

Windows Forms Programming

With C#

# C# Windows Forms 程序设计

(美) Erik Brown 著  
朱毓斌 吴飞 译



清华大学出版社

# C# Windows Forms 程序设计

(美) Erik Brown 著  
朱毓斌 吴飞 译

清华大学出版社  
北京

## 内 容 简 介

本书是打开 Windows Forms 编程之门的金钥匙，它所采用的编程语言是当前最为流行、最具潜力的 C#。本书内容详尽充实，共分为 3 个部分：初识 Windows Forms、Windows Forms 基础和高级 Windows Forms。通过本书的学习，读者能够迅速掌握通过 .NET Framework 开发 Windows Forms 程序的所有相关技术。

无论对开发桌面应用程序的 Windows 程序员，还是正在学习 Windows Forms 知识的开发人员，或者是毫无 Windows 开发经验的初学者，相信本书都是最佳的选择。

EISBN: 1-930110-28-6

WINDOWS FORMS PROGRAMMING WITH C#

Erik Brown

Copyright © 2002 by Manning Publications Co.

Original English Language Edition Published by Manning Publications Co.

All Rights Reserved.

本书中文简体字版由 Manning Publications Co. 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的任何部分。

**版权所有，翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。**

北京市版权局著作权合同登记号 图字：01-2002-6280

### 图书在版编目(CIP)数据

C# Windows Forms 程序设计/(美)布朗著；朱毓斌，吴飞译. —北京：清华大学出版社，2003

书名原文：Windows Forms Programming with C#

ISBN 7-302-06689-2

I. C… II. ①布…②朱…③吴… III. ①C 语言—程序设计②窗口软件，Windows—程序设计

IV. ①TP312②TP316.7

中国版本图书馆 CIP 数据核字(2003)第 041500 号

出 版 者：清华大学出版社

地 址：北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编：100084

社 总 机：010-62770175

客 户 服 务：010-62776969

组稿编辑：曹康

文稿编辑：陈宗斌

封面设计：康博

版式设计：康博

印 刷 者：清华大学印刷厂

发 行 者：新华书店总店北京发行所

开 本：185 × 260 印张：39.5 字数：1011 千字

版 次：2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号：ISBN 7-302-06689-2/TP · 4998

印 数：1 ~ 4000

定 价：78.00 元

# 前 言

我开始使用 Microsoft 的 .NET Framework 是在 2001 年初，那时我正和一个新成立的小公司合作一个项目。没料到情况发生了变化，我也因此有了比想象中更多的空闲时间。当 Manning Publications 问我是否有兴趣编写一本关于 .Net Framework 编程的书时，我欣然接受了他们的邀请。

对于如何组织本书的内容，我强烈建议把 Windows Forms 应用程序的编程技巧作为介绍的重点。我一直都非常喜欢 Jeff Proise 编写的 *Programming Windows 95 with MFC*，因此在我看来，写一本利用 .Net Framework 编写基于 Windows 的应用程序的著作是一个再好不过的选择。

我的建议中还包含着一个重要问题，那就是基于本书的课程内容相应地建立一个应用程序。在介绍各个主题时，这个应用程序也相应地不断变化。这样，当读者看完这本书时，也就得到了一个功能强大的应用程序。Manning Publications 也很赞同这个提议，这样，我就开始了编写本书的工作。

在写作过程中，我努力向着两个目标奋斗。第一就是尽可能全面地涵盖命名空间的各个类。我曾经读过很多类似的书——如果对于某个主题没有提供强有力的示例，往往使我很失望。所以，我将尽可能提供详尽的示例来说明如何在一个实际问题中正确地使用和巧妙地处理 Windows Forms 中的各个类。

第二个目标就是给一些高级用户介绍诸如树状视图以及拖放等有关界面功能的主题。在花很多的时间介绍一些基本的类(如菜单和按钮)的同时，对于基于 Windows 的编程中可能用到的一些更复杂的控件，也给出了比较详细的介绍。

这些设想、目标，再加上许多个不眠之夜，汇成了您眼前的这本书。通过创建一个基于本书的普通应用程序，对于应用程序的开发给出了一些指导，同时指出了进一步的研究工作中一些相关的类和主题。衷心希望我的工作能够提供足够的细节来说明如何在 .Net Framework 中组织这些基于 Windows 的应用程序，同时也为您发展和扩充自己的 .Net 项目提供尽可能多的帮助。

