

115

TP312BA  
5676

实用全书丛书

# Visual Basic .NET 技术内幕

Visual Basic .NET Unleashed

[美] Paul Kimmel 著

吕建宁 等译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书全面介绍了编程的基本原理,详细讲述了 Visual Basic.NET 中引入的新功能,如继承和多态、事件处理程序、结构化异常处理和 Web 服务。提供了一些建立在实践基础上的附加见解和建议,目的是帮助读者在使用 Visual Basic .NET 时获得比使用 Visual Basic 6 更大的收益。全书共分 20 章,涉及的内容包括:新的 Visual Basic 语言、高级面向对象编程、通用语言运行时、客户/服务器开发、Web 编程等。本书的代码都是作者精心编写和组织的,为了能提供高质量的范例,尽量采用了重构和简单的可重用的面向对象代码。

本书适合于计算机编程的初学者和愿意使用 Visual Basic.NET 语言的专业人士。

Authorized translation from the English language edition published by Sams Publishing. Copyright © 2002. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright © 2002.

本书中文简体版专有翻译出版版权由 Pearson 教育集团所属的 Sams Publishing 授予电子工业出版社。其原文版权及中文翻译出版版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

著作权合同登记号 图字:01-2001-4597

### 图书在版编目(CIP)数据

Visual Basic .NET 技术内幕/(美)金梅尔(Kimmel, P.)著;吕建宁等译.-北京:电子工业出版社,2002.8  
(实用全书丛书)

书名原文:Visual Basic .NET Unleashed

ISBN 7-5053-7942-9

I. V... II. ①金... ②吕... III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 062432 号

责任编辑:赵红燕

印刷者:北京大中印刷厂

出版发行:电子工业出版社 [www.phei.com.cn](http://www.phei.com.cn)

北京市海淀区万寿路 173 信箱 邮编:100036

经 销:各地新华书店

开 本:787 × 1092 1/16 印张:34 字数:870 千字

版 次:2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

定 价:56.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077

## 译者序

自从微软公司推出.NET以来,从.NET新特性的展示中,我们感到 Visual Basic .NET 确实可以称之为“一流的”语言。现在 Visual Basic .NET 包括一些 C++、Java 和 Delphi 程序员已经使用多年的东西。随着 Visual Basic .NET 的问世,将消除 VB6 中存在的种种编程限制。

Visual Basic .NET 是一种一流的语言,支持并含有一流语言所应有的功能,包括继承、虚方法、共享方法、事件处理、结构化异常处理、多线程以及在名字空间级的封装。在 Visual Basic .NET 中,继承意味着你所继承的是方法和属性,而不仅仅是一个界面。事件处理可通过委托和特殊类实现,允许你编写动态的事件处理程序并且可用来支持窗体。共享方法有时被称为类方法或静态方法,支持在元类级的调用行为,还支持多线程、名字空间和用参数表示的构造函数。你会发现使用 Visual Basic .NET 编写的应用程序远远超过你的想像,超过这门语言本身和.NET框架的限制。

本书全面系统地介绍了 Visual Basic .NET 技术,语言通顺、概念清晰、可读性极强。本书的作者具有12年的编程经验,全书内容贯穿了作者的心得体会。对任何希望学习 Visual Basic .NET 编程的人员来说都是一本极好的参考书。

本书由吕建宁组织翻译,参加翻译的人员有:范文斌、贾晓芳、周联芳、邓华、朱维芳、张文、何青、李迎、赵捷、陈正林、袁园、王子东、赵晓兵、张军、易杰、周仁林、江东、李文静、张春林、吕剑峰、张宇、朱丽、刘晓凤等。

由于时间和水平有限,书中译文一定有不妥之处,敬请广大读者批评指正。

# 前 言

撰写一本数百页的书是一项浩大的工程。但是,“千里之行始于足下”,我们即将一起踏上正是这样的漫长旅程。我的旅程是要把一门计算机语言一页一页地编写出来,而你们的旅程是要学会这门语言。对于有些人来说,这似乎是一项难以完成的任务,但是不用担心,只要我们共同努力,就一定能够成功地完成这次旅行。

## 概述

Visual Basic .NET并不仅仅是 Visual Basic 7。微软公司对 Visual Basic 6 这门数百万开发人员熟悉并喜爱的语言进行了全面的修改。或许是由于修改的原因,VB显然需要一本介绍自己的书,而最适合阐述这些变化的还是微软公司。在以往的VB版本中有一些东西显然是被遗漏了。这些被遗漏的东西看起来可能并不影响 Visual Basic 内容的完整性,但是存在着缺陷。

