

Pro C# 2010 and the .NET 4 Platform **Fifth Edition**

C#与.NET 4 高级程序设计 (第5版)

“C#圣经”
最新版

[美] Andrew Troelsen 著
朱 晔 肖 逵 姚琪琳 等译
张大磊 王少葵 范 睿



- Amazon超级畅销书,
- 全面涵盖C# 2010, 用
- 多位微软MVP联手翻译, 名著佳译相得益彰

TURING 图灵程序设计丛书

Pro C# 2010 and the .NET 4 Platform **Fifth Edition**

C#与.NET 4 高级程序设计 (第5版)

[美] Andrew Troelsen 著
朱 晔 肖 遼 姚琪琳 等译
张大磊 王少葵 范 睿

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C#与.NET 4高级程序设计 : 第5版 / (美) 特罗尔森 (Troelsen, A.) 著 ; 朱晔等译. — 北京 : 人民邮电出版社, 2011.4

(图灵程序设计丛书)

书名原文: Pro C# 2010 and the .NET 4 Platform, Fifth Edition

ISBN 978-7-115-25032-2

I. ①C… II. ①特… ②朱… III. ①C语言—程序设计②计算机网络—程序设计 IV. ①TP312②TP393.09

中国版本图书馆CIP数据核字 (2011) 第036929号

内 容 提 要

本书是C#领域久负盛名的经典著作, 深入全面地叙述了C#编程语言和.NET平台核心, 并以大量示例剖析相关概念。书中介绍了C#的各种语言构造、.NET 2.0的类、核心API、公共中间语言(CIL)、动态程序集和ASP.NET扩展等内容; 同时也介绍了.NET 3.0、.NET 3.5和.NET 4中的新的编程API, 包括WPF、WCF和WF的功能; 另外, 还介绍了最新的C# 2010编程语言、DLR、TPL、LINQ编程技术、COM与.NET的互操作性以及平台无关的.NET开发等。

本书由微软C# MVP Andrew Troelsen编写, 历经多次修订, 适合各层次.NET开发人员阅读。

图灵程序设计丛书

C#与.NET 4高级程序设计 (第5版)

- ◆ 著 [美] Andrew Troelsen
译 朱晔 肖遼 姚琪琳 张大磊 王少葵 范睿 等
责任编辑 王军花
执行编辑 丁晓昀
- ◆ 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 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 76.25
字数: 2148千字 2011年4月第1版
印数: 1-4 000册 2011年4月北京第1次印刷
著作权合同登记号 图字: 01-2010-4741号

ISBN 978-7-115-25032-2

定价: 149.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Pro C# 2010 and the .NET 4 Platform, Fifth Edition* by Andrew Troelsen, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2010 by Andrew Troelsen. Simplified Chinese-language edition copyright © 2011 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

致 谢

由于某个（或多个）疯狂的原因，这一版写起来比我预想的要困难得多。如果没有大家的帮助和支持，也许到.NET 6.0 Beta 1 发布的时候，本书才会问世。

非常感谢本书的技术编辑 Andy Olsen。他不仅指出了一个分号的错误，而且还对原始代码示例的改进提出了大量的建议。多谢 Andy！

更要感谢本书的文字编辑团队——Mary Behr、Patrick Meader、Katie Stence 以及 Sharon Terdeman。和他们工作非常有趣，并且他们还能够发现和纠正一些我根本没有注意到的语法错误。

特别感谢 Apress 出版社的 Debra Kelly 容忍我稿件逾期未交、错发邮件和反复修改。这是我们合作的第一个项目，我真诚地希望以后还能和她合作。谢谢你，Debra！

最后感谢我的妻子 Mandy，她总是能让我在写作时保持清醒。谢谢你，亲爱的。

对上一版的赞誉

“本书的前几版都取得了极大成功，新版当然也不会例外。”

——Peter Bromberg，微软 C# MVP

“学习 C# 的朋友，我推荐你准备一本 *Pro C# 2008 and the .NET 3.5 Platform, Fourth Edition*。”

——Suprotim Agarwal，微软 MVP

“请大家推荐一本 .NET Framework 方面的书！谢谢！”

“*Pro C# 2008 and the .NET 3.5 Platform, Fourth Edition*。”

——CSDN 论坛

“我购买了这本书，它大大加速了我的 .NET 3.5 学习过程。这本书深入研究 ASP.NET、ADO.NET 以及 LINQ，并对 WCF 和 WF 做了恰到好处的介绍。从 XAML 到 WPF 控件的使用，最后是 2D 图形、资源和主题。它清晰地展现了 C# 3.0 和 .NET 3.5 的增强特性。”

——Amazon.com 读者评论

“可以说这本书是 C# 最经典的书之一。”

——豆瓣

“Andrew Troelsen 绝对值得信赖。”

——Milan Negovan，微软 ASP.NET MVP，ASP.NET Resources 网站创始人

“这是一本不仅应该拥有，而且应该放在键盘旁边随时翻查的 C# 和 .NET 图书。”

