PROGRAMMER TO PROGRAMMER™

Professional C# 2008

C# 高级编程

(第 6 版)

- C# 经典名著
- C# 超级畅销书
- 曾荣获多个大奖
- 最详实的 C# 教程

Christian Nagel
(美) Bill Evjen 等著
Jay Glynn
李 铭 翻译
黄 静 审校

wrox™

WILEY

清华大学出版社

C#高级编程

(第 6 版)

Christian Nagel
(美) Bill Evjen 等著
Jay Glynn
李铭 翻译
黄静 审校

清华大学出版社
北 京

Christian Nagel, Bill Evjen, Jay Glynn, etal
Professional C# 2008
EISBN：978-0-470-19137-8

北京市版权局著作权合同登记号　图字：　01-2008-1921

图书在版编目(CIP)数据

C#高级编程(第 6 版)/(美)内格尔(Nagel, C.)等著；李铭 翻译；黄静 审校. —北京：清华大学出版社，2008.10
书名原文：Professional C# 2008
ISBN 978-7-302-18495-9

I. C…　II. ①内…　②李…③黄…　III. C 语言—程序设计　IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 136124 号

责任编辑：王　军　于　平
装帧设计：孔祥丰
责任校对：成凤进
责任印制：李红英
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：99.25　插　页：2　字　数：2477 千字
版　次：2008 年 10 月第 1 版　　印　次：2008 年 10 月第1 次印刷
印　数：1～5000
定　价：158.00 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：029028－01

前　言

对于开发人员来说，把 C#语言及其相关环境.NET Framework 描述为多年来最重要的新技术一点都不夸张。.NET 提供了一种新环境。在这个环境中，可以开发出运行在 Windows 上的几乎所有应用程序，而 C#是专门用于.NET 的新编程语言。例如，使用 C#可以编写出动态 Web 页面、XML Web 服务、分布式应用程序的组件、数据库访问组件、传统的 Windows 桌面应用程序，甚或可以联机/脱机运行的新型智能客户应用程序。本书介绍.NET Framework 3.5 版。如果读者使用 1.0、1.1、2.0 或 3.0 版本编码，本书的一些章节就不适用。本书将标注出只适用于.NET Framework 3.5 的新增内容。

不要被.NET 这个名称所愚弄，这个名称仅强调 Microsoft 相信分布式应用程序是未来的趋势，即处理过程分布在客户机和服务器上，但 C#不仅仅是编写 Internet 或与网络相关的应用程序的一种语言，它还提供了一种编写 Windows 平台上几乎任何类型的软件或组件的方式。另外，C#和.NET 都对编写程序的方式进行了革新，更易于实现在 Windows 上的编程。

这是一个相当重要的声明。毕竟，我们都知道计算机技术的发展速度非常快，每年 Microsoft 都会推出新软件、新的编程工具或 Windows 的新版本，并宣称这些对开发人员都非常有用，.NET 和 C#也不例外。

.NET 和 C#的重要性

为了理解.NET 的重要性，了解一下近 10 年来出现的许多 Windows 技术的本质，会有一定的帮助。尽管所有的 Windows 操作系统在表面上看来完全不同，但从 Windows 3.1(1992 年)到 Windows Server 2008，在内核上都有相同的 Windows API。在我们转而使用 Windows 的新版本时，API 中增加了非常多的新功能，但这是一个演化和扩展 API 的过程，并非替换它。

开发 Windows 软件所使用的许多技术和架构也是这样。例如，COM(Component Object Model，组件对象模型)是作为 OLE(Object Linking and Embedding，对象链接和嵌入)开发出来的。那时，它在很大程度上仅是把不同类型的 Office 文档链接在一起，所以利用它可以把一个小 Excel 电子表格放在 Word 文档中。之后，它逐步演化为 COM、DCOM(Distributed COM，分布式组件对象模型)和最终的 COM+。COM+是一种复杂的技术，它是几乎所有组件通信方式的基础，实现了事务处理、消息传输服务和对象池。