在我们完成这本书的时候,你会发现 Visual Basic .NET新的威力和魅力,并且我想你一定会有同感,那就是早就该对 VB 进行全面修改了。

本书全面论述了采用完全面向对象语言编程的基本原理,书中含有许多新术语和新功能,如继承和多态、事件处理程序、结构化异常处理和Web服务。除了介绍基本原理,本书还提供了一些建立在实践基础上的附加见解和建议,它们会帮助你在使用 Visual Basic .NET时获得比使用 Visual Basic 6 更大的收益。

Visual Basic .NET具有编程所需要的一切。Visual Basic .NET为读者展现的强大功能和潜力是以往任何 Basic 版本所无法提供的。

## 读者对象

撰写本书是为了帮助编程新手和专业人员转换到一门被完全修改了的语言。本书将帮助你顺利地做到 Visual Basic .NET 的转换,并针对 VB 编程的各种需要为你提供全面的资源。

如果你是一个 Visual Basic 6 (或 VB 早期版本)程序员、一个编程新手或者打算从另一种语言转换到 Visual Basic .NET,那么,这本书正是为你写的。你将获得从面向对象编程的基本原理(OPP)到一些最高级的面向对象术语和复杂的.NET 编程范例的各种信息。

从一开始我就非常精心地编写代码和组织相关范例并在 [www.sampublishing.com](http://www.sampublishing.com) 和 [www.softconcepts.com](http://www.softconcepts.com) 网站上提供了这些范例。我还花时间融入了 Extreme Programming (XP, 极端编程)中最精彩的部分之一:重构(Refactoring)。为了能提供高质量的范例,为了帮助你们解决问题,我尽量采用了重构和简单的可以重用的面向对象代码。这些代码范例看上去与VB6代码不同,它们更像 Visual Basic .NET 代码。我的目的是要帮助你们告别 VB6 并且最大程度地采用 Visual Basic .NET。

我采用 Martin Fowler 的 “Refactoring: Improving the Design of Existing Code” 作为重构的参考资料。任何由于误用重构方法而出现的错误都是我自己造成的。你们还会注意到本书彻底摒弃了你们中间大多数人可能非常熟悉的 Hungarian 符号。Hungarian 符号是为过去的编程语言而编写的并曾经给我们不少帮助。然而，Visual Basic .NET 面向对象和强有力的类型特性弱化了 Hungarian 符号的作用，因此，本书没有采用这一符号。

在你读完本书的前几章后，你将能成功地构建 Visual Basic .NET 应用程序。在你读完这本书的时候，你将能够非常专业、完整地使用 Visual Basic .NET 的一些最强大的功能。如果你的目的是使用 Visual Basic .NET 编写专业的解决方案，那么，本书正是为你编写的。

## Visual Basic .NET 是一流的语言

在我参加2000年11月举行的微软作者峰会时，微软公司的人士总是把 Visual Basic .NET 称为“一流的”语言。微软公司承认老的 VB 版本有待于改进，这是一个很大的进步。Visual Basic .NET 的确是一流的语言。

现在的 Visual Basic .NET 包括一些 C++、Java 和 Delphi 程序员已经使用多年的东西。那些已经使用上述其他语言编程并且也喜欢 VB 的人，正耐心地期待着那些能使他们走出“DLL 地狱”并取消 VB6 中实际存在的种种限制的变革。随着 Visual Basic .NET 的问世，这些限制将被消除。我的任务就是揭示这些曾经存在的限制，同时说明 Visual Basic .NET 所采用的方法。谈到新的功能，不知道我是否可以“显式地”说，“在 VB6 里你可不能这么做”，使用 VB6 编程的人一定懂得“显式地”的意思。

说 Visual Basic .NET 是一流的语言，是因为它显式地支持并含有一流语言所应有的东西，包括继承、虚方法、共享方法、事件处理、结构化异常处理、多线程以及在名字空间级的封装。在 Visual Basic .NET 里，继承意味着你所继承的是方法和属性，而不仅仅是一个界面。事件处理可通过委托和特殊类实现，它们允许你编写动态的事件处理程序并且可用来支持窗体。共享方法有时被称为类方法或静态方法，支持在元类级的调用行为，它还支持多线程、名字空间和用参数表示的构造函数。你会发现使用 Visual Basic .NET 编写的应用程序远远超过你的想像，超过这门语言本身和 .NET 框架的限制。

