

PROGRAMMER TO PROGRAMMER™



Professional WPF Programming

.NET Development with the Windows Presentation Foundation

WPF 高级编程



Chris Andrade
(美) Shawn Livermore 等著
Mike Meyers
王德才 吴明飞 译



清华大学出版社

WPF 高级编程

Chris Andrade

(美) Shawn Livermore 等著

Mike Meyers

王德才 吴明飞 译

清华大学出版社

北 京

Chris Andrade, Shawn Livermore, Mike Meyers, Scott Van Vliet
Professional WPF Programming: .NET Development with the Windows Presentation Foundation
EISBN: 978-0-470-04180-2

Copyright © 2007 by Wiley Publishing, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 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。

北京市版权局著作权合同登记号 图字：01-2007-4044

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，翻印必究。举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

WPF 高级编程/(美) 安德拉德(Andrade, C.) 等著; 王德才 吴明飞 译.
—北京: 清华大学出版社, 2008.10
书名原文: Professional WPF Programming: .NET Development with the Windows Presentation Foundation
ISBN 978-7-302-18356-3

I .W… II. ①安…②王…③吴… III. 窗口软件, Windows Vista—用户界面—程序设计 IV.TP316.7

中国版本图书馆 CIP 数据核字(2008)第 120359 号

责任编辑: 王 军 郑雪梅

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京密云胶印厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 25.75 插 页: 2 字 数: 627 千字

版 次: 2008 年 10 月第 1 版 印 次: 2008 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 56.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 025897-01

前 言

对于使用 Microsoft 技术的开发人员来说，这是一个激动人心的时刻。众多新平台、新技术以及新工具的不断涌现，使开发人员的舞台迅速扩大。其中，来自微软的一个新平台是 .NET Framework 3.0，Windows Presentation Foundation(WPF)是该平台的一个核心组件，也是本书的主题。WPF 为开发人员和设计人员提供了一个统一的平台，便于他们创建能够充分利用现代 PC 硬件设备的图形功能的多媒体(rich-media)应用程序。

在过去的几十年中，我们经历了使用命令行界面的漫长时间。今天的应用程序用户，希望使用新型的、吸引人的可视化交互界面，其中部分原因是他们在 Internet 上面对的是媒体和内容都很丰富的用户界面。WPF 正是为迎合这种需求而诞生的，它可以综合使用多种类型的媒体，包括动画、视频和音频，来创建富用户界面。此外，通过使用一种新的标记语言 XAML 和一套新的设计工具 Microsoft Expression Blend，设计人员和开发人员能够在开发同一个项目时实现无缝合作，这在以前是不可能的。在 WPF 推出之前，设计人员创建各种图形元素，并将这些图形元素以图像文件的形式交给开发人员，然后开发人员使用这些图形元素构建用户界面。现在，设计人员可以使用 Expression Blend 设计用户界面，并将设计模型保存为 XAML 文件，然后只需将 XAML 文件交给开发人员，开发人员再在 Visual Studio 中使用 WPF 编写代码即可。

本书将详细讲解 Windows Presentation Foundation 的概念和构成组件。通过本书可以学习如何创建富用户界面，如何使用 WPF 提供的各种新控件，以及如何利用 WPF 提供的新内容模型。本书将探讨 WPF 对象模型，以及 WPF 新提供的子系统，例如附属属性系统(dependency property system)和事件传递模型(routed event model)。在本书中还将介绍如何开发和配置用于桌面和浏览器的 WPF 应用程序。本书中还包括新的 XAML 语法(XAML 是一种为 WPF 应用程序定义用户界面的新标记语言，不管这些应用程序是基于 Windows 环境的还是面向 Web 的)、Expression Blend 使用基础和 Microsoft 新提供的图形设计工具。

WPF 是一个非常大的平台，本书将尽可能多地涉及其各个方面，我们的目标是使读者不仅了解 WPF 平台提供了哪些内容，而且学会如何使用该平台。当然，因为 WPF 是一个很大的平台，所以本书中不可能囊括全部内容，但是我们力争让读者掌握开始使用 WPF 的一些基本知识。本书不仅有趣、富有教育性，而且能够为读者使用 Windows Presentation Foundation 开发富用户界面打下坚实的基础。

