



Windows Forms 2.0 程序设计

Windows Forms 2.0 Programming

Microsoft[®]
.net[™]
开发系列

[美] Chris Sells 著
Michael Weinhardt

汪泳 译

微软公司客户端开发工具部门的项目经理和技术带头人

Mark Boulter 亲自作序



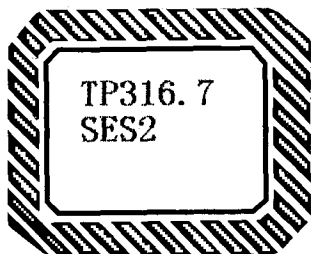
电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Windows Forms 2.0 程序设计

Windows Forms 2.0 Programming

[美] Chris Sells 著
Michael Weinhardt

汪 泳 译



電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

如果你想开发优秀的 Windows Forms 2.0 应用程序, 本书就是你书架上必不可少的参考书, 也是 Windows Forms 2.0 的最佳使用指南。

全书共包括 19 章和 6 个附录, 全面讲解了 Windows Forms 2.0 的各项技术, 还深入地探讨了各种类型应用程序的解决方案。在讲解这些技术的同时, 还列举了与 Windows Forms 之前版本不同的地方以及一些新增的内容。特别值得一提的是附录最后一部分讲解的文档管理功能, Windows Forms 应用程序的开发人员可以使用它来搭建应用程序的架构。

本书适合有 .NET 背景知识的 Windows Forms 应用程序学习者阅读, 有经验的 Windows Forms 开发人员也可以从中学到大量的高级技巧和内幕功能。由于本书讲解的内容非常全面, 本书还适合教学人员作为参考书使用。

Authorized translation from the English language edition, entitled WINDOWS FORMS 2.0 PROGRAMMING, 0321267966 by Chris Sells and Michael Weinhardt, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright©2006 Pearson Education, Inc.

Authorized translation from the English language edition, entitled WINDOWS FORMS 2.0 PROGRAMMING, 0321267966 by Chris Sells and Michael Weinhardt, 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 ELECTRONICS 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: 01-2007-1245

图书在版编目 (CIP) 数据

Windows Forms 2.0 程序设计 / (美) 魏尔斯 (Sells, C.), (美) 魏因哈特 (Weinhardt, M.) 著; 汪泳译.

—北京: 电子工业出版社, 2008.4

书名原文: Windows Forms 2.0 Programming

ISBN 978-7-121-05868-4

I. W… II. ①魏…②魏…③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 (2008) 第 010346 号

责任编辑: 陈元玉

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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: 787×1092 1/16 印张: 58.25 字数: 1000 千字 彩插: 4

印 次: 2008 年 4 月第 1 次印刷

定 价: 108.00 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 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

总有些书的独特内容是 Internet 上搜索不到的，那些书的作者拥有可以共享的真实体验，并且书中的代码可以用在正式的项目中。本书正是一本这样的书。

—Neal Myrddin, 独立开发人员

Chris 和 Michael 掌握了 Windows Forms 2.0 的灵魂。本书提供了对 Windows Forms 的设计以及与新的 2.0 特性的集成等内容的富有洞察力的见解。对于初学 Windows Forms 的程序员和经验丰富的商业应用程序开发人员，本书都同样适用。附录“2.0 中的新特性”也值得单独拿出来推荐。

—Mike Harsh, 微软公司, Windows Forms 程序经理

本书讨论了一种范围很广并且深入的技术，并将它分解为容易理解的可管理的片段。如果想创建复杂的 WinForms 应用程序，离开本书你将无法生存。

—Ryan Dorrell, AgileThought, 首席技术执行官

《Windows Forms 2.0 程序设计》提供了所有 .NET 开发人员都需要的一些东西。有经验的开发人员会喜欢本书中对新的 2.0 特性的深入的讨论，其中包括对 ClickOnce 部署技术、多线程和无所不在的新的设计器集成功能的最全面的讨论。初学 WinForms 程序设计的开发人员会喜欢本书对所有 WinForms 开发人员都需要知道的基础知识的讨论。不管是有经验的 WinForms 开发人员，还是初学者都需要这本书。

