

# Windows Forms 高级编程

Jason Bell

Benny B. Johansen 等著

袁勤勇 郑魏 等译

清华大学出版社

(京) 新登字 158 号

北京市版权局著作权合同登记号: 01-2001-4309

### 内 容 简 介

本书详细介绍了使用 Windows Forms 构建应用程序所需的各方面的信息, 包括窗体和控件的创建和处理, 也讨论了分布式编程的内容。本书还提供了许多实用的示例, 演示了运行中的 Windows Forms。示例的内容包括了如何在 Web 浏览器里显示 Windows Forms, 如何部署 Windows Forms 应用程序。最后给出了一个案例研究。

本书适用于具有 C# 或 VB .NET 初级或中级知识的开发人员。

Jason Bell, Benny B. Johansen: Professional Windows Forms

EISBN: 1-861005-54-7

Copyright© 2001 by Wrox Press Ltd.

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

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由清华大学出版社和英国乐思出版公司合作出版。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。

书 名: Windows Forms 高级编程

作 者: Jason Bell Benny B. Johansen 等著 袁勤勇 郑魏 等译

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.tsinghua.edu.cn>

责任编辑: 郭东青

封面设计: 康博

版式设计: 康博

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 38.25 字数: 979 千字

版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

书 号: ISBN 7-302-05521-1/TP·3248

印 数: 0001~5000

定 价: 65.00 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。

作为国内 IT 出版社中最知名品牌，清华大学出版社与 Wrox 公司合作引进了这套 Professional 系列，然后迅速组织了一批相关领域的知名专家学者进行翻译，经过编辑人员认真细致的加工后，现陆续奉献给广大读者。

读者可以从 [www.wrox.com](http://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](http://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

在谈论.NET 技术推动基于 Web 应用程序的方式时,很容易忽略这样一个事实,即 Microsoft 投入了大量的时间用于改进开发传统的 Windows 风格的应用程序的方法。

Windows Forms 是 Microsoft 的新成就,它允许我们创建复杂的、丰富的、数据感知用户界面应用程序。虽然 Windows Forms 令人印象深刻,但很多用户可能觉得昂贵的客户应用程序仍然不是切实可行的。在.NET 下,这个问题实际上不复存在,因为简单的 XCOPY 部署手段意味着 Windows Forms 应用程序的部署和支持更为便宜。这将导致 Windows Forms 应用程序开始取代以前使用的、基于浏览器的应用程序,即使 Windows Forms 应用程序存在用户界面较弱的缺点。

无论如何,由于大量的新特性,.NET Framework 提供的所有益处以及大量的改进和升级意味着 Windows Forms 将成为构造基于 Windows 界面的应用程序的工具,这是不容置疑的。预祝广大用户愉快地开发 Windows Forms 应用程序!

## 本书内容

第 1 章通过介绍.NET Framework 的概况将本书的内容划分成几个部分。该章讨论了 Framework 的各个组件,并对 Windows Forms 在何处保存这些组件加以特别关注。

第 2 章介绍 Visual Studio.NET 以及怎样使用它,以使编程更加容易。本章为您提供了有效地编制 Windows Forms 应用程序所需要了解的所有内容。从创建解决方案、项目和文件开始,处理诸如 intellisense 的特性,最后完成构建和调试。本章最后介绍了.NET 中的命令行编译。

第 3 章介绍在新的.NET 语言中固有的、重要的语言特性。所讨论主题是面向对象(Object Oriented)编程,但是即使经验丰富的 C 和 C++程序员也可能发现感兴趣的专题,如 Windows Forms 中的 Visual Inheritance(可视化继承)或者对已立法则的支持是如何升级的。

第 4 章是真正介绍 Windows Forms 的第 1 章。该章剖析了如何创建窗体,其中包括手工进行窗体编码的基本范例。在该章中,我们将仔细研究由 Visual Studio.NET 为一个基本窗体所生成的代码,以更好地了解内部所发生的一切。同时开始开发第一个应用程序,并特别介绍如何在窗体中放置、安排控件。

第 5 章介绍人机交互,演示 Windows Forms 如何隐性地支持优秀设计的原则。除了讨论界面设计的理论和实践以外,本章也将介绍一系列可用的窗体,并介绍怎样使用它们来控制应用程序的流程。

