

<<WPF高级编程>>

图书基本信息

书名：<<WPF高级编程>>

13位ISBN编号：9787302183563

10位ISBN编号：7302183562

出版时间：2008-10

出版时间：清华大学出版社

作者：安德拉德

页数：400

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

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.NETFramework3.0，WindowsPresentationFoundation (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MicrosoftExpressionBlend，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ExpressionBlend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VisualStudio中使用WPF编写代码即可。

本书将详细讲解WindowsPresentationFoundation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 (dependencypropertyssystem) 和事件传递模型 (routedeventmodel)。

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)、ExpressionBlend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VisualStudio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MicrosoftExpressionBlend工具。

通过使用ExpressionBlend，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VisualStudio中使用完全相同的标记语言编写

<<WPF高级编程>>

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.NET2.0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WindowsWorkflowFoundation (WF) 与WindowsCommunicationFoundation (WCF)，从而使读者可以熟悉.NETFramework3.0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ExpressionBlend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.NETFramework3.0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WindowsAPI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。

<<WPF高级编程>>

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MicrosoftExpressionBlend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MicrosoftExpressionBlend。

在本章中读者可以了解到ExpressionBlend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MicrosoftExpressionBlend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ExpressionBlend与VisualStudio2005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.NETFramework3.0的其他两个组件：WindowsWorkflowFoundation与WindowsCommunicationFoundation。

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：
WindowsSDK .NETFramework3.0运行时组件 带有OrcasCTPWinFX开发工具

的VisualStudio2005或OrcasreleaseofVisualStudio2005 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。

<<WPF高级编程>>

内容概要

《WPF高级编程》主要介绍WPF开发技术、模式和案例。

《WPF高级编程》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。

《WPF高级编程》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，《WPF高级编程》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《WPF高级编程》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。

《WPF高级编程》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《WPF高级编程》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。

《WPF高级编程》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，《WPF高级编程》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。

《WPF高级编程》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，《WPF高级编程》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。