Microsoft 选择这条道路的原因非常明显：它关注向后的兼容性。在过去的这些年中，第三方厂商编写了相当多的 Windows 软件，如果 Microsoft 每次都引入一项不遵循现有编码规则的新技术，Windows 就不会获得今天的成功。

向后兼容性是 Windows 技术的极其重要的特性，也是 Windows 平台的一个长处。但它有一个很大的缺点：每次某项技术进行演化增加了新功能后，都会比它以前更复杂。

很明显，对此必须进行改进。Microsoft 不可能一直扩展这些开发工具和语言，使它们越来越复杂，既要保证能跟上最新硬件的发展步伐，又要与 20 世纪 90 年代初开始流行的 Windows 产品向后兼容。如果要得到一种简单而专业化的语言、环境和开发工具，让开发人员轻松地编写优秀的软件，就需要一种新的开端。

这就是 C#和.NET 的作用。粗略地说，.NET 是一种在 Windows 平台上编程的新架构——一种新 API。C#是一种全新的语言，它可以利用.NET Framework 及其开发环境中的所有新特性，以及在最近 20 年来出现的面向对象的编程方法。

在继续介绍前，必须先说明，向后兼容性并没有在这个演化进程中失去。现有的程序仍可以使用，.NET 也兼容现有的软件。软件组件在 Windows 上的通信，现在几乎都是使用 COM 实现的。因此，.NET 能够提供现有 COM 组件的包装器(wrapper)，以便.NET 组件与之通信。

我们不需要学习了 C#才能给.NET 编写代码，因为 Microsoft 已经扩展了 C++，提供了一种新语言 J#，还对 Visual Baisc 进行了很多改进，把它转变成了功能更强大的 Visual Baisc .NET，并允许把用这些语言编写的代码用于.NET 环境。但这些语言都因有多年演化的痕迹，所以不能完全用现在的技术来编写。

本书将介绍 C#编程技术，同时提供.NET 体系结构工作原理的必要背景知识。我们不仅会介绍 C#语言的基础，还会给出使用各种相关技术的应用程序示例，包括数据库访问、动态的 Web 页面、先进的图形技术和目录访问等。唯一的要求是用户至少熟悉一门在 Windows 上使用的高级语言，例如 C++、Visual Baisc 或 J++。

.NET 的优点

前面阐述了.NET 的优点，但并没有说它会使开发人员的工作更易完成。本节将简要讨论.NET 的改进特性。

- 面向对象的编程：.NET Framework 和 C#从一开始就完全是基于面向对象的。
- 优秀的设计：一个基类库，它是以一种非常直观的方式设计出来的。
- 语言的无关性：在.NET 中，Visual Baisc .NET、C#、J#和托管 C++等语言都可以编译为通用的中间语言(Intermediate Language)。这说明，语言可以用以前没有的方式交互操作。
- 对动态 Web 页面的更好支持：ASP 具有很大的灵活性，但效率不是很高，这是因为它使用了解释性的脚本语言，且缺乏面向对象的设计，从而导致 ASP 代码比较凌乱。.NET 使用一种新技术 ASP.NET，为 Web 页面提供了一种集成式的支持。使用 ASP.NET，可以编译页面中的代码，这些代码还可以使用.NET 高级语言来编写，例如 C#或 Visual Baisc 2008。
- 高效的数据访问：一组.NET 组件，统称为 ADO.NET，提供了对关系数据库和各种数据源的高效访问。这些组件也可以访问文件系统和目录。.NET 内置了 XML 支持，可以处理从非 Windows 平台导入或导出的数据。
- 代码共享：.NET 引入了程序集的概念，替代了传统的 DLL，可以完美无暇地修补代码在应用程序之间的共享方式。程序集是解决版本冲突的正式系统，程序集的不同版本可以并存。