第 6 章介绍事件。由于使用事件驱动的应用程序时,理解如何实现事件的使用是非常重要的,因此本章介绍现象下的本质及其所包含的内部机制。例如,本章将介绍怎样用委托来编程,怎样用 VB.NET 和 C#来编写事件处理程序。

第 7 章深入探讨数据。本章将介绍 Visual Studio.NET 提供的用于连接到数据源的控件,还



将介绍 ADO.NET 体系结构。此外，我们还将通过一系列的范例，来介绍简单和复杂的数据绑定，以及怎样绑定到集合和数组。

第 8、9、10 章介绍 Windows Forms 可用的各种不同控件。这几章提供了在几个不同应用程序中可用的所有控件，然后讨论它们更重要的方法和属性。本章也介绍了如何在 Windows Forms 程序中使用 ActiveX 控件。但是，必须指出，由于包含在这些控件中的属性和方法的数量众多，所以本章只讨论重要的属性和方法。

第 11 章介绍 Windows Forms 组件。本章演示了所有可用组件的各种不同用途，以及怎样把它们合并到应用程序中，以增强应用程序的功能。所覆盖的专题从监视文件系统和事件日志到把水晶报表合并到窗体中。

第 12 章是关于 GDI+ 的，首先简单介绍图形绘制。然后，讨论绘图的重要问题，包括上色、缩放以及怎样绘制诸如贝济埃曲线的路径。本章将通过一系列范例来演示用不同方法生成的图形，例如用钢笔和纹理刷绘制的图形。

第 13 章涵盖了国际化的专题。现在，开发一个能支持多国用户的应用程序已经显得日益重要。正因为如此，Microsoft 现在提供了极好的支持。特别是，本章介绍了国际化的两个重要方面，它们分别是全球化和本地化，并演示如何在应用程序中有效地使用它们。

第 14 章演示怎样调试应用程序。本章讨论了 .NET 中的新调试特性，介绍怎样有效地使用调试器。然后，通过例子显示不同的调试窗口，最后介绍怎样优化应用程序。

第 15 章解决应用程序如何打包和部署的问题。本章着重介绍怎样使用 Visual Studio.NET 来包装 Windows Forms 应用程序。在其他内容中，我们也涵盖了 Windows Installer、Windows Forms 应用程序的卸载以及如何使用 XCOPY 进行部署等知识。

第 16 章介绍非 ASP.NET 专有属性的 Web 服务。本章创建了一个 Web 服务，然后用 ASP.NET Web Form 和 Windows Form 来使用它。本章也使用命令行编译和运行代码，来演示怎样执行非 Visual Studio 应用程序。

第 17 章介绍在浏览器中显示窗体和控件是何等的容易。尤其是，我们编写了一段移动代码，建立 IIS，并提供了几个演示移动代码不同功能的例子。

第 18 章是一个案例研究，在这个案例中，我们创建了复杂的 Windows Forms 应用程序。应用程序是一个 WDE(Website Development Environment, Web 站点开发环境)，它组合了本书中很多专题。特别是，它使用多文档界面(Multiple Document Interface)来表示单独页面的源、HTML 和浏览器视图。它也使用浏览器风格的窗体，而这些窗体允许我们记录站点中的 Web 页面，进而支持有效的站点维护。

## 本书读者对象

本书是为有一些 C# 或 Visual Basic.NET 开发经验的开发人员所撰写的。本书不是针对编程新手的，所以没有介绍 C# 或 VB.NET 的编程技巧或这些编程语言的基础。我们竭尽全力提供使用 Windows Forms 技术所需要的所有信息。

为此，本书首先在较高层次介绍 Windows Forms 的作用。接着，研究 Windows Forms 提供的特性，以及怎样使用 Visual Studio.NET (以及命令行) 来对它们进行操作。本书最后提供了一

些非常实用的 Windows Forms 应用程序。

一般说来，只要熟悉.NET 语言的基础知识，就能够使用本书，与 Visual Studio.NET 协同工作，不需要更多的参考资料。由于 Windows Forms 与其他知识融合在一起，因此为了使用演示的一些应用程序，您必须熟悉 Microsoft 软件。本书中范例代码都以 VB.NET 和 C# 方式提供，以在必要时指出这两种语言的区别。

