



计算机与信息科学系列规划教材
JISUANJI YU XINXI KEXUE XILIE GUIHUA JIAOCAI

WPF

桌面应用与开发

何敏藩 郑龙 邢立宁 肖丹 周舟
编著

湖南大学出版社

计算机与信息科学系列规划教材

WPF 桌面应用与开发

编 著 何敏藩 郑 龙 邢立宁
肖 丹 周 舟

湖南大学 出版社

内 容 简 介

本书对 WPF 进行了全面、详细的讲解。主要介绍了 WPF 与 XAML 语言, WPF 布局与依赖属性, WPF 控件, 传递事件和传递命令系统, 资源、风格及模板, 数据绑定等相关知识, 且每章都配有丰富的实例、要点和作业, 帮助读者理解和掌握书中的内容。本书适合作为计算机相关专业“WPF 桌面应用与开发”课程的培训教材, 也可以作为程序设计员或对 WPF 感兴趣读者的参考书, 供面向对象计算机爱好者和自学 WPF 的读者使用。

图书在版编目(CIP)数据

WPF 桌面应用与开发/何敏藩, 郑龙, 邢立宁, 肖丹, 周舟编著.

—长沙: 湖南大学出版社, 2018. 6

(计算机与信息科学系列规划教材)

ISBN 978-7-5667-1574-6

I. ①W… II. ①何…②郑…③邢…④肖…⑤周… III. ①Windows 操作系统—程序设计—高等学校—教材 IV. ①TP316.7

中国版本图书馆 CIP 数据核字(2018)第 140764 号

WPF 桌面应用与开发

WPF ZHUOMIAN YINGYONG YU KAIFA

编 著: 何敏藩 郑 龙 邢立宁 肖 丹 周 舟

责任编辑: 张建平

印 装: 北京虎彩文化传播有限公司印制

开 本: 787×1092 16 开 印张: 12.5 字数: 297 千

版 次: 2018 年 6 月第 1 版 印次: 2018 年 6 月第 1 次印刷

书 号: ISBN 978-7-5667-1574-6

定 价: 45.00 元

出版人: 雷 鸣

出版发行: 湖南大学出版社

社 址: 湖南·长沙·岳麓山 邮 编: 410082

电 话: 0731-88822559(发行部), 88821343(编辑室), 88821006(出版部)

传 真: 0731-88649312(发行部), 88822264(总编室)

网 址: <http://www.hnupress.com>

电子邮箱: presszhangjp@hnu.cn

版权所有, 盗版必究

湖南大学出版社凡有印装差错, 请与发行部联系

计算机与信息科学系列规划教材 编委会

主任：周忠宝

编委：周忠宝 郑 龙 何敏 藩

邢立宁 罗俊海 姚 锋

叶朝晖 邓劲生 姚煊道

邹 伟 王 浩 张 章

肖 丹 蔡 琴 付 艳

周 滔 周 舟

编著与设计单位：湖南大学

国防科技大学

电子科技大学

佛山科学技术学院

长沙学院

深圳华大乐业教育科技有限公司

前 言

时光荏苒,如白驹过隙,一转眼中国互联网已走过了 30 年的历程。回首过去,人工智能、云计算、移动支付,这些互联网产物不仅迅速占据了我们的生活,刷新了我们对科技发展的认知,而且也提高了我们的生活质量水平。人们谈论的话题也离不开这些,例如:人工智能是否会替代人类,成为工作的主要劳动力;数字货币是否会代替纸币流通于市场;虚拟现实体验到底会有多真实多刺激。从这种现象中不难发现,互联网的辐射面在不断变广,计算机科学与信息技术发展的普适性在不断变强,信息技术如化雨春风,润物无声地全面融入,颠覆了我们的生活。

1987 年,我国网络专家钱天白通过拨号方式在国际互联网上发出了中国有史以来第一封电子邮件,“越过长城,走向世界”,从此,我国互联网时代开启。30 年间,人类社会仍然遵循着万物自然生长规律,但互联网的枝芽却依托人类的智慧于内部结构中野蛮扩展,并且每一次主流设备、主流技术的迭代速度明显加快。如今,人们的生活是“拇指在手机屏幕方寸间游走的距离,已经超过双脚走过的路程”。

