

THE EXPERT'S VOICE IN .NET

Pro VB 2008 and the .NET 3.5 Platform

.NET 3.5与VB 2008

高级编程 (第3版)

使用VB 2008发挥.NET平台威力



(美) Andrew Troelsen 著
姜玲玲 唐明霞 译



清华大学出版社

.NET 3.5 与 VB 2008

高级编程

(第 3 版)

(美) Andrew Troelsen 著
姜玲玲 唐明霞 译

清华大学出版社

北 京

EISBN: 978-1-59059-822-1

Pro VB 2008 and the .NET 3.5 Platform

Andrew Troelsen

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright © 2008 by Apress L.P. Simplified Chinese-Language edition copyright © 2009 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司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: 01-2008-2286

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

.NET 3.5 与 VB 2008 高级编程(第3版)/(美)特洛尔森(Troelsen, A.) 著;姜玲玲,唐明霞 译.

—北京:清华大学出版社,2009.10

书名原文: Pro VB 2008 and the .NET 3.5 Platform

ISBN 978-7-302-20841-9

I. N… II. ①特… ②姜…③唐… III. BASIC 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 157791 号

责任编辑:王 军 郑雪梅

装帧设计:康 博

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:清华大学印刷厂

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:74 字 数:1990 千字

版 次:2009 年 10 月第 1 版 印 次:2009 年 10 月第 1 次印刷

印 数:1~3000

定 价:139.00 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:025422-01

前言

.NET 平台刚发布时(大约是 2001 年)在 Visual Basic 编程界引起了不小的轰动。一方面,很多顽固的 VB6 开发人员抗议 VB6 和 Visual Basic .NET(VB.NET)之间的区别太大。这个群体中有人惊讶地发现,VB.NET 实际上不是“VB7”(即与 VB6 具有相同的语法和编程结构,加上一些新的功能),其中有些东西已截然不同。

事情的真相是 VB.NET 与 VB6 的关系不大,最好将它看作 BASIC 家族的一门新语言。这一事实使有的人极度畏缩,以致于他们给 VB.NET 起了“VB.NOT”或“Visual Fred”等绰号来表达他们的不满。事实上,甚至有人开发了网站(<http://vb.mvps.org/vfred/Trust.asp>),专门写了请愿书,抨击 Microsoft 为了 VB.NET 这一新生事物而放弃 VB6 的决策。

VB.NET 不仅引入了重大的语法变化,而且有些 VB6 风格的行为在 .NET 平台下完全不见踪影,因为它只会增加混乱。数据访问、表单开发和 Web 站点结构的核心 .NET 编程模型完全不同于它们基于 COM 的对应物。

随着时间的推移,.NET 平台成为了主流的编程模型,即便是最顽固的 VB6 开发人员似乎也开始动摇了:VB6 很快成为遗留编程工具。甚至 Microsoft 本身也明确地表示会随着时间的推移逐渐停止对 VB6 的支持。这也许是好事,它迫使我们改变。

■注意:

在发布 .NET 2.0 时(大约是 2005 年),VB.NET 被重命名为“Visual Basic 2005”。在发布 .NET 3.5 时,Microsoft 的 BASIC 语言又被重新命名,这次称为“Visual Basic 2008”(是的,VB 重命名游戏玩得疯了)。在本书中,当看到术语 Visual Basic、VB 或 Visual Basic 2008 时,要知道它们都是指 .NET 平台中的 BASIC 语言。当指的是以 COM 为中心的 BASIC 语言时,会使用术语 Visual Basic 6.0 或简称为 VB6。

另一方面,很多 VB6 开发人员也为这众多的新语言功能所激动,而且开始虚心学习。这个群体的成员已准备好深入挖掘面向对象编程(OOP)、多线程应用程序开发以及从 .NET 基类库中发现的大量功能的细节。这些人很快意识到,在很多(如果不是大部分)情况下,现有 VB6 代码可以保留 VB6 代码,而新的开发可以用 .NET 平台和 Visual Basic 语言的最新迭代进行。

奇怪的是还有第三群人,随着 Visual Basic .NET 的发布而形成。因为 VB.NET 实际上是一门崭新的 OOP 语言,所以很多从来没有想过学习以 BASIC 为中心的语言的开发人员(通常是 C++、Java 和 C#程序员)现在思想更为开放,开始探索全无分号和花括号的语言了。

无论如何,不管您认同哪一种群体,都欢迎您阅读本书。本书将采用的总体方法是将 VB 2008 作为 BASIC 家族中的一个独特成员来对待。在学习了开头几章后,就会介绍 VB 2008 的

语法和语义。这里您将发现 OOP 的全面基础,包括了新 VB 2008 的所有语言功能(如对象初始化语法、匿名类型、扩展方法和语言集成查询),并指导如何使用 Visual Studio 2008 集成开发环境。

