

THE EXPERT'S VOICE® IN .NET

Pro WPF in C# 2010

Windows Presentation Foundation in .NET 4

WPF编程宝典

——C# 2010版

(美) Matthew MacDonald 著
王德才 译

Apress®



清华大学出版社

WPF 编程宝典

—— C# 2010 版

(美) Matthew MacDonald 著

王德才

译

清华大学出版社

北 京

Matthew MacDonald

Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4

EISBN: 978-1-4302-7205-2

Original English language edition published by Apress, Inc. Copyright © 2010 by Apress, Inc. Simplified Chinese-language edition copyright © 2011 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 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。

北京市版权局著作权合同登记号 图字: 01-2010-6329

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,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

WPF 编程宝典——C# 2010 版/(美) 麦克唐纳(MacDonald, M.) 著;王德才 译. —北京:清华大学出版社, 2011.6
书名原文: Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4

ISBN 978-7-302-25269-6

I. W… II. ①麦… ②王… III. ①Windows 操作系统—人机界面—程序设计 ②C 语言—程序设计
IV. ①TP316.7 ②TP312

中国版本图书馆 CIP 数据核字(2011)第 057051 号

责任编辑:王 军 韩宏志

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 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185×260 印 张:59 字 数:1587 千字

版 次:2011 年 6 月第 1 版 印 次:2011 年 6 月第 1 次印刷

印 数:1~4000

定 价:118.00 元

产品编号:036205-01

作者简介

Matthew MacDonald 是一位作家、教育家，也是微软最有价值专家。他撰写了十多本有关.NET 编程的书籍，包括 *Pro Silverlight 3 in C#*(2009 年 Apress 出版)和 *Pro ASP.NET 3.5 in C#*(2007 年 Apress 出版)以及本书的前一版本。目前他与妻子以及两个女儿居住在多伦多。

关于技术编辑

Fabio Claudio Ferracchiati 是一位前沿技术领域的多产作家。他已经撰写了十几本有关.NET、C#、Visual Basic 以及 ASP.NET 的编程书籍。他是.NET 领域微软认证的解决方案开发专家(MCSD)，现居意大利罗马。可以访问他的博客，网址为 www.ferracchiati.com。

致 谢

如果没有他人的帮助，没有哪位作者能够单独完成一书籍。在撰写本书时，我极大地受惠于整个 Apress 团队，包括 Anne Collett，他一直帮助我完成本书，还有 Kim Wimpsett 和 Marilyn Smith，他们负责复制编辑工作，还有多位幕后工作者，他们负责编写索引页面、绘图以及最后的校对工作。我还要特别感谢 Gary Cornell，他一直为我提供有关项目和出版方面的宝贵建议。

Fabio Claudio Ferracchiati 和 Christophe Nasarre 为本书提供了颇有见地且合乎时宜的技术分析评论，在此向他们表达最诚挚的感谢。本书也吸收了来自各个 WPF 团队的众多博客的部分精髓，他们一直热衷于挖掘 WPF 最深层次的技术。我鼓励所有希望学习 WPF 未来版本的读者向他们学习。最后衷心感谢爱妻以及 Nora、Razia、Paul 和 Hamid。感谢所有为本书做出贡献的人士！

前 言

.NET 问世之初便引入了一些重要的新技术,包括编写 Web 应用程序的全新方法(ASP.NET)、连接数据库的全新方法(ADO.NET)、新的类型安全的语言(C#和 VB.NET)以及托管的运行时(CLR)。在这些新技术中,其中的一项重要技术是 Windows 窗体,它是用于构建 Windows 应用程序的类库。

尽管 Windows 窗体是一个成熟且功能完整的工具包,但它使用的是在过去十年中基本没有变化的 Windows 技术。最重要的是,Windows 窗体依靠 Windows API 创建标准用户界面元素的可视化外观,如按钮、文本框和复选框等。所以,这些要素在本质上是不可定制的。

例如,如果希望创建一个外观时髦的光晕按钮,需要创建一个自定义控件,并使用低级的绘图模型为按钮(各种不同的状态)绘制各个方面的细节。更糟糕的是,普通窗口被切割成不同的区域,每个控件完全拥有自己的区域。所以,没有比较好的绘制方法可将一个控件的内容(如按钮背后的辉光效果)延伸到其他控件所占的区域中。更不要指望实现动画效果,如旋转文本、闪烁按钮、收缩窗口以及实时预览等,因为对于这些效果必须手工绘制每个细节。