虽然本书并不是专门介绍 C# 和 Visual Studio .Net 的，但还是尽可能地对 C# 的语法、用法以及 Visual Studio .Net 的特征和功能做了解释，当示例中用到相关的概念和特征时，就给出对这些主题的解释和说明。在本书最后的附录 A 中还给出了关于 C# 的一个概述。

## 关于本书

由于 .Net Framework 所涵盖的内容极其广泛，所以要在一本书里涉及所有的方面是不可能的。这一节给大家介绍一下本书的侧重点，并简单介绍书中各部分的内容。最后介绍一下所谓的“在线论坛”，关于本书的任何问题以及评论都可以在那里进行交流，并且可以通过 Internet

下载书中使用的一些源代码。

在讨论本书之前,我们首先介绍一下命名空间(namespaces)的概念。命名空间定义了一个组或者说一个范围,其中包含一些相关的类、结构以及其他一些类型。命名空间有点像一个家庭:它利用一个通用名称定义一组不同的成员,某些成员具有相同的功能。

.Net Framework 中所有的对象,确切地讲是 C#中所有的对象,组成了各个命名空间。例如, System 命名空间包含了和 Framework 本身相关的一些对象。由.Net 所定义的绝大多数命名空间都包含于 System 命名空间里。System.Windows 命名空间定义了和 Windows 操作系统相关的一些类型和命名空间,而 System.Web 命名空间则定义了和 Web 页面以及服务器相关的一些类型和命名空间。

这种对应于命名空间的组织结构允许两个具有相同基本名称的对象表示两个截然不同的概念,这就像两个人可以拥有相同的姓一样。例如,在 System.Web.UI.WebControls 命名空间中的 Button 类表示网页上的一个按钮,而 System.Windows.Forms 命名空间中的 Button 类则表示应用程序窗口中的一个按钮。.Net 中还有其他的命名空间,例如关于文件和目录对象的 System.IO 命名空间,关于图形对象的 System.Drawing 命名空间。在附录 B 中我们介绍了在.Net 中常用的一些命名空间。

## Windows Forms 命名空间

除了对.Net Framework 所支持的广大对象集合建立了一个结构之外,命名空间的概念对于如何写书也提供了一些指导。本书的重点是 System.Windows.Forms 命名空间,通常称之为 Windows Forms。Windows Forms 应用程序是被 Windows 操作系统执行的程序,它所使用的用户界面和随处可见的 Windows 用户桌面很相似。

本书希望较为系统地介绍 Windows Forms 命名空间,包括在这个命名空间中定义的绝大多数类型。

本书共 18 章,分成 3 个部分来介绍。

### 第 1 部分:初识 WINDOWS FORMS

本书的第 1 部分介绍了关于 C# 和 Windows Forms 的一些基本概念,前者比较粗略,后者比较详细。第 1 章利用一个文本编辑器创建了如图 0-1 所示的应用程序。我们将讨论如何通过 .Net Framework 来实现一个 Windows Forms 应用程序,以及如何利用 C# 来进行 Windows Forms 编程。



图 0-1 第 1 部分中的“Myphotos”应用程序

在第 2 章里，我们开始利用 Microsoft 的图形开发环境 Visual Studio .Net 在 .Net Framework 中创建应用程序。这一章里，我们重新创建了第 1 章中那个手工构建的应用程序，并将之称为“MyPhotos”。

## 第 2 部分：WINDOWS FORMS 基础

在这一部分里，我们开始系统介绍 Windows Forms 命名空间里的各个类。这部分是前面“MyPhotos”应用程序的继承和发展，如图 0-2 所示(第 13 章中也有相同的图示)。您将看到，这一部分将涉及创建一个 Windows Forms 应用程序时需要用到的主要的面向对象控件，包括菜单、状态栏、对话框、文本框以及组合框等。

“MyPhotos”应用程序将展示一个相册中的内容，它由一个或者多个图像文件或相片组成。程序将每个相册存储在一个文件里，并且允许用户逐个预览图像以及编辑相片或者相册的属性。