——Dr. Dobb's Journal

“个人感觉本书是目前国内最好的 C# 书，没有之一，红皮书我也看过，比本书要差一个档次。本书最大的特点就是细致，几乎每个知识点都有可运行的 DEMO，翻译也很用心，基本没有西方人惯常的长句子，比较流畅，看着不累。”

——china-pub.com 读者评论

“本书极为全面、细致深入地探讨了 C# 与 .NET 2.0 框架的各种特性。其中对 CIL 的介绍和运用尤为精彩，超过了任何其他文章和图书。绝对值得拥有。”

——Slashdot

“好书！内容恰到好处，是所有 C# 程序员的必由之路。”

——www.shanedeवानe.net

“本书是目前市场上学习 C# 的最佳图书。”

——csharp-online.net

“本书非常精彩，清晰透彻，覆盖面很广，且没有牺牲任何深度。虽然主要针对有经验的开发人员，但本书对于各层次程序员都很有价值，这在同类图书中绝无仅有。文字和主题的过渡都非常流畅，强烈推荐。”

——澳大利亚 Perth .NET 用户组

“本书的第 1 版面世不久，就很快成为我的必备参考书。与那些由一大群人合写的大部头不同，本书的作者只有一人，这种思路和语言的连续性对于编程书是至关重要的。最令人印象深刻的是，作者通过示意图和实例，非常到位而且易懂地解释了语言核心。”

——wagnerblog.com

“绝佳的参考，极为有用，所有 .NET 程序员都应该拥有。”

——John Glove，资深软件工程师

“本书的行文达到了非常高的水平，作者善于将非常难的主题讲解得极为明白易懂。对于有一定编程经验的开发人员甚至学生，本书绝对物超所值。”

——ACM Computing Reviews 在线杂志

“杰作！通过 IL 代码透视各种 C# 语法背后的本质，是学习一门语言的最佳方式。而且，作者还提供了大量从 C# 的角度理解 .NET 的深入见解。书中的许多实用建议弥足珍贵。”

——Chiradeep Chhaya，资深分析师

“最佳 C# 图书！为了学习 C#，在购买此书之前，我已经读了 3 本书，它们不仅没有解决任何问题，反而使我更加糊涂了。读了本书之后，所有的困惑都烟消云散，作者的阐述非常透彻详尽。”

——Amazon.com 读者评论

“有了本书之后，其他 C# 和 .NET 图书几乎都不再需要了。我已经预定了它的下一版本。”

——Dennis L. Hughes，资深 Visual C++ 程序员

“这是我极少数需要常常参考的书之一。强烈推荐给中高级 .NET 程序员，它不仅仅讲述了 C# 语言，而且及时而清晰透彻地阐释了 .NET。”

——Charles Wright，C# Programming Tips & Techniques 的作者

译者序

应用程序开发的目标始终如一，就是在最短时间内制作出最好的软件。当今最流行的快速开发平台是什么，毫无疑问，是微软公司的 .NET Framework。经过近 10 年的发展，微软的 .NET 已经变得非常庞大和成熟，而且 .NET 发展的步伐越来越快。

2002 年年初，微软发布了 Visual Studio .NET 2002 开发工具，也带来了 .NET 框架的第一个版本 1.0。2003 年春天，微软发布了 Visual Studio .NET 2003 以及 .NET Framework 1.1。尽管只是 1.1，但是 .NET Framework 1.1 比 1.0 多了几项重大更新，例如对移动设备的支持（精简版 .NET）和对 ODBC/Oracle 数据库的支持。2005 年年底，.NET Framework 2.0 随着 Visual Studio 2005 以及 SQL Server 2005 一起发布。无论是 Visual Studio 2005、C# 2.0 还是 ASP.NET 2.0，在易用性、安全性等方面都比前辈优秀很多。C# 2.0 中的泛型、迭代器、可空类型、匿名方法、分部类等特性确实给了我们很多方便。2006 年年底，微软随 Vista 操作系统推出了 .NET Framework 3.0。NET Framework 3.0 在 .NET Framework 2.0 的基础上增加了几个重要组件：WCF、WPF、WF 和 WCS（本书后面会对 WCF、WPF 和 WF 进行比较详细的介绍）。

- WCF: Windows Communication Foundation，最完整的集成化通信管理框架。WCF 编程模型把 Web 服务、.NET 远程技术、分布式事务和消息队列统一到单个面向服务的编程模型中，从而实现了真正意义上的分布式计算。
- WPF: Windows Presentation Foundation，最华丽的界面技术框架。WPF 提供了一种一致的方案来构建编程模型，并且支持使用更为丰富的控件和设计技术来开发 Windows 程序。
- WF: Windows Workflow Foundation，最灵活和最强大的工作流平台。WF 是微软用来定义、执行和管理工作流的编程模型、引擎和工具的总称。
- WCS: Windows CardSpace，最安全的个人数字标识解决方案。

2007 年年底，微软发布了 Visual Studio 2008 以及 .NET Framework 3.5。在 .NET Framework 3.0 的基础上，3.5 又新增了一些程序集，并引入了 LINQ、ASP.NET AJAX 等功能。