另外,本书将深入讨论构建 .NET 应用程序时将用到的各个主要 .NET 代码库。还将提供 .NET 桌面编程的各个架构(Windows Forms 和 Windows Presentation Foundation)、ADO.NET 的数据库编程、用 ASP.NET 进行的 Web 开发,以及其他关键 .NET 主题,如程序集配置、Windows Communication Foundation、Windows Workflow Foundation 和文件 IO 操作。

我们是一个小组,您和我

技术作者为一个小组的成员(我应当知道——我是他们中的一员)编写了本书。您知道用任何平台构建软件解决方案都是极其详细的,而且非常具体地针对您的部门、公司、客户群等。也许您在电子发行行业工作,为州或本地政府开发系统,或者在 NASA 或军事分支机构工作。就我而言,我开发了孩子的教育软件、各种 n 层系统,以及医药和法律行业的大量项目。很可能您在您的就业地点写的几乎 100% 代码都和我在我的工作地点写的代码不同(除非我们以前碰巧一起工作过)。

因此,在本书中我有意识地选择避免创建与特定行业或编程 vein 紧密相关的示例代码。鉴于此,我选择用与行业无关的示例来解释 VB 2008、OOP、CLR 和 .NET 基类库。我不会给每个示例填充网格数据、计算薪金或诸如此类的事,相反,我会坚持采用我们都相关的主题:汽车(它有一些几何结构,还有一些与员工有关的事情)。所以我们采用这个示例。

我的任务是解释 VB 2008 编程语言和 .NET 平台的核心方面。为此目的,当本书结束时,我将尽一切努力来为您提供继续学习所需的工具和策略。

您的任务是接受这些信息并将它应用到您的特定编程任务上。我当然知道您的项目很可能不是有关汽车和昵称的,但通过这些示例可以运用所学的知识!请放心,一旦您理解了本书中提出的概念,就可以很好地构建映射到您自己独有的编程环境的 .NET 解决方案。

本书读者对象

本书假设您目前不具备以 BASIC 为中心的语言或 Microsoft 的 .NET 平台的经验(然而,如果已经有一些经验,当然更好)。本书假定您要么是一名专业软件工程师,要么是计算机科学系的学生。因此,本书可能不是特别适合对软件开发完全陌生的人,因为我们将探索很多低级/高级主题,而且不会花很多时间来将数据绑定到网格(至少在第 22 章之前不会),也不会介绍 Visual Studio 2008 菜单项的每个选项。

尽管本书将深入讨论一些高级主题,但这并不是说这里包括的材料就是不实用的!本书着重关注您作为一名精通的 Visual Basic 2008 开发人员必须理解的细节。由于部分信息可能比较有挑战性(例如,理解委托和 lambda 表达式的作用),因此本书是以友好而平易近人的语气编写的。

本书假定您是希望了解 VB 2008 内部工作的开发人员,因此不会通过简单地“拖放”来编

写代码。尽管本书大部分是分析如何使用 Visual Basic 2008 来减少您必须手动创建的代码量，但本书通常只在您理解如何亲自编写代码后说明一下集成的向导的用法。这有助于您轻松地将集成了 IDE 的代码改为您喜欢的代码。

本书主要内容

本书从逻辑上划分成 8 个不同的部分，每个部分包括一些集中了特定技术集和/或特定任务的章节。为了帮助了解本书，下面将本书逐个部分和逐章地分解。

第 I 部分：Visual Basic 2008 与 .NET 平台

第 I 部分的目的是使您适应 .NET 平台、.NET 类型系统和构造 .NET 应用程序的过程中使用的各种开发工具的核心方面。在学习过程中，也会介绍 VB 2008 编程语言的一些基本细节。

第 1 章：.NET 基本原理

第 1 章的作用是作为本书其余部分的基石。首先，将介绍传统 Windows 开发的世界，并揭示以前的开发方式的缺点。然而，本章的基本目标是使您熟知一些 .NET 构件块，如公共语言运行库(Common Language Runtime, CLR)、公共类型系统(Common Type System, CTS)和公共语言规范(Common Language Specification, CLS)以及基类库。此外，还会初步介绍一下 VB 2008 编程语言和 .NET 程序集格式，还将介绍 .NET 平台的平台无关特性以及公共语言基础结构(Common Language Infrastructure, CLI)。

第 2 章：构建 Visual Basic 2008 应用程序

本章的目标是介绍用各种工具和技术编译 VB 2008 源代码文件的过程。首先，将介绍如何使用命令行编译器(vbc.exe)和 VB 2008 响应文件。在本章其余部分，将介绍各种 IDE，包括 SharpDevelop、Visual Basic 2008 Express 和 Visual Studio 2008。另外，将提供一些开放源工具，它们是很多 .NET 开发人员的常备工具。

第 II 部分：核心 VB 编程结构

这一部分探索 VB 2008 编程语言的核心方面，如固有数据类型、决策和迭代结构、构造(和重载)方法，以及操作数组、字符串、枚举和模块。接下来，将深入介绍关于结构化异常处理的作用，以及 CLR 如何处理内存管理细节。