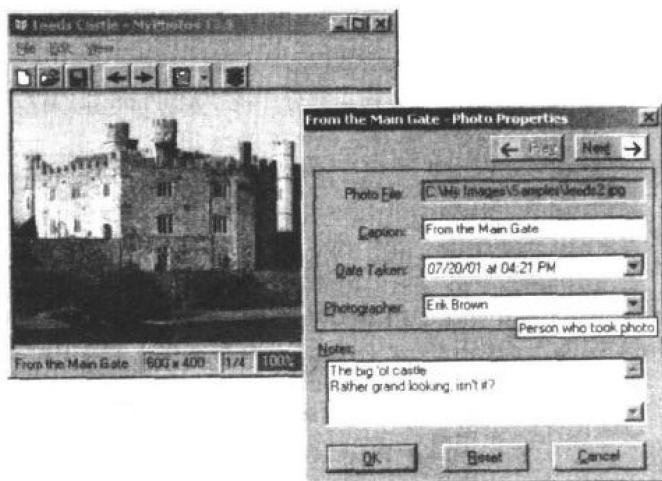


图 0-2 第 13 章中的“MyPhotos”应用程序。图中给出了主窗口和一个用于编辑选定相片属性的对话框

## 第 3 部分：高级 WINDOWS FORMS

诸如图像的列表查看、拖放等更多高级主题将在本书的第 3 部分中介绍。这一部分利用相册的概念创建了几个不同的应用程序，包括一个类似于 Windows 资源管理器的用来浏览相册的应用程序，以及一个展示如何将 Windows Forms 控件的内容与一个从数据源中得到的数值进行绑定的“数据驱动”型应用程序。

在第 18 章中的“MyPhotos”应用程序的主窗口如图 0-3 所示。它被转换成一个多文档界面，可以显示多个相册。同时，也添加了很多新的功能，例如在不同的相册之间拖动相片以及在程序中显示本书的 Web 站点，等等。

## 本书适用读者

每个作者都希望所有的人能阅读自己的著作，我当然也不例外。在编写本书的时候，我更多的是针对以下 3 类读者：

- 对于利用 .NET 技术开发桌面应用程序感兴趣的 Windows 程序员。



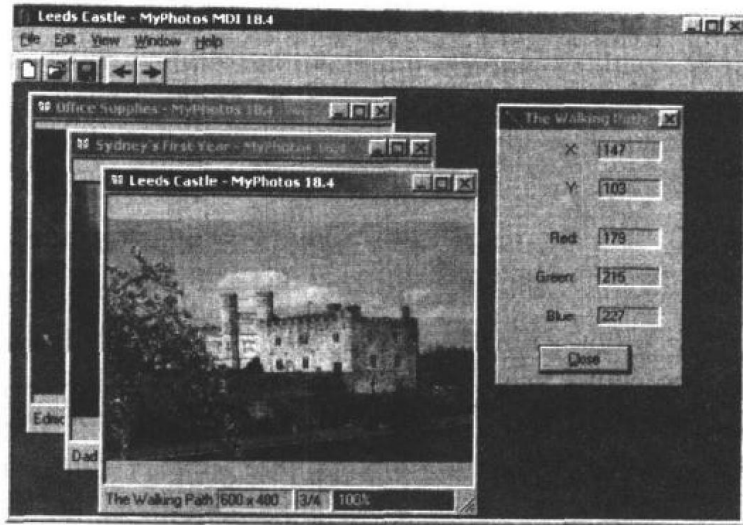


图 0-3 第 18 章中的“MyPhotos”应用程序。其中一个父窗口中显示的正是本书第 2 部分中的“MyPhotos”窗口

- 熟悉 .NET 或者 C# 技术，并且希望学到更多关于 Windows Forms 的类和编程知识的开发人员。
- 只有很少或者没有创建 Windows 应用程序经验的 C++ 程序员。

再次说明一下：本书涉及的只是 .NET Framework 的一部分内容，即 `System.Windows.Forms` 命名空间中包括的各个类。此外，还提供了关于 C# 以及 Visual Studio .NET 的很多信息，尤其值得一提的是，它将会指导您一步一步地利用 Visual Studio .NET 创建每一个应用程序示例。同时，本书在附录 C 中给出了关于 C# 和 .NET 的一个资源列表。

如果想要对 .NET Framework 有一个更加广泛、深入的了解，可以参考 Fergal Grimes 编著的 *Microsoft .NET for programmers*，也是由 Manning 出版社出版的。

## 源代码下载

只要您购买了本书，书中出现的所有程序的源代码都可以从 Manning 主页上下载。您可以登录 [www.manning.com/eebrown](http://www.manning.com/eebrown) 得到一些指导信息或者下载相关源代码。

## 作者在线

只要您购买了本书，就可以免费进入网上论坛，即“作者在线”。您可以登录主页得到关于论坛更加详尽的信息，可以注册获得进入论坛的资格，查看每一章节的代码，并可及时了解书中内容的更新和更正情况。您也可以对本书发表或褒或贬的评论，或者提一些技术问题。您可以从作者或者其他 Windows Forms 程序员那里获得帮助。登录 [www.manning.com/eebrown](http://www.manning.com/eebrown) 就可以找到论坛的相关信息。

