

Foreword by **Juval Löwy**, President, IDesign



# Windows Forms 2.0

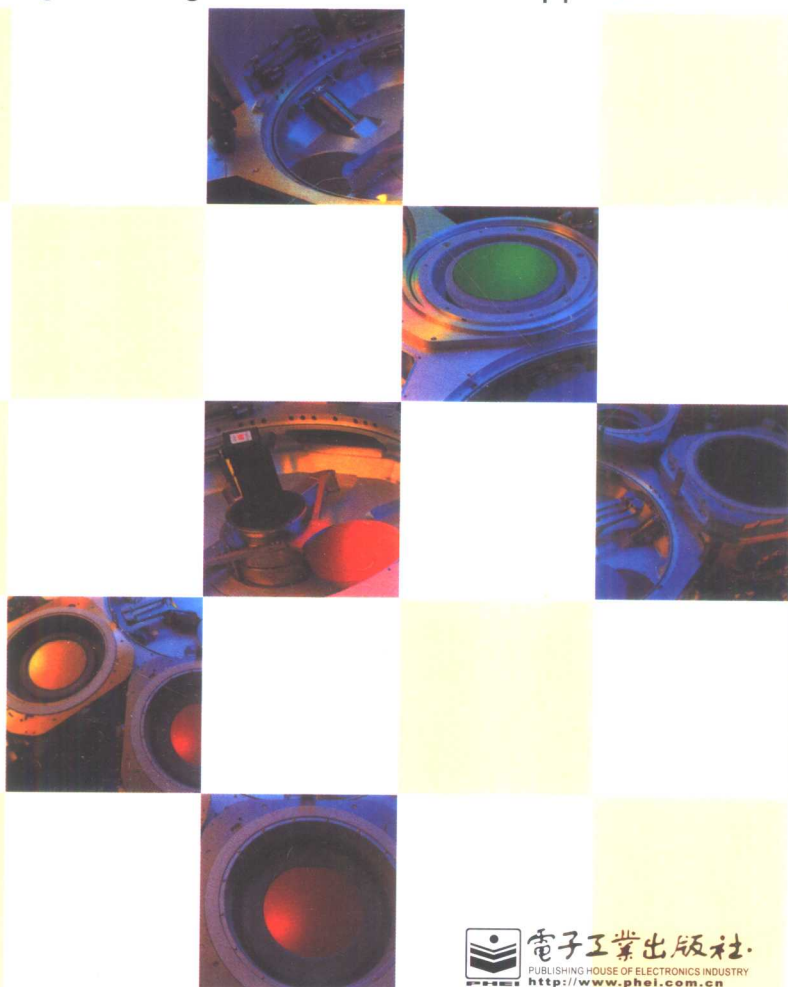
# 数据绑定

——.NET智能客户端数据应用程序设计

Data Binding with Windows Forms 2.0

Programming Smart Client Data Applications with .NET

Microsoft®  
**.net**™  
开发系列



[美] Brian Noyes 著  
汪泳 译

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# Windows Forms 2.0 数据绑定

— .NET 智能客户端数据应用程序设计

---

## Data Binding with Windows Forms 2.0

Programming Smart Client Data Applications with .NET

[美] Brian Noyes 著  
汪 泳 译

电子工业出版社

**Publishing House of Electronics Industry**

北京 • BEIJING

## 内 容 简 介

本书是 Windows Forms 2.0 数据绑定技术的最佳使用指南。

全书共包括 10 章和 4 个附录，对数据绑定技术进行了全面而深入的讲解。前半部分讨论了如何使用 Windows Forms 2.0 和 Visual Studio 2005 提供的强大的数据绑定特性。第 6 章专门讲述了 DataGridView 控件的使用。后面几章专注于高级特性，内容包括数据绑定接口、实现自定义数据绑定控件、自定义数据绑定商业对象和集合及数据验证等。附录分别介绍了 ASP.NET 和 WinFx（现在叫 WPF）的数据绑定，以及 Windows Forms 应用程序和 ADO.NET 的基础知识。

本书适合希望学习 Windows Forms 2.0 中新的数据绑定特性的 Windows Forms 程序设计和开发人员。Windows Forms 学习者可以快速地学到如何使用数据绑定功能，有经验的 Windows Forms 开发人员也可以学到大量的内部实现机制和高级技巧。