本书读者对象

本书主要面向准备学习使用 Microsoft 新的呈现架构(presentation framework)WPF 开发应用程序的 .NET Framework 开发人员。对于正在考虑使用新的 Expression Blend 系列设计

工具的设计人员，以及那些想了解这些工具是如何加强与设计人员合作的开发人员，本书也是很有帮助的。对于没有 .NET 开发经验的程序员，本书在学习新的 XAML 语法上有一定的参考价值。但是对于这部分读者，书中的大部分内容比较高级，因此建议先学习 .NET 2.0，作为学习 WPF 的第一步。

本书适合以下几类读者：

- 准备学习 WPF 新平台的 Microsoft .NET 应用程序开发人员。
- 准备研究 Microsoft Expression Blend 工具，学习使用 WPF 及相关工具所提供的协作特性的设计人员。

此外，熟悉以下相关技术的人员，也很适合阅读学习本书：

- Microsoft .NET Framework
- C#
- Web 服务
- HTML
- CSS

以上读者，对 .NET Framework 和 C# 有一个基本理解是必需的，而熟悉 Web 服务、HTML 标记语言以及 CSS 概念对于学习本书中的内容有帮助，但不是必需的。对于没有 .NET Framework 和 C# 基础的开发人员，可先学习《.NET Framework 2.0 高级编程》和《C# 2005 高级编程(第 4 版)》(均由清华大学出版社引进并出版)这一类的书籍。

读者并不需要逐章学习本书。我们建议先学习第 1 章至第 3 章，对 WPF 和 XAML 有一个基本理解，然后浏览后面的几章，从中挑选出您感兴趣或需要的部分进行学习。

本书主要内容

本书主要介绍 WPF 开发技术、模式和案例。

本书以“WPF 概述”作为开始。在第 1 章中，首先回答了“WPF 是什么？”、“如何开始使用 WPF？”以及“WPF 能够带来什么？”等几个问题，然后详细分析了 WPF 开发平台提供的各个子系统及其图形特征。

概述 WPF 之后，将进入 WPF 开发技术部分。本书通过使用 Visual Studio 创建几个示例程序，带领您快速进入应用程序开发。此外，还将介绍 XAML——创建用户界面的新标记语言。XAML 实际上是独立于 WPF 的一项技术，但在 WPF 中广泛使用了 XAML。

掌握了 WPF 的基本概念和开发技术之后，本书将带领读者进入设计工具的世界。Microsoft 提供了许多引人注目的新的设计工具系列。在本书中将学习使用新的 Microsoft Expression Blend 工具。通过使用 Expression Blend，可以创建高级的用户界面，学习实现样式、布局与动画。读者还将发现，将界面设计保存到 XAML 文件中，然后在 Visual Studio 中使用完全相同的标记语言编写应用程序逻辑是多么的容易。

接下来，将介绍如何使用 WPF 创建特殊效果，包括位图效果、变换，以及使用画刷对象创建玻璃或反射效果等。本书演示了这些技术，并为在应用程序中创建绚丽的可视化元素提供了基础。

在学习完特殊效果之后，将介绍如何使用 WPF 创建自定义控件。WPF 提供了一个极好的自定义对象模型，允许运用各种风格和已存在的各种元素模板。该模型是一个令人满意的新模型，允许将几乎任何元素放置在另一个元素中。通过这个新功能，在整个创建自定义控件的过程中，都不会遇到问题。当然，任何事物都不是绝对的，使用 WPF 创建自定义控件，也有可能遇到一些问题。本部分内容包括在什么情况下需要考虑创建自定义控件，以及如何创建自定义控件。

接下来，将进入 WPF 应用程序的企业应用开发部分。使用 WPF 可以创建两种风格的应用程序：基于 Windows 的单机运行的应用程序与基于 Web 的应用程序。这两种类型的程序基于相同的代码，即 XAML 与 .NET。这意味着为应用程序指定目标主机环境，只需要简单改变工程文件的设置并管理这些配置即可。这项功能是非常强大的，在本书的企业开发主题中，对这一功能进行了全面介绍。

在理解了应用程序模型与配置之后，读者可能想了解安全问题。WPF 的安全基于 .NET 2.0 的 CAS 安全模型，与所选择的应用程序模型以及运行环境也有关。如果 WPF 程序运行于浏览器中，将涉及到 Internet 区域安全设置。本书将深入介绍这些内容。