据估计,截至 2017 年 6 月,中国网民规模已达到 7.5 亿人,占全球网民总数的五分之一,而且这个数字还在不断地增加。

然而,面对快速发展的互联网,每一个互联网人亦感到焦虑,感觉它运转的速度已经快到我们追赶的极限。信息时刻在更新,科技不断被颠覆,想象力也一直被挑战,面对这些,人们感到不安的同时又对未来的互联网充满期待。

互联网的魅力正在于此,恰如山之两面,一面阴暗晦涩,一面生机勃勃,一旦跨过山之巅峰,即是不一样的风景。就是这样的挑战会让人着迷,并甘愿为之付出努力。而这个行业还有很多伟大的事情值得去琢磨,去付出自己的匠心。

本系列丛书作为计算机科学与信息科学中的入门与提高教材,在力争保障学科知识广度的同时,也统筹主流技术的深度,既介绍了计算机学科相关主题的历史,也涵盖国内外最新、最热门课题,充分呈现了计算机科学技术的时效性、前沿性。丛书涉及计算机与信息科学多门课程:JAVA 程序设计与开发、C# 与 WinForm 程序设计、SQL Server 数据库、Oracle 大型数据库、Spring 框架应用开发、Android 手机 APP 开发、JDBC/Jsp/Servlet 系统开发,等等;HTML/CSS 前端数据展示、jQuery 前端框架、JavaScript 页面交互效果实现,等等;大数据基础与应用、大数据技术概论、R 语言预测、PRESTO 技术内幕,等等;Photoshop 制作与视觉效果设计、网页 UI 美工设计、移动端 UI 视觉效果设计与运用、

CorelDraw 设计与创新,等等。

本系列丛书适合初学者,书中内容所涉及的知识点和相关信息应了解、掌握。开发人员可从本系列丛书中找到许多不同领域的兴趣点和各种知识点的用法。丛书实例内容选取市场流行的应用项目或产品项目,章后部分练习题模拟了大型软件开发企业实例项目。

本系列丛书在编写过程中,获得了国家自然科学基金委员会与中国民用航空局联合资助项目(U1733110)、湖南省科学“十三五”规划课题(XJK016BGD009)、湖南省教学改革研究课题(2015001)、湖南省自然科学基金(2017JJ1012)、国家自然科学基金(71371067)的资助,并得到了湖南大学、国防科技大学、电子科技大学、佛山科学技术学院、长沙学院和深圳华大乐业教育科技有限公司各位老师的大力支持,同时参考了一些相关著作和文献,在此向这些老师和文献作者深表感谢。

未来互联网信息技术已扑面而来,汹涌胜于往昔,你做好准备了吗?

作者
2017年9月

目次

理论部分

第 1 章 WPF 简介与 XAML 语言	2
1.1 WPF 简介	2
1.1.1 什么是 WPF	2
1.1.2 WPF 的作用	3
1.1.3 WPF 发展史	3
1.2 WPF 体系结构	4
1.2.1 WPF 核心组件	4
1.2.2 WPF 基本功能和服务	4
1.3 XAML 语言	5
1.3.1 创建 WPF 应用程序	6
1.3.2 什么是 XAML 语言	8
1.3.3 命名空间	8
1.3.4 XAML 的属性语法和属性元素语法	8
1.3.5 附加属性	9
1.4 WPF 应用程序生命周期	10
第 2 章 WPF 布局与依赖属性	15
2.1 什么是布局	15
2.1.1 WPF 布局基本元素面板	16
2.1.2 Canvas 布局控件	16
2.1.3 StackPanel 布局控件	17
2.1.4 WrapPanel 布局控件	18
2.1.5 DockPanel 布局控件	19
2.1.6 Grid 布局控件	20
2.1.7 UniformGrid 布局控件	22
2.1.8 ScrollViewer 控件	22
2.2 属性系统	23
2.2.1 CLR 属性	23