- C# 3.0: 提供了隐式类型变量、自动属性、扩展方法、对象初始化语法、Lambda 表达式、匿名方法、分部方法等新特性，其实几乎所有的特性都是因 LINQ 而生的。
- LINQ: 语言集成查询，官方组件包括 LINQ to SQL、LINQ to XML、LINQ to DataSet 等。当然，还有很多第三方公司或个人推出的 LINQ to XXX。可以通过这些组件使用统一的查询语言对各种基于磁盘、网络甚至服务的数据源进行查询。
- ASP.NET AJAX: 其实 ASP.NET AJAX 最早并不是在 .NET Framework 3.5 中出现的，只不过 .NET Framework 3.5 集成了 ASP.NET AJAX。
- 关于 .NET Framework 3.0 中四大技术的扩展，我们也可以认为 .NET Framework 3.5 提供了 WCF/WPF/WF/WCS 的 2.0 版本。

2010年4月,微软又发布了 Visual Studio 2010 和 .NET Framework 4, 这又是一个里程碑式的版本。如果说 C# 3.0 中的大多数特性都是语法糖, 那么 C# 2010 则建立在全新的 CLR 基础之上。

- DLR: Dynamic Language Runtime (动态语言运行时) 是 .NET 4 的重要组成部分, 它为 C# 这门静态强类型语言提供了类似 Smalltalk、LISP、Ruby 和 Python 那样的动态特性。
- TPL: 在多核计算机早已普及的今天, 充分利用每一个 CPU 可以显著提升程序的性能, 是任何一个编程语言都不能忽视的功能。Task Parallel Library (任务并行库) 应景而生, 你不必操作线程, 就可以编写细粒度、高可扩展性的并行程序。使用 PLINQ 还可以并行地进行 LINQ 查询。
- 改善 COM 互操作性: 动态特性、可选参数、命名参数、索引属性以及 PIA 的诞生, 都大大简化了与传统 COM 组件的互操作。
- WCF、WPF 和 WF 都升级到了 4.0 版本。

本书分 8 个部分, 从最底层的 CLR 开始, 到上层 C# 语言的各种结构, 再到 .NET Framework 中提供的各种 API, 一直到应用层面的 WCF、WF、WPF、ASP.NET 等, 甚至是 .NET 的一些互操作性都进行了全面的介绍。

感谢本书的作者 Andrew Troelsen 先生, 他为我们带来这样一本含大量精彩篇章的著作。本书是在前几个版本的中文版基础上翻译完成的。第三版主要由王少葵、张大磊、朱晔、范睿和金雪根(思归)翻译, 第四版的补充翻译由朱晔、肖逵和张大磊完成, 本版的补充翻译由姚琪琳完成。特别感谢图灵公司编辑王军花、丁晓昀等的辛勤劳动。另外也感谢很多在翻译过程中给过我们支持的朋友和网友, 请原谅不能一一列出他们的名字。

真诚地希望我们为这本书付出的劳动没有辜负大家的期望, 希望各位读者朋友在阅读后能够对 .NET Framework 和 C# 更了解。由于时间仓促, 且译者水平有限, 书中难免会出现一些问题或疏漏, 请各位读者不吝赐教, 我们非常感谢读者的任何批评和反馈, 并会基于此作出更正与改进。谢谢!

前言

本书历经了多个版本，第1版大约在2001年夏天几乎与.NET 1.0 Beta 2同时推出。从那以后，我一直感到非常幸福，心中充满了感激之情，因为我的书得到了出版社尤其是读者们的好评。这些年来，它曾入围 Jolt 大奖提名（可惜最后没得奖……）和 2003 年 Referenceware 编程类图书卓越大奖（这次我赢了？太酷了！）。

更重要的是，很高兴收到全世界读者发来的邮件。他们告诉我本书对他们的事业有一定的帮助，与他们交谈的感觉太棒了。事实上，本书比过去的任何版本都要好。感谢那些发来改进建议、指出排印错别以及其他错误的读者。

一些大学曾经或正在使用本书，并且本书也是计算机科学领域众多本科和研究生的必读书，对此，我受宠若惊。

对于出版商、读者和教授们，我只能说谢谢并祝你编程愉快！

无论如何，自从本书第1版面市，我一直努力工作，以使本书内容和.NET 平台的版本保持一致。现在你手中拿的是第5版，其中涵盖了 C# 2010 编程语言和.NET 4 平台的新特性，包括 DLR (Dynamic Language Runtime, 动态语言运行时)、TPL (Task Parallel Library, 任务并行库)、PLINQ (Parallel LINQ, 并行 LINQ)、ADO.NET Entity Framework (EF, 实体框架)，以及一些细微的（但很有用的）更新，如 C# 2010 命名参数、C# 2010 可选参数、`Lazy<T>` 类类型等。

除了涵盖所有这些新特性外，本书还为你打造了坚实的 C# 语言基础，介绍了面向对象编程 (OOP) 的核心、程序集配置、数据库访问 (ADO.NET)，以及构建桌面 GUI 应用程序、Web 应用程序和（在其他话题中穿插介绍的）分布式系统的过程。

