

Windows Forms 编程实战

Windows Forms
in Action

(美) Erik Brown 著
徐波 等译



Windows Forms

编程实战

Windows Forms
in Action

(美) Erik Brown 著
徐波 等译



本书由浅入深地介绍Windows Forms编程的技巧和各种实用方法。本书先详细介绍了菜单、状态条、可复用类库、文件对话框、文本框、按钮、列表框、组合框、选项卡控件等Windows Forms基础知识，然后在高阶应用中还介绍了自定义控件、列表视图、资源管理器、树型视图、多文档界面和数据绑定等扩展性内容。本书内容详实，实用性极强，通过大量的图表和程序实例形象而立体地说明了各种控件的操作和需要注意的重难点，可以使读者领略到使用Windows Forms编程的轻松和乐趣。

本书适合计算机编程开发人员参考。

Erik Brown: Windows Forms in Action (ISBN: 1932394-65-6)

Authorized translation from the English language edition published by Manning Publications Co..

Copyright © 2007 by Manning Publications Co..

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2007 by China Machine Press.

本书中文简体字版由Manning Publications Co.授权机械工业出版社独家出版。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。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2007-0288

图书在版编目（CIP）数据

Windows Forms编程实战 / (美) 布朗 (Brown, E.) 著；徐波等译. —北京：机械工业出版社，2008.1

书名原文：Windows Forms in Action

ISBN 978-7-111-22114-2

I. W… II. ①布… ②徐… III. 窗口软件，Windows—程序设计 IV. TP316.7

中国版本图书馆CIP数据核字（2007）第122084号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：吴怡 周茂辉

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2008年1月第1版第1次印刷

186mm×240mm·38.75印张

定价：75.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线电话（010）68326294

译者序

本书是美国.NET技术专家Erik Brown的力作。本书以生动翔实的语言，描述了在Visual Studio C# 2005开发环境中进行Windows Forms编程的各方面细节。

本书紧紧围绕Windows Forms这个主题，对各种Windows Forms特性进行了极为详细的介绍，而对其他细节涉墨甚少。本书并没有介绍面向对象编程，也没有花太多的篇幅介绍C#编程语言的细节（只是在附录A简单地描述了C#语言的结构）。关于Visual Studio C# 2005开发环境的许多使用技巧，则分散在具体的应用场合进行说明，而不是开辟专门的章节予以介绍。根据译者的体验，前者的效果要好得多。

如果读者想寻找一本教材，认真学习Windows Forms编程的细节，那么本书是一个非常理想的选择。本书以一个相册应用程序为主线，在学习过程中不断融入新的Windows Forms特性，使它从一个简单的只能显示单张照片的应用程序发展成为一个具有丰富特性的产品级相册应用程序，能够实现高级的浏览和打印功能。

本书的另一个特性是它所采用的表格代码，把实现每个特性所需要的代码以表格的形式分步骤显示，虽然对译者而言增加了很大的工作量，但对于读者而言，这样的代码形式看起来非常清楚，能够立即明白完成某一步骤需要哪些代码。

如果说这本书有什么缺陷的话，就是它的整体性非常强，需要读者耐心地从头到尾读完这本书，才能收到良好的效果。如果读者只是想了解某个Windows Forms特性（例如TreeView控件）的内容，可能因为本书内容的连贯性太强而无从下手。从这个意义上说，这本书更适合作为教材，而不是作为参考手册。

不管怎样，这是一本非常出色的Windows Forms专著，它在这个领域所发掘的深度远远超过普通的.NET编程书籍。如果读者能够耐心地读完本书，一定可以取得巨大的收获。

参加本书翻译工作的人员有徐波、陈永军、李福军、毛玉山、杨洁、张瑜、应巧敏、汤晓萍、胡柯铭、谢凯、郑培红、顾沂、袁静义、陈岚、张青等。

前 言

当我还在学校的时候，当我毕业后在不同的创业公司打拼的时候，我深信UNIX将统治这个世界，从来不曾想到自己会站在计算机行业的“公敌”Microsoft一边。我饶有趣味地发现自己出版了不止一本书，为Microsoft的技术传经布道。我想读者应该知道自己该走什么路，而我现在已经完全投身于Windows的开发之中。幸运的是，我对自己的转变深感快乐，而我原本以为自己是误入歧途。

至于出版这本书的想法，要想实现起来却并不容易。创作所带来的激情很快就会消退，你会意识到自己不仅想写一本书，而且想写一本好书。这会改变你的思维和方法，因为你必须付出极大的精力和时间使它变“好”。然后有一天，你又会发现一个事实，你不仅想写一本好书，而且想写一本人们喜欢并且愿意购买的书。

这正是我写作本书第1版和第2版的经历。第1版获得了相当的成功，因此为.NET 2.0推出第2版不仅顺理成章，而且看上去似乎很容易。遗憾的是，Microsoft那帮聪明的伙计们在Windows Forms中添加并修改了相当多的内容，使我感觉就像完全重写一本新书一样。