2.2.2	依赖项属性的概念	24
2.2.3	依赖属性的继承	27
2.2.4	自定义依赖属性	28
2.2.5	附加属性	34
第3章	WPF 控件	36
3.1	WPF 控件概述	36
3.2	内容控件	37
3.2.1	按钮基类	37
3.2.2	按钮(Button)	38
3.2.3	拨动按钮(ToggleButton)	38
3.2.4	重复按钮(RepeatButton)	38
3.2.5	复选框(CheckBox)	39
3.2.6	单选框(RadioButton)	39
3.2.7	标签控件(Label)	39
3.2.8	工具提示控件(ToolTip)	40
3.2.9	滚动条控件(ScrollViewer)	40
3.3	条目控件(ItemsControl)	40
3.3.1	菜单基类(MenuBase)	40
3.3.2	选择器基类(Selector)	41
3.3.3	状态条(StatusBar)	43
3.4	文本控件(TextControls)	43
3.4.1	密码框>PasswordBox)	43
3.4.2	文本框(TextBox)	43
3.4.3	富文本框(RichTextBox)	43
3.5	范围控件(RangeControl)	43
3.5.1	滚动条(ScrollBar)	43
3.5.2	滑动条(Slider)	44
3.5.3	进度条(ProgressBar)	46
第4章	传递事件和传递命令系统	47
4.1	WPF 中的元素树	47
4.1.1	逻辑树用途	47
4.1.2	可视化树	49
4.1.3	树遍历	50
4.1.4	“树”形式路由事件的路由	50
4.1.5	资源和树	51
4.2	传递事件	51
4.2.1	传递事件	51

4.2.2	终止事件传播	54
4.2.3	处理传递事件	55
4.2.4	附加传递事件	58
4.3	考察传递事件	62
4.4	传递命令	66
4.4.1	ICommand 接口	69
4.4.2	ICommandSource 与 CommandTarget	69
4.4.3	CommandBinding	69
4.4.4	传递命令(RoutedCommand)	70
4.4.5	WPF 命令仓库	70
第 5 章	资源、风格及模板	74
5.1	资源	74
5.1.1	统一资源标识(Unified Resource Identifier)	75
5.1.2	.NET 开发平台对资源国际化的支持	77
5.1.3	资源管理	77
5.1.4	WPF 元素中定义的资源	80
5.2	风格	87
5.2.1	Style 类	87
5.2.2	Setter	87
5.2.3	TargetType	89
5.2.4	BasedOn	90
5.2.5	触发器(Triggers)	91
5.3	模板	93
5.3.1	模板概述	93
5.3.2	控件模板(ControlTemplate)	94
5.3.3	数据模板(DataTemplate)	95
第 6 章	数据绑定	107
6.1	数据绑定的概述	107
6.1.1	什么是数据绑定	107
6.1.2	基本数据绑定概念	109
6.2	简单的数据绑定	110
6.2.1	简单的数据绑定	110
6.2.2	在 C# 中实现数据绑定	111
6.2.3	对非 FrameworkElement 和 FrameworkContentElement 实现 绑定	112
6.3	考察五种数据绑定模式	112
6.4	动态绑定	113

6.5 最简单的数据绑定——从 .NET 对象到界面元素	113
6.6 DataContext	115
6.7 控制绑定时刻	116
6.8 IValueConverter 开发	117
6.9 数据绑定的校验	119
6.9.1 开发业务规则类	119
6.9.2 在绑定中添加任意多个业务规则	121
6.9.3 清除控件上的错误信息	121
6.10 对集合对象的绑定	122
6.11 数据源	124
6.11.1 XML 数据源	125
6.11.2 .NET 对象数据源	127

上机部分

上机 1 WPF 简介与 XAML 语言	132
第 1 阶段 指 导	132
指导 1 创建 WPF 墨水应用程序	132
指导 2 为按钮添加点击事件	134
第 2 阶段 练 习	135
练习 创建 WPF 应用程序登录窗体	135
上机 2 WPF 布局与依赖属性	136
第 1 阶段 指 导	136
指导 1 使用 Canvas 进行布局	136
指导 2 StackPanel 布局控件	138
指导 3 布局的综合应用	140
指导 4 自定义布局控件	142
第 2 阶段 练 习	144
练习 1 使用 DockPanel 布局	144
练习 2 使用 Grid 布局控件	145
上机 3 WPF 控件	146
第 1 阶段 指 导	146
指导 1 完成窗体布局	146
指导 2 使用 ListView 显示数据	149
第 2 阶段 练 习	155
练习 在 ListView 中显示并能播放联系人的视频与铃声	155