Authorized translation from the English language edition, entitled DATA BINDING WITH WINDOWS FORMS 2.0 : PROGRAMMING SMART CLIENT DATA APPLICATIONS WITH .NET First Edition, 032126892X by NOYES,BRIAN published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright©2006 Pearson Education,Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTORNICS INDUSTRY Copyright ©2007

本书简体中文版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2007-0563

## 图书在版编目（CIP）数据

Windows Forms 2.0 数据绑定：.NET 智能客户端数据应用程序设计 /（美）诺伊斯（Noyes,B.）著；汪泳译。

—北京：电子工业出版社，2007.10

书名原文:Data Binding with Windows Forms 2.0: Programming Smart Client Data Applications with .NET

ISBN 978-7-121-04877-7

I. W… II.①诺…②汪… 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 126631 号

责任编辑：陈元玉

印 刷：北京智力达印刷有限公司

装 订：三河市金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：46.25 字数：830 千字

印 次：2007 年 10 月第 1 次印刷

定 价：88.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

# 对《Windows Forms 2.0 数据绑定》的赞誉

---

---

## Praise for Data Binding with Windows Forms 2.0

“Brian Noyes 的写作风格很容易引起你的兴趣，因为他在本书中详细说明了所有数据绑定方面的内容。在每章的开头，他首先会简要地介绍该章的主要内容，让读者感到头绪清晰；然后他再教给你合理的实践方法以及实现可扩展性需要思考的问题。更重要的是，当他解释数据绑定架构的方法、使用模式、数据集的价值、绑定控件以及其他内容的时候，总是描述他自己在这些问题上的看法。本书对于 .NET 2.0 的初学者来说非常合适，而且对于有经验的程序员，本书也是很适用的。任何关心应用程序中数据的程序员（几乎是他们中的任何人），都可以通过阅读 Brian 的书学到新的有用的东西。”

—Michele Leroux Bustamante, IDesign 首席架构师, Microsoft 区域经理, MVP

“Brian 帮我节省了大量的时间。我正在撰写 *The Hitchhiker's Guide to Visual Studio and SQL Server 2005 (7th Edition)* 一书，在该书中我不会深入地讲述数据绑定，因为 Brian 已经为我做了这些工作。他的书深入数据绑定的核心，并且使数据绑定看起来很容易。我还很高兴地看到本书深入了被误解和低估的 Windows Forms 架构。如果试图使你的应用程序更加具有交互性，并且能够利用到新的 Visual Studio 2005 技术，那么本书对你来说是必读之书。当我的读者希望深入了解数据绑定的内容的时候，我会介绍他们去看本书。”

—William Vaughn, Beta V 公司总裁

“Windows 应用程序中的数据绑定技术已经成熟。在以往 Visual Studio 6.0 中，我完全忽视了数据绑定功能，于是自己编写了重复的代码来封装商业逻辑。在 Visual Studio 2005 中，我们终于拥有了稳定并且吸引人的数据绑定技术。目前忽视数据绑定会使你的开发效率大打折扣并且落后于别人。Brian 对 Windows 开发这一核心话题做了清晰而简练的讨论：容易理解的对话式风格和技术语调、极好的例子和深入的解析，所有这些组合使得本书对于任何编写 Windows 应用程序的开发人员或初学者都是必读之书。”

—Stephen Forte, Corzen 公司首席技术官

“本书提供了技术清晰、可读以及深入的数据绑定讲解，还详细讨论了数据方面的最佳实践。Brian 表达了他对数据绑定机制的理解，讲述了创建复杂应用程序以及解决复杂问题所需了解的独一无二的基础知识。有效的数据绑定能大量减少应用程序中代码的数量，还能支持你开发更复杂的应用程序。因此，作为程序员务必阅读此书。”

—Jonathan Cogley, Thycotic 公司首席执行官, ASPInsider 以及 C# MVP

“.NET 框架 2.0、Visual Studio .NET 2005 和 Windows Forms 2.0 一起组成了当今最强大的数据绑定平台，我们绝对需要这样一本书来揭开它们的神秘面纱。Brian 不但贡献了他丰富的数据绑定知识和经验，还详细地讨论了数据绑定方面的许多内容，从基础知识一直到处理多种多样现实场景的方法都涉及了。我总是想要一本数据绑定方面的书，我很高兴看到 Brian 编写了本书。”