Manning 对于读者的一个义务，就是提供一个读者与读者之间，以及读者与作者之间可以相互交流的场所。但是对于作者来讲，在这种交流中需要投入多少精力并没有具体的规定，它完全是作者自愿的(也是无偿的)。

您也可以通过 [eebrown@eebrown.com](mailto:eebrown@eebrown.com) 或者个人主页 [www.eebrown.com](http://www.eebrown.com) 来联系 Erik。

## 关于.NET

.NET 的历史既长久又短暂。虽然 .NET Framework 基于的编程语言以及开发环境数年前就出现了，但相对而言它还是新生事物，并且它在市场上能否成功还有待于进一步的检验。本节将简要回顾 C# 编程语言的历史，以及 .NET Windows Forms 的功能。

C 语言最初是 70 年代早期在贝尔实验室发展起来的，当时是与 UNIX 操作系统联合使用的。它是从一种更早的所谓“B”语言发展而来的，而“B”语言本身是从一种被称为“BPCL”的语言派生出来的。70 年代这种语言很快就流行起来，并最终在 80 年代中期由美国国家标准协会(ANSI)制定出了一套标准。介绍 C 语言的较为权威的书籍依然是 Brian W. Kernighan 和 Dennis M. Ritchie 编写的 *The C Programming Language*，它最早出版于 1978 年。

C++ 语言是由 Bjarne Stroustrup 设计出来的，他的那本著名的 *The C++ Programming Language* 最早是于 1986 年出版的。这个语言标准的制定仍然是由 ANSI 以及其他组织完成的。如今它有了更加普遍的应用。

C 语言应用于 Windows 操作系统最早开始于 Windows 1.0 系统，大约是在 80 年代中期。Microsoft 关于人机交互开发环境(IDE)的一个最早的目标是 1990 年左右实现的，当时推出的是 Microsoft C 1.0。这个开发环境逐渐包括了 C++，并且成为 90 年代中期的 Visual C++ 的基础，并最终演化成为 Visual Studio 6.0，它支持 C++、Visual Basic 以及一个算不上标准化的 Java。Windows 操作系统以及 Win32 API 构成了这些产品创建以及扩展的基础。

对于 Microsoft Visual Studio 的另外两个值得一提的竞争产品就是 Borland C++ 以及 Borland Delphi。它们都是非常成功的可视化开发工具，对于 .NET Framework 的设计以及开发都有一定的影响。实际上，Microsoft 的 C# 语言负责人 Anders Hejlsberg 就是 Borland Delphi 的最初设计者。

其他贯穿于这个发展历史之中的产品还有：动态链接库、OLE 和 COM 的出现、诸如 ODBC 和 ADO 的数据库技术、Internet 的兴起与发展，以及将基于 DOS 的 Windows 重新设计成为 Windows NT 和 Windows XP。

### “打造” .NET

和一些技术以及产品相对的是，Internet 一直以来都在重新定义着程序员们设想的用户界面以及应用程序开发环境。随着诸如 Netscape 以及 Amazon.com 应用上取得的成功，一个产品或者一个界面越来越可能出现在一个网页浏览器上，而不再是一个传统的 Windows 应用程序中。Java 语言已经在 UNIX 系统和网页服务器上获得了一些成功，就目前来讲，它是很多大型网页应用程序的设计语言。

也许是由于这种变化的响应，也许是由于现在用户桌面和 Internet 之间的界限正越来越模糊，Microsoft 想发动一次“思想革命”——改变我们传统意义上对于应用程序开发的理解。其结果就是 .NET Framework 以及 C# 语言的出现<sup>1</sup>。

---

<sup>1</sup> Visual Basic 语言也已重新设计，影响了无数 Microsoft 产品。您可访问 [www.microsoft.com/net](http://www.microsoft.com/net) 以了解关于 .NET 的历史等详细情况。



理解 C# 的最容易的方法就是设想有人将 C++ 的所有烦人的缺点写下来, 然后去除掉其中的每一项。举个例子: 在 C++ 语言中, 指针的应用是件很麻烦的事情; 很多代码错误并不能被编译器检测到, 例如 `if (x = 5)`; 对于字符串的操作很困难; 没有一种好的方法可以安全地将一个对象“退化”成一个派生类型。C 和 C++ 语言的前身 B 和 BPCL 语言没有制定这样一个规范的类型系统——这种系统可以为语言中的整型、指针、字符型提供自由的空间。