—Fritz Onion, Pluralsight 公司的联合创始人, *Essential ASP.NET* 一书的作者,

—ASP.NET 最有价值专家

我不需要一本仅仅说明了 WinForms 组件的书。我可以在网上或其他书中获得这些内容。我需要的是可以让避免犯错误的指南以及对常见问题的有创意的解决方案。这正是本书的闪光点。将本书第 1 版作为必不可少的参考书的所有人都会喜欢讨论了 WinForms 2.0 功能的这一更新过的版本。

—Johan Ericsson, 软件工程师, Agilent Technologies 公司

Chris Sells 撰写的书籍总是富有洞察力的，这本最新的书籍也不例外。本书尤其适合希望不仅了解 Windows Forms 2.0 的漂亮界面，而且想了解它是如何与 .NET 环境相结合的人。读者将会长时间地将本书放在他们的书桌上，而不是束之高阁。

—Yoshimatsu Fumiaki, 住在日本东京的软件工程师

本书将开发成功的 Windows Forms 应用程序所需了解的信息展示给大家，Chris 和 Mike 为此做出了卓越的贡献。

—Jessica Fosler, 开发技术带头人, 微软公司

本书是 WinForms 2.0 教学和参考的必备书籍。

—Jim Rittenhouse, 高级软件工程师, 西门子公司

献给我的妻子 Melissa 和我的儿子 John 和 Tom。他们给了我全部的幸福生活。
献给我的父母，他们让我成为一名读者，还给了我写作的神秘基因，这很让我惊奇。

—Chris Sells

献给 Josef 和 Lili，为我是什么样的人以及我做过什么事

献给 Alex，为我希望成为的人和我希望能够做的事

—Michael Weinhardt

联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿信箱: bvtougao@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>

作者简介

About the Authors

Chris Sells 是互联系统部门的程序经理。他写过几本书，其中包括 *Programming Windows Presentation Foundation* (O'Reilly)、*Windows Forms Programming in C#* (Addison-Wesley)，以及 *ATL Internals* (Addison-Wesley)。在他的闲暇时间，Chris 主持着几个会议并且在 Microsoft 内部产品团队讨论列表上拥有自己的一个列表。关于 Chris 和他的几个项目的更多信息可以参见 <http://www.sellsbrothers.com>。

Michael Weinhardt 是 Microsoft 的程序员和作家，在 Windows Client SDK 部门供职。Michael 与 Chris 一起合作编写了多篇文章，为 MSDN 在线的“Wonders of Windows Forms”专栏做出了很多贡献，评论过几本 Windows 技术的书，并且喜欢与同事沟通技术的原由。Michael 确信由于他的父母在 20 世纪 80 年代中期购买了 Amiga 1000 机器而使他迷恋上了技术。

译序

毫无疑问，本书是迄今为止 Windows Forms 程序设计领域中最完整、最全面的一本书。

Chris Sells 和 Michael Weinhardt 独特的写作手法，就像是有一位大师在旁边向你娓娓道来，从编写 Windows Forms 的入门基础、控件的应用到完成各种类型的应用程序所需了解的技能的详细讲解，在内容的全面和细节的深入之间，他们找到了绝佳的平衡点。书中还提供了大量的代码示例和截图，一方面使得读者可以很轻松地掌握开发 Windows Forms 应用程序所需的各种技能，另一方面读者也可以将这些非常实用的代码用于自己的开发项目中。

全书覆盖了 Windows Forms 技术各个方面的重要概念，从窗体的基础、控件的使用、数据绑定、数据验证、组件和自定义控件到资源管理、本地化应用程序以及 ClickOnce 部署技术都有深入的探讨。

本书还有一个特点，作者将 Windows Forms 技术和 Visual Studio 2005 开发工具这对完美搭档结合在一起讲解，让读者可以充分地利用 VS05 提供的强大的功能，这在别的书籍上是找不到的。

如果你希望能够开发高质量的成功的 Windows Forms 应用程序，并且希望利用到 Windows Forms 2.0 的绚丽多彩的新特性，本书绝对应该是不二之选。而且，我认为这应该会成为 Windows Forms 应用程序开发人员必不可少的参考书。当你在开发过程中碰到不清楚的问题时，它都可以作为你可靠的指南。