Windows 演示基础(Windows Presentation Foundation, WPF)通过引入一个使用完全不同技术的新模型改变了所有这一切。尽管 WPF 也提供了熟悉的标准控件,但是它“自行”绘制每个文本、边框和背景填充。所以, WPF 能提供更强大的功能,可以改变渲染屏幕上所有内容的方式。使用这些特性,可以重新设置常见控件的样式(如按钮),并且通常不需要编写任何代码。同样,可以使用变换对象旋转、拉伸、缩放以及扭曲用户界面中的所有内容,甚至可以使用 WPF 动画系统对用户界面中的内容进行变换。并且因为 WPF 引擎将在窗口上渲染的内容作为单独操作的一部分,所以它能够处理任意多层相互重叠的控件,即使这些控件具有不规则的形状且是半透明的也同样如此。

在 WPF 这些新特性的背后是基于 DirectX 的功能强大的基础结构,DirectX 是一套硬件加速的图形 API,通常用于开发最前沿的计算机游戏。这意味着可以使用丰富的图形效果,而不会损失性能,而使用 Windows 窗体实现这类效果会严重影响程序运行的性能。实际上,甚至可以使用更高级特性,例如,对视频文件和 3D 内容的支持。使用这些特性(以及优秀的设计工具),可以创建出赏心悦目的用户界面和可视化效果,而使用 Windows 窗体技术是无法实现这些效果的。

尽管通常最关注 WPF 中最前沿的视频、动画以及 3D 特性,但有必要指出,可以使用 WPF 的标准控件和简单的可视化外观来构建常规 Windows 应用程序。实际上,使用 WPF 中的通用控件与使用 Windows 窗体中的通用控件同样容易。更值得一提的是, WPF 增强了商业开发人员所需要的特性,包括大幅改进的数据绑定模型、一套用于打印以及管理打印队列的新类,以及用于显示大量格式化文本的文档特性。甚至提供了用于构建基于页面的应用程序的模型,这种应用程序可以在 Internet Explorer 中无缝地运行,并且能够从 Web 站点加载,所有这些操作都不会出现常见的安全警告和令人讨厌的安装提示。

总之, WPF 将以前 Windows 开发领域中的精华和当今的创新技术融为一体, 来构建现代的、富图形用户界面。尽管 Windows 窗体应用程序还将继续维持数年, 但是从事新的 Windows 开发项目的开发人员应当首选 WPF。

关于本书

本书针对了解 .NET 平台、C# 语言以及 Visual Studio 开发环境的专业开发人员, 深入分析了 WPF。在学习本书前, 不需要具备使用以前版本 WPF 的经验, 而使用过 WPF 的开发人员可以通过阅读每章开头“新增功能”中突出显示的 WPF 新特性。

本书全面描述了所有 WPF 的主要特性, 从 XAML(用于定义 WPF 用户界面的标记语言)到 3D 绘图和动画。本书很少编写涉及 .NET Framework 其他特性的代码, 如用于查询数据库的 ADO.NET 类。本书中不讨论这些内容。但如果需要更多有关 .NET 而非特定于 WPF 的特性, 请参考 Apress 出版的许多专门介绍 .NET 的书籍。

内容概述

本书一共包括 33 章。如果刚开始学习 WPF, 将会发现按照章节顺序阅读本书是最容易的方法, 因为后续章节常用到前面章节中演示的技术。

下面是本书每一章的主要内容:

第 1 章: WPF 概述 介绍 WPF 的体系结构, WPF 的 DirectX 基础设施, 以及新的能够自动改变用户界面尺寸的设备无关度量系统。

第 2 章: XAML 介绍用于定义用户界面的 XAML 标准。该章将讨论为什么创建 XAML 以及它的工作原理, 并且将使用不同的编码方法创建一个基本的 WPF 窗口。