在探讨了 WPF 应用程序开发、配置基础，并全面分析了 WPF 应用程序的安全之后，本书将进入一些高级主题。这些高级主题之一就是如何混合使用 Win32 与 WPF 代码。本书中将涉及互操作的相关问题。在 Win32 程序中使用 WPF 以及在 WPF 程序中使用 Win32 都是可行的，本书将介绍如何实现这一特征，从而可以使读者快速掌握如何将 WPF 用于 Win32 程序中。

本书最后深入研究了体系结构、WPF 架构、XAML 以及多线程问题。此外，还介绍了 Windows Workflow Foundation(WF)与 Windows Communication Foundation(WCF)，从而使读者可以熟悉 .NET Framework 3.0 的其他组件。在理解了这些重要组件之后，读者还将学习如何构建一个简单的 WCF 服务与一个简单的 WF 工作流应用程序。

本书组织结构

本书先介绍了相关背景，包括 Microsoft 为什么以及怎样从长期以来使用的 Win32 平台转向 WPF。然后，对 WPF 及其核心组件和子系统进行了概述，接下来介绍了 Expression Blend 工具的使用，并通过示例演示了如何创建富用户界面。后续几章具体针对与 WPF 开发相关的各个主题。在本书的最后简要介绍了与 WPF 平行的其他两个组件，这些组件共同构成了 .NET Framework 3.0 这一重要而又强大的平台。

本书各章内容如下所示：

- 第 1 章，“WPF 概述”——本章探讨了近年来 Windows API 的演化，以及这种演化如何为 WPF 这一新平台铺平了道路。然后，从整体上介绍了 WPF 的体系结构，包括新的图形功能以及各个子系统，还介绍了相关的开发工具。
- 第 2 章，“WPF 与 .NET 编程”——本章首先介绍了构建与运行 WPF 应用程序的需求，然后是应用程序的各子系统，例如附属属性与事件传递系统。此外，还介绍了 WPF 提供的各种控件，以及如何使用平台提供的新结构修改和扩展这些控件。

- 第3章，“WPF 应用程序深入剖析”——本章介绍了在 WPF 中可以使用的各种应用程序模型，例如可以独立运行的与基于浏览器的应用程序。对相关程序文件及配置设置都进行了剖析，从而使读者可以快速理解并能开发 WPF 应用程序。
- 第4章，“使用 Microsoft Expression Blend 构建富用户界面：第一部分”——本章介绍了 WPF 应用程序设计人员可以使用的新工具，重点介绍了 Microsoft Expression Blend。在本章中读者可以了解到 Expression Blend 提供的设计环境，以及工程文件结构。本章内容还包括创建矢量图形、布局、动画和控件。
- 第5章，“使用 Microsoft Expression Blend 构建富用户界面：第二部分”——本章在第4章介绍的相关内容的基础上，介绍了如何为程序添加交互功能以处理用户输入、事件与命中测试。本章还给出了一系列的动画示例程序，以及与 WPF 绘图相关的 API。本章还演示了 Expression Blend 与 Visual Studio 2005 提供的协作特性。
- 第6章，“特殊效果”——本章探讨了 WPF 提供的丰富功能与特殊效果，包括画刷、位图效果以及变换等。有了这些功能，使用 WPF 可以很容易地快速创建富用户界面。
- 第7章，“自定义控件”——本章专门讲解了在 WPF 中自定义控件的开发。通过模板与类型的扩展支持，在 WPF 中自定义控件所需做的工作是很少的。但是在很多情况下需要创建自定义控件。本章不仅介绍如何创建自定义控件，还介绍了如何确定是否使用模板与类型等内置功能的支持。这些内置功能是非常强大的。
- 第8章，“WPF 企业开发”——本章探讨了使用 WPF 开发企业级应用程序时需要考虑的许多问题。应用程序模型、配置、运行环境、安全考虑、导航、状态管理以及本地化等内容，在本章中都进行了探讨。
- 第9章，“安全”——本章探讨了 WPF 的安全特性。WPF 应用程序的安全主要与所选择的程序类型和运行环境有关。本章内容包括操作系统安全、CAS 安全、Internet 安全沙箱以及 ClickOnce 安全等。
- 第10章，“WPF 与 Win32 交互”——本章主要针对 WPF 应用程序与已存在的 Win32 应用程序之间的交互问题。本章将解答如何在 Win32 应用程序中使用 WPF，以及如何在 WPF 应用程序中使用 Win32 等问题。通过学习本章，读者将了解到在 WPF 与 Win32 应用程序中 HWND 的区别。
- 第11章，“高级开发主题”——本章将深入研究本书中涉及的有关问题。首先深入分析 WPF 体系结构，介绍 WPF 架构中重要的类，例如 Application 类、DependencyObject 类。然后深入探讨 XAML 的语法与结构，以及 WPF 应用程序中的多线程问题。最后，深入分析了 .NET Framework 3.0 的其他两个组件：Windows Workflow Foundation 与 Windows Communication Foundation。