本书在翻译过程中得到了博文视点工作人员的热心帮助，在此向他们表示感谢。同时，我还要特别感谢我的妻子，没有她的支持，本书的翻译是不可能顺利完成的。本书的内容非常丰富，由于时间仓促，技术水平有限，不足之处还望读者一一指正。

汪泳

2007年12月

Foreword

很久很久以前，当我开始编写 PC 上的用户界面代码时，还没有现在这些很棒的框架。那时候的人们需要在 case 语句中编写大量的 C 代码，代码的长度比巨人的汗衫还长。我曾经在我们创建的一个应用程序中花费了几个星期去理解并实现 DDE（是的，是 DDE）（并且坦白地说，这并不是非常愉快的一次体验），之后我在一本杂志中看到了一篇文章，它演示了一个叫做“Smalltalk”的很棒的东西——使用几行代码就能够实现 DDE 了。我当时想：哇！真棒。这正是我喜欢的编程方式！从那以后，我使用它实现了很多的用户界面框架，最终导致我在 Microsoft 有了份与 Windows Forms 相关的工作。

对于 Windows Forms 的第一个版本来说，我们的目标是创建一个全面的用户界面框架，这个框架可以将 VB 的易用性与 MFC 的可扩展性和灵活性结合在一起。在开发过程中，我们又加入了更多的目标，其中包括丰富的设计期扩展性、GDI+ 支持以及对部分信任环境下的无接触部署（NTD）的支持。我相信我们已合理地实现了这些目标。尽管当我们第一次发布 Windows Forms 时，业界的焦点已经转移到了 Web 上，现在还是有大量的程序员在使用 Windows Forms 创建各种类型的应用程序，从照片管理软件到支持核心商业流程的应用程序。我发现看到人们使用 Windows Forms 创建有趣的应用程序是对我工作的最大奖赏。然而，说实话，还有些地方，我们可以做得更好。例如，Visual Studio 不支持 NTD 技术，并且当发生错误需要调试的时候可能会很复杂，所以总的来说，我会给 Windows Forms 的第一个版本做一个“显示承诺”的评价。

Windows Forms 2.0 试图实现这样的承诺。这是 Windows Forms 的一个重大改进。Windows Forms 的设计期和运行时功能的几乎所有部分都得到了改进。正如 Chris 和 Michael 在附录 A “Windows Forms 2.0” 的新特性中所指出的，我们将全新的特性与对现存的特性的改进很好地结合在了一起（大约有 329 个新类型、139 个更新过的类型以及 14 323 个新成员）。我不打算重复附录 A 的内容，我将特别指出并说明是如何达到 Windows Forms 这个版本目标的三个新特性的：

它们是部署问题的解决、更绚丽的应用程序和更高的开发效率。

部署

Deployment

我认为 .NET 框架 2.0 版的一个最重要的特性（不只是 Windows Forms，而是整个 .NET 框架）就是 ClickOnce。ClickOnce 实现了无接触部署的承诺，将容易的、可靠的、可管理的基于 Web 的部署方式带给了客户端应用程序。通过 Web 部署应用程序，在 Visual Studio 2005 中现在只需要使用向导一步步地设置就可以了。

绚丽的应用程序

Great-Looking Apps

从我加入 Microsoft 开始，顾客就一直要求具有内置的能够创建与 Microsoft Office 相似的应用程序的功能，使用新的菜单条控件 MenuStrip、工具条控件 ToolStrip 以及状态条控件 StatusStrip，我们在 Windows Forms 2.0 版中就能够做到这一点。工具条控件不仅支持标准 Windows 和 Office 外观，而且还可以自定义，使得它们看起来跟你所喜欢的样子一样。

开发效率

Productivity

我们添加了一整套改进的设计期和运行时，相信通过这些改进，可以大大提高大家的开发效率。其中一个我最喜欢的新的设计器特性就是 SnapLines，这允许我们在布局窗体时快速地对齐控件。只要用过具有 SnapLines 功能的设计器，我们就不想回到与设计器功能相同的 IntelliSense 上了。