读者可能会以为写作第2版要比第1版更为容易，其实不然。虽然我知道如何对正文进行组织，并且知道基本的格式是一样的。但是，作为一名“经验更丰富的作者”，我必须对本书以及书中的示例程序进行更合理的设计。每章的内容更为简捷，我试图利用自己在Unisys公司所积累的.NET企业级应用程序的开发经验，实现更好的程序设计。我在主要的MyPhotos应用程序之外添加了一些分支，覆盖了一些其他材料，为一些有趣的话题提供了示例代码。

另外一个原因是当我编写本书的第1版时还是一名顾问，因此我可以脱开几个星期，全身心地投入到写作之中。现在，作为一名全职的项目经理，找出充裕的时间来完成第2版的写作并不是一件容易的事情。我的家庭和朋友对这两个版本的写作提供了相当大的支持，对此我深表谢意。

最后，我发现自己对本书的第2版非常满意。有些关键的概念，例如加密、自定义控件、进度条等，以相当自然的方式融入到本书之中。我喜欢第2版的章节规划，并且从第1版的18章扩展到第2版的23章。我对索引部分也是相当满意，相信读者会发现它比第1版的索引更为有用。第1版和第2版的另一处较大修改是书名。这个版本被命名为“Windows Forms in Action”，以区别于其他书籍，并且与Manning新的“In Action”系列保持一致。书中大量使用的“活动-结果”表格正是这个新书名的由来。新的书名、新的封面、新的章节，就像推出一本新书一样。希望读者能够喜欢！

致谢

我不时地感到惊奇，为什么生活中那些纷乱的头绪能够产生真实的结果，摆在读者面前的这本书就是一个例子。虽然本书签署的是我的名字，但是还有许多已知或未知的人们为这本书作出了巨大的贡献。我特别感谢Manning出版社的排版人员Sydney Brown，她为本书的第1版的外观质量提供了非常大的帮助。Sydney现在已经没有和我们在一起了，但我们会一直怀念她，怀念她对本书以及其他Manning出版物所作出的贡献。

感谢我的家庭，感谢妻子Bridgett的耐心和爱。感谢Katie和Sydney给我的生活所带来的笑声和欢乐。感谢父母David和Janet，感谢我的老师们，以及在生活中给我提供支持的其他人。特别感谢Steve Cox和David Cobb，很久以前，他们带领我走入了计算机编程的殿堂。感谢我的许多朋友，感

谢他们所提供的只能朋友才能给予的支持和鼓励，尤其是Jean Siegel、Tommy McCracken、Tony Mason和Reggie Blue。

来自世界各地的许多审稿人奉献了他们的时间和精力阅读本书各章的早期版本，我向他们表示特别的感谢。如果没有他们的帮助和努力，本书肯定会逊色不少。尤其要感谢Dave Corun，他在本书最终出版之前对所有的正文进行了精心的技术审查。另外，非常感谢Erick Ellis以及Microsoft Windows Forms小组的其他成员，感谢他们在.NET 2.0框架中所作的努力，还感谢他们为本书所提供的许多建设性的意见。其他审稿人也向本书提供了具有深远意义的评价或想法，他们是Andrew Deren、Berndt Hamboeck、Jack Herrington、Shane Jervis、Joe Litton、Robert Marshall、Robert McGovern、Mark Monster、Alan Newson、Vipul Patel、Heath Stewart、Gary Udstrand和Andrew Varner。

我还想感谢本书第1版的审稿人，这是2001年的事情，感谢Steve Binney、Mark Boulter、Drew Marsh、Josh Mitts和Kunle Odutola。他们的建议对本书的第1版起到了关键的作用，并为第2版的重点内容和布局奠定了基础。

最后，我想感谢Manning出版社中那些我未曾谋面，但确实是在本书的出版过程中提供了关键支持的人们。尤其是Susan Capparelle，他从我最初的建议中看到了一些有价值的东西。Marjan Bace为本书的第1版提供了自己的想法和思路，并在对第2版也提供了一些建议。Karen Tegtmeyer负责协调所有的审稿人以及他们的反馈意见。Leslie Haimes为本书设计了非常酷的封面。Mary Piergies审查了整个项目，并回答了我的许多问题。Ann Navarro、Susannah Pfalzer和Lianna Wlasiuk帮助我完成本书的结构设计。Liz Welch对最终的文档进行了精心的复制和编辑。Barbara Mirecki和Katie Tennant对最后的手稿进行了细致的校对。Dottie Marsico为附录C创建了页号，并完成了本书最终的排版。Helen Trimes在本书的市场销售方面作出了贡献。最后，感谢Lee Fitzpatrick继续为我签署版税支票。

引 言

.NET框架所涵盖的主题相当广泛，用一本书的篇幅覆盖所有的主题是不可能的。本节介绍本书的焦点，并对正文的内容以及约定进行了简单的介绍。本节的末尾描述了本书的在线论坛，读者可以通过它了解和本书有关的各种问题和评论，并了解如何从Internet下载本书所使用的源代码。

介绍.NET

Microsoft善于改造自己。它最初是一家纯粹的操作系统公司，随后对业务进行了拓展，推出了像Word和Excel这样的办公软件，后来又通过Internet Explorer把目光瞄向了Internet。创建.NET的动力很可能来自Internet和Java作为大规模计算机解决方案的通用环境所获得的成功。.NET框架的第1个版定义了两个重要的概念。第1个是一种全新的语言C#，第2个是基于Windows的解决方案的标准框架。