在这本书里，我们还会发现很多事物。如果你只用过 VB6 编程，我猜想新的 VB 可能会让你吃惊。而如果你用过 Java、C++ 或 Delphi 编程并且喜欢 Visual Basic，你一定会为之而赞叹，VB 终于不再是使用其他语言编程的人士所认为的“二流语言”了。

从表面上看，开始学习 Visual Basic .NET 时似乎和学习 VB6 一样容易。但如果你只停留在表面上，你可能会感到失望，你编写的代码也不会有什么进步。和很多事物一样，正是这外表下面的内涵造就了千变万化的世界。

## 什么是通用语言运行时

从市场上销售的资料中，你会了解到通用语言运行时（Common Language Runtime, CLR）取代了令人乏味的 COM、MTS、VB 运行时 DLL 和其他一些技术。在实践方面，CLR 是由微软

公司开发出来的,微软公司所做的也正是我们大家应该做的,即寻求一种能够让一个代码体发挥其最大功效的方法,使它能在多个上下文里重用。

CLR 是一个类和代码的常见结构,它可以被 .NET 框架下的所有语言共享。这就是说,运行 Visual Basic .NET 的核心代码体的大部分可以在 C#、Visual C++ 和其他实现 .NET 的语言之间共享。

CLR 是 Visual Basic .NET 的基础,这使它成为其他 .NET 语言的同胞兄弟。其实,CLR 为你提供的功能与其他语言相同,但它使配置 Visual Basic .NET 应用程序变得更加简便。

## 转换到 Visual Basic .NET 的一些强制性理由

关于升级到 Visual Basic .NET 的强制性理由并不是用几句话能够解释清楚的。转换的理由包括 WebForms、Web 应用程序、Web 服务、CLR、结构化异常处理、真实面向对象编程和多线程,这里列举的只是一小部分。

在你读完本书后,你将对这一转换能给你带来的好处有一个更加清晰的认识,这不是我在这里用几句话把这些好处概括一下就能做到的。这里你要知道的就是转换到 VB .NET 的理由是确实存在的和真实的,它绝对不是为了达到推销的目的。

## 涉及的问题

转换到 Visual Basic .NET 所涉及的问题是你将不得不全部重新编写用以前的 VB 版本编写的应用程序。这并不是一件困难的事情,因为微软公司提供了一个升级工具。当你在 Visual Basic .NET 里打开 VB6 应用程序时,Upgrade Wizard (升级向导)就会自动运行。假如世界上的应用程序都能这样自动升级那该多好啊!运行用 VB6 编写的简单的 HelloWorld 应用程序,为支持兼容性,向导会生成大量的附加代码 (HelloWorld 的升级版及其原始应用程序可以从 [www.sampublishing.com](http://www.sampublishing.com) 网站下载)。

除了提供 VB6 的升级帮助外,微软公司还对 Visual Basic .NET 的兼容性做了一些不太重要的让步以便适当地支持向后兼容,但这一切并不意味着不需要全部重写 VB6 应用程序。重写的好处在于你将可以充分地利用 Visual Basic .NET 的优势,用更少的程序做更多的事情。不重写应用程序所涉及的问题包括代码会复杂得多,并且维护起来非常困难,为了兼容,你可能仍然不得不重写大部分升级代码。

与简单的升级相比,重写现存的应用程序可以延伸 Visual Basic .NET 的使用范围。因此,对于所有的用户和编程人员来说,它的好处是显而易见的,而且最终会获得丰厚的回报。

## 本书主要内容

本书涵盖了新的 Visual Basic .NET 语言、统一的 Visual Studio IDE、使用 WebForm 的 Web 编程、ADO.NET、Web 服务、GDI+ 等等。

Visual Studio .NET 是一个企业工具。要在 Visual Basic .NET 的上下文里演示这一工具的功能,你将接触到宏和 Automation Extensibility Model (自动可扩展性模型)的相关内容。

作为极端编程（Extreme Programming, XP）的一个方面，重构（Refactoring）的采用为你提供了切实可行的代码，进而帮助你从面向对象语言的编程中获得最大的收益。

最后，所有的代码程序清单（从无关重要的到复杂的）均可以从 [www.sampublishing.com](http://www.sampublishing.com) 网站下载。代码样例分布在各相关章节里。