与前几个版本一样，这一版用友好且平和的方式描述了 C# 编程语言和.NET 基础类库。同样，新版本继续重点阐述构建软件解决方案所需的内容，而不会在深奥且很少有人关心的内容上浪费时间。

你和我，我们是一个团队

技术作家所面对的是一群苛刻的读者（我知道，因为我就是他们中的一员）。无论使用什么平台（例如.NET、Java 和 COM），对部门、公司、客户和任何课题来说，构建软件解决方案都是非常复杂而且有针对性的事情。可能你在电子出版行业工作，或者为政府开发系统，或者是在科研机构或军队的某个部门工作。就我自己而言，我开发过儿童教育软件（Oregon Trail、Amazon Trail 等游戏软件）、各种 n 层系统以及许多医疗和金融行业的项目。你工作时编写的代码和我编写的代码百分之百是不同的（除非我们恰巧以前在一起工作）。

因此，在这本书中，我特意避免选择那些和具体行业紧密相关的例子，而是用与行业无关的例子来解释 C#、OOP、CLR 和.NET 4 基础类库。我不使用诸如数据填充表格、薪水计算或者其他的一些例子，而是坚持用与我们都有联系的主题：汽车，另外再加上几何结构和雇员薪水系统作为补充示例。你不用

担心会有什么陌生的背景知识。

我要做的是尽最大可能解释 C#编程语言和.NET 平台的核心内容。同时,我会尽可能把进一步学习本书的工具和策略提供给你。

你要做的是理解这些内容并将其付诸于具体编程工作中。我很清楚,你的项目可能与具有友好昵称的汽车(比如,BMW 的 Zippy 和 Yugo 的 Clunker)根本无关,但是所用到的知识是相通的。

放心,只要理解了这本书中的概念,你便能够很好地构建一个和实际情况紧密相关的.NET 解决方案了。

本书内容

本书从逻辑上分为 8 个部分,每个部分包含一些相关联的章节。如果读过本书的前几版,你很快就会发现很多变化。例如,我不再用一章单独介绍 C#语言的新特性,而是将这些新特性自然融入相关的章节。此外,根据读者的反馈,我扩展了 WPF 相关的内容。下面先按部分,再按章来分解本书。

第一部分: C#与.NET 平台

第一部分的目的在于让你初步适应并了解.NET 平台以及在构造.NET 应用中用到的各种开发工具(其中很多是开源的)。同时,你也会学习到 C#编程语言和.NET 类型系统的基本知识。

第 1 章: .NET 之道

这一章讲述本书其余部分的脉络。我们将首先回顾传统 Windows 开发领域,并阐述它们的缺点。然而,该章的主要目的是介绍许多.NET 相关的构建块,如 CLR (公共语言运行库)、CTS (公共类型系统)、CLS (公共语言规范)以及基础类库。该章让你对 C#编程语言和.NET 程序集格式有一个初步了解,同时概述了.NET 平台的平台无关性(附录 B 将会详细介绍这个主题)。

第 2 章: 构建 C#应用程序

这一章介绍使用各种工具和技术来编译 C#源代码文件。先介绍了如何使用命令行编译器(csc.exe)和 C#响应文件,接着介绍了许多代码编辑器和 IDE (集成开发环境),包括 Notepad++、SharpDevelop、Visual C# 2008 Express 以及 Visual Studio 2010。同时也会介绍了如何通过在本机安装.NET Framework 4 SDK 文档来配置自己开发用的电脑。

第二部分: C#核心编程结构

这部分很重要,因为所有类型的.NET 软件开发都必须用到它,如 Web 应用、GUI 桌面应用、代码库和 Windows 服务。你将会在这里了解到 C#语言的核心结构,包括 OOP 的详情。同时,这部分还分析了如何处理运行时异常,并详细介绍了.NET 的垃圾回收服务。

第 3 章: C#核心编程结构 I

这一章正式开始研究 C#编程语言,其中介绍了 Main()方法的作用和.NET 平台的内部数据类型,以及使用了 System.String 和 System.Text.StringBuilder 的文本数据操作。本章还讨论了迭代和选择构造、宽化和窄化操作以及 unchecked 关键字的用法。

第 4 章: C#核心编程结构 II

这一章完成了 C#核心方面的研究,首先介绍重载类型方法的结构以及如何通过 out、ref 和 params 关键字定义参数,接着介绍了 C# 2010 的两个新特性——参数和可选参数,然后介绍了如何创建和操作数据数组,定义可空数据类型(使用?和??操作符)以及值类型(包括枚举和自定义结构)

和引用类型之间的区别。

第 5 章：定义封装的类类型

从这一章开始，我们会研究如何使用 C# 编程语言进行面向对象编程（OOP）。在探讨了 OOP 的支柱概念（封装、继承和多态）之后，会介绍如何使用构造函数、属性、静态成员、常量以及只读字段来构建健壮的类类型。我们最后会研究分部类型定义、初始化对象的语法和自动属性。