在重新设计 C# 语言的时候, 既考虑到要保留 C 和 C++ 语言的灵活性, 又考虑到将类型系统以及语法规则正规化。很多 C++ 中的运行时错误在 C# 里都被认为是编译错误。另外, 还要具有一些截然不同的特性, 如内建的字符串类型, 不存在全局变量, 以及将评判系统和应用错误综合为通用异常模型(`common exception model`)。本书的附录 A 给出了对于 C# 语言的句法、关键词以及特征的概述。

尽管从设计的角度来讲, 对于 C# 语言和 .NET Framework 二者的关系没有严格的要求, 但两者可能将保持一段紧密的“交织”状态。 .NET Framework 是一个 Windows 操作系统的编程界面和运行环境, 它本身的很大部分就是用 C# 编写的。

.NET Framework 几乎包括了 Microsoft 的所有开发技术和环境——从 COM 到 XML 和 ASP, 再到 Visual Studio。这些技术在同一个系统里重新被创建和发明。尽管向后兼容性没有完全丧失, .NET Framework 对于这些技术以及使用这些技术的产品重新定义了类和方法。特别值得一提的是, 现在的框架还包括: 对 Windows 应用程序开发的新的支持、网址的获取和释放、异地程序之间的交互、数据库交互、安全、本地及远程安装以及其他技术。

我的目的并不是列举在 .NET 中可以找到的各种技术, 也不是为了证明它好或是不好。有很多书籍和文章都提供了这方面的信息, 如果您对其一无所知的话, 也许现在就不会阅读这个介绍了。我的目的仅仅是为了说明, Microsoft 正在对它的很多技术和产品进行全新的改造, 并努力想把这些不同的产品放到一个框架里, 即 .NET Framework。

## Windows Forms 概述

正如我们在“关于本书”那一节所介绍的, .NET Framework 中的所有对象, 实际上就是 C# 本身的对象, 是用所谓的命名空间来组织的。附录 B 中对于 .NET Framework 中比较常用的一些命名空间作了概述。

本书重点介绍其中的 `System.Windows.Forms` 命名空间, 它用来创建基于 Windows 的应用程序。本节将概括介绍这个命名空间所定义的各个类。

在介绍各个具体的类之前, 我们必须知道对于总体理解 .NET Framework 以及 Windows Forms 命名空间特殊性有所帮助的 3 个短语, 即组件、容器以及控件。本书对于这些短语做了详尽的介绍, 所以这一节中只通过几个例子进行简要介绍。

组件是允许应用程序间共享的对象。`Component` 类就是在这种意义上封装的, 它是 Windows Forms 命名空间大多数成员的基础。值得注意的另一点就是 `IComponent` 接口, 它定义了被所有组件支持的成员。我们将在第 5 章讨论这个接口, 在第 3 章中讨论 `Component` 类。

容器是能容纳零个或者多个组件的对象。它就是一个组合机制, 保证一些组件被封装在一起并被赋予相似的操作。在 Windows Forms 命名空间中, 在需要组合一些对象时就要用到容器。`Container` 类是基于容器这一概念封装的, `IContainer` 接口定义了所有容器需要的成员。

控件是一种可视化组件。在 Windows Forms 命名空间中，一个控件就是一个能够在 Windows 桌面上显示的图形化的界面组件。第 4 章中将要讨论的 Windows Forms Control 类，是所有 Windows Forms 控件的基础。对于我们来说，System.Web.UI 命名空间中定义一个 Control 类来表示在网页上显示的图形对象是值得关注的。

总的来讲，您在 Windows 桌面上可以看到的任何一个可视化界面就是一个控件，而任何隐藏在它背后的对象就是一个组件。举个例子，一个状态栏面板是用 StatusBarPanel 类来表示的，这就是一个组件。而您在界面上实际看到的状态栏是 StatusBar 类表示的，这就是一个控件。状态栏是第 4 章所要介绍的主题。

控件也可以是容器，从而使得它可以容纳一系列的控件或者组件。StatusBar 类就是一个容器，它容纳零个或者多个 StatusBarPanel 组件。另一个重要的容器控件就是 Form 类，它表示一个在 Windows 桌面上显示的应用程序窗口。我们将在第 1 章中介绍 Form 类，对它的讨论贯穿始终，第 7 章主要介绍“绘图和滚动”，第 8 章主要介绍“对话框”，而第 16 章主要介绍“多文档界面”。

图形界面中的大多数可视化元素，例如按钮、文本框、树以及对话框等都是用控件类表示的。一个例外就是菜单，我们将在第 3 章和第 16 章介绍这个主题。菜单对象都是组件，它是用 Form 类本身的一种特殊方式进行操作。