- 增强的安全性：每个程序集还可以包含内置的安全信息，这些信息可以准确地指出谁或哪种类型的用户或进程可以调用什么类的哪些方法。这样就可以非常准确地控制程序集的使用方式。
- 对安装没有任何影响：有两种类型的程序集，分别是共享程序集和私有程序集。共享程序集是可用于所有软件的公共库，私有程序集只用于某个软件。私有程序集是完全自包含的，所以安装过程非常简单，没有注册表项，只需把相应的文件放在文件系统的相应文件夹中即可。
- Web 服务的支持：.NET 集成了对开发 Web 服务的完全支持，用户可以开发出任何类型的应用程序。
- Visual Studio 2008：.NET 附带了一个开发环境 Visual Studio 2008，它可以很好地利用 C++、C#、Visual Baisc 2008 和 ASP.NET 进行编码。Visual Studio 2008 集成了 Visual Studio .NET 2002/2003/2005 和 Visual Studio 6 环境中的各种语言专用的所有最佳功能。
- C#：是使用.NET 的一种面向对象的新语言。

第 1 章将详细讨论.NET 体系结构的优点。

.NET Framework 3.5 中的新增特性

.NET Framework 的第 1 版(1.0 版)在 2002 年发布，赢得了许多人的喝彩。.NET Framework 的最新版本 2.0 在 2005 年发布，它对该架构进行了较大的改进。.NET Framework 3.5 尽管不像.NET Framework 2.0 那样是一个大的改进，但仍是一个相当重要的版本，包含了许多杰出的新功能。

Microsoft 每次发布新的架构时，总是试图确保对已开发出的代码进行尽可能少的修改。到目前为止，Microsoft 在这方面做得很成功。

注意：

一定要建立一个临时的服务器，来测试应用程序到.NET Framework 3.5 的升级，而不是直接升级当前运行的应用程序。

下面将详细描述 C# 2008 和.NET Framework 3.5 中的一些新变化，以及.NET Framework 3.5 的开发环境——Visual Studio 2008 的新增内容。

隐式类型化的变量

使用 C# 2008，可以声明一个变量，让编译器隐式地决定该变量的类型。LINQ 就使用这个功能处理所创建的查询。要使用这个新功能，需要使用 var 关键字：

```
var x = 5;
```

使用这个语句时，编译器会使用 5 来确定该变量的类型。这表示，该语句实际上应如下所示：

```
int x = 5;
```

自动实现的属性

在 C# 2008 中，声明属性这个常见任务变得更容易了。在以前的版本中，需要以如下方式声明属性：

```
private int _myItem;

  public int MyItem
  {
  get {
    return myItem
  }
  set {
    myItem = value;
  }
}
```

现在可以让编译器完成这个工作。不需要重复地把上面的结构放在代码中，只需使用自动实现属性的快捷方式：

```
public int MyProperty { get; set; }
```

使用这个语法会得到与前面较长示例相同的结果。编译器会把这个缩写形式转换为一般的格式，使代码更容易阅读和使用，并可以更快地编写解决方案。

对象和集合初始化器

C# 2008 现在允许在初始化对象的属性时，把值赋予属性。例如，假定代码中有如下对象：

```
public class MyStructure
{
  public int MyProperty1 { get; set; }
  public int MyProperty2 { get; set; }
}
```

使用 C# 2008，可以实例化 MyStructure 对象，如下所示：

```
MyStructure myStructure = new MyStructure() { MyProperty1 = 5,
 MyProperty2 = 10 };
```

这个功能允许一次声明集合中的多个项：

```
List < int > myInts = new List < int > () { 5, 10, 15, 20, 25 };
```

在这个例子中，所有的数字都加到 myInts 对象上，就好像使用了 Add()方法一样。

内置的 ASP.NET AJAX 支持

可以使用.NET Framework 2.0 创建 ASP.NET AJAX Web 页面，但这需要额外的安装。在 ASP.NET 3.5 和 Visual Studio 2008 中内置了 ASP.NET AJAX 支持。

现在，用 ASP.NET 和.NET Framework 3.5 创建的每个页面都支持 AJAX（在 Web.config 文

件中可以查看所有的 Ajax 配置)。在 ASP.NET 控件工具箱中有一些新的服务器控件，可以给 Web 站点添加 AJAX 功能。ASP.NET AJAX 的更多内容可参见第 39 章。