第 3 章: 布局 深入研究在 WPF 窗口中用于组织元素的布局面板。该章将分析不同的布局策略, 并且将构建一些普通类型的窗口。

第 4 章: 依赖项属性 介绍 WPF 如何使用依赖项属性来支持重要特性, 如数据绑定和动画。

第 5 章: 路由事件 介绍 WPF 如何使用事件路由在用户界面元素中发送冒泡路由事件或隧道路由事件。还介绍所有 WPF 元素都支持的一组基本鼠标、键盘以及多点触控事件。

第 6 章: 控件 分析所有 Windows 开发人员都非常熟悉的控件, 如按钮、文本框和标签, 还讨论它们在 WPF 中的区别。

第 7 章: Application 类 介绍 WPF 应用程序模型。在该章将会看到如何创建单实例和基于文档的 WPF 应用程序。

第 8 章: 元素绑定 介绍 WPF 数据绑定。在该章将会看到如何将任意类型的对象绑定到用户界面。

第 9 章: 命令 介绍 WPF 命令模型, 使用 WPF 命令模型可将多个控件连接到同一个逻辑操作。

第 10 章: 资源 介绍如何使用资源在程序集中嵌入二进制文件, 以及如何在整个用户界面中重用重要的对象。

第 11 章：样式和行为 解释 WPF 样式系统，使用 WPF 样式可以为一整组控件应用一套通用属性值。

第 12 章：形状、画刷和变换 介绍 WPF 中的 2D 绘图模型。在该章将学习如何创建形状、使用变换改变元素，以及使用渐变画刷、图像画刷和平铺图像画刷绘制特殊效果。

第 13 章：几何图形和图画 深入分析 2D 绘图。在该章将学习如何创建包含弧线和曲线的复杂路径，以及如何高效地使用复杂图形。

第 14 章：效果和可视化对象 介绍低级的图形编程。在该章将使用像素着色器应用 Photo 风格的效果，手动构建位图，并且为了优化绘图性能使用 WPF 的可视化层。

第 15 章：动画基础 研究 WPF 的动画框架，通过 WPF 动画框架可以使用简单的、声明式的标记将动态效果集成到应用程序中。

第 16 章：高级动画 研究了更高级的动画技术，如关键帧动画、基于路径的动画以及基于帧的动画。还将分析一个详明的示例，展示如何使用代码创建和管理动态的动画。

第 17 章：控件模板 介绍如何通过插入一个定制的模板为任意 WPF 控件提供动态的新外观(以及新行为)。还将看到如何使用模板构建能够换肤的应用程序。

第 18 章：自定义元素 研究如何扩展现有的 WPF 控件，以及如何创建自己的控件。在该章中将看到几个示例，包括基于模板的颜色拾取器、可翻转的面板、自定义的布局容器，以及执行自定义绘图的装饰元素。

第 19 章：数据绑定 展示如何从数据库获取信息，将获取的信息插入到自定义的数据对象中，并将这些数据对象绑定到 WPF 控件。还将学习如何使用虚拟化技术提高大型数据绑定列表的性能，以及如何使用验证方法捕获编辑错误。

第 20 章：格式化绑定的数据 展示将原始数据转换为包含图片、控件以及选择效果的富数据显示的一些技巧。

第 21 章：数据视图 分析如何在数据绑定窗口中使用视图在数据项中导航，以及应用过滤、分类和分组。

第 22 章：列表、网格和树 带您浏览 WPF 中的富数据控件，包括 ListView、TreeView 和 DataGrid。

第 23 章：窗口 分析 WPF 中窗口的工作原理。在该章还将学习如何创建不规则形状的窗口，以及如何使用 Vista 玻璃效果。还将通过定制任务栏跳转列表、缩略图以及图标重叠实现大部分 Windows 7 特性。

第 24 章：页面和导航 介绍如何使用 WPF 构建页面，以及保持跟踪导航历史。该章还将介绍如何构建驻留于浏览器中的 WPF 应用程序，这种应用程序可以从 Web 站点加载。

第 25 章：菜单、工具栏和功能区 分析面向命令的控件，如菜单和工具栏。在该章还将使用可免费下载的 Ribbon 控件尝试更富有现代气息的用户界面。

第 26 章：声音和视频 介绍 WPF 媒体支持。在该章将会看到如何控制声音和视频的播放，以及如何合成动画和生动鲜活的效果。

第 27 章：3D 绘图 研究 WPF 中对绘制 3D 图形的支持。在该章将学习如何创建和变换 3D 对象，以及如何为 3D 对象添加动画。甚至还会看到如何在 3D 表面上放置可交互的 2D 控件。

