

前 言

记得几年前，我建议Apress出版社出版一本书，介绍一种即将来临的软件SDK，代号NGWS（下一代Windows服务）。你可能已经想到了，NGWS最终成为了我们现在众所周知的.NET平台。从那时起，我对C#编程语言和.NET平台的研究就在这些最初的手稿中开始了。写这本书是一个令人着迷的过程，但是，我必须承认的是，对于这样一项在开发过程中正经历巨大变化的技术，写起来很伤脑筋。值得庆幸的是，在经过了许许多多个不眠之夜后，大约在2001年夏天，针对.NET 1.0 Beta 2的本书第1版出版了。

从那以后，我便感到非常幸福，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！）。

本书的第2版让我有机会能够详述那些关于.NET 1.1平台的内容。虽然第2版也包含了一些新的主题，但还是有部分章节和示例没能付诸出版。

本书的第3版加入了.NET 2.0所带来的大量吸引人的特性（新的C#编程结构、泛型以及核心API的更新等），而且还加入了一些写了很久但一直没有出版的新的内容，比如公共中间语言（CIL）、动态程序集和ASP.NET扩展等。

在这个特别版中，我添加了新的6章来描述.NET 3.0发布所带来的新的编程API。通过这些章节，你将理解Windows Presentation Foundation（WPF）、Windows Communication Foundation（WCF）和Windows Workflow Foundation（WF）的功能。另外，新章节还提供有关即将到来的C# 3.0编程语言和LINQ编程技术（LINQ到SQL和LINQ到XML）的详细描述。C# 3.0和LINQ目前还是beta版，最后两章为即将到来的改变提供了参考路线图。

与前3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基类库。我很不理解一些技术作者为何总是把书写成像GRE词汇表学习指南那样晦涩难懂。同样，新版本继续重点阐述构建软件解决方案所需的内容，而不会在深奥的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需求很高的读者（我应该知道，因为我就是他们的一员）。无论使用什么平台，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具体而且有针对性的事情。可能你工作在电子出版行业，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、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 2.0/3.0基类库。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教给你。

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，你的项目可能与汽车根本无关，但是所用到的知识是相通的。放心，一旦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6个部分，每个部分包含一些相关联的章节。如果读过本书的第3版，你会发现这一版的前25章和第3版是一样的（但是更正了第3版的错误），而第六部分（专门讲述建立.NET 3.0/C# 3.0和LINQ应用程序）是全新的。可以这么说，本书是按照部分、章节详细分类的。

第一部分：C#和.NET 平台简介

第一部分的目的在于让你了解.NET平台的核心、.NET类型系统和各种创建.NET应用的开发工具（其中很多是开源的），由此，你将对C#编程语言的基本情况有所了解。

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开发的模式，并阐述它们的缺陷。然而，该章的主要目的是介绍.NET核心部分的组成，比如公共语言运行库（CLR）、公共类型系统（CTS）、公共语言规范（CLS）以及基类库，让你对C#编程语言和.NET程序集格式有一个初始了解，并且使你领略.NET平台的平台独立性的本质和公共语言基础结构（CLI）的作用。

第2章：构建C#应用程序

这一章介绍使用不同工具和技术来编译和调试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IDE（集成开发环境），包括TextPad、SharpDevelop、Visual C# 2005 Express以及（当然少不了）Visual Studio 2005，你也将接触到许多开源工具（Vil、NAnt、NDoc等），这些开源软件是每一个.NET开发者所必备的。

第二部分：C#编程语言

这一部分探究了C#编程语言所有让人震惊的细节，包括.NET 2.0中新的语法结构。同时，第二部分还介绍了CTS的每一个成员（类、接口、结构、枚举和委托）和泛型结构。

第3章：C#语言基础

这一章分析了C#编程语言的核心结构，包括基本的类构造技术、值类型和引用类型之间的区别、装箱和拆箱，以及大家都喜欢的基类——System.Object的作用。该章也举例说明了.NET平台如何实现很多常见的功能，比如枚举、数组和字符串操作。该章最后分析了.NET 2.0的一些具体内容，比如“可空数据类型”。

第4章：C# 2.0面向对象编程

这一章详细分析了C# 2.0如何解释“OOP的支柱”——封装、继承和多态。一旦了解了构建类层次结构的关键字和语法，你就会看到XML代码注释的作用。

第5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)。

第6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)，而且还将理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 2005中不同的调试工具，这些工具能够让你调试那些被忽略的异常。

第7章：接口与集合

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。你将学到如何定义支持多行为的类型，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。该章剩下的内容分析了System.Collections命名空间，以显示接口类型的实用性。

第8章：回调接口、委托与事件

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。使用这个模式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(包括许多2.0特性，比如匿名方法)的使用后，将介绍C#的event关键字，通过使用这个关键字可以简化原始委托编程的操作。

第9章：高级C#类型构造技术

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，你将学到如何重载运算符、创建自定义类型转换例程(包括隐性的或显性的)、构建类型索引器，以及在*.cs代码文件中操作C风格的指针。