未来

The Future

发布完 Windows Forms 2.0 之后，我们的思路自然地转向了未来。预测未来是一件危险的事情。我孩提时的大多数预测都没有兑现，所以我对做任何长期预测都有点不安。然而，基于我们已经做到的事情和希望能够做到的事情，我可以发表一些看法。首先，对于 .NET 框架和托管代码，我的看法如下：它是现在和未来都要用到的编程模型。学习如何使用 .NET 框架和 Windows Forms 是对未来的可靠投资。其次，为了解释 Samuel Clemens 所说的话，“关于客户端应用程序已死的报告被过分地夸大了，”我对客户端应用程序的看法如下：客户端应用程序现在可以和 Web 应用程序一样容易被部署，提供了重大的商业价值，而且随着时间推移还将提供更多的价值。

xxviii

第三，作为对 Windows Forms 技术的持续投资的一部分，我们将确保 Windows Forms 和那些如 WinFX 这样的新的即将发布的技术能够很好地结合在一起。这允许我们使用这种技术创建今天的应用程序，并且还将在未来使用 Windows Forms 和新技术来增强那些应用程序的功能。最后，从 Windows Forms 的角度来看，我相信我们需要扩大框架和设计体验所提供的功能以支持创建客户端应用程序的整个过程。我们已经拥有了强大的设计器帮助我们创建用户界面，但是我们仍然需要编写太多的代码来创建整个应用程序。我希望看到我们能够提供强大的基于设计器的整个应用程序的设计体验，而不仅仅是为窗体提供设计器功能。

因此，希望我所说的关于 Windows Forms 至少让你对发现本书中包含的更多内容感到有点好奇。本书第 1 版提供了对 Windows Forms 的杰出的概述和介绍方面的内容。第 2 版同样如此。不管是第一次学习 Windows Forms，还是仅仅希望获得对 2.0 版本中所提供的新东西的了解，本书都将有助于你。它覆盖了所有的重要的特性类别，从创建窗体的基础知识、工具条框架和数据绑定，到 ClickOnce 部署技术。

本书在内容的权衡方面也做得很好：它既没有忽略 Visual Studio 2005 的存在，也没有将内容简化到对 Visual Studio 2005 特性的简化的“点击这里然后点击那里”这种类型的描述。本书不仅揭示了概念以及说明了在代码中如何使用这些概念，而且还说明了设计器是如何通过自动生成代码以帮助我们提高开发效率的。数据绑定部分的内容就是这种方式的一个很好的例子（第 16 章和第 17 章）：它们不会太短以至于没什么用，也不会太长以至于过于啰嗦。引用 Alan Cooper 的话来说，因为它们是“正确的”，所以它们是“黄金代码”示例。

我希望特别强调一下数据绑定部分的章节（第 16 章和第 17 章），这不仅是因为我非常喜欢数据绑定，而且是因为本书在解释数据绑定的工作原理以及如何有效地使用它这方面做得非常好。我还想强调一下编写控件和组件等的设计期行为方面的章节（第 11 章和第 12 章），这是因为这个话题经常被忽视。这些章节也使得本书成为我的必读之书。

xxix

因此，总的来说，本书不仅教会了大家如何有效地使用作为 Windows Forms 一部分的知识，而且还让大家了解到编写自己运行时和设计期扩展所需要了解的东西。

我将以一些致意和感谢的话作为我的结束。首先，感谢整个 Windows Forms 开发团队，他们竭尽所能地工作从而创建和发布了我相信是非常伟大的一个产品。我对我们获得的成就感到非常自豪。其次，要感谢 Michael 和 Chris，他们不仅仅是编写了一本书来解释我们的产品并使得这个产品对我们的客户来说更加容易使用，而且还为广泛的 Windows Forms 社区做出了卓越的贡献。感谢 Michael 在 MSDN 上发表的优秀文章以及对 Windows Forms 2.0 的反馈，尤其是对 ToolStrip 控件做出的反馈。感谢 Chris 对 Windows Forms 的几乎无限的激情，撰写的关于 Windows Forms 的优秀文章，对 Windows Forms、MFC 和 Windows 等技术的深入了解，以及他对产品各个方面的礼貌和有分寸但是热心而且丰富的反馈。Mike 和 Chris 在提升对产品的理解和帮助 Windows Forms 变得如此优秀等方面给予了我们巨大的帮助。最后，要感谢我们的客户：Windows Forms 2.0