全书始终都在讨论 Windows Forms 命名空间中的控件。很多普通的控件将在第 9 章“基本控件”、第 10 章“列表控件”、第 11 章“更多控件”、第 13 章“工具栏和提示”中介绍。诸如列表视图、树视图以及数据网格等高级控件将在本书第 3 部分中介绍。

Windows Forms 命名空间里除了控件、容器以及组件外，还提供了很多其他的对象来支持基于 Windows 的应用程序开发。我们将在第 12 章“.NET 分类”和第 18 章“.NET 琐碎”中介绍其中的一些。特别值得一提的是将在第 17 章中介绍的关于数据绑定的概念。

如果一本关于如何创建 Windows Forms 应用程序的书没有讨论“可重用库”的创建问题，就显得有些欠缺了。第 5 章中我们将通过创建一个相册库来讨论这个概念，这个库在书中后面的部分将会用到，特别是第 6 章“通用对话框”将用到这个库。

## 致谢

在我们的生活中，一切事情看起来都是那么纷繁芜杂，众多的线索最终导致一个可以感知的结果。对于这些，我从来没有停止过感到惊奇，对于您正在读的这本书也是一样。虽然书的封面印着我的名字，但还有许许多多我认识和不认识的人为它做出了贡献。

特别感谢我的家庭：我的妻子 Bridgett，她给了我耐心和爱；Katie 和 Sydney，谢谢他们定期的拜访和无条件的接受；Bianca，我忠实的伙伴，是他每天在办公室的另一张椅子上蜷缩着休息。

我同时感谢我的父母，David 和 Janet，我的老师以及其他在我的生命中支持过我的人。特别感谢 Steve Cox 和 David Cobb，他们在很久以前令我对计算机编程产生了兴趣。

同样感谢我的朋友，他们用朋友特殊的方式支持和鼓励我，尤其是 Jean Siegel, Janet Heffernan, Tony Mason 和 Marc Zapf。我还要感谢我的橄榄球队 Haymarket Outer Limits，它让

我在成稿期间获得了许多身体上的锻炼，给了我一段有趣而多产的时光。

感谢来自四面八方的校对者，他们贡献出许多的时间和精力阅读了不同章节的最初版本。没有他们的坚持和努力，这本书不能达到今天的水准。包括 Marc Zapf 帮我作技术校对和最终定稿；Javier Jarava 竭尽全力地帮我校正每一章的代码和文本；Josh Mitts 给了我深刻和鼓励性的评价；Andreas Haber 为我解决许多重要的技术难点(对不起，Andreas，我最终还是没有把 PropertyGrid 控件放进去)；Craig Fullerton 仔细地校对了本书的前半部分；Mark Boulter 在最初版本上进行了详尽的评价；Sam Raisanen 全面地校对了第 1、3 和 16 章；还有其他人给了我许多具有远见卓识的评价和批评，包括 Dharmesh Chauhan, Chris Muench, Tomas Restrepo 以及 Vijay Upadya。

我还要感谢看过本书最初概要的人，比如 Steve Binney, Mark Boulter, Drew Marsh, Josh Mitts 以及 Kunle Odutola。他们的建议很重要，对于以正确的方式开始本书的编写以至最终的完成都有着很大的帮助。

最后，我要感谢在整个出版过程中与我一起工作的人，包括那些我从没见过面，但在我的写作过程中给了我很大支持的人。特别是 Susan Capparelle，她在我的最初报告中认识到了本书的价值；Marjan Bace 通过无数次的电话讨论给了我很多深刻的评价和建议；Ted Kennedy 帮我和所有的校对者协调并收集反馈信息；Syd Brown 多次为我重新编辑表格而使它们最终成形；Leslie Haimes 在我已经同意了前一个封面设计后又为我重新设计了一个封面；Mary Piergies 负责和工作人员沟通并替我为他们回答问题；Rebecca Pepper 和 Lianna Wlasiuk 鼓励我修改书中很多的结构性元素；Lois Patterson 为我详尽地检查了最终版本的文字错误；最后，Lee Fitzpatrick 为我签属了版税支票。