## 使用本书的前提条件

要运行本书中的代码，需要具备下列条件：

- Windows 2000 或 Windows XP
- Visual Studio.NET

出于对本书中所介绍技术的考虑，我们强烈地建议您下载源代码，以供使用。

## 客户支持

笔者总是能从读者反馈中获得价值，笔者希望了解读者对本书的看法，以及读者喜欢什么，不喜欢什么，希望笔者在哪些方面做得更好。读者可以通过发送 e-mail 到 [feedback@wrox.com](mailto:feedback@wrox.com)，来提出自己的意见。请务必在您的邮件中加上本书的书名。

## 如何下载本书的范例代码

读者浏览 Wrox 站点：<http://www.wrox.com/>时，只需简单地通过搜索功能，或使用书名列表，来找到本书。单击 Code 栏中的 Download 按钮，或本书的细节页中的 Download Code 按钮，即可下载范例代码。

我们站点上可供下载的文件都已经用 WinZip 进行过压缩。读者把文件保存在自己硬盘上的文件夹时，应该用诸如 WinZip 或 PKUnzip 等的解压缩程序来解压缩文件。读者解压缩文件时，代码通常解压缩到各章的文件夹中。在开始解压缩文件时，要保证软件(WinZip、PKUnzip 等)设置为使用文件夹名字。

## 勘误表

我们尽力保证在文本和代码中没有错误。但是，没有任何东西是完美的，错误是不可避免的。如果读者在我们的书中发现错误，例如拼写错误或代码错误，我们衷心感谢您的反馈。通过发送勘误表，您可以减少另一个读者遭受挫折的时间，当然，您将帮助我们提供更高质量的信息。只需简单地把信息用电子邮件发送到 [support@wrox.com](mailto:support@wrox.com)，您的信息将得到检查，如果信息准确，将发送到本书的勘误表页中，或用于本书的后续版本中。

要在 Web 站点上找到勘误表，进入 <http://www.wrox.com/>，然后通过 Advanced Search 或书名列表来找到本书。单击 Book Errata 链接，Book Errata 链接位于本书的细节页中封面图形之下。



## 电子邮件支持

如果您希望就本书中的问题直接询问详细了解本书的专家, 就向 [support@wrox.com](mailto:support@wrox.com) 发送电子邮件, 在邮件的主题区填入书名和 ISBN 的后 4 位。典型的电子邮件应该包括下列内容:

- 在邮件的主题区填入书名、ISBN 的后 4 位和问题所在的页码。
- 在邮件中包括您的姓名、联系信息和问题。

我们不会向您发送邮寄宣传品。我们需要详细的信息以节省您和我们的时间。当您发送电子邮件时, 邮件将经历下列支持环节:

- 客户支持——您的邮件发送给客户支持人员, 他们是读到您的邮件的第一批人。他们拥有有关常见问题的文件, 他们将立即回答有关本书和 Web 站点的常见问题。

- 编辑——更深入的查询转发给负责本书的技术编辑。他们拥有编程语言和特定产品的经验, 能够回答更深入的技术问题。

- 作者——最后, 如果编辑不能回答您的问题, 他/她将问题转发给作者。我们尽量保护作者, 以免打扰其写作, 但是我们也乐意把特殊的问题转发给他们。所有的 Wrox 作者都提供对自己作品的支持。他们将把答案通过电子邮件发给客户和编辑, 所有读者将获益。

Wrox Support 过程只支持与我们的出版物直接相关的问题。通过在 <http://p2p.wrox.com/forum> 上的社团列表来提供对超出通常范围的问题的支持。

### p2p.wrox.com

欲与作者进行讨论, 或与同行进行讨论, 请加入 P2P 邮件列表。我们的系统在邮件列表、论坛和新闻组上提供了程序员与程序员之间的联系, 它们都对一对一的电子邮件支持系统进行补充。如果您在 P2P 上提出一个问题, 那么在我们的邮件列表中出现的众多作者和其他专家将对此进行研究。不仅仅在阅读本书的时候, 而且在开发应用程序的时候, 在 [p2p.wrox.com](http://p2p.wrox.com) 上, 您都将会找到很多对您有帮助的不同列表。特别适合本书的是 `pro_windows_forms` 和 `vs_dotnet` 列表。