第 6 章：继承和多态

这一章研究 OOP 其余的支柱概念（继承和多态），通过它，我们就能构建相关类类型的家族。我们还会研究虚方法、抽象方法（和抽象基类）以及多态接口的本质。这一章最后介绍 .NET 平台最高层基类 `System.Object` 的作用。

第 7 章：结构化异常处理

这一章的关键在于讨论如何使用结构化异常处理来处理运行时的异常情况。你不但可以学到 C# 控制这些异常的关键字（`try`、`catch`、`throw` 和 `finally`），还将了解应用程序级异常和系统级异常的区别。另外，该章还讲述了 Visual Studio 2010 中不同的调试工具，这些工具能让你调试那些被忽略的异常。

第 8 章：对象的生命周期

这一章分析了 CLR 如何使用 .NET 垃圾收集器来管理内存。你将会了解应用程序根、对象产生和 `System.GC` 类型的作用。理解了这些基础知识之后，剩下的内容将描述可处置的对象（通过 `IDisposable` 接口）和终结过程（通过 `System.Object.Finalize()` 方法）。这一章还将研究 .NET 4 的一个新类 `Lazy<T>`，它用来定义一个直到调用者发出请求时才会分配的数据。你将看到，当你的程序不希望托管堆散乱不堪时，这个特性将有助于确保堆的整洁。

第三部分：高级 C# 编程结构

这部分介绍了许多重要的高级技术，能让你对 C# 语言有更深入的理解。通过对接口和委托的学习，你将会了解 .NET 类型系统。同时你会学到泛型的作用和 C# 的各种高级特性（例如扩展方法、分部方法和指针操作），并对 LINQ 有初步的了解。

第 9 章：接口

这一章的内容建立在对面向对象开发的理解之上，涵盖了基于接口的编程问题。你将学到如何定义类和支持多行为的结构，如何在运行时发现这些行为，以及如何使用显式接口实现来选择性地隐藏特定的行为。除了创建大量自定义接口外，还介绍了如何实现 .NET 平台中的标准接口，以及使用这些接口构建可以排序、复制、枚举和比较的对象。

第 10 章：泛型

这一章首先研究泛型（generic）的概念。你将看到，泛型编程提供了一种创建类型和类型成员的方式，它包含由调用者指定的变量占位符。简而言之，泛型编程大大加强了应用程序的性能和类型安全。你不仅可以在 `System.Collections.Generic` 命名空间中看到各种泛型，而且可以学习如何创建自己的泛型方法和类型（有限制或没有限制）。

第 11 章：委托、事件和 Lambda

这一章的目的在于阐明委托（delegate）类型。可以简单地认为，一个 .NET 委托就是指向应用程序中其他方法的一个对象。使用这个类型，可以构建允许多个对象进行双向会话的系统。在分析了 .NET 委托的使用后，将介绍 C# 的 `event` 关键字，使用这个关键字可以简化原始委托编程的操作。随后研究 C# 3.0 Lambda 操作符 `=>`，并探讨了委托、匿名方法和 Lambda 表达式之间的联系。

第 12 章：高级 C# 语言特性

这一章介绍了许多高级的 .NET 编程技术，这些技术能够让你对 C# 编程语言有更深入的理解。例如，你将学到如何重载操作符、创建自定义类型转换例程（包括隐式的和显式的）、构建类型索引器并与之交互，使用扩展方法、匿名类型、分部方法以及使用不安全的代码上下文来操作 C# 指针。

第 13 章：LINQ to Object

这一章开始研究 LINQ（语言集成查询），它可以用来构建强类型的查询表达式，并且可以把它应用到很多 LINQ 目标来操作最广义的数据。我们还会学习把 LINQ 表达式应用到数据容器（例如，数组、集合和自定义类型）的 LINQ to Object。这对于本书其余部分所涉及的其他 LINQ API 同样适用（如 LINQ to XML、LINQ to DataSet、PLINQ 和 LINQ to Entities）。

第四部分：用 .NET 程序集编程

这一部分深入分析了 .NET 程序集格式的细节。你不仅会学到如何部署和配置 .NET 代码库，而且会理解 .NET 二进制映像的内部结构。这部分也阐述了 .NET 特性的作用和运行时解析类型信息的作用，还介绍了 .NET 4 动态语言运行时（DLR, Dynamic Language Runtime）和 C# 2010 中 `dynamic` 关键字的作用。后面的章节分析了一些较高层次的细节（如对象上下文、CIL 代码和在内存中构建程序集）。

第 14 章：.NET 程序集入门

从一个比较高的层次来看，程序集是用于描述托管的 *.dll 或 *.exe 二进制文件的一个术语，但是其真正内涵实际上远远不仅于此。通过这一章，你将学到单文件程序集和多文件程序集的区别，以及如何构建和部署每一个实体；你还会学到如何利用基于 XML 的 *.config 文件和发布策略程序集来配置私有的和共享的程序集。通过这些内容，你将看到 GAC（全局程序集缓存）的内部结构，其中包括 .NET 4 中对 GAC 的一些修改。