第 28 章：文档 介绍 WPF 富文档支持。在该章将学习如何使用流文档以尽可能容易阅读的方式展现大量文本，并将学习如何使用固定文档显示准备打印的页面。甚至还将学习如何使用 RichTextBox 控件提供文档编辑功能。

第 29 章：打印 演示 WPF 的打印模型，通过该打印模型可以在打印文档中绘制文本和图形。在该章还将学习如何管理页面设置和打印队列。

第 30 章：与 Windows 窗体的交互 分析如何在一个应用程序——甚至在一个窗口中，结合使用 WPF 和 Windows 窗体内容。

第 31 章：多线程 介绍如何创建具有良好响应性的、在后台执行耗时的 WPF 应用程序。

第 32 章：插件模型 展示如何创建可扩展的、能够动态发现和加载独立组件的应用程序。

第 33 章：ClickOnce 部署 展示如何使用 ClickOnce 安装模型部署 WPF 应用程序。

使用本书的前提条件

为了运行 WPF 4 应用程序，计算机中必须安装 Windows 7、Windows Vista 或带有 Windows XP SP2。还需要 .NET Framework 4。为了创建 WPF 4 应用程序(以及为了能够打开本书中所提供的示例项目)，需要安装 Visual Studio 2010，Visual Studio 2010 中包含了 .NET Framework 4。

还有一种选择。不使用任何版本的 Visual Studio，可以使用 Expression Blend(一种面向图形的设计工具)来构建和测试 WPF 应用程序。总之，Expression Blend 是面向图形设计人员的工具，他们使用该工具创建绚丽的内容；而对于编写大量代码的编程人员来说，Visual Studio 则是理想的工具。本书假定使用的是 Visual Studio。如果想学习更多有关 Expression Blend 的内容，可以参阅有关该主题的专门书籍(顺便提一下，为使用 WPF 4 创建应用程序，需要 Expression Blend 4，在撰写本书时它还是一个 beta 版本)。

代码示例和 URL

查看 Apress 网站或 www.prosetech.com 下载最新的示例代码是个好主意。测试在本书中介绍的大部分更复杂的示例需要用到这些代码示例，因为在本书示例中的那些不是很重要的细节通常被忽略了。本书主要关注最重要的部分，从而不需要为阐明一个概念而无谓地占用篇幅。

为了下载本书的源代码，可以访问 Web 站点 <http://www.prosetech.com> 查找本书的页面。也可从 <http://www.tupwk.com.cn/downpage> 下载本书的源代码。还将发现在本书中提及的链接的列表，从而不需要键入任何内容就可以找到重要工具和例子。

反馈

本书力争成为 WPF 编程爱好者最佳的辅导和参考资料。为了实现该目标，您的评论和建议对我们来说是非常有帮助的。您可以将本书的缺点、优点及其他反馈信息直接发送到邮箱 wkservice@vip.163.com，我们将不胜感激。