使用本书的条件

为了开发 WPF 应用程序，以及创建本书中的示例程序，必须首先安装以下软件：

- Windows SDK

- .NET Framework 3.0 运行时组件
- 带有 Orcas CTP WinFX 开发工具的 Visual Studio 2005 或 Orcas release of Visual Studio 2005

注意:

在本书第 2 章中将给出上述所需软件的更多细节,以及介绍如何获取这些必需的组件。

源代码

在练习本书中的示例时,读者可以手动输入所有的代码,也可以使用随书附带的源代码。本书中的所有源代码均可在 <http://www.wrox.com> 或 <http://www.tupwk.com.cn/downpage> 站点下载。登录 Web 站点 <http://www.wrox.com> 后,只需找到本书的标题(可以使用 Search 功能或标题列表)并单击显示本书详细内容的页面上的 Download Code 链接即可获取源代码。

由于很多书籍的名称类似,因此通过 ISBN 查找可能会更容易一些。本书的 ISBN 号是 978-0-470-04180-2。

在下载源代码之后,只需使用最喜欢的解压缩工具对其进行解压缩即可。另一个途径是到 Wrox 代码下载主页面 <http://www.wrox.com/dynamic/books/download.aspx>,这里有本书及其他所有 Wrox 书籍的源代码。

勘误表

我们尽最大努力确保本书在叙述和代码中没有错误。然而,没有人是完美的,错误时有发生。如果读者在本书中发现什么错误,例如拼写错误或编码错误,我们将会非常感谢您能反馈给我们。通过将错误添加到勘误表中,您也许能为其他读者节约数小时的时间,也可以帮助我们提供更高质量的书籍。

请给 wkservice@tup.tsinghua.edu.cn 发电子邮件,如果您的意见是正确的,我们将在本书的后续版本中采用。

要找到本书的勘误表,可以登录 Web 站点 <http://www.wrox.com>,然后搜索本书的书名或者使用标题列表。然后,在本书的详细内容页面上,单击 Book Errata 链接。在这个页面上读者可以查看到所有已提交的、由 Wrox 的编辑发布的错误信息。也可以在 <http://www.wrox.com/misc-pages/booklist.shtml> 页面找到一个完整的标题列表,这个列表包含了每本书的勘误表链接。

<http://p2p.wrox.com>

如果想参与讨论,可以加入 P2P 论坛,网址是 <http://p2p.wrox.com>。这些论坛是基于 Web 站点的系统,其作用是让读者发布与 Wrox 的书籍和相关技术有关的消息,并与其他

读者和技术用户联络。这些论坛提供订阅功能，当读者感兴趣的主题发布时，论坛会通过电子邮件把这些消息发送给读者。Wrox 的作家、编辑、其他行业专家及和您一样的读者都会出现在这些论坛上。

在 <http://p2p.wrox.com> 中，读者将找到很多不同的论坛，这些论坛不仅能帮助读者阅读本书，还可以帮助读者开发自己的应用程序。要加入这些论坛，可按如下步骤操作：

- (1) 登录 <http://p2p.wrox.com> 并单击 Register 链接。
- (2) 阅读用途条款并单击 Agree。
- (3) 填写加入论坛所必需的信息和可选信息并单击 Submit。
- (4) 读者将收到一封电子邮件，该邮件告诉读者怎样验证账户并成功加入。