理解C#的最简单方法也许是想象有人把C++的所有恼人之处都写下来，然后设计一种语言，把这些恼人之处都去掉。例如，在C++中，处理指针是件痛苦的事情，有很多代码错误（例如if (x = 5)）并没有被编译器所捕捉，操纵字符串比较困难，没有一种很好的方式把一个对象安全地“向下转换”为派生类型的对象。C和C++的前驱B和BCPL语言并没有定义正式的类型系统，这是因为考虑到在这些语言中整数、指数和字符的互通本质。

C#语言进行了精心设计，既保留了C和C++语言的灵活性，又对类型系统以及语言的语法进行正式化。C++中的许多常见运行时错误在C#中变成了编译错误。其他明显的区别是C#增加了内置的字符串类型，并且不再使用全局变量，把关键的系统和应用程序错误集成了一种通用的异常模型中。本书的附录A对C#语言的语法、关键字和特性进行了简单介绍。后面的“.NET 2.0的新特性”一节对.NET框架2.0版本所作的主要修改进行了总结。

.NET框架为大多数Microsoft的开发技术和环境提供了一个公共的主题，并随着时间而发生变化。除了把公司聚集在一个称为.NET的单一品牌之下这个好处之外，.NET框架还具有一个重要的技术目标。这个框架在本质上是在Windows平台上所运行的应用程序的一个执行环境。

Java环境的核心是提供一个环境，允许在任何操作系统中运行由单一的语言所编写的程序。.NET框架的核心则是提供一个环境，允许在单个操作系统中运行使用任何语言所编写的程序。很难说哪个更好，作者在这里只是为了指出Java和.NET的基本目标是不同的。

.NET框架是以命名空间的形式组织的。第1章将讨论这个概念，但是读者在这里只要简单地知道命名空间定义了相关联的类以及其他类型的逻辑集合。关于.NET框架所支持的各个命名空间的总结，可以参考本书后面的附录B。

命名空间对.NET框架所支持的大量对象集合进行结构化。对于作者而言，它指引了写作本书的方向和焦点。本书的焦点是System.Windows.Forms命名空间，可以简单地称为Windows Forms。Windows Forms应用程序就是由Windows操作系统所执行的程序，这种程序所使用的用户界面特性正是Windows桌面用户极为熟悉的。

本书试图对Windows Forms进行系统的介绍，将介绍这个命名空间所定义的大多数类型。附录C提供了Windows Forms命名空间的类图，并提供了参考表格或参考章节，帮助读者快速找到每个类或类型在书中定义的地方。

内容简介

本书共包括23章，分为3个部分。

第1部分：欢迎进入Windows Forms世界

本书的第1部分介绍C#的基本概念，尤其是在Windows Forms的概念。第1章使用一个文本编辑器创建了一个与图1相似的应用程序。我们讨论一些关键的C#术语，讨论.NET框架是如何执行Windows Forms应用程序的，并讨论用C#编写的Windows Forms程序的结构。

在第2章中，我们开始使用Microsoft的图形开发环境Visual Studio，利用.NET框架创建应用程序。第2章重新创建了第1章使用手工方式创建的那个应用程序。我们把这个应用程序称为MyPhotos。

第2部分：Windows Forms基础知识

第2部分开始对Windows Forms命名空间中的类进行系统的描述。我们继续对第2章所创建的MyPhotos应用程序进行开发，图2显示了到第16章时这个应用程序的外观。读者可以看到，第2部分覆盖了创建Windows Forms应用程序所需要的核心用户界面组件，包括菜单、工具条、对话框窗口、文本框和组合框等。

MyPhotos应用程序显示一个相册的内容，相册由一个或多个图像文件（即照片）组成。这个应用程序把每个相册存储在一个文件中，允许用户一次查看一张照片，并允许用户编辑相册和照片的属性。

第2部分还介绍了和Windows应用程序开发有关的各种重要概念，包括可复用类库、文件流、加密、后台处理、进度条和资源文件。第2部分的介绍部分提供了对从第3章到第16章所覆盖的话题的详细列表。

第3部分：高级Windows Forms

本书的第2部分讨论更高级的话题，例如自定义控件、列表视图和拖放功能。这个部分使用相册的概念创建了一些不同的应用程序，包括一个与Windows资源管理器相似的用于浏览相册的应用程序，以及一个数据驱动的应用程序，把Windows Forms控件的内容与取自数据源的值进行绑定。