目 录

第 1 章 WPF 概述	1	2.3.7 事件	39
1.1 Windows 图形演化	1	2.3.8 完整的 Eight Ball 示例	40
1.1.1 DirectX: 新的图形引擎	2	2.4 使用其他名称空间中的类型	41
1.1.2 硬件加速与 WPF	2	2.5 加载和编译 XAML	43
1.2 WPF: 高级 API	4	2.5.1 只使用代码	43
1.2.1 Windows 窗体将继续保留	5	2.5.2 使用代码和未经编译的 XAML	45
1.2.2 DirectX 也将继续保留	5	2.5.3 使用代码和编译过的 XAML	47
1.2.3 Silverlight	5	2.5.4 只使用 XAML	48
1.3 分辨率无关性	6	2.6 XAML 2009	49
1.3.1 WPF 单位	7	2.6.1 自动事件连接	50
1.3.2 系统 DPI	8	2.6.2 引用	51
1.3.3 位图和矢量图形	10	2.6.3 内置类型	51
1.4 WPF 体系结构	11	2.6.4 高级的对象创建	52
1.5 WPF 4	15	2.7 小结	53
1.5.1 新特性	15	第 3 章 布局	55
1.5.2 WPF 工具包	16	3.1 理解 WPF 中的布局	55
1.5.3 Visual Studio 2010	16	3.1.1 WPF 布局原则	56
1.6 小结	18	3.1.2 布局过程	57
第 2 章 XAML	21	3.1.3 布局容器	57
2.1 理解 XAML	21	3.2 使用 StackPanel 面板进行简单 布局	59
2.1.1 WPF 之前的图形用户界面	22	3.2.1 布局属性	60
2.1.2 XAML 变体	23	3.2.2 对齐方式	61
2.1.3 XAML 编译	23	3.2.3 边距	62
2.2 XAML 基础	24	3.2.4 最小尺寸、最大尺寸以及 显式地设置尺寸	63
2.2.1 XAML 名称空间	25	3.2.5 Border 控件	65
2.2.2 代码隐藏类	26	3.3 WrapPanel 面板和 DockPanel 面板	66
2.3 XAML 中的属性和事件	29	3.3.1 WrapPanel 面板	66
2.3.1 简单属性与类型转换器	30	3.3.2 DockPanel 面板	67
2.3.2 复杂属性	31		
2.3.3 标记扩展	33		
2.3.4 附加属性	33		
2.3.5 嵌套元素	35		
2.3.6 特殊字符与空白	37		

3.3.3 嵌套布局容器	68	5.2.1 RoutedEventArgs 类	108
3.4 Grid 面板	70	5.2.2 冒泡路由事件	109
3.4.1 调整行和列	72	5.2.3 处理挂起的事件	111
3.4.2 布局舍入	74	5.2.4 附加事件	111
3.4.3 跨越行和列	74	5.2.5 隧道路由事件	113
3.4.4 分割窗口	75	5.3 WPF 事件	114
3.4.5 共享尺寸组	79	5.3.1 生命周期事件	115
3.4.6 UniformGrid 面板	81	5.3.2 输入事件	116
3.5 使用 Canvas 面板进行基于坐标 的布局	82	5.4 键盘输入	117
3.5.1 Z 顺序	83	5.4.1 处理按键事件	118
3.5.2 InkCanvas 元素	83	5.4.2 焦点	120
3.6 布局示例	86	5.4.3 获取键盘状态	121
3.6.1 列设置	86	5.5 鼠标输入	122
3.6.2 动态内容	87	5.5.1 鼠标单击	123
3.6.3 组合式用户界面	88	5.5.2 捕获鼠标	124
3.7 小结	90	5.5.3 鼠标拖放	125
第 4 章 依赖项属性	91	5.6 多点触控输入	127
4.1 理解依赖项属性	91	5.6.1 多点触控的输入层次	127
4.1.1 定义依赖项属性	92	5.6.2 原始触控	128
4.1.2 注册依赖项属性	92	5.6.3 操作	130
4.1.3 添加属性包装器	94	5.6.4 惯性	133
4.1.4 WPF 使用依赖项属性的 方式	95	5.7 小结	134
4.1.5 共享的依赖项属性	96	第 6 章 控件	135
4.1.6 附加的依赖项属性	96	6.1 控件类	136
4.2 属性验证	98	6.1.1 背景画刷和前景画刷	136
4.2.1 验证回调	98	6.1.2 字体	138
4.2.2 强制回调	99	6.1.3 鼠标光标	143
4.3 小结	101	6.2 内容控件	144
第 5 章 路由事件	103	6.2.1 Content 属性	145
5.1 理解路由事件	103	6.2.2 对齐内容	147
5.1.1 定义、注册和包装路由 事件	104	6.2.3 WPF 内容原则	148
5.1.2 共享路由事件	104	6.2.4 标签	148
5.1.3 引发路由事件	105	6.2.5 按钮	149
5.1.4 处理路由事件	105	6.2.6 工具提示	152
5.2 事件路由	107	6.3 特殊容器控件	158
		6.4 带标题的内容控件	161
		6.4.1 GroupBox	161
		6.4.2 TabItem	162