## 新的 Visual Basic 语言

在 Visual Basic .NET 里保留了一些 Visual Basic 的关键字和语句，但大多数只是相似而已。本书全面介绍了 Visual Basic .NET 所实现的 VB 核心语言。

## 高级面向对象编程

在 Visual Basic 6 中类支持接口、继承或 COM、继承。接口继承与面向对象的继承不同。在 VB6 里你可以使用类把公有的方法和属性定义在一个类中，然后在另一个类中实现它们。虽然它的效果还算不错，但是没有面向对象继承中的超类和子类关系的功能强大。

在 Visual Basic .NET 里，可以从超类或父类获得方法和属性并在子类里扩展它们，这样做的好处是可以更大程度地重用代码。在 Visual Basic .NET 里，用于 OOP（面向对象编程）的术语和支持它们的语法是不同的。本书将全面介绍修改后的面向对象术语和范例。

除了基本的 OOP 术语外，本书还介绍并演示了修改 OOP 在 Visual Basic .NET 中对界面设计、使用名字空间、定义类、实现委托、继承、多态、封装和共享方法的影响。所有理论上的探讨都配有演示相关技术的样例。

在掌握了新的 OOP 术语后，本书会继续演示一些高级的概念，如映像、属性、多线程和 Web 编程。

在本书进行到演示如何优化和重用代码时，代码样例将被重构。

## 通用语言运行时

本书将演示 CLR 里大量的类，包括扩展的 System.Diagnostics 类，它为你提供了更好的测试和调试方法。

本书在适当的地方还将介绍 CLR 的其他方面。例如，在我们讨论界面设计和窗体时，我们将介绍 Windows Form 和 GDI+。

## 客户/服务器开发

客户/服务器编程仍然是编程中的主流。第 19 章介绍了 ADO.NET 编程方面的一些情况，讨论了 ASP.NET 和 Web 应用程序。

要介绍数据库编程，需要大量的篇幅，为此，我正在努力编写一本介绍 Web 编程和 ADO.NET 的新书。有关这本新书的情况请查阅 [www.sampublishing.com](http://www.sampublishing.com) 网站。

## Web 编程

对于 Web 编程来说，使用 Visual Basic .NET 是最佳的选择。ASP.NET 使用代码隐藏模块，像在 Windows Application 中一样，使用 Visual Basic .NET 就可以实现代码隐藏模块。第 18 章演

示了 Web 服务，第 19 章则介绍了 ASP.NET Web 应用程序和服务器控件。要介绍 Web 编程中的大量控件，需要一本专著，但愿在你们完全掌握了这本书的时候，我能为你们提供一本全面介绍 Web 编程的书。

## 小结

可能你会觉得我们太苛求了。在没有完整介绍一门新语言的情况下，要在这么一本书里介绍一门语言的新特性是比较困难的。但这不是一本薄书。本书是一本全面介绍 Visual Basic .NET 编程的书，因为在这门新语言里要学的东西很多。

本书的目的就是为读者提供全面的资料，随着经验和需要的增长可以反复阅读。第 1 章介绍了新的统一的 Visual Studio IDE，然后很快从基本原理和高级编程过滤到描述和解释 Visual Basic .NET 编程的一些最强大的特性。

# 目 录

## 第一部分 Visual Basic .NET 入门

|                                      |    |
|--------------------------------------|----|
| 第 1 章 使用统一的 Visual Studio IDE .....  | 2  |
| 1.1 用户自定义配置文件 .....                  | 3  |
| 1.2 创建一个项目 .....                     | 3  |
| 1.3 构建项目 .....                       | 8  |
| 1.4 窗体后面的代码 .....                    | 10 |
| 1.5 配置 IDE 选项 .....                  | 19 |
| 1.6 在新的 IDE 中调试 .....                | 21 |
| 1.7 项目模板概述 .....                     | 23 |
| 1.8 IntelliSense .....               | 28 |
| 1.9 使用视图 .....                       | 29 |
| 1.10 文档 .....                        | 34 |
| 1.11 小结 .....                        | 35 |
| 第 2 章 吐故纳新 .....                     | 36 |
| 2.1 文件扩展名 .....                      | 36 |
| 2.2 名字空间 .....                       | 38 |
| 2.3 引用 .....                         | 38 |
| 2.4 Option 语句 .....                  | 39 |
| 2.5 数据类型 .....                       | 42 |
| 2.6 变量声明 .....                       | 58 |
| 2.7 运算符 .....                        | 62 |
| 2.8 类型转换函数 .....                     | 64 |
| 2.9 在 VB .NET 中变量范围的变化 .....         | 65 |
| 2.10 流控制语句 .....                     | 65 |
| 2.11 数组和集合 .....                     | 66 |
| 2.12 结构化异常处理 .....                   | 70 |
| 2.13 Visual Basic .NET 中保留字的处理 ..... | 71 |
| 2.14 VB6 和 VB .NET 之间的兼容性 .....      | 71 |
| 2.15 小结 .....                        | 72 |
| 第 3 章 Visual Basic .NET 基础编程 .....   | 73 |
| 3.1 声明和初始化变量 .....                   | 73 |