中的所有特性都是基于你们的反馈来实现的。所以向所有花时间给予我们反馈和建议的人们表示感谢，请继续支持我们！

希望大家在使用 **Windows Forms** 的过程中找到快乐。

Mark Boulter

项目经理和技术带头人

微软公司客户端开发工具部门



xxx

前言

Preface

Windows 1.0 的程序员能够做的事情很简单。他们几乎不能选择实现程序的方法，可能有一个应用程序编程接口（API），也可能没有，而且大多数时候都没有。这意味着开发人员必需手工创建几乎所有的东西。在那个时候，这还不是问题。所有的 Windows 程序员都碰到过相同的限制，所以大家的应用程序或多或少地都在与相同的受限功能集合打交道。

另一方面，当时的 Windows 开发人员被淹没在选择的海洋中。Web 的开发为我们带来了静态超文本标记语言（HTML）、通过 ASP.NET 实现基于服务器的用户界面（UI）交互、通过 ActiveX 控件或 AJAX（代表了几样东西）¹ 实现客户端的用户界面交互。在客户端，我们还有原生框架，如 Microsoft 基础类库（MFC）、活动模板库（ATL）和被称为 Windows Presentation Foundation 的下一代的托管框架库（WPF 是 .NET 框架下一个版本 WinFX 的一部分）。开发人员如何选择呢？本书讨论的 Windows Forms 2.0 话题在这些技术中位于什么样的位置呢？

Windows 上的客户端技术提供了从支持跨平台的技术到具有丰富特性的技术这个范围中进行不同层次的选择。跨平台是用户界面在不同的操作系统上都能够运行的能力，而丰富性是用户界面利用它运行的操作系统功能的能力。因此，即使静态 HTML 几乎在任何地方都可以运行，但是，如果 HTML 中使用了客户端 JScript，跨平台的能力就缩减到那些支持脚本技术的浏览器中。另外一个就是 ActiveX 控件，它只能在 Windows 上运行，但支持所有的丰富特性。²

在另外一个极端上，我们有 WPF 技术，它是在 DirectX 基础上创建的一个全新的托管框架。

¹ AJAX 是英文词组 “Asynchronous JavaScript and XML”（异步 JavaScript 和 XML）的缩写，是由 Jesse James Garrett 在他的文章 *Ajax: A New Approach to Web Applications* 中定义的，这篇文章可以通过 <http://www.adaptivepath.com/publications/essays/archives/000385.php>（<http://tinysells.com/38>）访问。

² 组件对象模型（COM）和 ActiveX 曾经是跨平台运行的，但是在很久以前就已经限制在 Windows 操作系统中才能使用了。

WPF 在一个单一的并且内聚的整体中包含了文档、布局、动画、2-D、3-D、文本和图形等元素。这个框架提供了 Microsoft 创建的最丰富的客户端技术，它最大程度地利用了它运行的平台的能力，其中包括任何可用的硬件 3-D 图形加速卡。然而，当撰写本书时，WPF 还没有发布，它只能在 Windows XP、Windows Server 2003 和 Windows Vista 下运行。³

Windows Forms 2.0 还剩下多少空间可以发挥呢？它的丰富性不如 WPF，但是比起 HTML 技术来说又丰富得多。Windows Forms 2.0 的功能比 Windows Forms 1.0 也丰富了许多，其中包括大大增强的对灵活布局的支持、最新的菜单条和工具条控件、数据绑定、多线程、强类型资源和设置，以及最重要的 ClickOnce 部署。

有了 ClickOnce 技术，用户第一次能够和 Web 应用程序一样轻松地部署 Windows 应用程序。而且更好的是，ClickOnce 技术还允许 Windows 应用程序能够在本地安装、集成进 Start 菜单，并且在没有网络连接的情况下仍然可用。这些选项对于 Web 应用程序的用户是不可用的。

因此，Windows Forms 2.0 的特性还是很丰富的，但是在跨平台方面呢？和 Web 应用程序不同，Windows Forms 要求用户机器运行 Windows，明确地说，就是安装了 Windows 98 或之后版本的 PC 机。然而，它还支持其他设备，例如 Pocket PC 或 Smartphone，这是通过 .NET Compact 框架来实现的。这意味着我们可以编写一个 Windows Forms 2.0 应用程序并在 Windows 智能手机上运行这个应用程序。