第 15 章：类型反射、晚期绑定和基于特性的编程

这一章通过 `System.Reflection` 命名空间，分析了运行时类型发现的过程，来继续探讨 .NET 程序集。使用这些类型，你能够创建一个可以实时读取程序集元数据的应用程序。该章还将介绍如何在运行时使用晚期绑定来动态加载和创建类型。最后的主题是 .NET 特性（包括标准的和自定义的）的作用。为了说明每个主题的用法，该章最后将构造一个可扩展的 Windows Forms 应用程序。

第 16 章：进程、应用程序域和对象上下文

既然你已经对程序集有了一定的了解，这一章将深入探讨加载的 .NET 执行单元的组成。这一章的目标是阐明进程、应用程序域和上下文边界的关系。该章所叙述的内容为第 19 章做了很好的铺垫，下一章我们将讨论多线程应用程序的构造。

第 17 章：CIL 和动态程序集的作用

这一章有两个目的。在前半部分（大致）中，将会比之前章节更具体地介绍 CIL 的语法和语义，余下的部分主要讲述 `System.Reflection.Emit` 命名空间的作用。使用这些类型，可以构建一个能在运行时在内存中产生 .NET 程序集的软件。正式地说，一个能在内存中定义并执行的程序集称为动态程序集。不要将这个概念与第 18 章中的动态类型混淆。

第 18 章：动态类型和动态语言运行时

.NET 4 引入了一个新的 .NET 运行时环境，叫做动态语言运行时。使用 DLR 和 C# 2010 `dynamic` 关键字，可以定义直到运行时才真正处理的数据。这些特性可以显著地简化一些极其复杂的 .NET 编程任务。在这一章中，你将学习一些动态数据的实际用法，包括如何用简单的方式使用 .NET 反射 API，以及如何用最小的代价与遗留的 COM 库通信。

第五部分：.NET 基础类库

到本书的这一部分，你应该已经很好地掌握了 C# 语言以及 .NET 程序集格式的细节。这一部分将通过探索基础类库中的一些常用服务程序来讲授一些新的知识，包括多线程应用程序的创建、文件的输入/输出和利用 ADO.NET 的数据库访问，通过 WCF 构造分布式应用程序以及构建使用了 WF API 和 LINQ to XML API 的支持工作流的应用程序。

第 19 章：构建多线程应用程序

这一章介绍了如何构建多线程应用程序，演示了大量用于编写线程安全代码的技术。首先复习了 .NET 委托类型，解释了委托对于异步方法调用的内在支持。接下来，研究了 System.Threading 命名空间中的类型。最后，介绍了一个全新的 .NET 4 API——TPL。使用 TPL，.NET 开发者可以用一种极其简单的方式，将应用程序的工作分配给所有可用的 CPU。与此同时，你还将学习 PLINQ 的作用，它提供了一种在多个机器内核中执行 LINQ 查询的方法。

第 20 章：文件输入输出和对象序列化

System.IO 命名空间允许与机器的文件和目录结构交互。通过这一章的学习，你将学会如何以编程方式创建（和删除）一个目录系统，以及如何将数据从不同的（例如，基于文件的、基于字符串的、基于内存的等）数据流中移进移出。本章的后半部分探讨了 .NET 平台的对象序列化服务。简单地说，序列化（serialization）就是将一个对象（或一组相关对象）的状态持久化为流，以便今后使用。反序列化是一个从流中取出对象并放入供应用程序使用的内存中的过程。只要理解了基本原理，你就会学到如何通过 ISerializable 接口和一组特性来定制序列化过程。

第 21 章：ADO.NET 之一：连接层

本书中有 3 章介绍数据库相关操作，这是第一章，介绍 .NET 平台 ADO.NET 的数据库访问 API。具体而言，涉及 .NET 数据提供程序的作用以及如何使用由连接对象、命令对象、事务对象以及数据读取器对象构成的 ADO.NET 连接层与关系数据库进行通信。该章引导你创建一个在本书剩余部分会用到的自定义数据库和自定义数据访问类库（AutoLotDAL.dll）。

第 22 章：ADO.NET 之二：断开连接层

这一章将通过研究 ADO.NET 的断开连接层继续探讨数据库操作。在这里，我们会学习 DataSet 类型、数据适配器对象以及各种可以大幅简化创建数据驱动应用程序的 Visual Studio 2010 工具。此后，我们会学习如何把 DataTable 对象绑定到用户界面元素以及如何使用 LINQ to DataSet 将 LINQ 查询应用于内存对象 DataSet。

第 23 章：ADO.NET 之三：Entity Framework

这一章研究了 ADO.NET Entity Framework (EF) 的作用。实质上，EF 是一种用直接映射到业务模型上的强类型类编写数据访问代码的方式。在这里，你将了解 EF 对象服务的作用、实体客户端和对象上下文，以及 *.edmx 文件的构成。同时，还将学习如何使用 LINQ to Entities 与关系型数据库交互。你还将构建最终版本的自定义数据访问库（AutoLotDAL.dll），本书其他章节中将会使用这个库。