|              |                                     |            |
|--------------|-------------------------------------|------------|
| 3.2          | 使用块级范围 .....                        | 78         |
| 3.3          | 静态变量 .....                          | 79         |
| 3.4          | 使用静态变量 .....                        | 79         |
| 3.5          | 使用数组 .....                          | 80         |
| 3.6          | 使用新的抽象数据类型 .....                    | 83         |
| 3.7          | 隐藏变量 .....                          | 87         |
| 3.8          | 函数和子例程 .....                        | 89         |
| 3.9          | 定义结构 .....                          | 89         |
| 3.10         | 使用对象 .....                          | 89         |
| 3.11         | 异常处理 .....                          | 91         |
| 3.12         | 小结 .....                            | 96         |
| <b>第 4 章</b> | <b>宏和 Visual Studio 的可扩展性 .....</b> | <b>97</b>  |
| 4.1          | 自动化重复性任务 .....                      | 98         |
| 4.2          | 使用宏管理器 .....                        | 106        |
| 4.3          | 导出宏 .....                           | 106        |
| 4.4          | 使用 Macro IDE .....                  | 110        |
| 4.5          | 创建一个宏项目 .....                       | 112        |
| 4.6          | 从命令窗口管理 Visual Studio .....         | 113        |
| 4.7          | 响应 IDE 事件 .....                     | 114        |
| 4.8          | 定制 Visual Studio .....              | 115        |
| 4.9          | 创建插件 .....                          | 122        |
| 4.10         | 创建向导 .....                          | 128        |
| 4.11         | Visual Studio 集成项目 (VSIP) .....     | 131        |
| 4.12         | 小结 .....                            | 131        |
| <b>第 5 章</b> | <b>子例程、函数和结构 .....</b>              | <b>132</b> |
| 5.1          | 编写过程 .....                          | 132        |
| 5.2          | 定义过程参数 .....                        | 137        |
| 5.3          | 使用递归 .....                          | 149        |
| 5.4          | 定义结构 .....                          | 150        |
| 5.5          | 使用枚举类型 .....                        | 158        |
| 5.6          | 小结 .....                            | 158        |

## 第二部分 高级面向对象编程

|              |                       |            |
|--------------|-----------------------|------------|
| <b>第 6 章</b> | <b>映像 .....</b>       | <b>160</b> |
| 6.1          | Reflection 名字空间 ..... | 160        |
| 6.2          | 使用映像运行时发布类型 .....     | 174        |
| 6.3          | 映像使用的其他地方 .....       | 176        |
| 6.4          | 本地化 .....             | 177        |