—Michael Weinhardt, 自由作家和应用程序开发者

“在我的开发书库中《Windows Forms 2.0 数据绑定》占据了重要的位置。Brian 是一位杰出的技术、最佳实践和技巧方面的老师。我出席的每个会议中他都在传授高质量的知识，他的书也是如此。我发现本书包含有大量的信息，并且充满了学习这一技术必需的重要方法和实例。在本书中，Brian 表现出了他对概念的熟练掌握，我还非常欣赏他努力推荐的最佳实践。本书绝对值得从头到尾阅读。”

—Randy Hayes, Expert Network Solutions 公司总裁

“Brian 直接并且思路清晰的演示使得数据绑定这个非常容易误解的话题最终变得容易理解。”

—Sahil Malik, *Pro ADO.NET 2.0* 的作者, C# MVP

献给我的儿子 Nathan。你为我的生活打开了一个美妙的新空间，以前我甚至没有意识到我失去过这些东西。

献给我美丽的妻子 Robin。与你共享生活的每一天都带给我无限的幸福。

# 联系博文视点

您可以通过如下方式与本书（中文版）的出版方取得联系。

读者信箱：[reader@broadview.com.cn](mailto:reader@broadview.com.cn)

投稿邮箱：[broadvieweditor@gmail.com](mailto:broadvieweditor@gmail.com)

北京博文视点资讯有限公司（武汉分部）

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室

邮政编码：430074

电话：(027) 87690813 传真：(027) 87690813 转 817

若您希望参加博文视点的有奖读者调查，或对写作和翻译感兴趣，欢迎您访问：

<http://bv.csdn.net>

关于本书的勘误、资源下载及博文视点的最新书讯，欢迎您访问博文视点官方博客：

<http://blog.csdn.net/bvbook>

# 译序

---

今天，智能客户端架构已经成为独立 Windows 应用程序的主流架构，而数据绑定技术作为智能客户端架构的重要一环，其主要目标之一就是减少编写数据访问代码的复杂性，从而提高开发人员的生产效率。然而，要充分利用好 Windows Forms 2.0 提供的数据绑定方面的功能并不容易，幸运的是，本书详细地讲解了关于数据绑定的所有方面的内容，并为我们提供了清楚的实例，这使得我们能够更加容易地使用数据绑定技术。

如果说 .NET 平台上的数据绑定技术在之前的版本中还有些缺陷，那么我们可以说，在 Windows Forms 2.0 和 Visual Studio 2005 发布之后，数据绑定技术已经非常成熟。要想开发智能客户端应用程序以及数据访问功能，如果不使用数据绑定技术，我们就不可能充分地利用 Windows Forms 2.0 提供的优秀的特性，这样会大大落后于竞争对手。

可以说，在本书面世之前，还没有一本书能够如此清晰地和详细地介绍数据绑定相关的内容，是它为我们揭开了数据绑定的神秘面纱。如果你正在从事 Windows Forms 数据访问程序开发，本书绝对是你必须拥有的书籍之一。

虽然我力求完美，但限于种种主客观因素，难免会有疏漏或者错误的地方，欢迎各位读者批评指正。



最后，我要感谢博文视点的编辑和工作人员对我的充分信任和大力帮助。此外，还要感谢我的妻子和家人，他们给我的支持和鼓励，是我前进的动力。

汪 泳  
2007年7月

# 序言

---

## Foreword

如果没有“Q 博士”提供的那些工具和奇思异想的小道具，詹姆斯邦德还会是那个无所不能的特工吗？没有那些东西的帮助，他还能够在紧要关头死里逃生吗？或者它们仅使得他更具有威力，他还会拥有干掉敌人的绝招吗？如果拥有这些工具，其他人是否也能成为一个超级特工呢？是什么使得邦德、詹姆斯邦德无所不能？是这些工具，还是为了掌握这些高科技工具和装备——他所获得的训练以及学到的知识和技能呢？

我认为，同样的问题对于使用 Visual Studio 创建新型的应用程序来说也是合适的：是什么使得开发人员的生产力更高并且开发工作更有效率呢？是工具、向导、.NET 框架类，还是如何更好地设计和创建应用程序的知识呢？仅通过 Visual Studio 中的拖放操作真的能创建可维护的、稳健的、可重用的、可扩展的、安全的以及一致的应用程序吗？如何训练开发人员，使得他们的生产力更高并且可以使用层出不穷的新工具和相关技术的正确方法？当有需要自定义和修改生成的代码时，开发者有能力对开发工具和框架类的基本功能进行扩展和改进吗？在适当的时候，他们能将这些工具用在连 Microsoft 也想不到的不同的场景中吗？