第 3 章：VB 2008 编程结构(上)

本章首先介绍 VB 2008 模块类型的作用，以及可执行文件的入口点(Main()方法)的相关主题。本章还会介绍 VB 2008 的固有数据类型(以及它们的 CTS 等价物)、隐式和显式类型强制转换、迭代和决策结构，以及有效代码语句的结构。

第 4 章: VB 2008 编程结构(下)

本章将完成基本编码结构的分析。本章的主要推动力在于用 VB 2008 的语法挖掘构建子例程和函数的细节。接着,将会介绍 ByVal、ByRef 和 ParamArray 关键字的作用,并理解方法重载的主题。本章还介绍了如何构建和操作数组、枚举和结构,以及潜伏在后台中的底层类(System.Array、System.Enum 和 System.ValueType)。

第 5 章: 设计封装类类型

本章将深入挖掘第一个“OOP 支柱”:封装服务。本章不仅会介绍基本类结构(构造函数、共享成员和属性语法),而且还会研究几个辅助类设计技术,如 Partial 关键字的作用和 XML 代码文档语法。

第 6 章: 继承与多态

第 6 章的作用是分析 VB 2008 如何考虑 OOP 的其余“支柱”:继承和多态。这里您将了解如何用继承、虚拟方法、抽象方法(和类),以及各种类型强制转换操作构建相关类的家族。本章还会介绍.NET 基类库中的终极基类 System.Object 的作用。

第 7 章: 结构化异常处理

本章的重点是讨论如何用结构化异常处理来处理代码基中的运行时异常。本章不仅将介绍解决此类问题的 VB 2008 关键字(Try、Catch、Throw 和 Finally),而且会说明应用程序级和系统级的异常之间的区别。此外,本章将介绍 Visual Studio 2008 中的各种工具,利用这些工具可以调试逃出您的视线的异常。

第 8 章: 对象生存期

本章介绍 CLR 如何使用.NET 垃圾收集器管理内存。这里您将开始理解应用程序根、对象代和 System.GC 类型的作用。理解了这个基础后,本章其余部分将介绍构建“可诊断对象”的主题(通过 IDisposable 接口)以及如何与终结进程交互(通过 System.Object.Finalize()方法)。

第 III 部分: 高级 VB 编程结构

这一部分加深使用 VB 2008 时对 OOP 的理解。这里您将了解接口类型、委托和事件,以及几个高级主题(如运算符重载和泛型)的作用。另外,本部分深入挖掘了新的.NET 3.5 语言特性的细节,包括初步介绍 LINQ。

第 9 章: 使用接口类型

本章的内容构建在对基于对象的开发的理解之上,包括了基于接口编程的主题。这里您将学到如何定义支持多重行为的类型、如何在运行时发现这些行为,以及如何从对象级别有选择地隐藏特定行为。为了显示接口类型的有用性,本章会介绍如何使用接口构建自定义事件体系结构。

第 10 章：集合、泛型和可为空值的数据类型

本章首先介绍 `System.Collections` 名称空间的集合类型，它自从发布以来就是 .NET 平台的一部分。然而，自从 .NET 2.0 发布以后，VB 编程语言提供了对泛型的支持。正如您会看到的，泛型编程大大增强了应用程序性能和类型安全性。您不仅可以探索 `System.Collections.Generic` 名称空间中的各种泛型类型，而且还会了解到如何构建自己的泛型方法和类型(使用和不使用约束)。

第 11 章：委托、事件和 Lambdas

第 11 章的作用是揭开委托类型的面纱。简单地说，.NET 委托是一个指向应用程序中的其他方法的对象。使用这种模式，可以构建允许多个对象进行双向会话的系统。介绍了 .NET 委托的使用之后，将介绍 VB 2008 的 `Event`、`RaiseEvent`、`Handles` 和 `Custom` 关键字，它们被用来简化直接使用委托的编程操作。最后，将介绍 `lambda` 表达式和 VB 2008 的 `lambda` 运算符的作用。

第 12 章：运算符重载和自定义转换例程

本章通过引入一些高级编程技术来深入理解 VB 2008 编程语言。本章将包含值类型和引用类型的详细分析。接着，将学习如何重载运算符和创建自定义转换例程(隐式和显式)。本章最后对比了显式类型强制转换操作中 `CType()`、`DirectCast()` 和 `TryCast()` 的使用。

第 13 章：VB 2008 特有的语言特性

发布 .NET 3.5 时，VB 语言被增强为支持大量新的编程结构，其中很多被用来启用 LINQ API(从第 14 章开始分析)。本章将阐述局部变量的隐式类型、扩展方法、匿名类型和对象初始化语法的作用。