图3显示了第23章时MyPhotos应用程序的主窗口。这个应用程序被转换为多文档界面，可以显示多个相册。

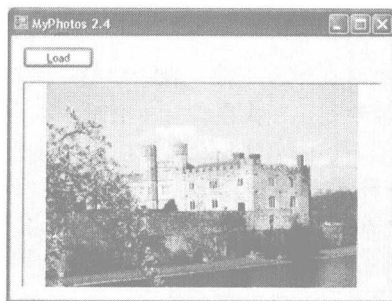


图1 在第1部分所出现的MyPhotos应用程序

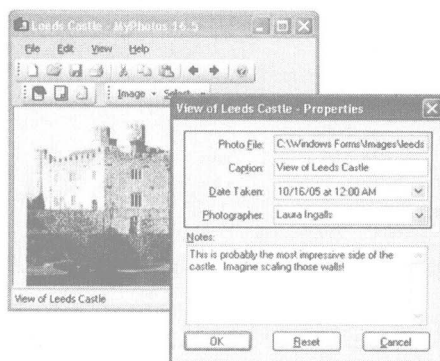


图2 第16章的MyPhotos应用程序。这张图显示了主窗口以及用于编辑一张特定照片的属性对话框

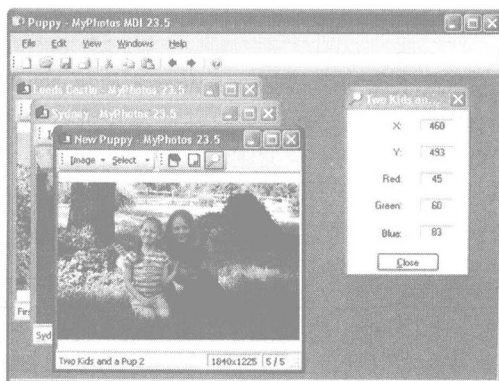


图3 第23章的MyPhotos应用程序。一个父窗口包含了一个或多个在本书的第2部分所创建的MyPhotos窗口

这个应用程序还添加了一些其他特性，例如在相册之间拖放照片以及在应用程序中显示本书的网站。

本书的读者

和所有作者一样，我希望每个人都能阅读本书。读的人越多，作者越感欣慰。但是，考虑到广大读者的利益，我在编写本书时，心中设想了三种类型的读者：

- 对使用.NET开发桌面应用程序感兴趣的Windows程序员。
- 熟悉.NET或C#，并有兴趣进一步学习Windows Forms编程的开发人员。
- 在Windows应用程序的开发上没什么经验甚至完全没有经验的C++程序员。

本书的第1版还适用于对有兴趣使用C#进行Windows应用程序编程的Visual Basic和Java程序员。如果读者符合这两个条件之一，也适合使用本书。

另外，需要指出的是，本书的内容只涉及.NET框架的一部分，即System.Windows.Forms命名空间。本书还提供了和C#和Visual Studio有关的大量信息，尤其是使用Visual Studio创建每个示例程序时所需要的详细步骤。为了方便读者了解其他信息，附录D列出了C#和.NET的资源列表。本书最后的参考书目部分列出了一些书籍，覆盖了应用程序开发的各个方面。

约定

表格约定

本书使用大量图表来描述正文的信息。从第2章开始，对示例程序进行的所有修改都是在“活动-结果”表格中说明的。这种表格以“手把手”的方式，提示如何在Visual Studio中对示例程序进行修改。下面是这种表格的一个例子。

描述这些步骤所描述的任务		
	活 动	结 果
1	描述需要执行的活动。	描述活动的结果。这可以是文本描述、图形或最终代码。
2	需要执行的第2个活动。 如何实现： a. 执行所描述的活动所需要的详细步骤。 b. 更多的步骤（如果需要）。	第2个结果。 备注：对结果的注释或解释。

除了这些表格之外，本书还使用.NET表格对.NET中的一些类和其他类型进行总结。这些表格对附近章节所讨论的类或其他类型进行总结，当读者以后回顾相关内容时，可以把它们作为快速参考。关于.NET框架的这些成员以及其他成员的详细信息，可以参阅在线文档。

当然，绝大多数.NET表格所描述的是Windows Forms命名空间的成员。.NET表格1使用PictureBox类作为例子说明了这种格式。

.NET表格1 PictureBox类

PictureBox类表示能够显示一幅图像的控件。当图像大于控件的客户区域时，并不会自动出现滚动条。因此，必须使图像正确地出现在控件的内部。这个类位于System.Windows.Forms命名空间，是从Control类继承的。参见.NET表格3-1，了解从这个基类所继承的成员。

公共属性	BorderStyle ErrorImage Image	获取或设置控件的边框风格。 获取或设置当错误发生时所显示的图像。 获取或设置图片框的图像。
------	------------------------------------	---

(续)

公共属性	SizeMode WaitOnLoad	获取或设置图像的显示方式。 获取或设置图像是否异步加载。
公共方法	CancelAsync	取消一个异步图像加载。
公共事件	SizeModeChanged	当SizeMode属性的值发生变化时发生。

注意这些表格的下面这些特性：

- 起始段落定义了这个类的用途、包含这个类的命名空间以及它的基类。如果未指定基类的命名空间，则被描述的类和基类位于同一个命名空间。如果未提示基类，这个被描述的类就是从System.Object类派生的。
- 这个类的成员（即这个类特有的属性、方法和事件[⊖]）在初始段落之后总结。从基类继续的成员并不在.NET表格中显示。在.NET表格1中，共显示了7个成员：5个属性、1个方法和1个事件。
- .NET框架2.0版本新增加的成员用斜体显示。在.NET表格1中，ErrorImage和WaitOnLoad属性以及CancelAsync方法是.NET 2.0新增加的。
- 如果是.NET 2.0新增加的类或其他类型，会在介绍的前面增加“.NET 2.0的新特性”前缀。另外，这些类型的所有成员都用斜体显示，以便与其他表格的新成员保持一致。