# 目 录

## 第 1 部分 初识 Windows Forms

<b>第 1 章 从 Windows Forms 开始</b> .....	<b>3</b>
1.1 用 C#编程 .....	3
1.1.1 命名空间和类 .....	5
1.1.2 构造函数和方法 .....	7
1.1.3 C#的类型 .....	8
1.1.4 入口点 .....	10
1.1.5 Application 类 .....	10
1.1.6 程序执行 .....	11
1.2 添加控件 .....	12
1.2.1 简写和完全限定名称 .....	13
1.2.2 字段和属性 .....	15
1.2.3 Controls 属性 .....	16
1.3 加载文件 .....	17
1.3.1 事件 .....	19
1.3.2 OpenFileDialog 类 .....	20
1.3.3 Bitmap 图像 .....	22
1.4 改变窗体的尺寸 .....	23
1.4.1 桌面布局属性 .....	26
1.4.2 Anchor 属性 .....	27
1.4.3 Dock 属性 .....	29
1.5 小结 .....	30
<b>第 2 章 Visual Studio .NET</b> .....	<b>32</b>
2.1 Visual Studio .NET 编程 .....	33
2.1.1 创建项目 .....	34
2.1.2 执行程序 .....	36
2.1.3 查看源代码 .....	36
2.2 添加控件 .....	40
2.2.1 AssemblyInfo 文件 .....	40
2.2.2 窗体重命名 .....	43
2.2.3 Toolbox 窗口 .....	43
2.3 加载文件 .....	48

2.3.1	Visual Studio .NET 中的事件处理程序	49
2.3.2	异常处理	52
2.4	改变窗体的尺寸	55
2.4.1	指定 Anchor 属性值	56
2.4.2	指定 MinimumSize 属性值	57
2.5	小结	58

## 第 2 部分 Windows Forms 基础

第 3 章	菜单	61
3.1	菜单的特征	62
3.1.1	Menu 类	63
3.1.2	Menu 类的层次结构	63
3.1.3	派生类	64
3.2	菜单栏	65
3.2.1	添加主菜单	65
3.2.2	添加 File 菜单	67
3.2.3	添加下拉菜单	69
3.2.4	添加 View 菜单	72
3.3	Click 事件	74
3.3.1	通过 Designer 窗口添加事件处理程序	75
3.3.2	通过 Properties 窗口添加事件处理程序	76
3.4	Popup 事件和共享处理程序	78
3.4.1	定义共享处理程序	78
3.4.2	处理 Popup 事件	82
3.5	上下文菜单	85
3.5.1	创建上下文菜单	86
3.5.2	添加菜单项	87
3.6	小结	89
第 4 章	状态栏	90
4.1	Control 类	90
4.2	StatusBar 类	92
4.2.1	添加状态栏	93
4.2.2	给状态栏中的文本赋值	95
4.3	状态栏面板	97
4.3.1	向状态栏里添加面板	98
4.3.2	给面板文本赋值	102
4.4	用户绘制的面板	103

4.4.1 DrawItem 事件 .....	104
4.4.2 绘制面板 .....	106
4.5 小结 .....	109
<b>第 5 章 可重用库 .....</b>	<b>111</b>
5.1 C#类和接口 .....	111
5.1.1 接口 .....	112
5.1.2 数据集合类 .....	113
5.2 类库 .....	116
5.2.1 创建类库 .....	117
5.2.2 使用命令行工具 .....	121
5.2.3 创建 PhotoAlbum 类 .....	122
5.2.4 创建 Photograph 类 .....	123
5.3 再谈接口 .....	127
5.3.1 支持 ICollection 接口 .....	128
5.3.2 支持 IList 接口 .....	129
5.3.3 实现相册的定位操作 .....	131
5.4 健壮性问题 .....	133
5.4.1 处理无效位图 .....	134
5.4.2 重写 Object 类方法 .....	137
5.4.3 清理系统资源 .....	140
5.4.4 将文件名与相册建立关联 .....	142
5.5 小结 .....	143
<b>第 6 章 通用文件对话框 .....</b>	<b>144</b>
6.1 设计问题 .....	145
6.1.1 改变菜单栏 .....	145
6.1.2 添加类变量 .....	148
6.2 选取多个文件 .....	148
6.2.1 向相册中添加图像 .....	149
6.2.2 从相册中删除图像 .....	151
6.3 Paint 事件 .....	152
6.3.1 绘制当前相片 .....	152
6.3.2 显示当前位置 .....	154
6.4 再谈上下文菜单 .....	155
6.4.1 显示下一幅相片 .....	156
6.4.2 显示前一幅相片 .....	156
6.5 文件和路径 .....	157
6.5.1 创建默认相册目录 .....	157

