

<<WPF编程宝典>>

图书基本信息

书名：<<WPF编程宝典>>

13位ISBN编号：9787302206569

10位ISBN编号：7302206562

出版时间：2009-8

出版时间：清华大学出版社

作者：麦克唐纳

页数：809

字数：1399000

译者：王德才

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

当.NET第一次出现时，它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）。

在这些新技术中最重要的是Windows窗体，它是用于构建Windows应用程序的一套类库。

尽管Windows窗体是一个成熟且非常完善的工具包，但它使用的是在过去十年中本质上并无变化的Windows技术。

最重要的是，Windows窗体依靠WindowsAPI来创建标准用户界面元素的可视化外观，如按钮、文本框、复选框等。

所以，这些要素在本质上是不可定制的。

例如，如果希望创建一个时髦的光晕按钮，就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，普通的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以将一个控件的内容（如按钮背后的发光效果）辐射到其他控件。

更不要指望实现动画效果，如旋转的文本、闪烁的按钮、收缩的窗口，以及生动的预览，因为对于这些效果必需手工绘制每个细节。

Windows呈现基础（Windows Presentation Foundation，WPF）通过一个使用完全不同技术的新模型改变了所有的一切。

尽管WPF也提供了熟悉的标准控件，但是它自己负责绘制每个文本、边框和背景填充。

所以，WPF能够提供更强大的功能，可以改变渲染屏幕上所有内容的方式。

使用这些特性，可以重新样式化通用控件，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内容，即使这些控件具有不规则的形状并且是半透明的。

在WPF这些新特性的背后是基于DirectX的新架构，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，而不会造成性能损失，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的特性，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），可以创建出非常绚丽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。

尽管通常会更加关注WPF中最前沿的视频、动画以及3D特性，但是关注使用WPF的标准控件和简单的可视化外观，构建常规Windows应用程序也是很重要的。

实际上，使用WPF中的通用控件和使用Windows窗体中的通用控件同样容易。

更好的是，WPF增强了商业开发人员所需要的特性，包括经过很大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。

当.NET第一次出现时，它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）。

在这些新技术中最重要的是Windows窗体，它是用于构建Windows应用程序的一套类库。

尽管Windows窗体是一个成熟且非常完善的工具包，但它使用的是在过去十年中本质上并无变化的Windows技术。

最重要的是，Windows窗体依靠WindowsAPI来创建标准用户界面元素的可视化外观，如按钮、文本框、复选框等。

所以，这些要素在本质上是不可定制的。

例如，如果希望创建一个时髦的光晕按钮，就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，普通的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以将一个控件的内容（如按钮背后的发光效果）辐射到其他控件。

更不要指望实现动画效果，如旋转的文本、闪烁的按钮、收缩的窗口，以及生动的预览，因为对于这些效果必需手工绘制每个细节。

Windows呈现基础（Windows Presentation Foundation，WPF）通过一个使用完全不同技术的新模型改变了所有的一切。

尽管WPF也提供了熟悉的标准控件，但是它自己负责绘制每个文本、边框和背景填充。

所以，WPF能够提供更强大的功能，可以改变渲染屏幕上所有内容的方式。

使用这些特性，可以重新样式化通用控件，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内容，即使这些控件具有不规则的形状并且是半透明的。

在WPF这些新特性的背后是基于DirectX的新架构，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，而不会造成性能损失，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的特性，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），可以创建出非常绚丽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。

尽管通常会更加关注WPF中最前沿的视频、动画以及3D特性，但是关注使用WPF的标准控件和简单的可视化外观，构建常规Windows应用程序也是很重要的。

实际上，使用WPF中的通用控件和使用Windows窗体中的通用控件同样容易。

更好的是，WPF增强了商业开发人员所需要的特性，包括经过很大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。

内容概要

本书在亚马逊网站上深受读者好评，由微软公司的最有价值专家Matthew MacDonald倾力而作，凝聚了Matthew多年来积累的丰富实践经验，是目前最全面的一本介绍WPF编程技术的书籍。书中不仅全面介绍了常见的图形界面编程技术，而且对WPF中非常有特色的文档和打印、音频和视频、动画、3D图形开发、多线程和插件等内容也进行了比较深入的介绍。