.NET Language Integrated Query(LINQ) Framework

LINQ 是最酷、最令人期待的特性，它提供了访问底层数据的功能。微软公司把 LINQ 提供为一个轻型的功能，为底层的数据库提供了一个强类型化的界面。LINQ 为开发人员提供了在他们习惯的编码环境下编写代码的方式，并可以把底层数据作为对象来访问，以利用 IDE、IntelliSense 甚至调试功能。

使用 LINQ 可以查询对象、数据集合、SQL Server 数据库、XML 等。无论底层数据源是什么，都可以用相同的方式获得数据，因为 LINQ 提供了一个查询数据的结构化方式。

下面的代码获得伪 XML 文档，提取 XML 文件中所有的顾客姓名：

```
XDocument xdoc = XDocument.Load(@"C:\Customers.xml");

var query = from people in xdoc.Descendants("CustomerName")
            select people.Value;

Console.WriteLine("{0} Customers Found", query.Count());
Console.WriteLine();

foreach (var item in query)
{
 Console.WriteLine(item);
}
```

提示：

第 11、27 和 29 章介绍了 LINQ 的各个方面。

Visual Studio 中的多目标

在许多情况下，.NET 开发人员现在使用多个.NET 应用程序，它们分别面向.NET Framework 2.0、3.0 和 3.5。继续在开发计算机上保留 Visual Studio 的多个版本，以使用.NET Framework 的多个版本似乎很愚蠢。

因此，Visual Studio 的最新版本 2008 现在支持使用自己需要的.NET Framework 版本。创建新的应用程序时，可以选择创建面向.NET Framework 2.0、3.0 或 3.5 的应用程序。

支持最新的应用程序类型

在.NET Framework 3.0 发布不久，就出现了一些非常新的功能。在这个版本中，允许使用 Windows Presentation Foundation(WPF)建立新的应用程序类型，以及基于 Windows Communication Foundation(WCF)和 Windows Workflow Foundation(WF)的应用程序和库。

在 Visual Studio 2008 中，可以创建所有这些应用程序，它们现在都可以用作项目类型，并带有新控件、Visual Studio 向导和功能。

C#的优点

C#在某种程度上可以看作是.NET 面向 Windows 环境的一种编程语言。在过去的十几年中，Microsoft 给 Windows 和 Windows API 添加了许多功能，Visual Baisc 2008 和 C++也经历了许多变化。虽然 Visual Baisc 和 C++最终已成为非常强大的语言，但这两种语言也存在问题，因为它们保留了原来的一些内容。

对于 Visual Basic 6 及其早期版本来说，它的主要优点是很容易理解，许多编程工作都很容易完成，基本上隐藏了 Windows API 和 COM 组件结构的内涵。其缺点是 Visual Basic 从来没有实现真正意义上的面向对象，所以大型应用程序很难分解和维护。另外，因为 Visual Baisc 的语法继承于 BASIC 的早期版本(BASIC 主要是为了让初学者更容易理解，而不是为了编写大型商业应用程序)，所以不能真正成为结构化或面向对象的编程语言。

另一方面，C++置根于 ANSI C++语言定义。它与 ANSI 不完全兼容，因为 Microsoft 是在 ANSI 定义标准化之前编写 C++编译器的，但已经相当接近了。但是，这导致了两个问题。其一，ANSI C++是在十几年前的技术条件下开发的，因此不支持现在的概念(例如 Unicode 字符串和生成 XML 文档)，某些古老的语法结构是为以前的编译器设计的(例如成员函数的声明和定义是分开的)。其二，Microsoft 同时还试图把 C++演变为一种用于在 Windows 上执行高性能任务的语言，为此不得不在语言中添加大量 Microsoft 专用的关键字和各种库。其结果是在 Windows 中，该语言成了一种非常杂乱的语言。让 C++开发人员说说字符串有多少种定义就可以说明这一点：char*、LPTSTR、string、CString (MFC 版本)、CString (WTL 版本)、wchar_t*、OLECHAR*等。