“尝试一下”建议

本书的另一个约定是使用特殊的段落强调一些话题，对Windows Forms和.NET框架进行进一步的探索。

尝试一下

这种段落提供了读者可以使用以前的章节所讨论的材料对示例程序所进行的修改。“尝试一下”段落向读者提供了对相关的话题进行更深一步理解的机会。本书并没有提供这些小节的代码，但可以从本书的网站中找到它们。

尝试一下段落遍及全书，有时候它所讨论的类成员并没有直接在示例代码中使用。

代码约定

最后值得一提的是本书所采用的各种代码约定。Microsoft定义了一些这方面的标准，读者可以通过google (www.google.com) 搜索“design guidelines for class library developers”，或者访问“<http://msdn.microsoft.com/library/en-us/cpgenref/html/cpconnetframeworkdesignguidelines.asp>”。本书遵循了其中的大部分标准，但也有一些例外和偏好，下面予以说明。

首先，作者以不同的方式处理Visual Studio中的变量和手写代码中的变量。在Visual Studio中，作者在变量前面添加了表示控件类型的前缀，例如btn表示按钮、txt表示文本框等。在手写代码中，作者喜欢在私有变量前面添加下划线前缀，清楚地表明它们是私有变量。采用这种做法的一个主要原因是使自动产生的事件处理方法更容易阅读，不使用下划线，并且更容易通过代码窗口来访问，因为它们与方法下拉列表中是按照控件的类型按字母顺序排列的。

考虑到书的篇幅，作者采用了一些简写方法，使代码更容易阅读，并且减少代码行的宽度和数量。作者避免使用this关键字，如果if或for循环语句体内只有1条语句，就不使用花括号。另外，使用缩写而不是完整的单词来表示变量名（例如，用“pwd”代替“password”）。这可以使代码更容易复制和阅读，并且使本书更容易进行排版，因为这些约定减少了代码的长度和宽度。

[⊖] 这些成员的含义将在本书的第1部分定义。

虽然作者尽量保持一致，但时不时仍然会违背这些以及其他约定。我可以说什么呢？我是个有时会违反常规的人，即使是在自己的书里。

源代码下载

凡是本书的购买者，都可以从Manning网站下载本书所有程序的源代码。读者可以访问www.manning.com/eebrown2，根据提示下载源代码。

Author Online

读者可以访问私人的Internet论坛Author Online，这是购买本书所附带的一项服务。读者可以访问本书的网站www.manning.com/eebrown2，了解访问这个论坛的详细规则、订阅或访问这个论坛、获取每个章节的代码以及查看本书的更新和修正材料。读者可以对本书进行评论，不管是赞扬还是批评。读者也可以询问技术问题，接受作者以及其他Windows Forms程序员的帮助。

Manning所作出的贡献就是提供了一个场所，使读者之间、读者和作者之间可以进行有益的对话。它对作者参加对话的次数并没有提供强制要求，因为这些活动是自愿的，并没有报酬。

.NET 2.0的新特性

本书实际上是《Windows Forms Programming in C#》的第2版，基于.NET框架的第2个版本。这里对.NET 2.0所提供的新特性进行简介，并介绍这个版本所覆盖的新材料。

无论是.NET 2.0还是本书，都发生了很大的变化。我们首先从C#说起。最大的变化之一是增加了泛型，它的精神与C++的模板相似，但并不相同。.NET框架还添加了一些泛型集合类。第5章讨论了泛型和泛型集合。C# 2.0的其他修改包括部分类，就是一个类的定义跨越多个文件；匿名方法能够把一个代码块作为参数传递。读者可以通过.NET文档了解更详细的信息。Windows Forms命名空间也发生了不少变化。所有的类现在（最终！）都支持极为有用的Tag属性，在控件中执行自定义绘制的功能也得到了极大的扩展。其他一些特性在.NET框架中得到了更好的支持，例如双缓冲、后台处理和声音文件的回放。

对菜单、状态栏和工具条的支持通过ToolStrip和ToolStripItem类进行了重写。旧的界面仍然可以使用，但推荐使用新的类。工具条是一个相当大的增强，将在第3、4、16和20章讨论。其他的新类包括MaskedTextBox、SplitContainer和WebBrowser控件，它们都将在本书中进行讨论。.NET 2.0通过TableLayoutPanel和FlowLayoutPanel类提供了更高级的窗体布局功能。第21章详细讨论了DataGridView以及相关的类，它们极大地改进了数据的显示。和.NET 1.x相比，第22章所讨论的BindingSource类简化了所有控件对数据绑定的支持。

Visual Studio也进行了改进，在设计器中添加了许多增强的布局帮助功能，包括快捷标签菜单和自定义控件的自动确认。另一个有用的特性是使用C#的部分类功能，把Visual Studio所生成的代码放在一个单独的文件中。在本书的学习过程中，将逐渐介绍Visual Studio的这些特性以及其他特性。