<<WPF高级编程>>

书籍目录

第1章 WPF概述 11.1 WindowsAPI的简要历史 11.1.1 平台演化 21.1.2 .NETFramework3.0介绍 21.2 初识WindowsPresentationFoundation 31.2.1 设计指导原则 31.2.2 WPF体系结构 51.3 XAML 131.4 可视化设计工具 161.4.1 XamlPad 161.4.2 MicrosoftExpressionBlend 171.4.3 WPF可视化设计器 181.4.4 ElectricRainZAM3D 191.4.5 MobiformAurora 191.5 小结 20第2章 WPF与.NET编程 232.1 开始 232.2 需要安装的软件 242.1.2 WPF应用程序类型 242.1.3 第一个WPF应用程序 252.2 WPF开发概念 272.2.1 XAML 272.2.2 Application对象 282.2.3 创建用户界面 302.2.4 事件处理 322.2.5 使用控件 382.2.6 触发器 472.2.7 语言支持 552.2.8 部署 572.3 小结 57第3章 WPF应用程序深入剖析 593.1 项目组成 593.1.1 App.Manifest 613.1.2 AssemblyInfo.cs 613.1.3 Resources.resx 633.1.4 Settings.settings 633.1.5 MyApp.xaml 653.2 .NETFramework3.0Windows应用程序 663.3 XAML浏览器应用程序 683.4 WCF服务库 703.5 WPF自定义控件库 733.6 小结 77第4章 使用MicrosoftExpressionBlend构建富用户界面：第一部分 794.1 概述 794.2 设计环境 804.2.1 工作空间面板 814.2.2 艺术板 824.2.3 配置环境 834.3 项目结构 844.3.1 添加一个新项目文件 844.3.2 构建与一个项目 854.4 创建向量对象 854.4.1 形状 864.4.2 路径 874.5 使用属性面板操作对象 884.5.1 变换面板 884.5.2 画刷面板 884.5.3 透明性、透明度与可见度 904.6 操作文本 914.7 管理布局 924.7.1 用户界面布局面板 934.7.2 其他布局控件 954.7.3 嵌套布局面板 964.8 动画 964.8.1 时间线子面板 984.8.2 关键帧 984.8.3 为对象的属性添加动画 984.8.4 运动路径 984.8.5 触发器 994.8.6 创建控件 1004.8.7 模板 1024.8.8 样式 1034.8.9 状态 1034.8.10 导入音频/视频媒体 1044.9 小结 104第5章 使用MicrosoftExpressionBlend构建富用户界面：第二部分 1055.1 ExpressionBlend工作流 1055.2 处理用户输入 1085.2.1 键盘和鼠标类 1085.2.2 事件与事件处理 1085.2.3 定位 1105.2.4 命中测试 1145.3 使用动画API 1245.3.1 动画类 1245.3.2 使用代码通过编程创建动态动画 1245.3.3 程序动画 1275.4 使用故事板进行交互 1325.5 WPF绘图API 1345.5.1 Geometry类 1345.5.2 Shape类 1345.5.3 画刷 1385.6 小结 141第6章 特殊效果 1436.1 画刷 1446.1.1 SolidColorBrush 1446.1.2 GradientBrush 1506.1.3 ImageBrush 1536.1.4 DrawingBrush 1556.1.5 VisualBrush 1566.2 位图效果 1586.3 变换 1636.3.1 TranslateTransform类 1656.3.2 ScaleTransform类 1686.3.3 SkewTransform类 1716.3.4 RotateTransform类 1736.3.5 透明掩码 1766.4 综合运用——组合效果 1786.4.1 具有反射效果的跳跃小球示例程序 1786.4.2 动画图像观察器示例程序 1846.5 小结 190第7章 自定义控件 1917.1 概述 1917.2 控件类 1927.2.1 UserControl类 1937.2.2 创建用户控件 1937.3 WPF中的数据绑定 2027.3.1 绑定标记扩展 2027.3.2 绑定模式 2027.3.3 数据模板 2047.3.4 数据转换 2057.4 创建和编辑样式 2077.4.1 为树指定目标类型 2087.4.2 继承和覆盖样式 2097.4.3 样式触发器 2107.5 使用模板定制已有的控件 2127.5.1 小结 215第8章 WPF企业开发 2178.1 WPF应用程序模型 2188.1.1 单机应用程序 2198.1.2 基于浏览器的应用程序 2218.1.3 安全考虑 2308.2 状态管理 2318.2.1 Application对象 2318.2.2 独立存储 2358.3 状态管理示例程序 2358.3 导航 2418.3.1 导航元素 2418.3.2 结构化导航 2428.3.3 导航拓扑 2528.4 应用程序本地化 2698.4.1 自动布局指导原则 2708.4.2 使用网格以达到灵活性 2728.4.3 本地化属性和注释 2738.5 WPF部署模型 2748.5.1 生成应用程序 2758.5.2 部署单机Windows应用程序 2768.5.3 部署XAML浏览器应用程序 2778.5.4 部署.NETFramework3.0运行库 2788.6 小结 280第9章 安全性 2819.1 WPF安全模型 2819.2 受信任的应用程序与Express应用程序 2829.3 核心操作系统安全性 2829.3.1 LUA 2829.3.2 虚拟化 2839.3.3 沙箱 2839.3.4 下一代密码 2839.4 CLR安全性 2849.4.1 代码访问性 2849.4.2 临界代码方法 2919.4.3 验证 2929.5 MicrosoftInternetExplorer的安全性 2939.5.1 区域限制 2939.5.2 XBAP沙箱工作区 2949.5.3 XAML浏览器应用程序的安全性 2949.6 ClickOnce安全性 3019.6.1 受信任的发布者 3019.6.2 个人证书文件 3029.7 .NET3.0安全性实用程序 3059.8 小结 309第10章 WPF与Win32交互 30710.1 Win32用户界面概述 30710.2 WPF和HWND如何交互操作 30810.3 在WPF中使用Win32HWND 30910.4 在Win32应用程序中使用WPF 31210.5 为WPF添加Windows窗体控件 31410.5.1 在代码中添加WindowsFormsHost 31410.5.2 在XAML代码中添加HwndHost 31610.5.3 为WPF添加ActiveX控件 31610.5.4 在XAML代码中添加ActiveX控件 32110.5.5 为Windows窗体添加WPF控件 32210.5.6 影响控件属性 32610.6 小结 328第11章 高级开发主题 32911.1 WPF体系结

<<WPF高级编程>>

构 33011.1.1 核心子系统 33111.1.2 WPF线程模型 33411.1.3 桌面窗口管理器 33511.2 WPF框
架 33611.2.1 Dispatcher对象 33711.2.2 DependencyObject/DependencyProperty类 33711.2.3 Application
类 33811.2.4 Freezable类 33911.2.5 Visual类 33911.2.6 UIElement类 34511.2.7 FrameworkElement
类 34511.2.8 Control类 34611.3 XAML深入分析 34711.3.1 XAML的后台工作 34811.3.2 动态操
作XAML 35211.3.3 解析Window1.xaml 35611.3.4 XAML标记扩展 36011.3.5 XAML与自定义类
型 36011.4 WPF多线程 36211.4.1 单线程应用程序模型 36311.4.2 线程关联度
与DispatcherObject 36311.4.3 WPF分发器 36411.4.4 使用单线程应用程序 36511.4.5 异步线程 36711.5
Windows通信基础WCF 37211.5.1 服务定向 37211.5.2 WCF体系结构 37311.5.3 WCF基础 37411.5.4
构建一个WCF服务 37611.6 Windows workflow基础 38311.6.1 workflow定义 38411.6.2 WF体系结
构 38411.6.3 WF基础 38511.7 小结 399

章节摘录

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Win FX）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。

编辑推荐

《WPF高级编程》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 workflow应用程序。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>