当谈到数据绑定和数据访问时，这些问题再合适不过了。Visual Studio 2005 中的“数据源”窗口生成了大量专门的代码，设计器还将这些代码与可视的控件连接起来，并且全部以无缝的自动的方式整合在一起。但是拥有这些就足够了吗？我认为，要达到像可维护

性、可靠性和可扩展性这些目标，关键在于理解工具到底生成了什么以及为什么工具生成了这些代码，并且还要理解和领会整个设计方法以及隐含的相关的最佳实践。只有了解了这些内容，你才有可能获得成功。原因非常简单，自动生成的代码以及框架内置的类（如 `DataGridView`）可以用在几乎所有类型的应用程序和使用案例中。一旦你偏离了正确的路径（我们不可避免地都会碰到），只能靠你自己，也只有靠你自己的技能和知识，才能指引你正确前进。我相信向导的作用不在于让任何人都可以开发应用程序，而在于能帮助有经验的开发人员能高效完成那些消耗时间的、琐碎的、普通的以及重复的任务，使得他们的开发工作更有效率并且能够更专注于实现应用程序的逻辑和要求完成的重要功能。我认为，要用好这些工具，你需要了解这些工具，只有理解了设计器生成的代码以及为什么生成这些代码，你才能很好地利用它们。

本书的内容是与使用 Visual Studio 2005 创建 Windows Forms 应用程序中绑定数据源相关的知识。它非常详细地解释了设计器生成的代码背后的基本原理，推荐了最佳实践，给出了技巧和诀窍，揭开了自动生成代码的神秘面纱，讲解了如何扩充这些代码，以及演示了如何突破这些代码的限制。不仅如此，本书还可以让你充分发挥当今的以及未来的智能客户端应用程序的强大功能和用户友好性的优势。它就是你的终极武器，让你可以成为数据绑定的“詹姆斯邦德”，我认为这句话说明了一切。

Juval Löwy

2005 年 8 月

# 前言

---

## Preface

当我首次开始跟 Addison Wesley 出版社的编辑讨论本书的时候，我有点怀疑。我心里的反应是，真的有人需要一本只关注数据绑定技术的书籍吗？我的意思是，Windows Forms 只是有关 GUI 的一些东西，难道不是吗？你可以拖这个，放那个，还可以挂接一些事件处理程序，然后继续创建企业应用程序的其余部分：所有那些让你头痛的中间层代码。

在我深入思考以后，我意识到人们在 Windows Forms 应用程序中所做的大部分工作都是围绕着数据绑定展开的，开发人员碰到的大多数问题都是如何让数据绑定功能能够正确地运行。在 Windows Forms 2.0 和 Visual Studio 2005 中新增的大多数功能都与数据绑定相关，所以我很快确信撰写这样一本书是一个好主意。希望在你阅读完本书之后，也会这样认为。

通过 .NET 框架类提供的功能和 Visual Studio 2005 提供的丰富的设计器支持，数据绑定在 Windows Forms 2.0 中最终成为成熟的技术，并且具有非常强大的功能。正确地使用数据绑定，你可以不用编写大量不必要的代码，就能向用户提供能够正常运作的与数据交互的丰富体验，编写容易维护的代码。为了让数据绑定在多种不同的使用案例中都能正确运行，你不仅要了解如何设置控件的多个属性，还要理解数据绑定后是如何运作的，尤其

是当你想支持的应用场景的复杂性超过了 .NET 框架中数据绑定组件的基本功能时。

由于智能客户端架构的不断成熟，Windows Forms 应用程序在商业系统中变得更加重要。基于 Web 浏览器的应用程序还有大量需要完善的地方，它们目前还不支持大量的常用场景。由于不能利用客户端机器的能力，受到基于浏览器应用程序的请求-响应模式以及与此模式相关的连接问题的限制，所以在 Windows Forms 中编写复杂的数据应用程序场景的能力显得越来越重要。幸运的是，.NET 服务中的功能正在快速地成熟，跟上了这一步伐。

## 谁应该阅读本书

### Who Should Read This Book?

本书的主要读者应该是中级和高级的 Windows Forms 开发人员，以及那些希望学习 Windows Forms 下新的数据绑定特性并希望提升数据绑定应用程序的编码实践能力的人。本书深入讲解了 Windows Forms 中数据绑定机制的高级特性，包括数据绑定控件、使用数据源以及创建自定义数据绑定对象和集合等内容。如果你在编写 Windows Forms 应用程序的时候，花费了大量的时间与数据打交道，那么你就应该阅读本书。