没有加入 P2P 论坛也可以阅读该论坛上的信息，但是如果希望发布自己的消息，则必须加入该论坛。

加入之后，就可以发布新消息或者回复其他用户发布的消息了。可以在任何时间阅读 Web 站点上的消息。如果希望某个论坛能将最新的消息通过电子邮件发送给自己，则可以单击论坛列表中该论坛名称旁边的 Subscribe 图标。

要获得如何使用 P2P 论坛的更多信息，可以阅读 P2P FAQ 列表中的问题及其答复，这些问题与论坛软件的工作原理及很多与 P2P 和 Wrox 相关的常见问题有关。要阅读 FAQ，可以单击任意 P2P 页面上的 FAQ 链接。

目 录

第 1 章 WPF 概述..... 1	第 3 章 WPF 应用程序深入剖析..... 59
1.1 Windows API 的简要历史..... 1	3.1 项目组成..... 59
1.1.1 平台演化..... 2	3.1.1 App.Manifest..... 61
1.1.2 .NET Framework 3.0 介绍..... 2	3.1.2 AssemblyInfo.cs..... 61
1.2 初识 Windows Presentation Foundation..... 3	3.1.3 Resources.resx..... 63
1.2.1 设计指导原则..... 3	3.1.4 Settings.settings..... 63
1.2.2 WPF 体系结构..... 5	3.1.5 MyApp.xaml..... 65
1.3 XAML..... 13	3.2 .NET Framework 3.0 Windows 应用程序..... 66
1.4 可视化设计工具..... 16	3.3 XAML 浏览器应用程序..... 68
1.4.1 XamlPad..... 16	3.4 WCF 服务库..... 70
1.4.2 Microsoft Expression Blend..... 17	3.5 WPF 自定义控件库..... 73
1.4.3 WPF 可视化设计器..... 18	3.6 小结..... 77
1.4.4 Electric Rain ZAM 3D..... 19	第 4 章 使用 Microsoft Expression Blend 构建富用户界面: 第一部分..... 79
1.4.5 Mobiform Aurora..... 19	4.1 概述..... 79
1.5 小结..... 20	4.2 设计环境..... 80
第 2 章 WPF 与 .NET 编程..... 23	4.2.1 工作空间面板..... 81
2.1 开始..... 23	4.2.2 艺术板..... 82
2.1.1 需要安装的软件..... 24	4.2.3 配置环境..... 83
2.1.2 WPF 应用程序类型..... 24	4.3 项目结构..... 84
2.1.3 第一个 WPF 应用程序..... 25	4.3.1 添加一个新项目文件..... 84
2.2 WPF 开发概念..... 27	4.3.2 构建与部署一个项目..... 85
2.2.1 XAML..... 27	4.4 创建向量对象..... 85
2.2.2 Application 对象..... 28	4.4.1 形状..... 86
2.2.3 创建用户界面..... 30	4.4.2 路径..... 87
2.2.4 事件处理..... 32	4.5 使用属性面板操作对象..... 88
2.2.5 使用控件..... 38	4.5.1 变换面板..... 88
2.2.6 触发器..... 47	4.5.2 画刷面板..... 88
2.2.7 语言支持..... 55	4.5.3 透明性、透明度与可见度..... 90
2.2.8 部署..... 57	4.6 操作文本..... 91
2.3 小结..... 57	4.7 管理布局..... 92

