

PROGRAMMER TO PROGRAMMER™



Professional Windows GUI
Programming using C#

Windows GUI 高级编程

—— C# 编程篇

Wahid Choudhury

Richard Conway

徐燕华 毛尧飞

等著

译



清华大学出版社

TP316.7
9

Windows GUI 高级编程

—C#编程篇

Wahid Choudhury
Richard Conway 等著

北京市版权局著作权合同登记号：01-2002-3177

内 容 简 介

本书以 GUI 编程为重点, 详细介绍了 .NET 下的 Windows 编程技术。主要内容涉及使用或不使用 Visual Studio.NET 创建 Windows 应用程序、运用 Windows Forms 类和控件、高级的数据绑定、处理用户输入、使用 GDI+ 绘图、打印和打印预览、创建自定义控件、.NET 中的多线程编程, 以及部署 Windows 应用程序等。

本书适合于具有一定 .NET 和 C# 编程经验, 但希望了解有关如何在 Windows 应用程序中使用新的开发技术的程序员。并且读者应有使用 MFC 或 Visual Basic 编写 Windows 应用程序的经验。

Wahid Choudhury, Richard Conway et al: Professional Windows GUI Programming using C#
EISBN: 1-86100-766-3

Copyright©2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司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内容。

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。

图书在版编目(CIP)数据

Windows GUI 高级编程——C#编程篇/(美)查德亨利等著; 徐燕华, 毛尧飞译. —北京: 清华大学出版社, 2003

书名原文: Professional Windows GUI Programming using C#

ISBN 7-302-06309-5

I. W... II. ①查...②徐...③毛... III. ①窗口软件 Windows—程序设计②C 语言—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字(2003)第 008014 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

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

责任编辑: 于平

封面设计: 康博

版式设计: 康博

印 刷 者: 北京通州大中印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 27.75 字数: 575 千字

版 次: 2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号: ISBN 7-302-06309-5/TP·4766

印 数: 0001~4000

定 价: 50.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前言

尽管 Web 编程是近年来的热门话题，但是 Windows 编程却在 .NET 下获得新生。部分原因是因为 Web 服务允许利用 Internet，甚至在 Windows 应用程序中也可以利用 Internet。但时至今日，人们才意识到 Windows 程序可以比基于浏览器的应用程序提供更加丰富的技巧。

.NET Framework 最引人注目之处在于它结合了以前只有 C++ 程序才拥有的强大功能和 Visual Basic 的简洁的特点。实际上，使用 Visual Studio .NET，在 C# 中创建功能性 Windows 应用程序和在 VB6 中创建一样的简单。Visual Studio .NET 所采用的拖放式操作说明了 .NET 提供的功能比 VB 提供的功能要强大得多。例如，现在要定制控件的可视化外观就比较简单，这样既可以利用它们的功能，同时也可以创建出我们所希望的外观。即使要重新创建控件也很容易，而不会像以前那样花费大量时间。

的确，如果要利用这些功能，就要稍微多做一些工作，但是这种努力可以做到事半功倍的效果。本书将向您展示 Windows Forms 的功能以及如何利用它们。.NET 的优点就是可以准确地选择所需的复杂程度。无论在什么地方要进行自我控制，都有选项可供使用（在处理 Windows 消息一级的正下方）。在需要标准行为的地方，.NET 会很好地实现我们的要求。本书旨在帮助我们理解 .NET 的功能及通过自我控制能实现什么操作。

本书主要内容

本书针对如何在 Windows 环境中使用 C# 编写程序进行简明扼要而深入的指导。GUI 编程是重点讨论的问题，但对重要的其他相关内容也进行了深入讨论。本书并不是有关 GUI 设计的艺术专著。

本书的使用对象是那些既了解 .NET 和 C# 基本知识，又需要更多有关在实际应用中使用 C# 知识的程序员。本书不仅可以让 .NET 程序员理解 Windows Forms 在 .NET 中的工作原理，还可以了解 .NET 的底层图形 API，GDI+。

在讲述 VS.NET 如何简化 Windows 应用程序的创建工作时，重点讲述实际代码，因此通过学习本书，不使用 VS.NET，读者也可以得心应手地创建成熟的 Windows 应用程序。