按照下列步骤订购邮件列表:

- (1) 进入 <http://p2p.wrox.com/>。
- (2) 在左边的菜单栏中选择合适的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照指令订阅, 并填入电子邮件地址和密码。
- (5) 回复所接收的确认电子邮件。

使用订阅管理器以加入更多列表, 并设置电子邮件参数。

### 为什么我们的系统提供了最佳支持

您可以选择加入邮件列表, 或接收它们作为每周文摘。如果您没有时间或设备来接收邮件列表, 那么您可以搜索我们联机文档。垃圾邮件都被删除, 您自己的电子邮件地址由 Lyris 系统保护。有关加入和离开邮件列表的查询和其他有关列表一般查询, 应该发送邮件到 [listsupport@p2p.wrox.com](mailto:listsupport@p2p.wrox.com)。

# 目 录

<b>第 1 章 .NET Framework 简介</b> .....	1
1.1 .NET 的伟大之处 .....	2
1.1.1 Windows Forms 类层次结构 .....	2
1.1.2 .NET 分布式应用技术 .....	2
1.1.3 程序员效率 .....	4
1.1.4 公共语言运行时环境 .....	5
1.1.5 基于证据的安全性 .....	10
1.1.6 无用存储单元收集 .....	12
1.1.7 CLR 中缩写和术语汇总 .....	13
1.2 在.NET 中创建用户界面 .....	13
1.2.1 Windows Forms .....	14
1.2.2 Web Forms .....	15
1.2.3 我们应该使用什么: Windows Forms 还是 Web Forms .....	15
1.3 XML .....	16
1.4 Web 服务 .....	17
1.5 通过 COM+的组件服务 .....	18
1.5.1 事务处理 .....	18
1.5.2 对象存储池 .....	18
1.5.3 排队组件 .....	19
1.6 Visual Studio.NET .....	19
1.7 小结 .....	20
<b>第 2 章 Visual Studio .NET 概述</b> .....	21
2.1 创建第一个项目 .....	21
2.2 解决方案、项目和文件 .....	23
2.3 File 菜单 .....	24
2.4 View 菜单 .....	25
2.4.1 Solution Explorer 窗口 .....	26
2.4.2 Class View 窗口 .....	27
2.4.3 Code 菜单项 .....	28
2.4.4 Designer 菜单项 .....	29
2.4.5 Toolbox 菜单项 .....	29



2.5	编写程序代码	31
2.5.1	智能感知	31
2.5.2	使用程序代码向导	32
2.6	Edit 菜单	33
2.7	项目管理和 Project 菜单	35
2.7.1	项目内管理	36
2.7.2	项目属性	38
2.7.3	添加到项目	40
2.8	Build 菜单	41
2.9	Debug 菜单	42
2.10	Tools 菜单	42
2.11	Help 菜单	44
2.12	命令行开发	44
2.13	小结	49
<b>第 3 章</b>	<b>继承和其他重要的新语言特性</b>	<b>50</b>
3.1	面向对象程序设计介绍	50
3.1.1	对象和类	51
3.1.2	关联	52
3.1.3	继承	54
3.1.4	多态性	55
3.1.5	抽象、封装和程序代码重用	56
3.2	WroxGraph: 扩展控件	57
3.2.1	Controls 和 GDI+ 的基础知识简介	58
3.2.2	完整的应用程序	59
3.2.3	声明继承	61
3.2.4	重写能力的说明	62
3.2.5	抽象基类和接口	62
3.2.6	访问和属性的说明	63
3.2.7	“Me”、“MyBase”和“MyClass”	66
3.2.8	函数重载	68
3.2.9	对象构造函数	70
3.2.10	无用存储单元收集	72
3.2.11	基本对象	73
3.2.12	强制转换	75
3.2.13	使用委托	76
3.3	可视化窗体继承	78
3.3.1	测试 Framework	78