6.4.3	Expander	163	7.4.3	翻译过程	202
6.5	文本控件	165	7.5	小结	207
6.5.1	多行文本	166	第 8 章	元素绑定	209
6.5.2	选择文本	167	8.1	将元素绑定到一起	209
6.5.3	拼写检查	167	8.1.1	绑定表达式	210
6.5.4	PasswordBox	169	8.1.2	绑定错误	211
6.6	列表控件	170	8.1.3	绑定模式	211
6.6.1	ListBox	170	8.1.4	使用代码创建绑定	213
6.6.2	ComboBox	173	8.1.5	多绑定	214
6.7	基于范围的控件	173	8.1.6	绑定更新	217
6.7.1	Slider 控件	174	8.2	绑定到非元素对象	218
6.7.2	进度条	175	8.2.1	Source 属性	218
6.8	日期控件	176	8.2.2	RelativeSource 属性	219
6.9	小结	178	8.2.3	DataContext 属性	220
第 7 章	Application 类	179	8.3	小结	221
7.1	应用程序的生命周期	179	第 9 章	命令	223
7.1.1	创建 Application 对象	180	9.1	理解命令	223
7.1.2	派生一个自定义的 Application 类	181	9.2	WPF 命令模型	224
7.1.3	应用程序的关闭方式	182	9.2.1	ICommand 接口	225
7.1.4	应用程序事件	183	9.2.2	RoutedCommand 类	225
7.2	Application 类的任务	185	9.2.3	RoutedUICommand 类	226
7.2.1	显示初始界面	185	9.2.4	命令库	227
7.2.2	处理命令行参数	186	9.3	执行命令	228
7.2.3	访问当前 Application 对象	187	9.3.1	命令源	228
7.2.4	在窗口之间进行交互	188	9.3.2	命令绑定	229
7.2.5	单实例应用程序	190	9.3.3	使用多命令源	231
7.3	程序集资源	195	9.3.4	微调命令文本	232
7.3.1	添加资源	195	9.3.5	直接调用命令	232
7.3.2	检索资源	196	9.3.6	禁用命令	233
7.3.3	pack URI	198	9.3.7	具有内置命令的控件	235
7.3.4	内容文件	199	9.4	高级命令	237
7.4	本地化	200	9.4.1	自定义命令	237
7.4.1	构建能够本地化的用户 界面	200	9.4.2	在不同的位置使用相同的 命令	238
7.4.2	使应用程序为本地化做好 准备	201	9.4.3	使用命令参数	240
			9.4.4	跟踪和翻转命令	241
			9.5	小结	245

第 10 章 资源	247
10.1 资源基础	247
10.1.1 资源集合	247
10.1.2 资源的层次	249
10.1.3 静态资源和动态资源	250
10.1.4 非共享资源	251
10.1.5 通过代码访问资源	251
10.1.6 应用程序资源	252
10.1.7 系统资源	253
10.2 资源字典	254
10.2.1 创建资源字典	254
10.2.2 使用资源字典	254
10.2.3 在程序集之间共享资源	255
10.3 小结	258
第 11 章 样式和行为	259
11.1 样式基础	259
11.1.1 创建样式对象	262
11.1.2 设置属性	263
11.1.3 关联事件处理程序	264
11.1.4 多层样式	266
11.1.5 通过类型自动应用样式	267
11.2 触发器	268
11.2.1 简单触发器	268
11.2.2 事件触发器	270
11.3 行为	272
11.3.1 获取行为支持	272
11.3.2 理解行为模型	273
11.3.3 创建行为	273
11.3.4 使用行为	275
11.3.5 Blend 中的设计时行为支持	276
11.4 小结	277
第 12 章 形状、画刷和变换	279
12.1 理解形状	279
12.1.1 Shape 类	280

12.1.2 矩形和椭圆	282
12.1.3 改变形状的尺寸和放置形状	283
12.1.4 使用 Viewbox 控件缩放形状	284
12.1.5 直线	287
12.1.6 折线	288
12.1.7 多边形	288
12.1.8 直线线帽和直线交点	290
12.1.9 点划线	291
12.1.10 像素对齐	292
12.2 画刷	293
12.2.1 SolidColorBrush 画刷	294
12.2.2 LinearGradientBrush 画刷	294
12.2.3 RadialGradientBrush 画刷	296
12.2.4 ImageBrush 画刷	297
12.2.5 平铺的 ImageBrush 画刷	299
12.2.6 VisualBrush 画刷	301
12.2.7 BitmapCacheBrush 画刷	302
12.3 变换	303
12.3.1 变换形状	304
12.3.2 变换元素	306
12.4 透明	307
12.4.1 使用元素半透明	307
12.4.2 透明掩码	308
12.5 小结	310
第 13 章 几何图形和图画	311
13.1 路径和几何图形	311
13.1.1 直线、矩形和椭圆图形	312
13.1.2 使用 GeometryGroup 组合形状	313
13.1.3 使用 CombinedGeometry 融合几何图形	315