第 1 章：.NET 和 Windows

首先简要介绍 Windows 的工作原理，以及操作系统如何使用消息与单个窗口通信。



另外还介绍在使用和不使用 VS.NET 的两种情况下,如何在.NET 中创建 Windows 程序。

第 2 章: 使用 Windows Forms

本章详细介绍 Form 类、如何将标准功能(如菜单、工具栏和对话框)添加到 Windows Forms 上,以及如何在.NET 中创建 MDI 应用程序。

第 3 章: Windows 常用控件

在讲述了在.NET 中构建 Windows 应用程序的基本知识后,本章就开始介绍其中最有用的预定义的控件,这些控件随.NET Framework 一起装载,主要是一些更高级的功能和控件,以及.NET 的新功能和新控件。另外,还会介绍如何将现有 ActiveX 控件结合到 Windows Forms 应用程序中。

第 4 章: 数据绑定

将控件绑定到数据源的能力极其有用,但是过去只是将该功能局限于简单的客户机/服务器应用程序使用。.NET 则改变了这一切,它允许用户将控件绑定到许多不同的对象类型,而不仅仅是数据集。本章将分析如何在多层应用程序中使用该功能利用数据绑定。

第 5 章: 用户输入

Windows 控件都提供键盘和鼠标事件的标准处理,但是在许多情况下需要更高级别的控制。本章将剖析如何才能捕获键盘事件来发现已被按下/释放的组合键,如何才能使用国际化键盘,以及如何使用鼠标事件跟踪鼠标位置、按下鼠标按钮和滚动鼠标轮。

第 6 章: 在.NET 环境下绘图

.NET 提供了一组类专门用于将图形绘制到输出设备上,如用户屏幕。这些类充当名为 GDI+ 的非托管 DLL 的包装器。本章将讲述如何使用 GDI+ 的强大的图形功能,将文本和图像绘制到屏幕上,其中包括使用不同的画笔、画刷和颜色,以及执行图像变换功能,如旋转、缩放和滚动图像。

第 7 章: 打印

如果不向用户提供选项来输出窗口内容,那么 Windows 程序就不能称为完整的。在.NET 中,采用与将窗口内容显示到屏幕上的相同的 GDI+ 库就可以实现这点。本章将讲述如何提供全面的打印支持,其中包括访问页面和打印机设置,提供打印预览,和打印内容。

第 8 章 自定义控件

.NET 提供了 3 种方法来创建自定义控件——扩展现有控件、将几个控件合并成一个复合控件以及重新创建控件,包括其用户界面。本章将介绍这 3 种类型的实例。除了剖

析如何创建控件外，还要介绍如何在设计时通过修改属性网格来更方便地使用它们。

第 9 章：优化 Windows 应用程序

本章将介绍与 Windows Forms 应用程序有关的一些编程问题，尤其是线程技术问题。讨论如何在 .NET 中创建多线程 Windows 应用程序以及相关的问题和缺陷。另外，还要剖析如何在 Windows Forms 应用程序中，通过访问 Web 服务和调用 API 函数来使用其他编程模型。最后，介绍如何使用事件存取器来提高 .NET 的事件处理模型的效率。

第 10 章：部署 Windows 应用程序

既然知道了如何创建功能齐全的 Windows 应用程序，现在就该将它部署到客户机器上。尽管 .NET 夸大了 XCOPY 部署功能，但在实际应用中，终端用户需要更为友好的安装程序来安装和维护应用程序。本章将讨论如何使用 Visual Studio .NET 创建安装项目，使用 Windows Installer 向用户提供友好的安装应用程序。另外还将讨论如何通过 Internet 部署应用程序。

本书读者对象

本书的读者对象是具有一定的 .NET Framework 和 C# 编程经验，又想了解更多如何在实际的 Windows 应用程序中使用新技能的开发人员。本书尤其适合那些想深入了解 Windows Forms 知识，突破 Visual Studio .NET 提供的简单拖放式技术，而超越开发人员可获得的标准界面的程序员。本书的读者应已具有使用 MFC 或 Visual Basic 编写 Windows 应用程序的经验。

使用本书的要求

使用本书必须有一台安装 .NET Framework 的计算机。也就是说，需要运行 Windows 2000 Professional(或更高版本)，或 Windows XP 操作系统。本书采用的图片表明这两个操作系统中的任何一个都与所探讨的示例相兼容。

可以从 <http://msdn.microsoft.com/netframework/downloads/howtoget.asp> 免费下载 .NET Framework。 .NET Framework 有下面两个版本：

- .NET Framework Redistributable——完整的架构，其中包括运行 .NET 应用程序所需的一切内容，约 20Mb。
- .NET Framework SDK(软件开发工具包)——完整的架构和创建 .NET 应用程序的工具、技术文档、示例和教程，要了解更多有关 .NET 的信息可以参考这些资