4.7.1	用户界面布局面板	93	6.1.2	GradientBrush	150
4.7.2	其他布局控件	95	6.1.3	ImageBrush	153
4.7.3	嵌套布局面板	96	6.1.4	DrawingBrush	155
4.8	动画	96	6.1.5	VisualBrush	156
4.8.1	时间线子面板	96	6.2	位图效果	158
4.8.2	关键帧	98	6.3	变换	163
4.8.3	为对象的属性添加动画	98	6.3.1	TranslateTransform 类	165
4.8.4	运动路径	98	6.3.2	ScaleTransform 类	168
4.8.5	触发器	99	6.3.3	SkewTransform 类	171
4.8.6	创建控件	100	6.3.4	RotateTransform 类	173
4.8.7	模板	102	6.3.5	透明掩码	176
4.8.8	样式	103	6.4	综合运用——组合效果	178
4.8.9	状态	103	6.4.1	具有反射效果的跳跃 小球示例程序	178
4.8.10	导入音频/视频媒体	104	6.4.2	动画图像观察器示例程序	184
4.9	小结	104	6.5	小结	190
第 5 章	使用 Microsoft Expression Blend 构建富用户界面:		第 7 章	自定义控件	191
	第二部分	105	7.1	概述	191
5.1	Expression Blend workflow	105	7.2	控件基类	192
5.2	处理用户输入	108	7.2.1	UserControl 类	193
5.2.1	键盘和鼠标类	108	7.2.2	创建用户控件	193
5.2.2	事件与事件处理	108	7.3	WPF 中的数据绑定	202
5.2.3	定位	110	7.3.1	绑定标记扩展	202
5.2.4	命中测试	114	7.3.2	绑定模式	202
5.3	WPF 动画 API	124	7.3.3	数据模板	204
5.3.1	动画类	124	7.3.4	数据转换	205
5.3.2	使用代码通过编程创建 动态动画	124	7.4	创建和编辑样式	207
5.3.3	程序动画	127	7.4.1	为样式指定目标类型	208
5.4	使用故事板进行交互	132	7.4.2	继承和覆盖样式	209
5.5	WPF 绘图 API	134	7.4.3	样式触发器	210
5.5.1	Geometry 类	134	7.5	使用模板定制已有的控件	212
5.5.2	Shape 类	134	7.6	小结	215
5.5.3	画刷	138	第 8 章	WPF 企业开发	217
5.6	小结	141	8.1	WPF 应用程序模型	218
第 6 章	特殊效果	143	8.1.1	单机应用程序	219
6.1	画刷	144	8.1.2	基于浏览器的应用程序	221
6.1.1	SolidColorBrush	144	8.1.3	安全考虑	230
			8.2	状态管理	231

11.2.4	Freezable 类	339
11.2.5	Visual 类	339
11.2.6	UIElement 类	345
11.2.7	FrameworkElement 类	345
11.2.8	Control 类	346
11.3	XAML 深入分析	347
11.3.1	XAML 的后台工作	348
11.3.2	动态操作 XAML	352
11.3.3	解析 Window1.xaml	356
11.3.4	XAML 标记扩展	360
11.3.5	XAML 与自定义类型	360
11.4	WPF 多线程	362
11.4.1	单线程应用程序模型	363
11.4.2	线程关联度与 DispatcherObject	363
11.4.3	WPF 分发器	364
11.4.4	使用单线程应用程序	365
11.4.5	异步线程	367
11.5	Windows 通信基础 WCF	372
11.5.1	服务定向	372
11.5.2	WCF 体系结构	373
11.5.3	WCF 基础	374
11.5.4	构建一个 WCF 服务	376
11.6	Windows 工作流基础	383
11.6.1	工作流定义	384
11.6.2	WF 体系结构	384
11.6.3	WF 基础	385
11.7	小结	399

第 1 章

WPF 概述

对于曾经开发过 Windows 应用程序的程序员而言，本书展示了开发 Windows 应用程序的一些精彩的新变化。此外，对曾经开发过 Web 应用程序的程序员来说，本书提供的开发方式更令人激动。

Windows Presentation Foundation，也称做 WPF，是下一代图形平台。基于该平台可以构建运行于 Windows Vista 之上的 Windows 应用程序和 Web 应用程序，Windows Vista 是 Microsoft 最新发布的新一代操作系统。WPF 为建立优雅的并且非常灵活的用户体验提供了基础，这种用户体验是通过同时使用用户界面、应用程序逻辑、文档以及多媒体实现的。

注意：

尽管 WPF 最初的目标是用于 Windows Vista 平台，但是作为 .NET Framework 3.0(即以前的 WinFX)的一部分，WPF 也可以在 Windows XP 和 Windows Server 2003 平台上使用。

通过使用 XAML 与 Microsoft Expression Designers 等工具，应用程序的界面和功能可以得到进一步加强，这些工具使设计人员和开发人员可以同时开发用户界面与应用程序逻辑，从而共同提供完美的用户体验。

本章对 WPF 进行概述，包括以下内容：

- Windows API 的演变。
- .NET Framework 3.0，基于 Windows 开发的下一代 API。
- WPF 体系结构与开发模型。
- XAML，支持 WPF 开发的声明式语言。
- 开发 WPF 应用程序的相关工具。