|                                    |            |
|------------------------------------|------------|
| 6.5 小结 .....                       | 177        |
| <b>第7章 创建类 .....</b>               | <b>178</b> |
| 7.1 定义类 .....                      | 178        |
| 7.2 封装和信息隐藏 .....                  | 181        |
| 7.3 添加字段和属性成员 .....                | 182        |
| 7.4 给类添加方法 .....                   | 194        |
| 7.5 添加类事件 .....                    | 203        |
| 7.6 定义嵌套类 .....                    | 204        |
| 7.7 创建类的实例 .....                   | 213        |
| 7.8 小结 .....                       | 214        |
| <b>第8章 添加事件 .....</b>              | <b>215</b> |
| 8.1 理解事件和事件处理程序 .....              | 215        |
| 8.2 在 Windows 窗体设计器中创建事件处理程序 ..... | 224        |
| 8.3 在代码编辑器中创建事件处理程序 .....          | 226        |
| 8.4 创建运行时事件处理程序 .....              | 228        |
| 8.5 在 Web 窗体设计器中创建事件处理程序 .....     | 229        |
| 8.6 声明和引发事件 .....                  | 230        |
| 8.7 小结 .....                       | 235        |
| <b>第9章 理解委托 .....</b>              | <b>236</b> |
| 9.1 使用 EventHandler 委托 .....       | 237        |
| 9.2 委托成员概述 .....                   | 244        |
| 9.3 定义委托 .....                     | 246        |
| 9.4 把委托作为参数传递 .....                | 247        |
| 9.5 多转换委托 .....                    | 255        |
| 9.6 跨项目使用委托 .....                  | 258        |
| 9.7 小结 .....                       | 260        |
| <b>第10章 继承和多态性 .....</b>           | <b>261</b> |
| 10.1 继承的基础知识 .....                 | 261        |
| 10.2 什么是继承 .....                   | 262        |
| 10.3 定义必须被子类化的类 .....              | 267        |
| 10.4 定义不能被子类化的类 .....              | 270        |
| 10.5 多态 .....                      | 270        |
| 10.6 动态类型转换 .....                  | 275        |
| 10.7 定义接口 .....                    | 277        |
| 10.8 小结 .....                      | 280        |
| <b>第11章 共享成员 .....</b>             | <b>281</b> |

|      |                |     |
|------|----------------|-----|
| 11.1 | 声明共享字段 .....   | 281 |
| 11.2 | 定义共享属性 .....   | 282 |
| 11.3 | 使用共享方法 .....   | 285 |
| 11.4 | 定义共享构造函数 ..... | 290 |
| 11.5 | 实现构件方法 .....   | 296 |
| 11.6 | 重载共享成员 .....   | 298 |
| 11.7 | 引发共享事件 .....   | 300 |
| 11.8 | 小结 .....       | 303 |

|               |                                |            |
|---------------|--------------------------------|------------|
| <b>第 12 章</b> | <b>定义属性 .....</b>              | <b>304</b> |
| 12.1          | 理解属性的作用 .....                  | 305        |
| 12.2          | 注释类型和成员 .....                  | 313        |
| 12.3          | 用 MSIL Disassembler 查看属性 ..... | 322        |
| 12.4          | 使用映像提取属性 .....                 | 322        |
| 12.5          | 创建自定义属性 .....                  | 323        |
| 12.6          | 组件属性 .....                     | 326        |
| 12.7          | COM 互操作属性 .....                | 327        |
| 12.8          | 小结 .....                       | 327        |

## 第三部分 用户界面设计

|               |                        |            |
|---------------|------------------------|------------|
| <b>第 13 章</b> | <b>创建控制台应用 .....</b>   | <b>330</b> |
| 13.1          | 控制台应用基础 .....          | 330        |
| 13.2          | 使用 Console 类 .....     | 336        |
| 13.3          | 定义 FileSort 演示程序 ..... | 341        |
| 13.4          | 控制台应用名字空间 .....        | 347        |
| 13.5          | 多线程化控制台应用 .....        | 348        |
| 13.6          | 调试控制台应用 .....          | 350        |
| 13.7          | 跟踪文件系统 .....           | 353        |
| 13.8          | 小结 .....               | 354        |

|               |                          |            |
|---------------|--------------------------|------------|
| <b>第 14 章</b> | <b>多线程应用 .....</b>       | <b>355</b> |
| 14.1          | 没有线程的异步处理 .....          | 356        |
| 14.2          | 带有线程池的轻型线程 .....         | 357        |
| 14.3          | 重型线程 .....               | 370        |
| 14.4          | 应用 ThreadStatic 属性 ..... | 372        |
| 14.5          | Windows 窗体中的多线程 .....    | 374        |
| 14.6          | 小结 .....                 | 377        |

|               |                            |            |
|---------------|----------------------------|------------|
| <b>第 15 章</b> | <b>使用 Windows 窗体 .....</b> | <b>378</b> |
| 15.1          | Forms 名字空间概述 .....         | 378        |