如果你是一位 Windows Forms 初级开发人员，本书会帮助你快速地学到如何支持数据绑定。Windows Forms 2.0 中的大量特性可以让开发人员使用向导和设计器特性来完成工作，这些特性对初学者也是有帮助的，并且你可以从本书中学到这些特性。此外，附录 C 和附录 D 适合初学者学习 Windows Forms 和数据访问的基础知识。

## 约定

### Conventions

开发应用程序不仅与代码相关，更多的是与开发工具相关。然而，为了让读者更容易阅读本书中的大量代码，笔者用了一些公共的约定。代码中类、变量、命名空间以及其他

编程产物的引用都使用了等宽字体,这样做会帮助你辨别 DataSet 类的实例与数据集的概念性的讨论之间的差别。短的代码清单使用另外一种等宽字体嵌入在文中:

```
private void Form1_Load(object sender, EventArgs e)
{
    m_EmployeesTableAdapter.Fill(northwindDataSet.Employees);
}
```

长的代码清单不但使用更小的字体,而且使用了清单编号来标识,例如:代码清单 4.1。在代码清单中,加粗字体突出了代码中特定的相关部分,尤其是那些“不断进化的代码”。当我从清单中移除与讨论无关的细节的时候,你将会看到使用省略号 (//...) 标识的一个注释。这代表要完成某个例子需要更多的代码或由设计器生成的更多代码已经存在,但是要理解讨论的概念,你不需要去关注它们。偶尔,也会有解释性的注释用于说明上下文:

```
public partial class CustomersDataSet : System.Data.DataSet {
    // ...

    // 嵌套的强类型数据表定义
    public partial class CustomersDataTable : System.Data.DataTable,
        System.Collections.IEnumerable {
        // 用于访问数据行的强类型索引器
        public CustomersRow this[int index] { ... }

        // ...
    }
}
```

当我在讨论数据绑定场景中需要使用的各种对象的时候使用了一种对话式的语调,比如在本书中描述 DataSet 类的时候。然而,当讨论数据集的时候,我并不是在讨论 DataSet 类的实例,而是在讨论派生的强类型 DataSet 类的实例。虽然将这些类称作 DataSet 在技术上是正确的,因为通过继承特性,它确实是一个 DataSet,但是我发现在文章中有太多的代码字眼会令人讨厌。所以只有在一些东西确实是代码,并且只在这段上

下文中才正确的时候，我才使用等宽字体来标识它们。当讨论这些对象类型代表的概念的时候，我喜欢使用术语**数据集**、**数据表**和**表适配器**等字眼，而只在引用特定类型或实例以及由代码定义的功能的时候，才会使用 `DataSet`、`DataTable` 和 `CustomersTableAdapter` 等字眼。

讨论组件和控件也可能会让人混淆，这取决于使用的语言定义上的精确程度。从技术上讲，**Windows Forms** 中所有的控件都是**组件**，因为 `Control` 类派生自 `Component` 类。数据绑定的大量概念对控件和组件同时有效，例如本书中将深入讨论的 `BindingSource` 组件和 `DataGridView` 控件。不幸的是，许多人认为**组件**是由窗体使用的不可视对象，而**控件**是在窗体上具有可视输出的对象。为了避免控件和组件之间的混淆，当我讨论对不可视组件和控件同时有效的概念的时候，我会使用**组件**这个字眼。所以当你看到**组件**这个词的时候，应该这样想：这对控件也有效，因为控件继承自组件。

## 系统要求

### System Requirements

本书是使用 .NET 2.0 的代码库和 Visual Studio 2005 的 Beta1、几个 CTP 版本和最终 Beta 2 版本来编写的。本书中讲解的代码可以使用 Beta 2 运行。我与 Microsoft 的 Windows Client 产品团队联系紧密，在 Beta 2 和产品正式版之间没有任何计划中的特性上的改变。然而，当 .NET 2.0 生成以及发行的时候，还是可能会有一些新的语法上的变化。如果这些变化影响了代码或概念，我将通过本书的网站 ([www.softinsight.com/databindingbook](http://www.softinsight.com/databindingbook)) 提供勘误，并且还会在 Visual Studio 2005 正式版发布的时候更新相应的代码。