所有这些变化使本书也发生了极大的变化。虽然有些材料仍然和第1版相同，但所有的内容都进行了审查，大部分都进行了重写。章节的内容更为简洁，第1版的有些章节在第2版中被分成2章。第1版共有18章，第2版增加到23章。新书一般会增加新的主题。第2版所增加的主题（或者在第1版中只进行了很少的讨论）包括流、加密、用户控件、进度条、自定义控件和应用程序的部署。

本书的总体方法和原来相似，它仍然是一本适合从头到尾阅读的教程，并对示例程序中的每一行代码进行了详细的解释。作者相信示例程序的整体设计得到了极大的改进，代码的逻辑和表现形

式得到了更好的分离，更适合用户使用，并且可以方便地使用自定义控件。

封面说明

本书封面图画是一个中世纪步兵，他是一位弩手“Pieton, Arbalétrier”。这幅画取自Sylvain Maréchal所收集的历史和区域服饰画集。这个四卷的画集最初出版是在1788年（这是于巴黎出版的，法国大革命之前的一年），每张图都是专家绘制，并手工着色的。

有时候，很难把一本计算机书籍与其他计算机书籍分清楚。Manning出版社在计算机书籍方面有许多发明和创新，根据几个世纪以前的丰富历史以及服饰的多样性来设计书籍的封面，显示很久以前未知艺术家的绘画。如果这些艺术家地下有知，肯定会为他们的作品在两百年以后出现在书籍的封面上而吃惊不已。

目 录

译者序
前 言
引 言

第1部分 欢迎进入Windows Forms世界

第1章 初识Windows Forms	2
1.1 用C#编程	2
1.1.1 命名空间和类	4
1.1.2 构造函数和方法	5
1.1.3 C#的类型	5
1.1.4 Main方法	7
1.1.5 Application类	8
1.1.6 程序的执行	9
1.2 Windows Forms控件	9
1.2.1 using指令	11
1.2.2 控件类	12
1.2.3 Controls属性	13
1.3 加载文件	13
1.3.1 事件	15
1.3.2 OpenFileDialog类	16
1.3.3 Bitmap图像	18
1.4 更改窗体的大小	19
1.4.1 Anchor属性	20
1.4.2 Dock属性	21
1.5 回顾	23
第2章 欢迎使用Visual Studio	24
2.1 使用Visual Studio编程	24
2.1.1 创建项目	25
2.1.2 执行程序	27
2.1.3 观察源代码	27
2.2 Windows Forms控件	31
2.2.1 AssemblyInfo文件	31
2.2.2 对窗体进行重命名	33

2.2.3 工具箱窗口	34
2.3 加载文件	38
2.3.1 在Visual Studio中处理事件	38
2.3.2 处理异常	41
2.4 改变窗体的大小	43
2.4.1 设置Anchor属性	43
2.4.2 设置Dock属性	44
2.5 回顾	44
第2部分 Windows Forms基础知识	
第3章 菜单	49
3.1 控件和容器	49
3.1.1 控件类	49
3.1.2 容器类	50
3.2 菜单的本质	51
3.2.1 菜单术语	52
3.2.2 .NET中的菜单	52
3.3 菜单栏	53
3.3.1 添加菜单条	54
3.3.2 添加菜单项	56
3.3.3 添加下拉菜单项	58
3.4 菜单处理	62
3.4.1 通过设计窗口添加事件处理方法	62
3.4.2 通过属性窗口添加事件处理方法	63
3.5 回顾	65
第4章 上下文菜单和状态条	66
4.1 上下文菜单条	66
4.1.1 创建上下文菜单	67
4.1.2 向上下文菜单添加菜单项	68
4.1.3 共享上下文菜单	69
4.2 下拉事件和事件参数	71
4.2.1 处理子菜单项点击	72
4.2.2 在显示子菜单之前对它进行更改	75