第 24 章：LINQ to XML 简介

第 14 章介绍核心的 LINQ 编程模型，特别是 LINQ to Object。在这里，我们会通过研究如何把 LINQ 查询应用到 XML 文档来深入理解 LINQ。我们首先学习在使用 System.XML.dll 程序集的类型进行 .NET XML 操作时所暴露出的一些“缺点”，然后探索如何使用 LINQ 编程模型（LINQ to XML）在内存中创建 XML 文档、将文档持久化到硬盘驱动以及对其内容进行导航。

第 25 章：WCF

到现在为止，所有的样例程序都是一台计算机上执行的。在这一章中，我们将学习 WCF API 以便

无须考虑底层管道，以统一的方式构建分布式应用程序。该章会介绍WCF服务、服务器端以及客户端的构建。我们还会看到，WCF服务非常灵活，在客户端和服务端中都可以使用基于XML的配置文件以声明方式指定地址、绑定和契约。本章还会介绍大量在.NET 4中非常有用的快捷方式。

第 26 章：Windows Workflow Foundation 4.0 简介

在所有.NET API中，WF（Windows Workflow Foundation）给初学者带来的困惑是最大的。实际上，在.NET 4中，(.NET 3.0中引入的)原始WF API已经被完全改写。你先将学习启用工作流的应用程序的作用，随后你将了解使用.NET 4 WF API对业务流程进行建模的不同方式，接下来，你将学习WF 4.0活动库的范围，以及如何构建自定义活动。在这个活动中，将使用本书前面所创建的自定义数据库访问库。

第六部分：使用 WPF 构建桌面用户界面

.NET 3.0 向程序员介绍了一个绝妙的 API——WPF（Windows Presentation Foundation），该 API 很快就成为 Windows Forms 桌面编程模型的继承者。实质上，WPF 所构建的桌面应用程序可以包含向量图形、交互式动画，以及使用声明式标记语法 XAML 所进行的数据绑定操作。

此外，WPF 控件架构提供了一种非常简单的方式，仅仅使用一些格式良好的 XAML，就可以彻底改变常用控件的外观。我很高兴地告诉各位，这一部分包含 5 章内容，专门介绍 WPF 编程模型。

第 27 章：WPF 和 XAML

本质上，WPF 可以为桌面应用程序（以及非直接的 Web 应用程序）构建交互性极好的富媒体前端。和 Windows Forms 不同，这个超强的 UI 框架把很多关键服务（例如 2D 和 3D 图形、动画、富文档等）整合到了一个统一的对象模型中。在这一章中，我们会从 WPF 以及 XAML（可扩展应用程序标记语言）开始研究，还会学习如何创建不使用 XAML、只使用 XAML 以及两者结合的 WPF 程序。最后，我们创建了在其余 WPF 相关章节中都会用到的自定义 XAML 编辑器。

第 28 章：使用 WPF 控件编程

这一章将介绍使用内置的 WPF 控件和布局管理器的过程，例如，构建菜单系统、拆分窗口、工具条和状态条，还将介绍大量 WPF API（及其相关控件），包括 WPF Documents API、WPF Ink API 和数据绑定模型。同样重要的是，这一章将开始研究 Expression Blend IDE，它将简化为 WPF 应用程序创建富 UI 的任务。

第 29 章：WPF 图形呈现服务

WPF 是一个图形密集型 API，它提供了 3 种呈现图形的方式：形状、绘图和几何图形、可视化。在这一章中，我们将介绍这几种方法，并讨论大量重要的图形基元（如画刷、画笔和图形变换），还将学习 Expression Blend 用于简化创建 WPF 图形过程的多种方式，以及如何对图形数据执行命中测试操作。

第 30 章：WPF 资源、动画和样式

这一章介绍了 3 个重要的（也是相互关联的）主题，它们将加深你对 WPF API 的理解。第一个主题是逻辑资源的作用。你将看到逻辑资源（也叫对象资源）系统提供了一种方式，可以在 WPF 应用程序中命名和引用常用的对象。接下来，你将学习如何定义、执行和控制动画序列。你可能会认为 WPF 动画仅局限于视频游戏和多媒体应用，但实际上绝不是这样。最后你将学习 WPF 样式的作用。与使用 CSS 或 ASP.NET 主题引擎的 Web 页面类似，WPF 应用程序也可以定义常用控件的外观。

第 31 章：WPF 控件模板和用户控件

这一章通过介绍构建自定义控件的过程结束对 WPF 编程模型的研究。先介绍了创建自定义控件

时涉及的两个重要话题：依赖属性和路由事件。理解了这些内容之后，我们将学习默认模板的作用，以及如何在运行时以编程的方式查看它们。打好这些基础之后，最后将学习如何使用 Visual Studio 2010 和 Expression Blend（包括 .NET 4 Visual State Manager）构建自定义的 UserControl 类。

第七部分：使用 ASP.NET 构建 Web 应用程序

这一部分主要研究使用 ASP.NET 编程 API 来构建 Web 应用程序。我们会看到，ASP.NET 基于标准的 HTTP 请求/响应对事件驱动的面向对象框架分层，以此来对桌面用户界面的创建进行建模。