基于这一层次的丰富性和跨平台特性，我们应该使用 Windows Forms 来创建哪种类型的应用程序呢？答案是，人们使用 Windows Forms 创建过几乎所有类型的应用程序，其中包括生产力应用程序、工具软件、企业工具软件，甚至游戏。Windows Forms 可以被用来创建小一点的软件，例如 Microsoft 自己内部的人力资源应用程序（叫做 HeadTrax），或者也可以被用来创建大一点的软件，例如 Visual Studio 2002 版在 Visual Studio 开发工具之上创建的大多数用户界面增强功能。之所以能够创建这些类型的应用程序，是因为 Windows Forms 丰富的特性和工具、跨平台性以及这个版本之后基于社区使用的和反馈的实现成熟度特性。

实际上，.NET 开发人员社区非常喜欢 Windows Forms，甚至到了质疑“早期的 .NET 是集中在基于 Web 的应用程序”这种说法的程度。基于我在培训行业中的朋友和本书第 1 版的销售情况，我认为，有大量的并且数量不断增加的开发人员在编写 Windows Forms 应用程序。正是为了这些开发人员，我们才编写了本书。

谁应该阅读这本书

Who Should Read This Book?

当编写本书的时候，我们希望同时为已经在 .NET 中写过程的程序员和还没有写过程的

³ 当此书中文版出版时，WPF 已经发布。—编者注

程序员提供实际的 Windows Forms 知识。因此，当每个核心的 .NET 话题出现时，我们都会做一个简单的介绍。然而，我们并不试图包括全部 .NET 框架的内容，它太庞大了。相反，当我们需要更多的有用信息的时候，会引用其他的提供了完整细节的文章。特别地，大家会发现我们大量地引用了由 Don Box 和 Chris Selss 所著的 *Essntial .NET* 一书，这是一本很好的补充读物。同样地，我们还推荐了由 Ingo Rammer 所著的 *Advance .NET Remoting* 一书和由 Jeffrey Richter 所著的 *Applied Microsoft .NET Framework Programming* 一书（关于这些书的更多细节，请参见“参考资料”）。

.NET 的一个核心话题对 Windows 程序员具有特殊的意义，我们在附录 C “委托和事件”中详细讨论了这些内容。如果大家是 .NET 编程的初学者，这部分内容尤其重要。但是在学到 Windows Forms 的一些框架性知识之前，我们也不会推荐大家深入研究这一话题（这些框架性的知识可在第 1 章“你好，Windows Forms”中学到三分之一左右）。

我们提供了附录 B “从 MFC 转移到 Windows Forms”和附录 F “文档管理”，这些内容对以前的 MFC 程序员（或者就快成为以前的）来说尤其重要。虽然 Windows Forms 没有提供 MFC 的所有特性，但我们努力地在本书的材料和代码示例中填补了这些缺陷。

另外一个值得注意的地方是：几年前，一位作者编写了他的第一个 5 天培训课程。这个培训课程的话题就是 Windows 95，其中包括了需要花几个小时对 Windows 95 的新控件的讨论：控件的样子、属性、方法和事件以及如何访问它们。这些时间对参与的人来说会觉得很长。某个特定控件的细节只有当我们在需要使用它的时候才令人感兴趣，而且是在当真正需要使用控件的时候，控件的文本和 Visual Studio 的智能提示帮助功能很好地为我们提供了所需的信息。

因此，本书没有完整地介绍任何标准控件。相反，如果某个控件在当前话题的场景下令人感兴趣，我们就会相应地介绍那个控件，就像在第 17 章“应用数据绑定”中介绍 DataGridView 控件一样。另外，为了列出所有的组件和控件，并对每个控件的主要功能做一介绍，附录 D “组件和控件概览”提供了标准控件和组件的列表。我们不会浪费大家的时间去试图提供比 .NET 框架的 SDK 和 Visual Studio 2005 内置的参考文档还全面的内容。相反，本书专注于实际的开发场景，这些内容在别的地方是找不到的。最后，为了帮助大家理解组件和控件的区别以及如何创建它们，我们提供了第 9 章“组件”和第 10 章“控件”。

xxxiii

约定

Conventions

如果你决定尝试阅读本书，我们会感谢你对我们的信任并表达我们希望你顺利完成的心愿。为了帮助你阅读本书，希望你了解我们使用的一些约定。

首先，贯穿全书都使用了一种指代模式（除了本部分）。因为 Chris 和 Michael 工作时联系

紧密并且努力斟酌书中的每句话，我“**I**”这个词的使用实际上意味着“两个作者”，而使用我们“**we**”实际上意味着“作者和读者你”。虽然我们有时会直接使用“你”，但 Chris 和 Michael 倾向于使用“我们”这个词。