4.3 状态条	77	6.5 回顾	133
4.3.1 创建状态条	77	第7章 对话框	135
4.3.2 添加状态条标签	78	7.1 消息框	135
4.4 回顾	82	7.1.1 创建OK消息框	136
第5章 可复用类库	84	7.1.2 创建YesNo消息框	137
5.1 类库	84	7.1.3 创建YesNoCancel消息框	139
5.1.1 创建类库	85	7.2 Form.Close方法	141
5.1.2 使用命令行工具	87	7.2.1 比较Close和Dispose	141
5.1.3 创建Photograph类	88	7.2.2 拦截Form.Close方法	143
5.1.4 定义类的属性	91	7.3 窗体和面板	145
5.2 object类	93	7.3.1 Form类	145
5.2.1 比较object和Object	93	7.3.2 创建对话框	146
5.2.2 重写object类的方法	94	7.4 无模式对话框	147
5.3 接口	96	7.4.1 创建无模式对话框	148
5.3.1 接口和集合	96	7.4.2 向窗体添加面板和标签	149
5.3.2 理解集合类	97	7.4.3 更新PixelFormat控件	152
5.4 泛型	101	7.4.4 显示无模式对话框	155
5.4.1 泛型类	101	7.4.5 更新PixelFormat窗体	157
5.4.2 理解泛型集合	102	7.4.6 追踪鼠标移动	158
5.4.3 资源的销毁	105	7.5 回顾	159
5.5 回顾	107	第8章 文本框	160
第6章 文件和常用对话框	109	8.1 窗体的继承	160
6.1 标准菜单	109	8.1.1 创建基类窗体	161
6.1.1 更改菜单栏	109	8.1.2 创建派生类窗体	163
6.1.2 管理相册	112	8.2 标准文本框	165
6.2 常用文件对话框	117	8.2.1 添加文本框	165
6.2.1 创建新相册	117	8.2.2 添加多行文本框	169
6.2.2 打开相册	119	8.2.3 连接对话框	173
6.2.3 保存相册	121	8.3 掩码文本框	174
6.2.4 打印和退出	123	8.3.1 处理KeyPress事件	174
6.3 流和写入器	124	8.3.2 设置文本的掩码	176
6.3.1 Stream类	124	8.4 回顾	180
6.3.2 把相册写入磁盘	126	第9章 按钮	181
6.3.3 从磁盘中读取相册	128	9.1 下压按钮	181
6.4 相册管理	130	9.1.1 Button类	182
6.4.1 添加和删除图像	130	9.1.2 扩展PhotoAlbum类	183
6.4.2 浏览相册	131	9.1.3 存储相册数据	184
6.4.3 显示相册的状态	133	9.2 单选按钮	186

9.2.1	RadioButton类	186	11.1.2	创建一个新的解决方案	231
9.2.2	创建相册对话框	187	11.1.3	重构应用程序的代码	231
9.3	复选框按钮	189	11.2	简单的列表框	236
9.3.1	CheckBox类	189	11.2.1	ListBox类	237
9.3.2	使用复选框按钮	190	11.2.2	显示数据源	238
9.3.3	完成AlbumEditDialog窗体	191	11.2.3	选择列表元素	241
9.3.4	再次连接对话框	193	11.3	多选列表框	243
9.4	工具提示和错误提示	194	11.3.1	启用多选功能	243
9.4.1	ToolTip类	195	11.3.2	改变列表项的顺序	246
9.4.2	显示工具提示	195	11.3.3	删除列表元素	249
9.4.3	创建扩展器提供程序	198	11.4	回顾	250
9.4.4	ErrorProvider类	200	第12章	组合框	251
9.5	回顾	202	12.1	标准组合框	251
第10章	处理用户输入和加密	203	12.1.1	ComboBox类	251
10.1	键盘事件	203	12.1.2	创建组合框	252
10.1.1	3种事件	203	12.1.3	选择列表框中的列表项	254
10.1.2	处理KeyPress事件	203	12.1.4	修改数据源	255
10.1.3	处理KeyDown和KeyUp事件	205	12.2	可编辑的组合框	257
10.1.4	预处理命令键	206	12.2.1	再次创建组合框	257
10.2	鼠标事件	207	12.2.2	动态更新组合框	260
10.2.1	MouseEventArgs类	208	12.3	自动完成	261
10.2.2	使用鼠标事件实现 FlybyTextProvider	209	12.3.1	理解自动完成	261
10.2.3	使用FlybyTextProvider类	212	12.3.2	在组合框中使用自动完成功能	264
10.3	加密	214	12.4	列表项的格式化	264
10.3.1	Cryptography命名空间	214	12.4.1	格式化接口	264
10.3.2	对数据进行加密	215	12.4.2	实现IFormattable接口	265
10.3.3	写入加密数据	217	12.4.3	在列表控件中使用格式化	268
10.3.4	读取加密数据	218	12.4.4	实现自定义格式化	270
10.3.5	存储经过加密的相册	219	12.5	回顾	272
10.4	链接标签	222	第13章	选项卡控件和选项卡页面	273
10.4.1	LinkLabel类	222	13.1	选项卡控件的基础知识	273
10.4.2	保存加密相册	223	13.1.1	TabControl类	273
10.4.3	打开经过加密的相册	224	13.1.2	创建选项卡控件	274
10.5	回顾	227	13.2	选项卡页面的基础知识	277
第11章	列表框	229	13.2.1	TabPage类	277
11.1	列表控件	229	13.2.2	创建选项卡页面	277
11.1.1	ListControl类	229	13.2.3	在选项卡页面中封装现有的控件	280
			13.3	自绘制的选项卡	282