第10章：泛型

从.NET 2.0开始，C#编程语言加强了对CTS中新特性——泛型(generic)的支持。你将看到，泛型编程大大加强了应用程序的性能和类型安全。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)。

第三部分：.NET 程序集编程

第三部分深入分析了.NET程序集格式的细节。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属性的作用和多线程应用程序的结构。后面的章节分析了一些较低层次的细节(如对象上下文)和CIL的语法及语义。

第11章：.NET程序集入门

从一个比较高的层次来看，程序集(assembly)是用于描述托管的*.dll或*.exe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，以及如何创建和部署每一个实体；将学到如何利用基于XML的*.config文件和发布策略程序集来配置私有和共享的程序集。通过这些内容，你将看到全局程序集缓存(GAC)的内部结构和.NET Framework 2.0配置工具的作用。

第12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，从而继续探讨.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激活和操作类型，最后的主题是.NET属性（包括标准的和自定义的）的作用。例如每个主题的用法，该章最后将构造一个可扩展的Windows窗体应用程序。

第13章：进程、应用程序域、上下文和CLR宿主

现在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。第一个目标是阐明进程、应用程序域和上下文边界的关系。一旦界定了这些术语，就能够准确理解CLR本身是如何驻留在Windows操作系统中的，并加深对mscoree.dll的理解。该章所叙述的内容为第14章做了很好的铺垫。

第14章：构建多线程应用程序

这一章叙述了如何构建多线程的应用程序，并且演示了一系列可以用于确保线程代码安全的技术。该章从回顾.NET委托类型开始，以便更好地理解委托在本质上对异步方法调用的支持，之后介绍System.Threading命名空间中的众多类型（Thread、ThreadStart等），以便轻松创建额外的执行线程。

第15章：CIL和动态程序集的作用

这一章有两个目的。在前半（大致）部分中，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。

第四部分：使用.NET 库编程

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程序集格式的细节了。第四部分将通过探索一系列基类库的命名空间来讲授一些新的知识，包括文件的输入/输出、.NET远程处理层、Windows窗体开发和利用ADO.NET的数据库访问。

第16章：System.IO命名空间

如同这个名字所表达的意思一样，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，你将学会如何编程创建（和删除）一个目录系统，以及如何将数据从不同的（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。

第17章：对象序列化

这一章介绍了.NET平台的对象序列化服务。简单地说，序列化（serialization）就是允许你将一个对象（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。一旦理解了基本原理，你就会学到如何通过ISerializable接口和一组.NET 2.0的新特性来自定义序列化过程。

第18章：.NET远程处理层

不同于通俗的理解，XML Web服务不是在.NET平台上建立分布式应用仅有的方法。在这一章中将学习关于.NET远程处理（remoting）层的内容。可以看到，通过使用MBV（值封送）和MBR（引用封送）语义，CLR提供了便于在应用程序和机器之间传输对象的可能。由此，你将学习如何利用XML配置文件来以声明的方式改变分布式.NET应用程序的运行时行为。

第19章：使用System.Windows.Forms构建更好的窗体

这一章由System.Windows.Forms命名空间开始，介绍构造传统的GUI桌面应用程序的细节，这样的应用程序支持菜单系统、工具栏和状态条。如你所希望的，我们将会讨论各种Visual Studio 2005设计

时的特性，以及一些.NET 2.0 Windows窗体类型（MenuStrip、ToolStrip等）。

第20章：使用GDI+绘制图形

这一章介绍了如何在Windows窗体环境中动态呈现图形数据。除了讨论如何处理字体、颜色、几何图像和图像文件，还会讨论单击测试和基于GUI的拖放技术。该章还将介绍新的.NET资源格式，学到这里或许你已经猜到了，它是基于XML数据表现形式的。

第21章：Windows窗体控件编程

这是以Windows为中心的最后一章，将讨论许多.NET Framework 2.0提供的GUI部件。这一章不仅介绍如何依靠各种Windows窗体控件来编程，还将介绍对话框的开发和窗体的继承。另外，这一章还将讲述如何构造集成到IDE中的自定义Windows窗体控件。

第22章：使用ADO.NET访问数据库

ADO.NET是.NET平台上访问数据库的API。你将会看到，可以使用一个连接的和没有连接的层与ADO.NET类型交互。在这一章中，你将有机会使用ADO.NET的这两种模式进行开发，还将学习到几个新的.NET 2.0 ADO.NET的主题，包括数据提供者工厂模式、连接字符串构建器和异步数据库访问。

第五部分：Web 应用程序和 XML Web 服务

第五部分着力于叙述ASP.NET Web应用程序和XML Web服务的构造。在这一部分的前两章中可以看到，ASP.NET 2.0主要从ASP.NET 1.x升级而来，并且包含了许多新的吸引人的内容。

第23章：ASP.NET 2.0网页和Web控件

这一章介绍了在.NET平台下使用ASP.NET的Web技术。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#、VB.NET，诸如此类）所替代。这一章将介绍关键的ASP.NET主题，诸如用（或者不用）代码隐藏文件、ASP.NET Web控件的作用、验证控件以及与ASP.NET 2.0提供的新的“母版页”的交互。