13.1.4	使用 PathGeometry 绘制 曲线和直线	317	15.3.3	重叠动画	368
13.1.5	微语言几何图形	321	15.3.4	同步的动画	368
13.1.6	使用几何图形进行剪裁	323	15.3.5	控制播放	369
13.2	图画	324	15.3.6	监视动画进度	373
13.2.1	显示图画	326	15.4	动画缓动	375
13.2.2	导出插图	328	15.4.1	使用缓动函数	375
13.3	小结	329	15.4.2	在动画开始时应用缓动 与在动画结束时应用 缓动	376
第 14 章	效果和可视化对象	331	15.4.3	缓动函数类	377
14.1	可视化对象	331	15.4.4	创建自定义缓动函数	379
14.1.1	绘制可视化对象	332	15.5	动画性能	381
14.1.2	在元素中包装可视化 对象	334	15.5.1	期望的帧速率	382
14.1.3	命中测试	336	15.5.2	位图缓存	383
14.1.4	复杂的命中测试	338	15.6	小结	385
14.2	效果	341	第 16 章	高级动画	387
14.2.1	BlurEffect 类	342	16.1	动画类型回顾	387
14.2.2	DropShadowEffect 类	343	16.1.1	动态变换	388
14.2.3	ShaderEffect 类	344	16.1.2	动态改变画刷	391
14.3	WriteableBitmap 类	345	16.1.3	动态改变像素着色器	394
14.3.1	生成位图	346	16.2	关键帧动画	395
14.3.2	写入 WriteableBitmap 对象	347	16.2.1	离散的关键帧动画	397
14.3.3	更高效的像素写入	348	16.2.2	缓动关键帧	397
14.4	小结	350	16.2.3	样条关键帧动画	398
第 15 章	动画基础	351	16.3	基于路径的动画	399
15.1	理解 WPF 动画	351	16.4	基于帧的动画	401
15.1.1	基于时间的动画	352	16.5	使用代码创建故事板	404
15.1.2	基于属性的动画	353	16.5.1	主窗口	405
15.2	基本动画	353	16.5.2	Bomb 用户控件	406
15.2.1	Animation 类	354	16.5.3	投炸弹	407
15.2.2	使用代码创建动画	356	16.5.4	拦截炸弹	410
15.2.3	同时发生的动画	360	16.5.5	统计炸弹和清理工作	412
15.2.4	动画生命周期	360	16.6	小结	413
15.2.5	Timeline 类	361	第 17 章	控件模板	415
15.3	故事板	364	17.1	理解逻辑树和可视化树	416
15.3.1	故事板	365	17.2	理解模板	420
15.3.2	事件触发器	365	17.2.1	修饰类	422
			17.2.2	剖析控件	423