上机 4 传递事件和传递命令系统	157
第 1 阶段 指 导	157
指导 1 使用 WPF 命令部分命令	157
指导 2 绑定并处理 WPF 命令库中部分命令	159
第 2 阶段 练 习	163
练习 完成记事本剩余功能	163
上机 5 资源、风格及模板	164
第 1 阶段 指 导	164
指导 1 将通讯录登录页面的按钮的风格放入资源中	164
指导 2 为按钮设计模板	170
第 2 阶段 练 习	175
练习 给控件模板添加触发器	175
上机 6 数据绑定	176
第 1 阶段 指 导	176
指导 1 DataGrid 的使用	176
指导 2 设计 DataGrid 样式	182
第 2 阶段 练 习	186
练习 用 DataGrid 对数据进行编辑	186
参考文献	187

理论部分

理论部分

第 1 章 WPF 简介与 XAML 语言

学习目标

- 理解 WPF 是什么
- 了解 WPF 工作原理
- 了解 WPF 运行环境
- .NET 3.0 新增的主要组件

1.1 WPF 简介

WPF 是基于 DirectX 的新一代开发技术,其利用 XAML(应用程序扩展语言)做界面描述,后台采用各种 .NET 语言作为业务逻辑开发。成功应用其图像处理技术,可以实现复杂的图像渲染。WPF 测试版首次应用在 2006 .NET 3.0 中,后连同 VS2008 .NET 3.5 一同正式发布。由于使用 XAML 页面处理技术,Windows 桌面应用程序与 Web 应用程序之间的间隙变得模糊。

1.1.1 什么是 WPF

WPF 为 Windows Presentation Foundation 的首字母缩写,中文译为“Windows 呈现基础”,其原来代号为“Avalon”,因与“我佩服”拼音首字母组合一样,国内有人调侃地称之为“我佩服”。由 .NET Framework 3.0 开始引入,与 Windows Communication Foundation 及 Windows Workflow Foundation 并行为新一代 Windows 操作系统以及 WinFX 的三个重大应用程序开发类库。

WPF 是微软新一代图形系统,运行在 .NET Framework 3.0 及以上版本下,为用户界面、2D/3D 图形、文档和媒体提供了统一的描述和操作方法。基于 DirectX 9/10 技术的 WPF 不仅带来了前所未有的 3D 界面,而且其图形向量渲染引擎也大大改进了传统的 2D 界面,比如 Vista 中的半透明效果的窗体等都得益于 WPF。程序员在 WPF 的帮助下,要开发出媲美 Mac 程序的酷炫界面已不再是遥不可及的奢望。WPF 相对于 Windows 客户端的开发来说,向前跨出了巨大的一步,它提供了超丰富的 .NET UI 框架,集成了矢量图形,丰富的流动文字支持(flow text support),3D 视觉效果和强大无比的控件模型框架。

Windows Presentation Foundation(以前的代号为“Avalon”)是 Microsoft 用于 Windows 的统一显示子系统,它通过 WinFX 公开。它由显示引擎和托管代码框架组成。Windows Presentation Foundation 统一了 Windows 创建、显示和操作文档、媒体和用户界面(UI)的方式,使开发人员和设计人员可以创建更好的视觉效果、不同的用户体验。

Windows Presentation Foundation 发布后,Windows XP、Windows Server 2003 和以后所有的 Windows 操作系统版本都可以使用它。

WPF 是一套 API 函数库,在 .NET Framework 3.0 以上版本类库运行,是 .Net Framework 3.0 里新推出的主要功能模块之一。