---

3.3.2	基本窗体	78
3.3.3	使用 Inheritance Picker	81
3.3.4	“OOMathTest”窗体	82
3.3.5	共用成员变量和方法	84
3.4	接口	85
3.4.1	声明接口	85
3.4.2	实现接口	85
3.4.3	实现多重接口	88
3.5	结构化错误处理	89
3.5.1	“Try、Catch、Finally”	89
3.5.2	产生和显示异常	90
3.6	小结	91
<b>第 4 章</b>	<b>Windows Forms 简介</b>	<b>92</b>
4.1	Windows Forms 与 Web Forms	92
4.2	Windows Forms 的特性	93
4.3	窗体类型	95
4.4	.NET Framework 中的 Windows Forms	97
4.5	创建窗体	98
4.6	在 Visual Studio.NET 中自动生成窗体的代码	102
4.7	“Form”类属性	107
4.8	WroxEdit	108
4.8.1	在窗体上放置控件	109
4.8.2	设置窗体的位置	116
4.9	向应用程序中增加窗体	118
4.10	WroxWarnings	118
4.11	小结	124
<b>第 5 章</b>	<b>Windows Forms 和界面设计</b>	<b>125</b>
5.1	设计一个用户界面	125
5.1.1	易学性	126
5.1.2	有效性	126
5.1.3	灵活性	127
5.1.4	健壮性	127
5.2	实用用户界面设计	128
5.2.1	单文档界面窗体	129
5.2.2	多文档界面窗体	130
5.2.3	Explorer 样式的界面	132



5.2.4 窗体元素.....	140
5.3 小结.....	142
<b>第 6 章 事件处理</b> .....	<b>143</b>
6.1 常用的 Windows Forms 事件.....	144
6.2 理解委托.....	145
6.2.1 C#中的委托.....	146
6.2.2 VB.NET 中的委托.....	151
6.3 理解事件.....	154
6.3.1 用 C#编写事件处理程序.....	155
6.3.2 用 VB .NET 编写事件处理程序.....	158
6.4 通过重写方法实现事件.....	162
6.4.1 .NET 事件和委托的基本原理.....	165
6.4.2 用 C# 验证文本框.....	166
6.4.3 对两个事件使用一个事件处理程序.....	168
6.4.4 在 VB .NET 中验证文本框.....	168
6.5 实现和引发事件.....	169
6.6 小结.....	172
<b>第 7 章 Windows Forms 和数据</b> .....	<b>173</b>
7.1 例子.....	173
7.2 ADO.NET 体系结构概述.....	174
7.3 数据绑定.....	193
7.4 向导.....	199
7.4.1 DataForm 向导示例.....	199
7.4.2 利用 DataForm 向导创建 DataGrid.....	205
7.4.3 利用两个 DataGrid 控件的 Parent/Child.....	212
7.4.4 参数化查询的例子.....	213
7.5 绑定到数组和集合.....	215
7.5.1 CollectionListBox 示例.....	216
7.5.2 DataTableListBox 示例.....	217
7.6 小结.....	219
<b>第 8 章 基础控件和对话框控件</b> .....	<b>220</b>
8.1 WroxCount.....	222
8.1.1 “Form” .....	222
8.1.2 “Panel” .....	225
8.1.3 “Label” .....	227

8.1.4	“CheckBox”	229
8.1.5	“TextBox”	232
8.1.6	“Button”	233
8.1.7	“LinkLabel”	236
8.2	WroxTranslator	238
8.2.1	“RadioButton”	239
8.2.2	“GroupBox”	241
8.2.3	“ListBox”	242
8.2.4	“TrackBar”	245
8.2.5	“PictureBox”	247
8.2.6	“RichTextBox”	248
8.2.7	“HScrollBar” 和 “VScrollBar”	250
8.3	WroxEditThis	252
8.3.1	对话框	252
8.3.2	“OpenFileDialog”	254
8.3.3	“SaveFileDialog”	261
8.3.4	“FontDialog”	264
8.3.5	“ColorDialog”	267
8.3.6	“PrintDocument”	269
8.3.7	“PrintPreviewControl”	275
8.3.8	“PageSetupDialog”	276
8.3.9	“PrintPreviewDialog”	278
8.3.10	“PrintDialog”	280
8.4	小结	281
<b>第 9 章</b>	<b>外部控件和 ActiveX 控件</b>	<b>283</b>
9.1	WroxMenu	283
9.1.1	术语	285
9.1.2	“MainMenu” 和 “MenuItem”	285
9.1.3	上下文菜单	297
9.1.4	“ToolBar” 和 “ToolBarButton”	298
9.1.5	Tooltip	303
9.1.6	“StatusBar” 和 “StatusBarPanel”	305
9.1.7	“Timer”	310
9.1.8	“ImageList”	311
9.1.9	使用 ActiveX 控件	314
9.2	小结	315