17.3	创建控件模板	426	18.4.4	使用 FlipPanel	481
17.3.1	简单按钮	426	18.4.5	使用不同的控件模板	482
17.3.2	模板绑定	428	18.5	自定义面板	483
17.3.3	改变属性的触发器	429	18.5.1	两步布局过程	484
17.3.4	使用动画的触发器	431	18.5.2	Canvas 面板克隆	486
17.4	组织模板资源	433	18.5.3	更好的换行面板	487
17.4.1	分解按钮控件模板	433	18.6	自定义绘图元素	490
17.4.2	通过样式应用模板	435	18.6.1	OnRender() 方法	491
17.4.3	自动应用模板	437	18.6.2	评估自定义绘图	492
17.4.4	由用户选择的皮肤	438	18.6.3	自定义绘图元素	493
17.5	构建更复杂的模板	440	18.6.4	自定义装饰元素	495
17.5.1	嵌套的模板	440	18.7	小结	496
17.5.2	修改滚动条	443	第 19 章	数据绑定	497
17.5.3	控件模板示例	447	19.1	使用自定义对象绑定到 数据库	497
17.6	可视化状态	448	19.1.1	构建数据访问组件	498
17.7	小结	449	19.1.2	构建数据对象	500
第 18 章	自定义元素	451	19.1.3	显示绑定对象	501
18.1	理解 WPF 中的自定义 元素	452	19.1.4	更新数据库	503
18.2	构建基本的用户控件	454	19.1.5	更改通知	504
18.2.1	定义依赖项属性	455	19.2	绑定到对象集合	505
18.2.2	定义路由事件	457	19.2.1	显示和编辑集合项	506
18.2.3	添加标记	458	19.2.2	插入和移除集合项	509
18.2.4	使用控件	460	19.2.3	绑定到 ADO.NET 对象	510
18.2.5	命令支持	460	19.2.4	绑定到 LINQ 表达式	511
18.2.6	深入分析用户控件	463	19.3	提高大列表的性能	513
18.3	创建无外观控件	464	19.3.1	虚拟化	513
18.3.1	修改颜色拾取器的 代码	464	19.3.2	项目容器再循环	515
18.3.2	修改颜色拾取器的 标记	465	19.3.3	延迟滚动	515
18.3.3	精简控件模板	467	19.4	验证	515
18.3.4	特定主题样式和默认 样式	469	19.4.1	在数据对象中进行 验证	516
18.4	支持可视化状态	471	19.4.2	自定义验证规则	519
18.4.1	开始编写 FlipPanel 类	472	19.4.3	响应验证错误	521
18.4.2	选择部件和状态	474	19.4.4	获取错误列表	522
18.4.3	默认控件模板	475	19.4.5	显示不同的错误指示 符号	523

19.4.6 验证多个值	525	21.2.1 过滤集合	580
19.5 数据提供者	528	21.2.2 过滤 DataTable 对象	583
19.5.1 ObjectDataProvider	529	21.2.3 排序	584
19.5.2 XmlDataProvider	531	21.2.4 分组	585
19.6 小结	533	21.3 小结	589
第 20 章 格式化绑定的数据	535	第 22 章 列表、网格和树	591
20.1 数据绑定回顾	535	22.1 ListView 控件	591
20.2 数据转换	536	22.1.1 使用 GridView 创建列	592
20.2.1 StringFormat 属性	537	22.1.2 创建自定义视图	596
20.2.2 值转换器简介	539	22.2 TreeView	603
20.2.3 使用值转换器设置字符串 的格式	539	22.2.1 使用数据绑定的 TreeView	604
20.2.4 使用值转换器创建 对象	541	22.2.2 将 DataSet 对象绑定到 TreeView	606
20.2.5 应用条件格式化	543	22.2.3 即时创建节点	607
20.2.6 评估多个属性	545	22.3 DataGrid	610
20.3 列表控件	546	22.3.1 改变列的尺寸与重新 安排列	612
20.4 列表样式	548	22.3.2 定义列	613
20.4.1 ItemContainerStyle	548	22.3.3 设置列的格式和样式	617
20.4.2 包含复选框或单选按钮的 ListBox 控件	549	22.3.4 设置行的格式	618
20.4.3 交替条目样式	552	22.3.5 行细节	620
20.4.4 样式选择器	553	22.3.6 冻结列	621
20.5 数据模板	556	22.3.7 选择	621
20.5.1 分离和重用模板	558	22.3.8 排序	622
20.5.2 更高级的模板	559	22.3.9 DataGrid 编辑	622
20.5.3 改变模板	561	22.4 小结	624
20.5.4 模板选择器	562	第 23 章 窗口	625
20.5.5 模板与选择	564	23.1 Window 类	625
20.5.6 改变条目布局	568	23.1.1 显示窗口	627
20.6 ComboBox 控件	570	23.1.2 定位窗口	628
20.7 小结	573	23.1.3 保存和还原窗口位置	629
第 21 章 数据视图	575	23.2 窗口交互	631
21.1 View 对象	575	23.2.1 窗口所有权	632
21.1.1 检索视图对象	576	23.2.2 对话框模型	633
21.1.2 视图导航	576	23.2.3 通用对话框	634
21.1.3 以声明方式创建视图	579	23.3 非矩形窗口	635
21.2 过滤、排序与分组	580	23.3.1 简单形状窗口	635