第24章：ASP.NET 2.0 Web应用程序

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。同ASP一样，ASP.NET允许轻松创建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。一旦知道了使用ASP.NET处理状态的多种方式，你就会明白System.HttpApplication基类（位于Global.asax文件中）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25章：XML Web服务

这一章将介绍.NET XML Web服务的作用。简单地说，Web服务就是一个由标准HTTP请求激活的程序集。这一方法的好处在于，HTTP是一个在它接受范围内几乎通用的电报协议，因而它是一个用来建造平台独立和语言独立的分布式系统的极好选择。你同样也会看到一些外围的技术（WSDL、SOAP和UDDI），这些技术使得Web服务和外部客户端能协调通信。

第六部分：.NET 3.0 扩展编程

第六部分专注于介绍.NET 3.0中的新API：WPF、WCF和WF，除此之外，还详细介绍了C# 3.0语言和LINQ编程模型，这些（在写本书时）都是Beta版。

第26章：建立.NET 3.0编程环境

这一章介绍了在构建.NET 3.0相关软件，或探讨C# 3.0和LINQ的开发前，首先需要完成的任务是安装一系列免费下载的SDK和Visual Studio 2005社区预览版模型。这一章详细解释了在进行开发的计算机上安装.NET 3.0/C# 3.0/LINQ的步骤，随后介绍如何修复安装过程中一个危险的bug。

第27章：WPF介绍

WPF (Windows Presentation Foundation)，是一个崭新的构建.NET桌面应用程序的模型。这一章将带你到这个新API的幕后，看看WPF试图解决什么问题，桌面应用中标记（简称XAML）的作用，代码隐藏文件的用法。随后，还将阐述WPF中主要服务（图像呈现、动画等），讨论新的控件编程模型，以及XBAP应用程序的概念。

第28章：WCF介绍

WCF (Windows Communication Foundation)，是专门针对分布式应用程序开发的.NET 3.0特定连接的编程API。在该章中你还将了解到，WCF的主要目标是整合一系列先前独立的API（COM+、MSMQ、.NET远程处理、XML Web服务等），把它们放到一个统一（而且可以扩展）的对象模型中。尽管WCF实际上是一个新的API，但你在第18章中掌握的.NET远程处理知识和在第25章掌握的XML Web服务的知识会有助于你加深对WCF幕后的理解。

第29章：WF介绍

WF (Windows Workflow Foundation)，是另一个.NET 3.0重要组件。这一章首先确切地定义了 workflows的概念以及在编程中什么地方会使用到工作流；之后讲述了一些WF的活动、WF运行库引擎的任务，以及如何利用Visual Studio 2005“Orcas”中的工作流设计器。

第30章：C# 3.0的语言功能

在写本书时，C# 3.0还是测试产品。然而，一旦安装了正确的开发工具（在第26章中描述），你就可以了解所有在.NET主要编程语言的未来版本中可能出现的新的结构。这一章还将介绍隐式类型的局部变量、对象初始化语法、扩展方法、匿名类型和lambda表达式的作用。这里讲述的内容为本书的最后一章做了完美的铺垫。

第31章：LINQ介绍

最后一章深入探讨了LINQ编程模型的细节，它将在2007年中期与C# 3.0一起发布（我们期望是这样^①）。简单地讲，LINQ尝试提供单一对称的模型来访问数据，无论数据位于何处。你将看到，LINQ允许建立查询表达式（有意地把它设计得与SQL查询类似）来访问和操作位于数组、集合、关系型数据库或XML文档中的数据。

本书源代码

本书里所包含的所有代码示例（除了一些零碎的小段代码）都可以从Apress网站上免费下载。访问网址<http://www.apress.com>，选择Source Code链接，然后按书名查找。一旦找到了本书的主页，你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，本书中如下的注释是在提示你，文中讨论的例子可以加载到Visual Studio 2005中，以便进一步检验和修改。

① 已经按时发布。——译者注

源代码 这里给出了源代码所在的具体目录。

只需找到正确的目录，打开*.sln文件就可以了。

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）。如果是这样的话，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# SpEd”）。

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

在已有文字上完成这个“特别版”比从头写一本新书要幸福许多，没有大家的帮助和贡献，这本书是不可能完成的。首先，我非常感谢Apress全体员工，是他们一如既往的精心工作，让我那粗糙的原稿变成了精良的作品。接下来，要感谢我的技术审稿人，他们做了真正伟大的工作，保证了我的作品的准确。当然，那些可能隐藏在书中的错误（拼写、编码或其他）都是我的责任。

感谢我的朋友和我的家庭，感谢你们一次又一次地容忍我的繁忙和有时的坏脾气。更需要感谢的是我在Intertech Training公司的朋友和同事，是你们直接或间接地支持了我。最后，感谢我的妻子Mandy和所有给我爱和鼓励的人们。

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买本书时看了它一眼）。我希望你能喜欢这本书，并且能有效利用这些新知识。

Andrew Troelsen