料, 约 130Mb。

本书设计成“编辑器中性”的。使用简单文本编辑器(如写字板)就可以创建、运行并理解书中所有示例。虽然我们指出使用 Visual Studio .NET 比较好, 但是使用本书不需要 Visual Studio .NET。尽管其他最新的安装应用程序(如 InstallShield)可以和 .NET 一起使用, 但是本书第 10 章还是具体讨论如何使用 Visual Studio .NET 提供的工具部署 Windows 应用程序。

客户技术支持

我们一贯重视读者的意见, 并想知道每位读者对本书的看法, 包括读者喜欢和不喜欢的内容, 以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(邮件地址为 feedback@wrox.com)向我们反馈意见。请确保反馈信息提到本书的书名。

如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时, 通过 Search 工具或书名列表, 可以方便地找到需要的书名。然后, 单击 Code 列中的 Download 超链接, 或者单击本书详细页面中的 Download Code 超链接, 就可以下载相应的示例代码。

当您单击下载本书的代码时, 将会看到带有以下三个选项的 Web 页面:

- 如果您是 Wrox Developer Community 的成员(即如果您已经在 ASPToday、C#Today 或者 Wroxbase 上注册), 就可以使用一贯的用户名和密码的组合进行登录以下载代码。
- 如果不是它们的成员, 则会向您询问是否愿意注册为会员, 以便可以免费下载代码。此外, 也可以从 Wrox Press 下载免费的文章。注册为会员后, 可以得到本书升级版本和新版本的有关信息。
- 第三个选项是完全绕过注册过程, 直接下载代码。

对于本书而言, 不注册也能下载代码, 但是, 如果您愿意注册后下载代码, 您的注册信息不会泄漏给第三方。关于这方面的详细条款和条件, 可以通过单击下载页面上的相关链接来查看。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存文件到本地硬盘上的文件夹中后, 需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时, 通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中, 应确保解压缩程序(WinZip、PKUnzip, 等等)被设置为使用原有的文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请反馈信息给我们，我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。只需将信息发送到 support@wrox.com，您的反馈信息将被检查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 <http://www.wrox.com/>，并通过 Advanced Search 或者书名列表轻松找到本书书名。然后，单击 Book Errata 链接即可，该链接位于本书详细页面中的封面图的下面。

E-Mail 支持

如果您希望直接向详细了解本书的专家咨询本书中的问题，可以发送电子邮件到 support@wrox.com，要求在邮件的主题中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一个典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页码。
- 邮件正文中应包括读者的姓名、联系信息和问题。

我们将不返回您的无用邮件，因为我们仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一封电子邮件时，它将得到下面一系列支持：

- 客户支持——首先，您的信息将被递送到我们的客户支持人员手中，并由他们阅读。他们备有常见问题的文件，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持——接着，一些深层次的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。
- 作者支持——最后，如果编辑不能回答您的问题(这种情况很少发生)，他们会将问题交给本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴地转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们写的书提供技术支持。他们会将答复通过电子邮件发送给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书籍内容直接相关的问题提供支持，对于超出常规书籍支持的问题，您可以从 <http://p2p.wrox.com/forum> 中的公共列表中获得支持信息。



p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们独特的系统将 programmer to programmer™(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家看到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

- (1) 登录 <http://p2p.wrox.com> 站点。
- (2) 从左边的主菜单栏选择一个适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照说明订阅并填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用预订管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最好支持的原因

您可以加入整个邮件列表，也可以接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表，以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目 录

第 1 章 .NET 和 Windows	1
1.1 Windows 应用程序和.NET	2
1.2 消息、句柄和窗口	4
1.3 Windows Forms 命名空间	13
1.4 不使用 Visual Studio .NET 创建窗体	16
1.5 在 Visual Studio .NET 中创建窗体	19
1.6 小结	27
第 2 章 使用 Windows Forms	29
2.1 创建基于窗体的应用程序	29
2.1.1 自定义窗体	32
2.1.2 通过继承获得更好的窗体	33
2.1.3 连接窗体上的事件	34
2.1.4 有模式和无模式窗体	36
2.1.5 处理继承的窗体上的事件	40
2.1.6 使用 VS.NET 向导创建窗体	41
2.2 将菜单添加到窗体中	42
2.2.1 主菜单	42
2.2.2 上下文菜单	45
2.2.3 菜单处理程序	47
2.3 多文档界面的窗体	48
2.3.1 使用属性管理 MDI 窗体	50
2.3.2 改变菜单	51
2.4 常用对话框	52
2.4.1 打开文件的对话框	53
2.4.2 保存文件的对话框	56
2.4.3 颜色选择的对话框	58
2.4.4 VS.NET 和常用对话框	59
2.5 工具栏	60
2.5.1 添加工具栏	61