|               |                             |            |
|---------------|-----------------------------|------------|
| 15.2          | System.Drawing 名字空间概述 ..... | 390        |
| 15.3          | 使用 Form 类 .....             | 396        |
| 15.4          | 使用 GDI+ 定制窗体绘画 .....        | 401        |
| 15.5          | 小结 .....                    | 401        |
| <b>第 16 章</b> | <b>设计用户界面 .....</b>         | <b>402</b> |
| 16.1          | 管理控件布局 .....                | 402        |
| 16.2          | 使用菜单 .....                  | 404        |
| 16.3          | 高级窗体技术 .....                | 410        |
| 16.4          | 组件和控件类能为我们做什么 .....         | 413        |
| 16.5          | 动态添加控件 .....                | 414        |
| 16.6          | 创建自定义的 UserControls .....   | 415        |
| 16.7          | 创建自定义控件 .....               | 419        |
| 16.8          | 小结 .....                    | 421        |
| <b>第 17 章</b> | <b>用 GDI+ 进行编程 .....</b>    | <b>423</b> |
| 17.1          | GDI+ 基础 .....               | 423        |
| 17.2          | 高级绘图 .....                  | 430        |
| 17.3          | Windows 窗体中的自定义绘图 .....     | 441        |
| 17.4          | 图形打印 .....                  | 441        |
| 17.5          | 小结 .....                    | 442        |

## 第四部分 构建 Web 服务

|               |                             |            |
|---------------|-----------------------------|------------|
| <b>第 18 章</b> | <b>使用和实现 Web 服务 .....</b>   | <b>444</b> |
| 18.1          | Web 服务概述 .....              | 444        |
| 18.2          | 使用 UDDI 查找 Web 服务 .....     | 446        |
| 18.3          | 调用 Web 服务 .....             | 448        |
| 18.4          | 实现 Web 服务 .....             | 452        |
| 18.5          | 选择 Web 访问方法 .....           | 460        |
| 18.6          | 管理状态信息 .....                | 461        |
| 18.7          | 在 Web 服务中处理和抛出异常 .....      | 463        |
| 18.8          | 调试 Web 服务 .....             | 464        |
| 18.9          | 编译和配置 Web 项目 .....          | 465        |
| 18.10         | 小结 .....                    | 466        |
| <b>第 19 章</b> | <b>ASP.NET Web 编程 .....</b> | <b>467</b> |
| 19.1          | Web 窗体 .....                | 467        |
| 19.2          | 使用请求和响应 .....               | 472        |
| 19.3          | ASP.NET 和 ADO.NET .....     | 472        |
| 19.4          | 输出缓存 .....                  | 477        |

|               |                                     |            |
|---------------|-------------------------------------|------------|
| 19.5          | 性能考虑 .....                          | 481        |
| 19.6          | 呈现控件 .....                          | 482        |
| 19.7          | 给页面动态添加控件 .....                     | 483        |
| 19.8          | 创建自定义 Web 控件 .....                  | 488        |
| 19.9          | 小结 .....                            | 496        |
| <b>第 20 章</b> | <b>写到事件日志中 .....</b>                | <b>498</b> |
| 20.1          | 管理事件源 .....                         | 498        |
| 20.2          | 把事件写到现有的日志中 .....                   | 502        |
| 20.3          | 写到一个自定义事件日志中 .....                  | 503        |
| 20.4          | 获得事件日志的内容 .....                     | 505        |
| 20.5          | 清除事件日志 .....                        | 506        |
| 20.6          | 接受事件通知 .....                        | 506        |
| 20.7          | 使用远程事件日志 .....                      | 507        |
| 20.8          | 使用 EventLogTraceListener .....      | 507        |
| 20.9          | 小结 .....                            | 508        |
| <b>附录 A</b>   | <b>在 VB.NET 中 VB6 编程元素的改变 .....</b> | <b>509</b> |

# 第一部分

## Visual Basic .NET 入门

- 第 1 章 使用统一的 Visual Studio IDE
- 第 2 章 吐故纳新
- 第 3 章 Visual Basic .NET 基础编程
- 第 4 章 宏和 Visual Studio 的可扩展性
- 第 5 章 子例程、函数和结构

# 第1章 使用统一的 Visual Studio IDE

## 本章要点：

- 用户自定义配置文件
- 创建一个项目
- 构建项目
- 窗体后面的代码
- 配置 IDE 选项
- 在新的 IDE 中进行调试
- 项目模板概述
- IntelliSense
- 使用视图
- 文档
- 小结

起初，一个新的体系结构诞生了。当设计师将该模型公诸于众时，比尔·盖茨认为它是一个很好的体系，便说，看，我们应该给这个体系起名叫.NET。后来经济崩溃了，电子商务公司失败了，据新闻报道，很多人因购买技术股而损失惨重。