1.1 Windows API 的简要历史

Windows API 为应用程序开发提供了 Windows 操作系统支持的核心功能，主要用于 C/C++ 语言的开发，Windows API 是应用程序和 Windows 之间最直接的交互机制。

Windows API 包含以下功能组合：

- **基本服务**——提供对计算机核心资源的访问，例如内存、文件系统、设备、处理器以及线程等。
- **通用控件库**——Windows 操作系统中广泛使用的通用控件集合，通过这些控件为应用程序提供统一的 Windows 视觉风格。
- **通用对话框库**——用于完成通用任务的对话框集合，包括打开文件对话框、保存文件对话框和打印对话框等。
- **图形设备接口(GDI)**——为应用程序提供在显示器、打印机及其他设备上图形输出的功能。
- **网络服务**——提供的 Windows 操作系统中的各种联网功能，包括 RPC(远程过程调用)与 NetBIOS(网络基本输入输出系统)等。
- **用户界面(UI)**——提供一种管理机制，用于管理应用程序中的窗口与控件，以及来自鼠标和键盘等设备的输入信息。
- **Windows 外壳**——用于组织和显示所有 Windows 用户界面的容器，包括桌面、任务栏和 Windows 浏览器等。

通过上面这些服务，开发人员可以拥有极大的灵活性，用以创建运行在 Windows 操作系统之上的功能强大的应用程序。但是，这种灵活性是以低级的和单调乏味的底层操作为代价的。

在每一次新版本的 Windows 操作系统发布时，总是会修改和增加一些 Windows API。而每次发布新版本操作系统时，Microsoft 总是力争保持向后兼容，因此许多最初的 Windows API 函数在今天的 Windows XP 和 Windows Server 2003 中仍然存在。

以下是 Windows API 的主要版本：

- **Win16**——第一个 16 位版本 Windows 操作系统提供的 API。
- **Win32**——包含在 Windows NT 中，用于 32 位版本 Windows 的 API。
- **Win32 for 64-bit Windows**——以前称之为 Win64，用于 64 位版本的 Windows XP 和 Windows Server 2003 的 API。

1.1.1 平台演化

自从 20 多年前发布 Windows 1.0 版本以来，Windows API 的图形设备接口(GDI)和用户界面(UI)服务为 Windows 应用程序提供了可靠的图形平台。我们日常使用的许多应用程序，包括 Microsoft Office 系列与 Internet 浏览器等，都是构建在这一基础之上。尽管这一图形平台有许多成功之处，但是在这 20 多年的过程中，它在技术上已经发生了根本变化。随着个人电脑的激增，以及因特网带宽的增加，对丰富视觉界面体验的需求也在快速增加。此外，图形硬件技术的进展，为数字媒体和交互娱乐产品铺平了道路。面对新的丰富视觉体验的需求，Microsoft 斥巨资开发了下一代 Windows API——.NET Framework 3.0 来完成这一体验。

1.1.2 .NET Framework 3.0 介绍

.NET Framework 3.0 是开发运行于 Windows 操作系统之上的应用程序的一个革命性的

里程碑。.NET Framework 3.0 是一套托管的 API 集合，构建于 .NET Framework 2.0 之上，它为消息传递、工作流和呈现提供了强大的功能。

.NET Framework 3.0 的核心组件包括：

- Windows Presentation Foundation(WPF)——与用户界面所有内容相关的图形子系统。
- Windows Communication Foundation(WCF)——.NET Framework 3.0 的信息传递子系统，保证应用程序通过一套统一的 API 进行通信。
- Windows Workflow Foundation(WF)——为运行于 Windows 之上的应用程序提供工作流服务。

作为针对 Windows Vista 的新编程模型，.NET Framework 3.0 将 Windows 平台和使用 Visual Studio 2005 开发的应用程序融合在一起。通过直接访问低级的操作系统服务与硬件功能，.NET Framework 3.0 为创建具有丰富用户体验的应用程序提供了完美的解决方案。

注意：

尽管 WCF 与 WF 是 .NET Framework 3.0 中与 WPF 同等重要的组件，但是它们都超出了本书的范围。有关 WCF 与 WF 的更多信息，请参考 .NET Framework 开发中心的 MSDN (<http://msdn2.microsoft.com/en-us/netframework>)。