2.5.2 处理工具栏事件	65
2.5.3 VS.NET 和工具栏	69
2.6 状态栏	69
2.7 小结	72
第 3 章 Windows 常用控件	73
3.1 控件基类	73
3.1.1 控件属性	73
3.1.2 控件容器	75
3.2 常用控件	76
3.2.1 TextBox 控件	76
3.2.2 ListBox 控件	81
3.2.3 ListView 控件	83
3.2.4 ComboBox 控件	85
3.2.5 DomainUpDown 控件	87
3.3 按钮控件	89
3.3.1 CheckBox 控件	92
3.3.2 RadioButton 控件	94
3.3.3 Scrollbar 控件	95
3.3.4 Timer 控件	98
3.3.5 DateTimePicker 控件	100
3.3.6 MonthCalendar 控件	103
3.3.7 TreeView 控件	105
3.4 自定义绘制	108
3.4.1 自定义绘制 ComboBox	108
3.4.2 自定义绘制 Button	113
3.5 在.NET 中使用 ActiveX 控件	117
3.5.1 使用 AxImp	118
3.5.2 简单的 Web 浏览器应用程序	121
3.5.3 Windows 媒体播放器	127
3.5.4 Microsoft Agent 控件	128
3.6 小结	131
第 4 章 数据绑定	132
4.1 .NET 中的数据绑定	132
4.1.1 BindingContext	133

4.1.2	BindingManagerBase	133
4.1.3	Binding 类	135
4.2	简单的数据绑定	136
4.3	更新数据	147
4.4	复杂的绑定	149
4.4.1	绑定要求	149
4.4.2	ITypedList	164
4.4.3	IBindingList 和 IEditableObject	169
4.5	绑定到自定义控件中	180
4.6	小结	183
第 5 章	用户输入	184
5.1	介绍键盘	185
5.1.1	关于编码	185
5.1.2	KeyboardSample 应用程序	186
5.1.3	加速键	191
5.1.4	开发键盘计时器	192
5.1.5	检查键盘修正键	194
5.2	将键盘国际化	194
5.3	失去焦点和得到焦点	198
5.4	响应鼠标	200
5.4.1	鼠标功能	200
5.4.2	跟踪鼠标	203
5.5	使用鼠标绘图	207
5.6	使用工具提示	214
5.7	小结	215
第 6 章	在 .NET 环境下绘图	216
6.1	GDI+ 概述	217
6.2	事件和画图	221
6.3	图形对象	222
6.4	颜色、画笔和画刷	224
6.4.1	透明度	225
6.4.2	已知的颜色	227
6.4.3	渐变颜色	229
6.4.4	影线画刷	232



6.4.5	颜色对话框	233
6.5	形状、路径和区域	234
6.5.1	线条和曲线	235
6.5.2	变换	238
6.5.3	路径	241
6.5.4	阴影和裁剪	242
6.5.5	区域	244
6.6	绘制文字——字体	245
6.6.1	字体对话框	246
6.6.2	已安装的字体族	247
6.6.3	文本方向	249
6.7	处理图像	251
6.7.1	缩放	253
6.7.2	旋转和翻转	254
6.7.3	PictureBox 控件	256
6.7.4	滚动	257
6.8	小结	259
第 7 章	打印	260
7.1	System.Drawing.Printing	260
7.2	在.NET 环境下打印	261
7.2.1	图形	263
7.2.2	PrinterSettings	264
7.2.3	PrintDocument	268
7.2.4	PrintDialog	282
7.2.5	PageSetupDialog	287
7.3	示例：打印 DataGridView	292
7.3.1	设计目标	292
7.3.2	GridPrinting	292
7.4	小结	309
第 8 章	自定义控件	310
8.1	扩展控件	311
8.1.1	创建 TextBox 扩展控件	311
8.1.2	使用 TextBox 扩展控件	321
8.1.3	设计时使用特性和 XML 文档	324