第 14 章：LINQ 概述

本章首先介绍 LINQ，它被认为是 .NET 3.5 的最有吸引力的方面。在本章您将发现，LINQ 允许构建强类型的查询表达式，它可以被应用到一些 LINQ 目标来操作意义最广泛的“数据”一词。这里，您将了解 LINQ to Objects，它允许向数据的容器应用 LINQ 表达式(数组、集合和自定义类型)。在后面的第 24 章分析如何向关系数据库(通过 LINQ to ADO)应用 LINQ 表达式和 XML 文档(按 LINQ to XML 的方式)时，这些信息将很有用。

第 IV 部分：使用 .NET 程序集编程

第 IV 部分深入讨论 .NET 程序集格式的细节。这一部分不仅介绍如何部署和配置 .NET 代码库，而且将介绍 .NET 二进制映像的内部组成。本书的这一部分还解释了 .NET 属性的作用和多线程应用程序的结构以及使用互操作性的程序集访问遗留 COM 应用程序。

第 15 章：.NET 程序集概述

从一个很高的水平来说，程序集是用来描述托管*.dll 或*.exe 文件的术语。然而，.NET 程

序集的真正意义远不止于此。本章将讨论单个文件和多文件程序集之间的区别，以及如何构建和部署各个实体。本章将说明私有和共享程序集可以使用基于 XML 的*.config 文件和发布策略程序集配置。本章还将研究 .NET Framework 配置实用程序的作用。

第 16 章：类型反射、迟绑定和基于属性的编程

第 16 章继续分析了通过 System.Reflection 名称空间进行运行时类型检查的过程。使用这些类型，可以构建可以即时读取程序集的元数据的应用程序。本章将讨论如何动态地使用迟绑定激活和操作运行时类型。本章的最后一个主题探索了 .NET 属性的作用(标准和自定义)。为了说明各个主题的有用性，本章以可扩展的 Windows Forms 应用程序的构造结束。

第 17 章：进程、AppDomains 和对象上下文

现在您已对程序集有了扎实的理解，本章开始深入挖掘加载的 .NET 可执行文件的成分。本章的目标是为了说明进程、应用程序域和上下文的边界之间的关系。这些主题为第 18 章的内容提供了适当的基础。第 18 章将介绍多线程应用程序的结构。

第 18 章：构建多线程应用程序

本章介绍了如何构建多线程应用程序，并说明了一些可用来编写线程安全代码的技术。为了理解委托对异步方法调用的固有支持，本章首先回顾了 .NET 委托类型。接下来将研究 System.Threading 名称空间中的类型。您会看到很多允许轻松地创建其他执行线程的类型(Thread、ThreadStart 等)。最后介绍 BackgroundWorker 类型，它大大简化了从桌面用户界面中线程的创建。

第 19 章：.NET 互操作性程序集

本部分的最后一章介绍了一个名为互操作性程序集的独特 .NET 程序集类型。这些二进制映像用来允许 .NET 应用程序使用经典 COM 类型。一旦深入了解了 .NET 应用程序如何使用 COM 服务器的细节，然后将学习相对的功能：使用 .NET 对象的 COM 应用程序。学完本章后，就会对互操作性层有了扎实的理解。

第 V 部分：.NET 基类库

到本书的这个部分，您已经对 VB 2008 语言、.NET 类型系统和程序集格式有了相当好的理解。本部分利用了您新得到的知识，探索了大量常用的基类库中发现的服务，包括文件 IO 和使用 ADO.NET 的数据库访问。本部分还包括使用 Windows Communication Foundation(WCF) 的分布式应用程序和使用 Windows Workflow Foundation(WF)API 的启用工作流的应用程序。

第 20 章：文件和目录操作

顾名思义，System.IO 名称空间允许与机器的文件和目录结构交互。在本章的学习过程中，您将学会如何通过程序创建(和销毁)目录系统以及将数据移进及移出各种流(基于文件、基于字符串、基于内存等)。

第 21 章：对象串行化

本章介绍 .NET 平台的对象串行化服务。简言之，串行化允许将一个对象(或一组相关对象)的状态保留到流中以便以后使用。反串行化是将对象从流中取到内存中以便应用程序使用的过程。一旦了解了基础知识，然后将学习如何通过 `ISerializable` 接口和一些 .NET 属性定制串行化过程。

第 22 章：ADO.NET 第 1 部分：连接层

在这前两章以数据库为中心的章节中，将学习 ADO.NET 编程 API。具体地说，本章将介绍 .NET 数据提供者的作用，以及如何使用 ADO.NET 的连接层与关系数据库通信，通过连接对象、命令对象、事务对象和数据读者对象来表示。此外，本章还将介绍自定义数据库以及本书其余部分将用到的数据访问库的创建方法。

第 23 章：ADO.NET 第 2 部分：断开连接层

本章继续通过分析 ADO.NET 的断开连接层来研究数据库操作。本章将讨论 `DataSet` 类型、数据适配器对象和 Visual Studio 2008 的大量工具的作用，这些工具大大简化了数据驱动应用程序的创建。接着将介绍如何将 `DataTable` 对象绑定到用户接口元素，如 Windows Forms API 的 `GridView` 类型。