如果你打算运行本书网站上的示例程序，或者试验本书中的讨论步骤和代码清单，需要在你的机器上安装 Visual Studio 2005，并且还需要能访问到 SQL Server 2000 或 2005 数据库服务器，这个数据库服务器应该安装了 Northwind 示例数据库。此外，为了运行某些实例程序，你还需要创建新数据库的权限。

Visual Studio 2005 有多个版本可以选择。本书中讨论的所有特性在 Visual Studio 2005 的免费 Express 版中也可以运行。你可以使用 Visual C# 2005 Express 或 Visual Basic 2005 Express 加上 SQL Server 2005 Express 来开发本书的所有示例程序，但是因为 Visual Studio 的 Express 版本不支持使用服务器路径的数据连接（它们只支持基于文件路径的 SQL Express 数据库的连接），你要在 SQL Express 中创建示例数据库和数据，然后修改连接字符串，并使用 Express 版提供的设置数据连接的方法。

本书中包含的示例程序和脚本假设你使用的机器上安装了 Visual Studio 2005 的 Standard、Professional 或 Enterprise 版，以及本机的 SQL Server 2000 或 2005 的默认实例。如果没有这些配置，要运行示例程序的话，需要为使用数据库的所有示例程序修改连接字符串设置。所需的修改在本书的网站上可以找到，连接字符串的差别突出地显示在示例代码中的许多地方。

此外，虽然 Northwind 没有与 SQL Server 2005 一起发布，但是在 MSDN 下载网站 [www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46A0-8DA2-EEBC53A68034&displaylang=en](http://www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46A0-8DA2-EEBC53A68034&displaylang=en) 这个链接中有 SQL Server 2005 可以使用的一个单独的可安装的下载包。这个下载包中提供了脚本代码以及可以附加到 SQL Server 2005 或 SQL Server 2005 Express 上的 MDF 文件。

## 编程语言的选择

### Choice of Language

我选择使用 C# 撰写本书。下载代码中包括 C# 和 Visual Basic 两种语言的代码。事实上将来很长一段时间内，还会继续有大量的文章、书籍和示例程序混合使用 C# 和 Visual Basic。就算我自己更喜欢使用 C#，但我并不是某种语言的盲目信仰者，并且我感觉 Visual Basic 是那些在之前版本的 Visual Basic 上有很多经验的开发人员的首选。

我相信，要成为一个高效的 .NET 开发人员，需要能够阅读这两种语言的代码，就算大多数时候你只会使用一种。如果阅读 C# 代码对你来说有点困难，我建议你使用这个机



会来熟悉 C# 代码。通过一定数量的可以获取的参考资料，它将会扩展你的视野，也许它还会帮助你的工作。另外，它还会让你在那些无法阅读 Visual Basic 代码的笨拙的并且思想闭塞的 C# 开发人员面前拥有可以炫耀的资本。

## 编码标准

### Coding Standards

是的，我使用了编码标准，你也应该有一个。它就是那么简单，但是不幸的是，开发组织并没有足够重视这一点。编码标准是一个重要的工具，可以确保减少代码中的错误，而且对于确保代码库可维护来说，它显得更加重要。在你的团队中由某个开发人员编写的代码看起来应该与所有其他开发人员编写的代码一样，以便在需要的时候可以维护和扩展这些代码。代码评审应该与编码规范并行使用，它也应该成为你开发过程中的日常工作。

在 IDesign，我们有一个优秀的编码标准，从我们的网站 [www.idesign.net](http://www.idesign.net) 可以下载到。你可以直接使用，或者你可以使用它来开发你自己的编码规范。我们的标准包括大量的信息，而不仅仅是简单的语法。它包含了大量 .NET 开发和设计的最佳实践。

本书的演示代码中，我使用了 IDesign 编码标准，用于成员变量、属性、方法和事件处理程序的命名。类的任何成员，不管是变量还是窗体上的控件都有一个前缀 `m_`，剩下的名称使用 PascalCasing 命名规范。虽然这与 Microsoft 的标准不同（Microsoft 在不同的产品团队中采用了不同的标准），但是也没问题。你可以使用你喜欢的任何标准，关键是要采用统一的标准。我不想加入编码标准的争论，我也不希望收到有关这方面的任何投诉或电子邮件。

你在本书中会看到有些代码段没有遵循这一约定（在我讨论设计器生成的代码的时候），因为由设计器生成的代码的命名规范与我们的编码标准不同。所以当讨论由设计器