8.2	复合控件	325
8.2.1	创建 AddressControl 复合控件	326
8.2.2	使用 AddressControl	340
8.3	自定义绘制的控件	344
8.3.1	创建 RoundButton 控件	344
8.3.2	使用 RoundButton 控件	352
8.4	深入介绍 PropertyGrid	353
8.4.1	自定义 TypeConverters	353
8.4.2	自定义 UITypeEditors	355
8.5	小结	358
第 9 章	优化 Windows 应用程序	360
9.1	线程的基本问题	360
9.2	应用程序锁定	362
9.3	线程性能	367
9.4	编写多线程应用程序	371
9.4.1	数据库应用程序	371
9.4.2	网络应用程序	378
9.5	不同的 Windows Forms 编程模型	383
9.6	使用 P/Invoke	388
9.7	事件存取器	391
9.8	小结	395
第 10 章	部署 Windows 应用程序	396
10.1	使用 Windows Installer	398
10.1.1	安装向导	398
10.1.2	文件系统编辑器	400
10.1.3	其他编辑器	402
10.1.4	注册表编辑器	403
10.1.5	文件类型编辑器	404
10.1.6	UI 编辑器	404
10.1.7	自定义操作编辑器	409
10.1.8	启动条件编辑器	410
10.1.9	修复和删除	411
10.1.10	重新分布 .NET Framework	412
10.1.11	安装到 GAC	413



10.1.12	通过 Web 使用 Windows Installer	414
10.2	合并模块项目	415
10.3	CAB 文件	416
10.4	处理部署	417
10.5	代码访问安全——保护客户	423
10.6	小结	424

第1章 .NET和Windows

在过去的几年中,关于如何构建应用程序的体系结构,开发人员的想法已经发生了改变。曾经有一段时间,常用的手段是开发一个 Windows 应用程序,使它包含全部 GUI(图形用户接口)代码和大多数业务逻辑,或者至少是包含一个存储过程无法包容的那些内容。这种结构称作“客户机-服务器”体系结构。那时,人们还认为它会代替大型机系统。后来,我们就有了在组件(可能位于不同的机器上,介于人和数据之间)中执行业务逻辑的能力。诸如 MTS、MSMQ 以及 MQSeries 之类的技术,在如何构建应用程序结构方面,给予了开发人员新的思考方式。通过将业务逻辑和表示逻辑分离,可以更容易地实现业务规则中的修改。实现纯业务逻辑的代码,其重用机会比那些与表示代码混合在一起的代码更高。现在,多数业务逻辑位于“中间层”,所以就诞生了多层编程方法。人们认为,这种方法肯定会替代大型机系统。当多层编程技术正在革新的时候,被人们称作万维网的事物也是如此。由于 Web 开发更复杂,所以客户机变得越来越瘦,直到最后达到某些行业权威所声明的那样:PC 机将被废弃,人们希望一切事物都可以从 Internet、内联网或者本地网络上运行。客户机将只不过是显示 HTML 的一种终端类型设备。不是代替大型机,而是回归为类似于大型机的系统,也就是针对集中式系统而工作的非常瘦的“哑”客户机。这一点实际上并没有产生太好的效果,所以我们后退一步考虑。瘦客户机仍然是要考虑的,但是它应该是基于 PC 机的浏览器。这样,就使得能够根据需要将脚本或者小程序下载到客户端并执行。采用这种类型的体系结构,其主要原因之一是,基于 Windows 的应用程序难于安装和维护。如果网络有着数千台客户 PC 机要管理,这种情况尤其突出。那时,就会出现 DLL hell,使应用程序变得不可操作。所以,尽可能尝试着从服务器而不是客户端上执行程序是有意义的。如果不想在客户端上安装任何事物,那么就不必管理任何事物。

这种体系结构的问题是,我们只在客户机上显示 HTML。仅仅借助浏览器,很难给予用户丰富的用户体验。如此一来,我们就必须依赖脚本或者小程序,或者必须在服务器和客户机之间不断来回传送数据。所以,无论是从性能上还是从安全角度出发,这种体系结构都不是吸引人的做法。.NET Framework 的设计人员意识到这一点并确信,高度忠诚的智能客户应用程序会受到人们的欢迎。Web 永远是应用程序开发的重要部分。可是,不是每项应用程序都能够在浏览器中生存,而且它也不应该如此。当今的 PC 机所具有的处理能力是短短几年以前所想不到的。使用这些 PC 机显示 HTML 和其他短小的内容似乎有些浪费。.NET Framework 中的 System.Windows.Forms 命名空间允许开发人员利用这种处理能力的优点。