第 24 章：使用 LINQ API 编程

第 14 章介绍了 LINQ 编程模型，特别是 LINQ to Object。本章将通过分析如何向关系数据库、`DataSet` 对象和 XML 文档应用 LINQ 查询来深入理解语言集成查询。接着将介绍数据上下文对象、`sqlmetal.exe` 实用程序和 Visual Studio 2008 的各种 LINQ 特有的方面的作用。

第 25 章：Windows 通信基础

.NET 3.0 引入了一个崭新的 API：WCF，它允许以一种对称的方式构建分布式应用程序，不管它们的底层管道技术是什么。本章将提供 WCF 服务、主机和客户端的构造。WCF 服务极其灵活，其中客户端和主机可以利用基于 XML 的配置文件来声明性地指定地址、绑定和契约。

第 26 章：Windows 工作流基础

除了 WCF 外，.NET 3.0 还引入了另一个 API：WF，它允许定义、执行和监控对复杂业务过程建模的工作流。本章将阐述 Windows Workflow Foundation 的整体作用，以及活动、工作流设计器、工作流运行时引擎的作用，而且还将介绍启用了工作流的代码库的创建。

第 VI 部分：使用 Windows Forms 创建桌面应用程序

本书的这一部分介绍如何使用 Windows Forms API 构建图形桌面应用程序。Windows Forms 是原始桌面 GUI 架构，自从版本 1.0 以来，它就是 .NET 基类库的一部分。虽然 .NET 3.0 确实自带了一个新的 GUI 架构(Windows Presentation Foundation)，但是 Windows Forms 仍然是 .NET 开发的关键部分，并且很可能是您为很多传统业务应用程序选择的 UI 工具箱。

第 27 章: Windows Forms

本章首先介绍 System.Windows.Forms 名称空间。在本章中,将学习构建支持菜单系统、工具栏和状态栏的传统 GUI 桌面应用程序的细节。本章还将介绍 Visual Studio 2008 的各种设计时的细节。

第 28 章: 使用 GDI+渲染图形数据

本章包括如何在 Windows Forms 环境中动态地渲染图形数据。除了讨论如何操作字体、颜色、几何图像和图像文件外,本章还介绍了点击测试和基于 GUI 的拖放技术。此外,本章还将介绍 Windows Forms 资源格式,它允许将图形图像文件、字符串数据和桌面应用程序的其他方面嵌套进可执行文件本身中。

第 29 章: 使用 Windows Forms 控件编程

关于 Windows Forms 的最后一章将介绍 .NET Framework 3.5 自带的大量 GUI 窗口小部件。在本章中,不仅将学习如何使用各种各样的 Windows Forms 控件编程,而且也将学习对话框开发和 Form 继承。另外,本章还介绍了如何构建集成到 IDE 中的自定义 Windows Forms 控件。

第 VII 部分: 使用 WPF 创建桌面应用程序

本书的这一部分集中介绍名为 Windows Presentation Foundation(WPF)的全新桌面编程模型。WPF 是“超动力的”UI 工具箱,允许构建高交互性和富媒体的桌面应用程序。在本部分中将讨论 WPF 的作用、名为 XAML 的基于 XML 的语法,以及几个 WPF 功能,如动画、图形渲染和数据绑定。

第 30 章: WPF 和 XAML

.NET 3.0 引入了一个全新的 GUI 工具箱,即 WPF。本质上,WPF 允许构建交互性极强的富媒体桌面应用程序(以及间接地构建 Web 应用程序)前台。与 Windows Forms 不同,这个超动力的 UI 架构将大量关键服务(2D 和 3D 图形、动画、富文档等)集成到一个统一的对象模型中。本章将首先介绍 WPF 和 Extendable Application Markup Language(XAML)。在本章中将学习如何使用 XAML 构建 XAML-free 的 WPF 程序,以及两者的结合。最后将构建一个自定义 XAML 查看器,它会在以 WPF 为中心的章节的其余部分使用。

第 31 章: 使用 WPF 控件编程

在本章中,将介绍如何使用 WPF 控件内容模型,以及一些相关的以控件为中心的主题,如依赖性属性和例程事件。本章包括了使用大量 WPF 控件;而且,本章还将解释布局管理程序、控件命令行和 WPF 数据绑定模型的使用。

第 32 章: WPF 二维图形渲染、资源和主题

本部分的最后一章将通过分析 3 个表面上相互独立的主题来介绍 WPF。然而,WPF 的图形渲染服务通常要求定义自定义资源。使用这些资源,可以生成自定义 WPF 动画,以及使用

图形、资源和动画，还可以为 WPF 应用程序构建自定义主题。为了将所有这些主题结合在一起，本章最后说明了如何在运行时应用自定义图形主题。

第 VIII 部分：用 ASP.NET 构建 Web 应用程序

本书的最后一部分致力于用 ASP.NET 构造 Web 应用程序。ASP.NET 被有意识地设计成通过在标准 HTTP 请求/响应上放置一个事件驱动的且面向对象的架构来为桌面用户界面创建建模。