现在进入.NET 时代——一种全新的环境，它对这两种语言都进行了新的扩展。Microsoft 给 C++添加了许多 Microsoft 专用的关键字，并把 Visual Baisc 演变为 Visual Baisc .NET，再演变为 Visual Baisc 2008，保留了一些基本的 Visual Baisc 语法，但在设计上完全不同，从实际应用的角度来看，Visual Baisc 2008 是一种新语言。

在这里，Microsoft 决定给开发人员另一个选择——专门用于.NET、具有新起点的语言，即 C#。Microsoft 在正式场合把 C#描述为一种简单、现代、面向对象、类型非常安全、派生于 C 和 C++的编程语言。大多数独立的评论员对 C#的说法是“派生于 C、 C++ 和 Java”。这种描述在技术上是非常准确的，但没有涉及到该语言的真正优点。从语法上看，C#非常类似于 C++和 Java，许多关键字都是相同的，C#也使用类似于 C++和 Java 的块结构，并用括号({})来标记代码块，用分号分隔各行语句。对 C#代码的第一印象是它非常类似于 C++或 Java 代码。但在这些表面的类似性后面，C#学习起来要比 C++容易得多，但比 Java 难一些。其设计与现代开发工具的适应性要比其他语言更高，它同时具有 Visual Basic 的易用性、高性能以及 C++的低级内存访问性。C#包括以下一些特性：

- 完全支持类和面向对象编程，包括接口和继承、虚函数和运算符重载。
- 定义完整、一致的基本类型集。
- 对自动生成 XML 文档说明的内置支持。
- 自动清理动态分配的内存。

- 可以用用户定义的特性来标记类或方法。这可以用于文档说明，对编译有一定的影响(例如，把方法标记为只在调试时编译)。
- 对.NET 基类库的完全访问权，并易于访问 Windows API。
- 可以使用指针直接访问内存，但 C#语言可以在没有它们的条件下访问内存。
- 以 Visual Basic 的风格支持属性和事件。
- 改变编译器选项，可以把程序编译为可执行文件或.NET 组件库，该组件库可以用与 ActiveX 控件(COM 组件)相同的方式由其他代码调用。
- C#可以用于编写 ASP.NET 动态 Web 页面和 XML Web 服务。

应该指出，对于上述大多数特性，Visual Baisc 2008 和托管 C++也具备。但 C#从一开始就使用.NET，对.NET 特性的支持不仅是完整的，而且提供了比其他语言更合适的语法。C#语言本身非常类似于 Java，但其中有一些改进，因为 Java 并不是为应用于.NET 环境而设计的。

在结束这个主题前，还要指出 C#的两个局限性。其一是该语言不适用于编写时间急迫或性能非常高的代码，例如一个要运行 1000 或 1050 次的循环，并在不需要这些循环时，立即清理它们所占用的资源。在这方面，C++可能仍是所有低级语言中的佼佼者。其二是 C#缺乏性能极高的应用程序所需要的关键功能，包括保证在代码的特定地方运行的内联函数和析构函数。但这类应用程序非常少。

编写和运行 C#代码的环境

.NET Framework 3.5 运行在 Windows XP、2003、Vista 和最新的 Windows Server 2008 上。要使用.NET 编写代码，需要安装.NET 3.5 SDK。

除非要使用文本编辑器或其他第三方开发环境来编写 C#代码，否则一般使用 Visual Studio 2008。运行托管代码不需要安装完整的 SDK，但需要.NET 运行库。需要把.NET 运行库分布到还没有安装它的客户机上。

本书内容

在本书中，首先在第 1 章介绍.NET 的整体体系结构，给出编写托管代码所需要的背景知识，此后本书分几部分介绍 C#语言及其在各个领域中的应用。

第 I 部分——C#语言

本部分给出 C#语言的背景知识。这一部分没有指定任何语言，但假定读者是有经验的编程人员。首先介绍 C#的基本语法和数据类型，再介绍 C#的面向对象特性，之后是 C#中的一些高级编程论题。