第 32 章：构建 ASP.NET 网页

这一章介绍了使用 ASP.NET 的 Web 应用开发。如你所见，服务器端的脚本代码现在由“真正的”面向对象的语言（如 C# 和 VB.NET 等）所替代。这一章将介绍 ASP.NET 网页的构造、基础的编程模型以及 ASP.NET 的其他关键主题，如 Web 服务器的选择和 Web.config 文件的使用。

第 33 章：ASP.NET Web 控件、母版页和主题

由于前几章介绍了 ASP.NET 页面对象的构建，这一章将会关注组成内部控件树的控件。在这里，我们会研究包括验证控件、内置站点导航控件以及各种数据绑定操作在内的核心 ASP.NET Web 控件。同样，还会演示母版页的作用以及相对于传统样式表的服务器端 ASP.NET 主题引擎。

第 34 章：ASP.NET 状态管理技术

这一章将通过研究 .NET 下处理状态管理的多种方式来扩展你对 ASP.NET 的理解。和传统的 ASP 一样，使用 ASP.NET 可以轻松创建 cookie 以及应用程序级变量和会话级变量。然而，ASP.NET 还引入了一项新的状态管理技术：应用程序高速缓存。只要知道了使用 ASP.NET 处理状态的多种方式，你就会明白 HttpApplication 基类的作用，以及如何用 Web.config 文件动态改变 Web 应用程序的运行时行为。

第八部分：附录

本书最后一部分介绍了两个重要主题，说实话它们不是很适合作为正文，因此就成了附录。在这里我们通过概览 .NET 平台（WF）的初始 GUI 框架来完成对 C# 和 .NET 平台的研究。你还将学习如何使用 Mono 平台在非微软 Windows 操作系统上构建 .NET 应用程序。

附录 A：Windows Forms 编程

.NET 平台最初发布的桌面 GUI 工具包是 Windows Forms。该附录将介绍这个 UI 框架的作用，并演示如何构建主窗口、对话框和菜单系统，还将说明窗体继承的作用，以及如何使用 System.Drawing 命名空间呈现二维图形数据。最后，该附录将构建一个绘图程序（半成品），演示附录中讨论的各个主题。

附录 B：使用 Mono 进行平台无关的 .NET 开发

最后，附录 B 介绍了一个叫做 Mono 的 .NET 平台的开源实现。使用 Mono 可以在 Mac OS X、Solaris、AIX 以及各种 Linux 版本的操作系统中创建、部署和执行富特性的 .NET 应用程序。由于 Mono 和微软的 .NET 平台非常相似，因此你也应该知道 Mono 提供的大部分内容。因此，附录 B 会重点介绍 Mono 的安装过程、Mono 的开发工具以及 Mono 运行时引擎。

本书源代码

本书所包含的所有代码示例都可以从 Apress 网站上的 Source Code/Download 中免费下载（也可以从图灵网站 www.turingbook.com 本书主页上免费注册下载）。访问网址 <http://www.apress.com>，选择

Source Code/Download 链接, 然后按书名查找。找到本书的主页后, 就可以下载一个压缩的*.zip 文件。解开压缩文件就会看到, 所有代码都是按章编排的。

需要提醒你注意的是, 本书中的各章都包含有如下所示的源代码说明, 书中讨论的例子都可以依此线索下载, 并加载到 Visual Studio 2010 中, 以便进一步讨论和修改。

源代码 这里给出了源代码所在的具体目录。

要打开一个 Visual Studio 2010 解决方案, 可以使用 File→Open→Project/Solution 菜单选项, 然后导航到解压缩文档所在的子目录, 找到正确的*.sln 文件。

获取更新信息

在阅读本书时, 你或许会发现一些偶然的语法错误或代码错误 (很显然我不希望看到这些)。如果真发现了, 我在此道歉。作为一个凡人, 尽管我已经很尽力了, 但是一两个小错误总是难免的。你可以从 Apress 的网页上获得勘误表 (还是在这本书的“主页”上)。如果你发现错误的话, 请在那里找到我的联系方式, 与我联系。

与我联系

如果你有任何关于本书源代码的问题, 或者需要进一步阐明这里所举的例子, 亦或者只是想简单地向我传达你关于 .NET 平台的想法, 请通过以下电子邮件地址与我联系: atroelsen@intertech.com (为了确保你的邮件不会被我的信箱划为垃圾邮件, 请在主题栏中包含 “C# FifthEd”)。

请你们相信, 我会尽我所能在较短的时间里回复你们。但是, 就如同各位一样, 我有时也会比较忙。如果我没能在一周或两周的时间里回复你们, 请不要认为我是一个怪异的人或者不屑与你们交流, 我可能只是比较忙而已 (或者, 如果足够幸运的话, 在某处度假也不一定)。

致谢

感谢购买本书 (或者至少在你决定是否购买的时候, 曾经在书店里翻看过)。我希望你喜欢阅读本书, 并且可以灵活运用书中所学的知识。

最后, 感谢大家购买此书 (或者至少也要感谢你在书店里决定是否购买本书时看了它一眼)。我希望你喜欢这本书, 并且能有效利用这些新知识。

Andrew Troelsen