第 33 章：构建 ASP.NET Web 页面

本章首先研究在使用 ASP.NET 的 .NET 平台下支持的 Web 技术。服务器端脚本代码现在用“真正的”面向对象语言(如 VB 2008、C#等)替代了。本章将介绍关键 ASP.NET 主题，如使用(或不使用)code-behind 文件、ASP.NET 3.5 目录结构以及 web.config 文件的作用。

第 34 章：ASP.NET Web 控件、主题和 Master 页面

本章将深入挖掘 ASP.NET Web 控件的详细信息。一旦理解了这些 Web 窗口小部件的基本功能，然后就会用各种各样的 .NET 3.5 功能(Master 页面、*.sitemap 文件、主题和皮肤)构建一个简单然而有说明性的 Web 站点。另外，本章将介绍有效性验证器控件和增强的数据绑定引擎的使用。

第 35 章：ASP.NET 状态管理技术

本章将介绍在 .NET 下处理状态管理的各种方式，从而扩展对当前 ASP.NET 的理解。像经典 ASP 一样，ASP.NET 允许轻松地创建 cookies，以及应用程序级和会话级变量。一旦学习了使用 ASP.NET 处理状态的各种方式，然后就开始学习 System.HttpApplication 基类(潜伏在 Global.asax 文件中)的作用。最后我们介绍 ASP.NET 配置文件管理 API。

获取本章的源代码

本书包含的所有代码示例(散布在全书中的小块代码)都可以免费使用，并可以立即从 Apress Web 站点的 Source Code 区域下载。简单地导航到 <http://www.apress.com>，选择 Source Code/Download 链接，并按照书名查找本书。一旦找到 *Pro VB 2008 and the .NET 3.5 Platform* 的页面，就可以下载一个自解压的*.zip 文件。当解压缩内容时，将发现代码已经按照各章逻辑性划分了。此外，示例代码还可以从 www.tupwk.com.cn/downpage 下载。

注意，各章中像下面这样的“源代码”指示正在讨论的示例的线索，可以将它们加载到 Visual Studio 2008 中以便进一步分析和修改。

源代码：

这是将您指向特定目录的源代码通知！

要做到这一点，简单地打开在正确的子目录中找到的*.sln 文件。如果使用的不是 Visual

Studio 2008(参见第 2 章的其他 IDE)，可以将提供的源代码手动加载到所选择的开发工具中。

获取本书的更新

当您阅读本书时，可能发现一个偶然的图形化或代码错误(虽然笔者很希望没有)。如果出现这种情况，笔者表示抱歉。作为一个人，尽管笔者尽了努力，但仍然可能存在一些小错误。如果有这种情况，可以从 Apress Web 站点中获得本书的勘误表(也位于本书的主页上)，以及关于如何将您发现的任何错误通知笔者的信息。

联系作者

如果您对关于本书的源代码有任何问题，需要弄明白给定的示例，或者只是希望提供您关于.NET 的想法，请给作者发电子邮件(为了确保您的邮件不进入垃圾邮件文件夹，请在主题中包括“VB TE”字样)：atroelsen@intertech.com 或 wkservice@vip.163.com。

总之，感谢您购买本书(或者至少在书店浏览本书以决定是否购买)。希望您喜欢阅读本书，并充分利用您新学到的知识。

目 录

第 I 部分 Visual Basic 2008 与 .NET 平台

第 1 章 .NET 基本原理	3
1.1 历史状态	3
1.1.1 C/Win32 API 程序员的生活	3
1.1.2 C++/MFC 程序员的生活	4
1.1.3 Visual Basic 6.0 程序员的生活	4
1.1.4 Java/J2EE 程序员的生活	4
1.1.5 COM 程序员的生活	5
1.1.6 Windows DNA 程序员的生活	5
1.2 .NET 解决方案	6
1.3 .NET 平台的构件块 (CLR、CTS 与 CLS)	6
1.4 Visual Basic 2008 的新内容	7
1.5 其他 .NET 支持语言	8
1.6 .NET 程序集概览	10
1.6.1 单文件与多文件程序集	11
1.6.2 公共中间语言的作用	11
1.6.3 .NET 类型元数据的作用	14
1.6.4 程序集清单的作用	15
1.7 公共类型系统	15
1.7.1 CTS 类类型	16
1.7.2 CTS 接口类型	16
1.7.3 CTS 结构类型	16
1.7.4 CTS 枚举类型	17
1.7.5 CTS 委托类型	17
1.7.6 CTS 类型成员	17
1.7.7 固有 CTS 数据类型	18
1.8 公共语言规范	18
1.9 了解公共语言运行库	20

1.10 程序集/名称空间/类型 之间的区别	21
1.10.1 通过编程访问名称空间	24
1.10.2 引用外部程序集	25
1.11 使用 ildasm.exe	25
1.11.1 查看 CIL 代码	26
1.11.2 查看类型元数据	26
1.11.3 查看程序集元数据 (也称为清单)	27
1.12 使用 Lutz Roeder 的 reflector	27
1.13 部署 .NET 运行库	28
1.14 .NET 的平台无关特性	29
1.15 小结	30