第 10 章 高级控件 .....	316
10.1 日程安排应用程序 .....	316
10.1.1 “NotifyIcon” .....	317
10.1.2 “TabControl” 和 “TabPage” .....	318
10.1.3 “Splitter” .....	322
10.1.4 “MonthCalendar” .....	324
10.1.5 “DateTimePicker” .....	328
10.1.6 “ComboBox” .....	331
10.1.7 “CheckedListBox” .....	334
10.1.8 “TreeView” .....	339
10.1.9 “ListView” .....	349
10.1.10 “NumericUpDown” .....	356
10.1.11 “DomainUpDown” .....	359
10.1.12 “ErrorProvider” .....	361
10.1.13 “HelpProvider” .....	365
10.1.14 “ProgressBar” .....	367
10.2 小结 .....	368
第 11 章 组件和报表 .....	369
11.1 WroxAdmin .....	369
11.1.1 MessageQueue 组件 .....	370
11.1.2 FileSystemWatcher 组件 .....	377
11.1.3 EventLog 组件 .....	382
11.1.4 PerformanceCounter 组件 .....	387
11.1.5 ServiceController 组件 .....	393
11.1.6 Process 组件 .....	396
11.1.7 DirectoryEntry 组件 .....	399
11.1.8 DirectorySearcher 组件 .....	405
11.1.9 CrystalReportViewer 和 ReportDocument .....	409
11.2 小结 .....	415
第 12 章 GDI+介绍 .....	416
12.1 绘图概述 .....	417
12.1.1 “Graphics” 类 .....	417
12.1.2 坐标系统 .....	420
12.1.3 颜色 .....	424
12.1.4 使用 Pen 绘图 .....	425
12.1.5 使用 Brush 绘图 .....	431

12.1.6	使用 Font 绘制文本	435
12.1.7	使用 Image 绘图	438
12.1.8	用 Texture Brush 绘图	440
12.1.9	用带纹理刷的 Pen 绘图	441
12.1.10	使用 Image 绘制文本	442
12.1.11	几何转换和 Alpha 组件	443
12.1.12	水平和垂直滚动	444
12.1.13	打印	447
12.2	GDI+的高级功能	451
12.3	小结	453
<b>第 13 章</b>	<b>国际化</b>	<b>454</b>
13.1	全球化	454
13.1.1	文化介绍	455
13.1.2	改变文化	455
13.1.3	“CultureInfo”类	459
13.1.4	文化相关的数据	460
13.2	本地化	462
13.2.1	创建和使用资源	462
13.2.2	构建本地化窗体	466
13.2.3	使用 WinRes 来支持本地化	468
13.3	小结	469
<b>第 14 章</b>	<b>调试和优化</b>	<b>470</b>
14.1	在.NET 中调试新特性	470
14.1.1	本机 WIN32 应用程序的调试	471
14.1.2	调试器设置	471
14.2	使用调试器	474
14.3	调试托管代码	478
14.3.1	调试和跟踪类	478
14.3.2	断言	479
14.3.3	调试窗口	480
14.3.4	把调试器附着在运行进程上	487
14.4	异常处理	488
14.5	优化和运行时概要记录	489
14.6	监控性能	491
14.6.1	性能阈值	491
14.6.2	性能计数器	491



