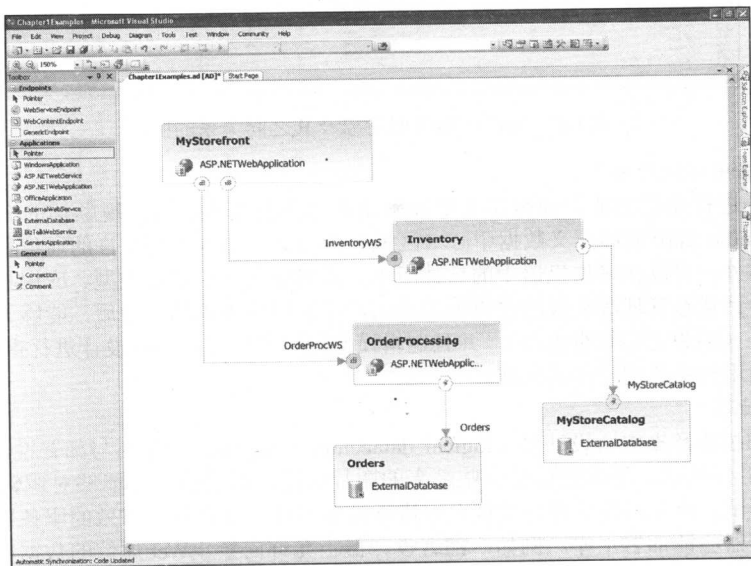


图1-1 在应用程序设计器中查看一个电子商务应用程序

● 类设计器

Visual Studio 2005的类设计器(class designer)提供在图形界面里编写和修改对象的方法。可以利