作者简介

Matthew MacDonald，曾荣获微软最有价值专家称号，并通过了微软的MCSD认证，是一位知名作家、教育家。

他在编程方面作出了许多重要贡献，撰写了多本有关.NET编程的经典书籍。

除本书外，Matthew的著作还包括Pro.NET 2.0 Windows Forms and Custom Controls (Apress)、Pr0.NET

书籍目录

第1章 WPF概述 第2章 XAML 第3章 Application类 第4章 布局 第5章 内容 第6章 依赖项属性和路由
第7章 经典控件 第8章 窗口 第9章 页面和导航 第10章 命令 第11章 资源 第12章 样式 第13章 形状
和画刷 第14章 图形、图画和可视化对象 第15章 控件模板 第16章 数据绑定第17章 数据模板、数据视图
和数据提供者 第18章 列表、树、工具条和菜单 第19章 文档 第20章 打印 第21章 动画 第22章 声音和
视频 第23章 3D绘图 第24章 自定义元素 第25章 与Windows窗体的互操作 第26章 多线程和插件

章节摘录

第1章 WPF概述 Windows呈现基础 (Windows Presentation Foundation, WPF) 是一个用于Windows平台的全新的图形显示系统。

WPF是针对 .NET 而设计的, 它受现代显示技术, 如HTML和Flash, 以及硬件加速技术的影响。WPF也完全不同于自从Windows 95以来一直使用的Windows用户界面。

本章将介绍WPF的体系结构, 首先会介绍WPF的工作原理, 以及它如何为开发下一代Windows应用程序提供支持。

1.1 理解Windows图形 15年以来, Windows开发人员一直在使用本质上相同的显示技术, 如果您不了解这一实际情况, 那么就很难理解WPF的变化有多么大。

标准的Windows应用程序依赖于Windows操作系统的如下两个部分来创建用户界面, 这两个部分已经使用了很长时间:

- User32该部分为许多元素提供了类似的窗口外观, 如窗口、按钮、文本框等

- GDI/GDI+该部分为渲染简单形状、文本以及图像提供绘图支持, 但是非常复杂 (而且通常性能较差)。

在过去的几年里, 这两种技术一直在改进, 并且开发人员使用的与之交互的API也已经发生了很大的变化。

但不管是使用 .NET 和Windows窗体, 还是使用过去的Visual Basic 6或基于C++代码的MFC, 在底层都是使用Windows操作系统的相同部分来工作的。

新的框架工具为使用User32和GDI/GDI+进行交互提供了更好的封装。

这些框架工具提高了开发效率, 降低了复杂性, 并且提供了更多的特性, 使开发人员不必再自己编写底层代码, 但是这些框架工具不可能避免系统组件在设计上的基本限制, 这些限制已经存在了十多年

编辑推荐

尽管wPF的视频、动画和3D功能被大肆宣传，但是wPF仅适用于构建不需要丰富图形的企业应用程序。

《WPF编程宝典：使用C# 2008和.NET 3.5(第2版)》将覆盖wPF对主要企业功能的增强，如数据绑定、打印管理以及文档显示，并展示了使用wPF创建专业企业应用程序所需要的所有技术。

WPF (Windows Presentation Foundation) 通过引入一个新的模型，使用完全不同的技术改变了这一切。

尽管wPF提供了熟悉的标准控件，但是它使用高性能的Directx引擎绘制所有细节。

所以，WPF能够提供大量冲击视觉的新功能，《WPF编程宝典：使用C# 2008和.NET 3.5(第2版)》就包括了所有这些功能。

例如，我们不用编写任何代码就可以自定义外观精美的通用控件。

我们能够使用动画效果旋转、拉伸、缩放以及扭曲用户界面中的所有内容。

使用windows Forms不可能实现或实现起来非常困难的任务——例如，播放媒体文件或创建具有纹理的3D形状——现在都可以轻松实现。

当阅读完《WPF编程宝典：使用C# 2008和.NET 3.5(第2版)》后，您将能够掌握如何完成这些以前不可能完成的任务。

多年来，.NET开发人员一直在使用Windows Forms工具包创建windows应用程序。

尽管windows Forms是一项成熟且相当完善的技术，但它本质上仍是windows的一部分，而这一部分在过去10年中基本上没有发生变化。

因此，即使用windows Forms完成简单的任务也需要做大量的工作。

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>