14.7	小结	492
<b>第 15 章</b>	<b>程序打包和部署</b>	<b>493</b>
15.1	打包和部署过程	493
15.1.1	打包	493
15.1.2	部署	494
15.1.3	Windows Installer	494
15.1.4	安装打包文件“.msi”	495
15.1.5	使用 VS.NET 打包和部署	496
15.1.6	定制安装	502
15.1.7	使用 CAB 文件打包和部署	505
15.1.8	使用自展开可执行文件打包和部署	507
15.1.9	使用“XCOPY”部署	507
15.1.10	卸载 Windows Forms 应用程序	507
15.1.11	使用命令行部署	507
15.2	小结	508
<b>第 16 章</b>	<b>Web 服务和 Windows Forms(带 SDK)</b>	<b>509</b>
16.1	什么是 Web 服务	509
16.1.1	创建 Web 服务	511
16.1.2	使用 Web 服务	517
16.2	小结	531
<b>第 17 章</b>	<b>Web 页中的 Windows Forms</b>	<b>533</b>
17.1	改善和约束	533
17.2	移动编码概述	534
17.2.1	安全性	535
17.2.2	实现概述	536
17.3	示例	536
17.3.1	安装和配置 IIS	537
17.3.2	移动编码示例：“Hello World”	539
17.3.3	定制对话框	542
17.3.4	Web 页里的窗体	544
17.3.5	写入文件	546
17.3.6	防止哄骗控件	548
17.4	小结	548
<b>第 18 章</b>	<b>案例研究：WebMate Web 站点开发环境</b>	<b>549</b>
18.1	WebMate 介绍	550

---

18.2	WebMate 使用	552
18.3	源代码	559
18.3.1	面向对象的体系结构	559
18.3.2	项目编码	561
18.3.3	WebMate 类的层次结构	563
18.4	保存项目	565
18.5	打开项目	569
18.6	拖放文件	577
18.7	编辑 Web 页	583
18.7.1	查看 Web 页的编辑结果	589
18.7.2	关闭 WebMate	592
18.8	小结	594

# 第1章 .NET Framework简介

根据一位主导产业的 CIO 所说, 在所有 Internet 应用软件中有 99% 已经写就。考虑到未来 5 年、10 年以及 20 年中技术进步的速度, 市场方面的变化, 以及 Internet 用户期望值方面的变化, 这很可能是一个保守的估计。

.com 经历了不合理的极大繁荣(以及萧条), 随着围绕在其周围的炒作以及兴奋趋于平静, 我们进入了另一个时代, Internet 被理解为它本来的意义——信息技术人员工具箱中的另一个工具(尽管是非常重要的一个工具)。这一技术将是将来若干代应用程序的平台, 而不仅仅是创建基于浏览器的应用程序的平台。用户的期望值已经变化, 并且常常更频繁地要求新型应用程序与整个世界交互作用, 而不是依靠本地计算机资源。也就是通过网络同其他的计算机通信, 它通常的意思是使应用程序能够与 Internet 通信。

实际上, 不能访问 Internet 上资源的孤立应用程序的时代已经结束了, 贯穿 Microsoft.NET 战略整体结构的线索是分布计算的简化。分布式应用程序的各部分运行在不同的计算机上, 并且出现于应用程序中的数据可以来自网络上许多不同的来源。

.NET Framework 是一个开发平台, 目的是使创建大量应用程序的工作相对简单。即使如此, 如果它仅仅稍优于 MFC、Visual Basic、PowerBuilder、Visual Foxpro 或过去的任何一个软件开发框架, 那么它或许将不是这么重要的事物了。它在 Windows 开发人员中引起如此轰动, 存在着若干原因: 一个是上面强调的分布式计算; 另一个是分层类库以及相关的 .NET 工具的高质量, 其中最主要的工具是 Visual Studio .NET。

不论我们在 .NET 中是创建 Windows 应用程序, 还是创建 Web 应用程序, 都存在某些 Framework 的特性, 这些特性几乎一定会在某些时刻开始起作用, 它们在 .NET 战略中嵌入得太深了。同时, 它们为基于 .NET 的应用程序带来空前水平的 Internet 意识, 而这是整个 .NET 战略所不可缺少的。所以, 虽然在我们开始创建 Windows Forms 应用程序之前不必了解这些技术, 但我们还是应该对它们进行一个简要的、高层次的了解。

## 本章主要内容:

- 公共语言运行时环境(CLR)
- 在 .NET 中创建用户界面
- XML
- Web 服务
- 通过 COM+ 的组件服务

最终, 利用 Visual Studio .NET, Microsoft 提高了关于集成开发环境(IDE)质量和性能的门槛。在本章中, 我们将快速了解一下此工具新特性的一些要点。本书中的其他章节将对 Visual Studio .NET 的用法和功能进行详细研究。