Windows Forms 最优秀的地方就在于它的可视化特性，所以我们在本书中使用了大量的插图来说明它的特性。其中一些图像真的需要彩图才能够达到效果，所以你阅读时请确保查看书中的彩色插图。

和插图一样，我们认为代码同样重要。代码是以等宽字体列出的：

```
System.Console.WriteLine("Hello, Windows Forms.");
```

命令行应用程序的启动也以等宽字体列出：

```
C:\> csc.exe hello.cs
```

当代码段的一部分或命令行启动需要特别留意时，我们会以粗体来显示，通常还会提供一个注释：

```
// 注意对 .NET System 命名空间的使用  
System.Console.WriteLine("Hello, Windows Forms.");
```

xxxiv

当我们希望吸引大家对某段代码的注意时，我们会将多余的代码替换为省略号：

```
class MyForm : System.Windows.Forms.Form {  
    ... // 字段  
    private void MyForm_Load(  
        object sender, System.ComponentModel.EventArgs e) {  
  
        MessageBox.Show("Hello from MyForm");  
    }  
}
```

另外，为了让打印出来的代码可读性更好，当命名空间和保护性关键字不能提供更多的信息时，我们通常会将它们去掉：

```
// 缩短的 "System.Windows.Forms.Form" 基类  
class MyForm : Form {  
    ... // 字段  
  
    // 去掉了 "private" 修饰符和 "System.ComponentModel" 命名空间  
    void MyForm_Load(object sender, EventArgs e) {  
        MessageBox.Show("Hello from MyForm");  
    }  
}
```

当显示 .NET 特性时，我们会使用它们的缩写名称：

```
[Serializable] // 而不是[SerializableAttribute]
class MyCustomType {...}
```

C#是为了方便让我们去掉“Attribute”后缀的，但是当我们在网上文档中查找特性类的细节时，要记得添加这个后缀。

为了清晰，我们有时会在打印出来的代码中省略错误检查代码，但是我们会尽量将它们放在可以在 Web 站点上找到的示例代码中。

在正文中，我们通常在需要说明一个将要定义的新术语时，会将这个单词或词组设置为黑体。作为这种类型的术语和声明的一个例子，霸权是指占有优势的影响力或权威，以及可能的商业做法。

有时，当发现键盘快捷键用起来很方便时，我们会提到它们。我提到的键盘快捷键是 Visual Studio 2005 (VS05) 为 C#程序员提供的默认键盘绑定。如果大家没有使用这些键盘绑定，则需要自己映射这些键盘快捷键。

最后，由于这是本书的第 2 版，我们在页边空白中使用图形、在代码中使用注释标记新类型 (New) 和改进过 (Updated) 的类型。另外，如果大家想了解 Windows Forms 2.0 中的新特性，可以阅读附录 A “Windows Forms 2.0 的新增功能”。

◀ xxv

联系

Contact

本书的最新信息，其中包括源代码和勘误，可以在 <http://www.sellbrothers.com/writing/wfbook> 里找到。这个站点还可以让大家发送关于本书的反馈，赞美和批评都可以。

Michael 的致谢

Michael's Acknowledgments

撰写书籍不只是用笔在纸上写字那么简单，也不仅仅是作者的创作。尽管当事人不一定知道，有许多人帮助我们一起完成了本书。

我的父母 Josef 和 Lili:毫无疑问的是，如果没有我父母的无法估量的并且无条件的爱和支持，我不可能完成本书。对我来说，从我出生那天起，他们就一直爱护着我，而我深深地感动着。我该如何感谢他们呢？不能。相反，由于他们提供给我的美好生活，我只能尽我所能地做出更多的贡献。由于这一点，我为本书倾尽了全力。

我的兄弟 Alex:当他申请进某个大学读书的时候，他完全满足入学条件。然而，学校却告诉他，他很可能不能成功地完成学业。当然，他们犯了个很大的错误。当我在写作过程中碰到困难