第 II 部分——Visual Studio

本部分介绍全世界 C#开发人员都使用的主要 IDE：Visual Studio 2008。本部分的两章探讨使用工具建立基于.NET Framework 3.5 的应用程序的最佳方式，另外，还讨论项目的部署。

第Ⅲ部分——基类库

本部分介绍在.NET 环境中编程的规则。特别是安全性、线程本地化、事务处理、建立 Windows 服务的方式，以及将自己的库生成为程序集的方式。

第Ⅳ部分——数据

本部分介绍如何使用 ADO.NET 和 LINQ 访问数据库，以及与目录和文件的交互。我们还详细说明.NET 对 XML 的支持、对 Windows 操作系统的支持，以及 SQL Server 2008 的.NET 特性。在 LINQ 部分，特别关注 LINQ to SQL 和 LINQ to XML。

第Ⅴ部分——显示

本部分讨论传统 Windows 应用程序的创建，在.NET 中这种应用程序称为 Windows 窗体。Windows 窗体是应用程序的胖客户版本，使用.NET 创建这些类型的应用程序是实现该任务的一种快捷、简单的方式。除了介绍 Windows 窗体之外，我们还将论述 GDI+，这种技术可用于创建包含高级图形的应用程序。本部分还阐述如何编写在网站上运行的组件，如何编写网页。其中包括 ASP.NET 3.5 提供的许多新特性。最后，我们还将陈述如何建立基于 WPF 和 VSTO 的应用程序。

第Ⅵ部分——通信

这一部分介绍通信，主要论述独立于平台进行通信的 Web 服务、在.NET 客户机和服务器之间通信的.NET Remoting 技术、在后台上运行的 Enterprise Services 和 DCOM 通信。有了消息异步排队技术，可以进行断开连接的通信。本部分还介绍如何利用新的 WCF 和 WF。

第Ⅶ部分——附录

这一部分介绍如何建立应用程序，来利用 Windows Vista 中的新功能，并探讨未来的 ADO.NET Entities 技术，以及如何在 C#应用程序中使用它。

如何下载本书的示例代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 http://www.wrox.com/ 和 http://www.tupwk.com.cn/downpage 上下载。登录到站点 http://www.wrox.com/上，使用 Search 工具或书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

许多图书的书名都很相似，所以通过 ISBN 查找本书是最简单的，本书的 ISBN 是 978-0-470-19137-8。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 http://www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

要在网站上找到本书的勘误表，可以登录 http://www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。请给 wkservice@tup.tsinghua.edu.cn 发电子邮件，我们就会检查您的信息，如果是正确的，我们将在本书的后续版本中采用。

p2p.wrox.com

P2P 邮件列表是为作者和读者之间的讨论而建立的。读者可以在 p2p.wrox.com 上加入 P2P 论坛。该论坛是一个基于 Web 的系统，用于传送与 Wrox 图书相关的信息和相关技术，与其他读者和技术用户交流。该论坛提供了订阅功能，当论坛上有新贴子时，会给您发送您选择的主题。Wrox 作者、编辑和其他业界专家和读者都会在这个论坛上进行讨论。

在 http://p2p.wrox.com 上有许多不同的论坛，帮助读者阅读本书，在读者开发自己的应用程序时，也可以从这个论坛中获益。要加入这个论坛，须执行下面的步骤：

(1) 进入 p2p.wrox.com，单击 Register 链接。

(2) 阅读其内容，单击 Agree 按钮。

(3) 提供加入论坛所需的信息及愿意提供的可选信息，单击 Submit 按钮。

然后就可以收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的信息，但只有加入论坛后，才能发送自己的信息。

加入论坛后，就可以发送新信息，回应其他用户的贴子。可以随时在 Web 上阅读信息。如果希望某个论坛给自己发送新信息，可以在论坛列表中单击该论坛对应的 Subscribe to this Forum 图标。

对于如何使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作原理，以及许多针对 P2P 和 Wrox 图书的常见问题解答。要阅读 FAQ，可以单击任意 P2P 页面上的 FAQ 链接。

目　录

第Ⅰ部分　C# 语 言