第 2 章 构建 Visual Basic 2008 应用程序	31
2.1 .NET Framework 3.5 SDK 的作用	31
2.2 VB 2008 命令行编译器 (vbc.exe)	32
2.3 使用 vbc.exe 构建 VB 2008 应用程序	32
2.3.1 使用 vbc.exe 引用外部程序集	34
2.3.2 使用 vbc.exe 编译多个源文件	35
2.3.3 使用 vbc.exe 引用多个外部 程序集	36
2.4 使用 vbc.exe 响应文件	36
2.5 使用 SharpDevelop 构建 .NET 应用程序	38
2.6 使用 Visual Basic 2008 Express 构建 .NET 应用程序	41

2.7 使用 Visual Studio 2008 构建 .NET 应用程序	42
2.7.1 Visual Studio 2008 的一些特有功能	42
2.7.2 使用 New Project 对话框选择 .NET Framework 的版本	43
2.7.3 使用 Solution Explorer 实用程序	43
2.7.4 Class View 实用程序	45
2.7.5 Object Browser 实用程序	45
2.7.6 Visual Studio 2008 代码片断技术	45
2.7.7 可视化类设计器	46
2.7.8 Object Test Bench	49
2.7.9 综合帮助系统	50
2.8 Visual Basic 6.0 兼容性程序集的作用	51
2.9 其他 .NET 开发工具的部分目录	52
2.10 小结	53

第 II 部分 核心 VB 编程结构

第 3 章 VB 2008 编程结构(上)

3.1 Module 类型的作用	57
3.1.1 带多个模块的项目	58
3.1.2 模块是不能创建的	59
3.1.3 重命名初始模块	60
3.1.4 模块的成员	60
3.2 Main 方法的作用	61
3.2.1 使用 System.Environment 处理命令行参数	61
3.2.2 使用 Main() 处理命令行参数	62
3.2.3 Main() 作为函数(而不是作为子例程)	62
3.2.4 使用 Visual Studio 2008 指定命令行参数	63
3.3 一个有趣的附加作用: System.Environment 类的一些其他成员	63
3.4 System.Console 类	64

3.4.1 使用 Console 类的基本输入输出	65
3.4.2 格式化控制台输出	65
3.4.3 .NET 字符串格式化标记	66
3.4.4 使用 String.Format() 格式化字符串	67
3.5 系统数据类型与 VB 简洁表示法	67
3.5.1 变量声明和初始化	68
3.5.2 数据类型的默认值	69
3.5.3 数据类型类层次结构	70
3.5.4 固有数据类型和 New 关键字	71
3.5.5 体验数值数据类型	71
3.5.6 System.Boolean 的成员	72
3.5.7 System.Char 的成员	72
3.5.8 分析字符串数据中的值	73
3.6 System.String 类型	73
3.6.1 基本字符串操作	74
3.6.2 字符串连接(以及 Newline 常量)	74
3.6.3 字符串和相等	76
3.6.4 字符串是不能改变的	76
3.6.5 System.Text.StringBuilder 类型	77
3.7 窄化(显式)或宽化(隐式)的数据类型转换	78
3.7.1 Option Strict	80
3.7.2 显式转换函数	82
3.7.3 System.Convert 的作用	83
3.8 构建 Visual Basic 2008 代码语句	83
3.8.1 语句延续字符	83
3.8.2 在单行上定义多个语句	84
3.9 VB 2008 流程控制语句结构	85
3.9.1 If/Then/Else 语句	85
3.9.2 构建复杂条件表达式	86
3.9.3 Select/Case 语句	87
3.10 VB 2008 迭代结构	87
3.10.1 For/Next 循环	88
3.10.2 For/Each 循环	88