6.5.2	设置标题栏	160
6.5.3	处理 New 菜单	161
6.6	保存文件对话框	162
6.6.1	写入相册数据	163
6.6.2	将相册保存为新的文件	166
6.6.3	保存现有相册	168
6.7	打开文件对话框	169
6.7.1	读取相册数据	169
6.7.2	打开相册文件	171
6.8	小结	172
<b>第 7 章</b>	<b>绘图与滚动</b>	<b>174</b>
7.1	Form 类的层次结构	174
7.1.1	ScrollableControl 类	175
7.1.2	Form 类	176
7.2	图像绘制	177
7.2.1	删除 PictureBox 控件	177
7.2.2	处理图像菜单	177
7.2.3	实现 Stretch To Fit 选项	180
7.2.4	实现 Scale To Fit 选项	183
7.2.5	当调整窗体大小时重新绘制	187
7.3	自动滚动	190
7.3.1	滚动属性	191
7.3.2	实现自动滚动	191
7.4	面板	193
7.4.1	添加面板	194
7.4.2	更新菜单处理程序	195
7.4.3	绘制状态栏面板	196
7.4.4	绘制图像	197
7.5	小结	200
<b>第 8 章</b>	<b>对话框</b>	<b>201</b>
8.1	消息框	201
8.1.1	MessageBox.Show 方法	203
8.1.2	创建 OK 对话框	204
8.1.3	创建 YesNo 对话框	205
8.1.4	创建 YesNoCancel 对话框	207
8.2	Form.Close 方法	209
8.2.1	Close 和 Dispose 方法之间的关系	209



8.2.2 拦截 Form.Close 方法	211
8.3 模态对话框	213
8.3.1 为图片添加标题	214
8.3.2 保存标题值	215
8.3.3 创建 CaptionDlg 窗体	217
8.3.4 添加属性到 CaptionDlg 窗体	222
8.3.5 在 MainForm 类中显示对话框	224
8.4 非模态对话框	226
8.4.1 创建 PixelDlg 窗体	227
8.4.2 添加类成员到 PixelDlg	228
8.4.3 显示非模态的 PixelDlg 窗体	230
8.4.4 更新 PixelDlg 窗体	232
8.4.5 随着鼠标的移动更新 PixelDlg	233
8.5 小结	235
<b>第 9 章 基本控件</b>	<b>236</b>
9.1 窗体继承	237
9.1.1 创建基类窗体	237
9.1.2 创建派生窗体	241
9.2 标签和文本框	243
9.2.1 扩充 Photograph 类	244
9.2.2 创建 PhotoEditDlg 面板区域	249
9.2.3 创建多行文本框	252
9.2.4 为主窗体添加 PhotoEditDlg 对话框	256
9.2.5 使用 TextBox 控件	258
9.3 按钮类	260
9.3.1 扩展 PhotoAlbum 类	262
9.3.2 使用新的相册设置	265
9.3.3 创建 AlbumEditDlg 面板区域	268
9.3.4 使用单选按钮	269
9.3.5 使用复选框按钮	273
9.3.6 为主窗体添加 AlbumEditDlg 对话框	278
9.4 小结	281
<b>第 10 章 列表控件</b>	<b>282</b>
10.1 列表框	283
10.1.1 创建列表框	283
10.1.2 处理被选取的项	289
10.2 多选列表框	291

10.2.1	启用多选功能	292
10.2.2	处理 Move Up 和 Move Down 按钮	294
10.2.3	处理 Remove 按钮	297
10.3	组合框	299
10.3.1	创建组合框	300
10.3.2	处理被选项	302
10.4	组合框的编辑	304
10.4.1	替换 Photographer 控件	305
10.4.2	动态更新组合框	307
10.5	自绘制列表	309
10.5.1	添加一个上下文菜单	310
10.5.2	设置选项的高度	311
10.5.3	绘制列表项	313
10.6	小结	316
<b>第 11 章</b>	<b>更多控件</b>	<b>317</b>
11.1	Tab 控件	317
11.1.1	TabControl 类	318
11.1.2	创建 tab 控件	319
11.2	Tab 页面	322
11.2.1	动态创建 tab 页面	323
11.2.2	利用 Visual Studio 创建 tab 页面	325
11.3	日期和时间	328
11.3.1	日期和时间	329
11.3.2	自定义 DateTimePicker 控件	331
11.4	日历	332
11.4.1	添加 MonthCalendar 控件	333
11.4.2	初始化日历控件	335
11.4.3	处理日历控件中的鼠标单击事件	337
11.5	小结	341
<b>第 12 章</b>	<b>.NET 分类</b>	<b>342</b>
12.1	键盘事件	342
12.1.1	处理 KeyPress 事件	343
12.1.2	处理其他键盘事件	344
12.2	鼠标事件	346
12.2.1	MouseEventArgs 类	346
12.2.2	处理鼠标事件	347
12.3	图像按钮	351