1.2 初识 Windows Presentation Foundation

Windows Presentation Foundation(WPF)以前被称为 Avalon，是 Windows Vista 的图形子系统，提供了整合图形用户界面的完整方法，整合对象还包括二维与三维图形、文档以及数字媒体等。WPF 构建在 .NET Framework 之上，为 Windows 应用程序开发提供了一个托管的环境。它充分利用了 Microsoft 以前推出的 .NET Framework 的优点，并使熟悉 .NET 技术的开发人员能够很容易地转到使用 WPF 开发应用程序。

1.2.1 设计指导原则

为增强用户体验，提高开发人员和设计人员的能力。Microsoft 在构建 WPF 时遵循了以下设计原则：

- 集成
- 向量图形
- 声明式编程
- 简化的部署
- 可移植文档

1. 集成

目前，开发 Windows 应用程序可能需要使用多种不同的技术，从用于开发二维图形的 GDI/GDI+、用户界面服务(User32 或 WinForms)，到用于开发三维图形的 Direct3D 或

OpenGL 等。另一方面, WPF 被设计成开发应用程序的单一模型, 可以通过类似的程序代码来实现动画、数据窗体以及三维模型的开发。

2. 向量图形

WPF 实现了一个向量图形引擎, 以充分利用新图形硬件的强大功能。向量图形引擎可以在不损失图形显示质量的情况下, 根据特定的显示器分辨率对图形进行缩放, 而这对于原来的固定大小的光栅化图形来说几乎是不可能的。WPF 利用 Direct3D 对向量图形进行绘制, 并利用支持 DirectX7 或更高版本显卡的图形处理单元(GPU)进行图形渲染。考虑到未来新技术的发展, 例如高分辨率显示以及可能出现的其他未知的窗体显示因素, WPF 实现了一个基于浮点数的逻辑像素系统, 并支持 32 位的 ARGB 颜色。

3. 声明式编程

WPF 引入了一种新的基于 XML 的语言来表示用户界面与用户交互, 称为 XAML(可扩展应用程序标记语言, eXtensible Application Markup Language, 发音为“zammel”)。与 Macromedia 的 MXML 语言类似, XAML 使用 XML 标记来表示用户界面元素, 所以 XAML 使应用程序可以在编译或运行时动态地解析和操纵用户界面元素, 从而提供了一个灵活的用户界面组成模型。

借鉴 ASP.NET 的成功经验, XAML 也采取了后台编码模型, 允许设计人员和开发人员平行工作, 并无缝组合他们的工作成果以创建卓越的用户体验。在设计时工具(例如 Visual Studio 2005 中的 Windows Presentation Foundation 可视化设计器插件)的辅助下, 开发基于 XAML 的应用程序的过程和 WinForm 的开发非常类似。此外, 习惯于使用 Macromedia 的 Flash 8 这一类可视化工具的设计人员, 可以很方便地使用这些可视化设计工具(例如 Microsoft Expression Blend)来构建基于 XAML 的解决方案。这些工具在本章的后面以及整本书中都将涉及到。

4. 简化部署

WPF 应用程序可以部署为可独立运行的应用程序或运行于 Internet 浏览器的 Web 应用程序。作为智能客户端应用程序, 基于 Web 的 WPF 应用程序只在信任沙箱(sandbox)内工作, 以保护客户端计算机免遭恶意应用程序的攻击。

此外, 运行于 Internet 浏览器的 WPF 应用程序能够利用本地客户计算机的硬件功能, 从而通过三维图形、数字媒体以及其他形式等, 提供一个丰富的 Web 体验, 这是今天基于 Web 的应用程序开发中的一个热点问题。

5. 可移植文档

WPF 引入了一组新的文档和打印技术。WPF 使用开放打包约定(Open Packaging Conventions)——一种随 Microsoft Office 12 发布的约定, 该约定支持压缩、自定义元数据、数字签名以及权限管理等功能。WPF 还包括了 XML 文件规范(XML Paper Specification, XPS)。与可移植文档格式(Portable Document Format, PDF)类似, XPS 允许文档在多台计算机间共享, 而且不需要安装相应的应用程序。