3.10.3 Do/While 和 Do/Until 循环结构	89	5.3.2 观察构造函数流程	125
3.10.4 With 结构	90	5.4 Shared 关键字	126
3.11 小结	90	5.4.1 定义共享方法(和字段)	126
第 4 章 VB 2008 编程结构(下)	91	5.4.2 定义共享数据	128
4.1 定义子例程与函数	91	5.4.3 定义共享构造函数	130
4.1.1 ByVal 参数修饰符	92	5.5 定义 OOP 的支柱	131
4.1.2 ByRef 参数修饰符	93	5.5.1 封装的作用	132
4.1.3 定义可选参数	94	5.5.2 继承的作用	132
4.1.4 使用 ParamArrays	95	5.5.3 多态的作用	133
4.1.5 方法调用约定	96	5.6 Visual Basic 2008 访问修饰符	134
4.1.6 包含静态局部变量的方法	97	5.6.1 访问修饰符与嵌套类型	135
4.2 成员重载	98	5.6.2 默认访问修饰符	135
4.2.1 Overloads 关键字	99	5.6.3 访问修饰符与字段数据	136
4.2.2 方法重载的细节	99	5.7 第一个支柱: VB 2008 的 封装服务	136
4.3 VB 2008 中的数组操作	100	5.7.1 使用传统的取值和赋值封装	137
4.3.1 VB 2008 数组初始化语法	101	5.7.2 使用属性封装	138
4.3.2 定义 Objects 的数组	101	5.7.3 属性的内部表示	140
4.3.3 定义数组的下边界	102	5.7.4 控制属性可见性级别的 Get/Set 语句	142
4.3.4 Redim/Preserve 语法	103	5.7.5 只读和只写属性	142
4.3.5 使用多维数组	104	5.7.6 共享属性	142
4.3.6 System.Array 类	105	5.8 常量数据	143
4.4 VB 2008 的枚举	106	5.9 只读字段	145
4.4.1 控制枚举的底层存储	107	5.10 局部类型定义	146
4.4.2 声明和使用枚举	107	5.11 通过 XML 文档化 VB 2008 源代码	147
4.4.3 System.Enum 类型(以及解析关 键字命名冲突的一个教训)	108	5.12 使我们的概念形象化	151
4.4.4 动态地发现枚举的名/值对	110	5.13 小结	151
4.5 VB 2008 结构类型	111	第 6 章 继承与多态	152
4.6 小结	113	6.1 继承的基本机制	152
第 5 章 设计封装类类型	114	6.1.1 Inherits 关键字	153
5.1 VB 2008 类类型简介	114	6.1.2 关于多个基类	154
5.2 类构造函数	117	6.1.3 NotInheritable 关键字	155
5.2.1 默认构造函数的作用	119	6.2 修订 Visual Studio 2008 类图	156
5.2.2 定义自定义构造函数	119	6.3 第二个支柱: 继承的细节	157
5.2.3 再次介绍默认构造函数	120	6.3.1 使用 MyBase 控制基类的创建	159
5.3 Me 关键字的作用	121	6.3.2 Protected 关键字	160
5.3.1 使用 Me 链接构造函数调用	123		

6.3.3 添加密封类	161
6.4 包容/委托模型的编程	162
6.5 第三个支柱: VB 2008 的多态支持	165
6.5.1 Overridable 和 Overrides 关键字	166
6.5.2 用 Visual Studio 2008 进行重写	167
6.5.3 NotOverridable 关键字	168
6.5.4 抽象类和 MustInherit 关键字	169
6.5.5 用 MustOverride 构建一个多态接口	170
6.5.6 成员屏蔽	173
6.6 基类/派生类的类型强制转换规则	174
6.7 终极父类: System.Object	176
6.7.1 重写 System.Object.ToString()	179
6.7.2 重写 System.Object.Equals()	179
6.7.3 重写 System.Object.GetHashCode()	180
6.7.4 测试修改后的 Person 类	181
6.7.5 System.Object 的共享成员	181
6.8 小结	182
第 7 章 结构化异常处理	183
7.1 关于错误、bug 和异常	183
7.2 .NET 异常处理的作用	184
7.2.1 .NET 异常处理的原子性	184
7.2.2 System.Exception 基类	185
7.3 最简单的示例	186
7.3.1 抛出一个简单的异常	187
7.3.2 捕获异常	188
7.4 配置异常的状态	189
7.4.1 TargetSite 属性	189
7.4.2 StackTrace 属性	190
7.4.3 HelpLink 属性	190
7.4.4 Data 属性	191
7.5 系统级异常 (System.SystemException)	192

7.6 应用程序级异常 (System.ApplicationException)	193
7.6.1 构建自定义异常(I)	193
7.6.2 构建自定义异常(II)	195
7.6.3 构建自定义异常(III)	195
7.6.4 按 Visual Studio 方式构建自定义异常	196
7.7 处理多个异常	196
7.7.1 广义 Catch 语句	198
7.7.2 重新抛出异常	199
7.7.3 内部异常	199
7.8 Finally 块	200
7.9 谁在抛出什么	201
7.10 未处理异常的结果	202
7.11 使用 Visual Studio 2008 调试未处理异常	202
7.12 融合 VB6 错误处理与结构化异常处理	203
7.13 小结	204

第 8 章 对象生存期	205
8.1 类、对象和引用	205
8.2 对象生存期基础	206
8.2.1 CIL 的 New 关键字	207
8.2.2 将对象引用设置成 Nothing	208
8.3 应用程序根的作用	209
8.4 对象代	210
8.5 System.GC 类型	210
8.6 构建可终结对象	213
8.6.1 重写 System.Object.Finalize()	214
8.6.2 详细介绍终结过程	215
8.7 构建可诊断对象	216
8.8 构建可终结和可诊断类型	218
8.9 小结	220

第 III 部分 高级 VB 编程结构

第 9 章 使用接口类型	223
9.1 接口类型	223
9.2 定义自定义接口	225