13.3.1 DrawItem事件	282	15.4 计时器	335
13.3.2 绘制页面的选项卡	284	15.4.1 Timer类	336
13.3.3 使用自绘制的选项卡控件	286	15.4.2 使用计时器	336
13.4 自绘制的列表框	287	15.5 回顾	338
13.4.1 自定义的列表项	287	第16章 工具条	339
13.4.2 设置列表项的大小	288	16.1 工具条的基础知识	339
13.4.3 绘制列表项	289	16.1.1 ToolStrip类	340
13.5 回顾	291	16.1.2 创建工具条	341
第14章 日期、日历和进度条	292	16.2 工具条项的基础知识	342
14.1 显示日期和时间	292	16.2.1 ToolStripItem类	342
14.1.1 DateTimePicker类	292	16.2.2 添加标准工具条项	342
14.1.2 创建DateTimePicker控件	294	16.2.3 工具条项的对齐、溢出和其他特性	345
14.2 日期和时间的格式	296	16.3 图像列表	346
14.2.1 对日期和时间值进行格式化	296	16.3.1 ImageList类	346
14.2.2 自定义DateTimePicker控件	299	16.3.2 创建图像列表	346
14.3 日历	300	16.3.3 添加自定义工具条按钮	348
14.3.1 MonthCalendar类	300	16.4 工具条容器	349
14.3.2 UserControl类	301	16.4.1 ToolStripContainer类	350
14.3.3 创建日历控件	302	16.4.2 在容器中嵌入一个工具条	351
14.3.4 初始化日历	303	16.5 特殊的按钮项	353
14.3.5 在日历控件中处理鼠标点击	305	16.5.1 切换按钮	353
14.4 进度条	308	16.5.2 下拉按钮	356
14.4.1 创建image用户控件	308	16.5.3 分隔按钮	357
14.4.2 ProgressBar类	311	16.6 回顾	360
14.4.3 执行后台任务	313		
14.5 回顾	316	第3部分 高级Windows Forms	
第15章 装饰和美化	317	第17章 自定义控件	364
15.1 图像和光标	317	17.1 扩展现有的控件	364
15.1.1 Bitmap类	317	17.1.1 设计可滚动图片框	365
15.1.2 图标	320	17.1.2 创建可滚动图片框	366
15.1.3 光标	324	17.1.3 实现可滚动图片框	368
15.2 嵌入式资源和声音	325	17.2 基于Control类的自定义控件	371
15.2.1 嵌入资源	325	17.2.1 创建一个基于Control的类	371
15.2.2 使用项目资源文件	327	17.2.2 自定义控件的公共特性	373
15.2.3 播放声音	329	17.3 ScrollablePictureBox类	375
15.3 滑动条	331	17.3.1 绘制控件	375
15.3.1 TrackBar类	331	17.3.2 改变控件的大小	377
15.3.2 使用滑动条	332	17.3.3 处理属性的变化	378

17.4 用户控件	380	19.3.3 激活列表项	443
17.5 Visual Studio中的自定义控件	381	19.4 虚拟项	445
17.5.1 对类的行为进行自定义	381	19.5 回顾	447
17.5.2 对类成员的行为进行自定义	383	第20章 多文档界面	448
17.5.3 对设计时的行为进行自定义	385	20.1 MDI窗体	448
17.6 回顾	387	20.1.1 创建MDI容器	449
第18章 资源管理器界面和树形视图	389	20.1.2 创建MDI子窗体	450
18.1 界面风格	389	20.1.3 更新应用程序的入口	451
18.1.1 单文档界面	389	20.2 合并菜单	453
18.1.2 多文档界面窗口	390	20.2.1 设置合并操作	454
18.1.3 资源管理器界面	391	20.2.2 设置合并索引	455
18.2 .NET的资源管理器界面	391	20.2.3 打开和关闭子窗体	457
18.2.1 SplitContainer类	392	20.3 MDI子窗体	459
18.2.2 TreeView类	394	20.3.1 合并工具条	459
18.2.3 创建资源管理器界面	395	20.3.2 显示像素数据	463
18.3 树节点	397	20.3.3 两次打开同一个相册	466
18.3.1 TreeNode类	397	20.3.4 更新标题栏	468
18.3.2 创建树节点	398	20.4 MDI子窗体管理	469
18.4 自定义树形视图	401	20.4.1 排列MDI子窗体	470
18.4.1 创建自定义树形视图	402	20.4.2 显示MDI子窗体列表菜单	472
18.4.2 创建自定义的树节点	402	20.5 回顾	473
18.4.3 使用接口的自定义节点	404	第21章 数据绑定	474
18.4.4 展开和收缩树节点	409	21.1 数据网格视图	475
18.4.5 选择和编辑树节点	411	21.1.1 DataGridView类	475
18.4.6 集成自定义树形控件	415	21.1.2 创建数据网格视图	476
18.5 回顾	417	21.1.3 填充数据网格视图	478
第19章 列表视图	419	21.2 列和行	481
19.1 列表视图的基础知识	419	21.2.1 创建列	482
19.1.1 ListView类	419	21.2.2 创建组合框列	486
19.1.2 创建列表视图	421	21.2.3 理解带区和行	487
19.1.3 生成列表视图	423	21.3 单元格	491
19.2 列表视图的列	426	21.3.1 DataGridViewCell类	491
19.2.1 定义列标题	427	21.3.2 使用单元格类	492
19.2.2 生成列表视图的列	429	21.3.3 保存经过修改的相册	495
19.2.3 对列表视图的列进行排序	432	21.4 回顾	495
19.3 列表视图的特性	436	第22章 双向数据绑定和绑定源	497
19.3.1 选择列表项	437	22.1 绑定列表	497
19.3.2 编辑列表项的标签	439	22.1.1 IBindingList接口	497