现在，让我们从历史的角度看待过去的一年，回顾一下过去。新经济是以信息为基础的，而旧经济则以制造业为基础。我们正处在信息时代，或如 Shoshana Zuboff 认为的那样，是信息商务将各种活动、事件及对象的描述和度量转变为信息。

Visual Basic .NET 是为了满足使信息成为无所不在（即“信息化”）的需要而推出的新的体系结构产品。商业需要跟踪交易和购买习惯，通过各种设置、包括蜂窝电话提供产品和服务。

我是伴随着 Basic 长大的，历经 ROM-BASIC，BASICA，GW-Basic，Basic for DOS，Visual Basic for DOS 和 Visual Basic for Windows 等六七个版本。Visual Basic .NET 的问世归功于它以前的版本，但 Visual Basic .NET 却是至今变化最大、最深刻的版本。从统一的集成开发环境（Integrated Development Environment，IDE）及其工具到通用语言运行时（Common Language Runtime，CLR）以及面向对象的语法，一切都改变了。如果你是一个有经验的 VB6 程序员，你将在 VB .NET 里发现很多熟悉的元素，使你感觉像在家里一样；另外，VB.NET 中还有一些新的特性。

由于所有这些原因，我们从 IDE 开始。在本章中你将看到 Visual Studio .NET 中一些最出色和最有用的特性以及 .NET Framework 中的工具。此外，我们还将创建几个应用程序。

## 1.1 用户自定义配置文件

第一次打开 Visual Studio.NET 时, 首先要注意的是没有启动 Visual Basic 的单独菜单项。要创建一个 Visual Basic 应用程序, 需要选择 Start, Programs, Microsoft Visual Studio.NET 7.0, 然后, 选择 Microsoft Visual Studio .NET 7.0 ( 以下称为 Studio 或 Visual Basic )。

当第一次运行 Studio 时, 会出现一个启动页( Start Page )——一个起 IDE 导航器作用的 HTML 页面。这个菜单是可用的, 但在开始编程以前, 还需先做一些准备工作。如果不小心关闭了 Start Page, 可以通过选择 Help ( 帮助 ), Show Start Page in Studio 重新打开它。Start Page 首先出现是因为微软具有统一的 Visual Studio, 一个惟一适用于所有 .NET 开发的环境。统一的 IDE 是指 .NET 语言之间的互用性。

Visual Studio.NET 包括配置文件管理。在 Start Page 上选择 My Profile 可允许你定义用户配置文件, 它的基本含义是你可以使 Visual Studio 在外观和操作上尽可能接近你所期待的 Visual Basic 环境。把配置文件设置到 Visual Basic 作为第一步会使你的环境尽可能与 Visual Basic 6 一致, 从而允许 VB .NET 大大改变。

对于 Visual Basic 用户来说, 要自定义用户配置文件, 需按以下步骤操作:

1. 如果 Start Page 未被显示, 从 Studio Help 菜单中选择 Help, Show Start Page。
2. 在 Get Start 下方的左栏中, 选择 My Profile ( Start Page 是一个 HTML 页面, 所以使用它。一切都采用“指向+单击”式的超链接 )。
3. 在 Profile 组合框中, 选择 Visual Basic Developer ( 如果其余的组合框包含你需要的值——参见 Visual Basic 6 样式配置, 可以跳过其余步骤 )。
4. 在 Keyboard Scheme 组合框中, 选择 Visual Basic 6。
5. 在 Window Layout 组合框中, 选择 Visual basic 6。
6. 在 Help Filter 组合框中, 选择 Visual Basic and Related。
7. 对于 Show Help 单选按钮, 任选一个你喜欢的。我选择 Internal Help。
8. 对于 At Startup Show 组合框, 选择 Show Start Page, 或者可以选择用 Studio 打开 New Project 对话框, 和使用 VB6 一样。

在此可以探索 Start Page 的其他一些选项, 但你已经准备好了打开你的第一个项目。让我们创建一个简单的控制台应用程序, 同时探讨一下该项目的一些特性。

## 1.2 创建一个项目

为了表示对前辈的敬意, 我们将检测修订后的项目管理工具并创建著名的“Hello World”应用程序作为控制台的应用程序 ( 关于项目模板 ( Project Templates ) 的更多信息, 参见本章后面的“项目模板概述”一节 )。

到现在为止, 你已经配置了使用 Visual Basic 设置的配置文件。在这里选择 File, New, Project in Visual Studio .NET。New Project ( 新项目 ) 窗口将列出项目类型。选择 Visual Basic Projects