WPF 是 Windows 操作系统中一次重大变革,与早期的 GDI+/GDI 不同。WPF 是基于 DirectX 引擎的,支持 GPU 硬件加速,在不支持硬件加速时也可以使用软件绘制;能对高级别的线程进行绘制,提高使用者的体验;能自动识别显示器分辨率并进行缩放。而 Vista 就是一个非常典型的例子。

1.1.2 WPF 的作用

程序人员和设计人员是有明确分工的。设计人员使用 Expression Studio 中的可视化套装工具进行界面设计,将设计界面对应的 XAML 文件交给程序人员即可。程序人员将 XAML 文件直接套用到开发环境,就不需要考虑界面设计了。程序员可以用任何一种 .NET 编程语言进行开发。

WPF 最重要的特色是对矢量图的超强支持,并兼容支持 2D 绘图,比如矩形、自定义路径、位图等绘制。对三维运算支持也很强大,包括 3D 控件及事件,支持将 3D 与 2D 及视频进行合并打造更立体的效果。使用高精度 (ARGB) 颜色,并支持浮点类型的像素坐标。这些功能都是远远强于 GDI+ 的。

1.1.3 WPF 发展史

在最新版 Visual Studio 中,微软加入了 git 源码管理工具,和之前的 TFS 大体上类似。.NET 发展到现在已经有十多年的历史了,为了更好地理清 .NET 的发展历史,特总结如表 1-1。

表 1-1 .NET 版本发展历程

版本	版本号	发布日期	Visual Studio	Windows 集成
1.0	1.0.3705.0	2002-02-13	Visual Studio .NET	
1.1	1.1.4322.573	2003-04-24	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	2005-11-07	Visual Studio 2005	
3.0	3.0.4506.30	2006-11-06		Windows Vista, Windows Server 2008
3.5	3.5.21022.8	2007-11-19	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	2010-04-12	Visual Studio 2010	4.0

1.2 WPF 体系结构

1.2.1 WPF 核心组件

WPF 可以认为是 MS 利用原有 .NET 框架的一些特色,加上 DirectX 的产物。从图 1-1 的 WPF 组件中,我们可以看出最底层仍然是一些内核 API。

其中 PresentationFramework、PresentationCore、milcore 组件是 WPF 的核心。milcore 是一个和 DirectX 交互的非托管组件,非托管代码能带给我们更高效的处理,能更好地和 DirectX 交互。WPF 的所有显示都是由 DirectX 完成的。milcore 中一个非常重要的功能就是 Composition 引擎,这个引擎对效率的要求很高,它的具体作用稍后介绍。所以 milcore 放弃了一些 CLR 的特征来换取效率。而另外两个组件都是建立在 CLR 基础之上,利用了 .NET 的优势。

至于其中的 User32 组件有什么作用,作者目前知道的就是在 WPF 的某些应用场景中为了某些兼容需要而使用 User32,例如 DWM(桌面窗口管理)。DWM 涉及的内容非常多,感兴趣的同学可以看 SDK 文档。

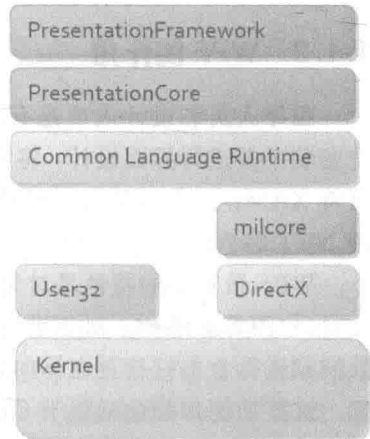


图 1-1 WPF 核心组件结构

1.2.2 WPF 基本功能和服务

我们除了关心 WPF 的基本结构外,更重要的是要知道 WPF 提供了什么功能,请看图 1-2:

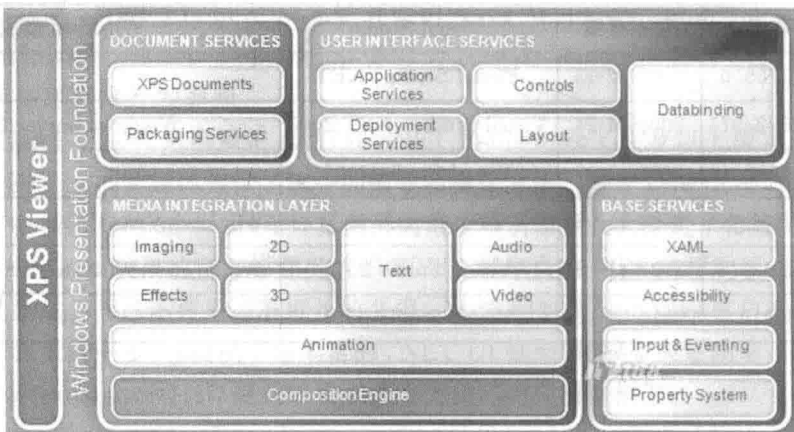


图 1-2 WPF 基本功能和服务视图

- Core Presentation

包含了所有的图形效果,如图形、2D 图形、3D 图形、文本、音频、视频和显示效果。同时还包括强大的动画效果,动画可以应用前面的所有元素。最下面的那个就是视觉基本元素。

- User Interface Services

包含了 Application Services(应用程序服务)、Deployment Services(部署服务)、Controls(控件库)、Layout(布局)和 Data Binding(数据绑定)。

- Base Services

提供了 XAML 支持、Accessibility(提高开发效率)、Input&Events(输入和事件的支持)、Property System(属性系统)。

- Document Services 和 XPS Viewer

提供了基本的打印和报表服务,可以通过这些组件实现自定义的打印和显示效果。

基于上面的特性以及我们使用的情况可以总结出:

在输入方面,WPF 支持键盘、鼠标、手写笔、触摸屏,蓝牙录入、视频捕捉等;在和外部设备的交互上,能与有驱动、COM、Win32 或通信协议支持的任何设备进行交互;打印和报表方面,可访问打印对话框、打印队列、打印窗体、控件以及很多可选操作,同时对报表也支持得很好。同时支持流文档和固定文档,支持对文件的无限制访问,对动画、图形图像、3D、影像的支持也相当好(其他技术是很难做到的),所以 WPF 除了在性能方面存在着缺陷以外,其他方面可以说是做得非常优秀,同时我们有信心相信微软也会在性能方面下工夫。

WPF 与现有技术的对比,见表 1-2。

表 1-2 WPF 与现有技术的对比

WPF	现在的技术
XAML	XHTML
Layout	WinForms 2.0 Panels
Styles	CSS
Control Templates	
3D	Managed DirectX
Animation	Flash
Data Binding	ASP.NET Syntax
Data Templates	ASP.NET Templates(Atlas)

从表中可以看出,WPF 技术和其他的技术有很多相似之处,比如 XAML 和 XHTML、Styles 和 CSS 等尤其相似,所以对我们的开发人员来说,学习 WPF 也不是一件难事。

1.3 XAML 语言

WPF 有两套 API,一套用于普通的编码访问,比如 C#、VB.NET 等其中 .NET 支持的语言。而另外一套就是基于 XML 的 API,被称为 XAML(Extensible Application Markup Language)。

XAML 实现 UI 代码和应用程序逻辑代码的分离。在 .NET 3.0 和 Windows Vista 中, XAML 与 WPF 一起建立整个 UI。由于 XAML 是基于 XML 的, 所以每个 XAML 代码都肯定是一个完整的 XML 文件。XAML 继承了 XML 所有的定义和规则。XAML 与其他 XML 扩展不同之处就是他所表示的意义。每个 XAML 元素是一个 .NET CLR 类。基于 XML 使得我们非常容易扩展和操作 XAML。利用 XAML 的 WPF 这种关系, 开发人员可以单独设计漂亮的 UI。我们可以把程序逻辑写在单独的文件或者是内联嵌入到 XML 文件。

1.3.1 创建 WPF 应用程序

默认开发环境为 Windows7 + Visual Studio. Net 2008。

(1) 在搭建好开发环境之后, 打开 VS2008, 选择创建项目(Create Project), 出现如图 1-3 所示的选择项目模板的对话框:



图 1-3 选择模板对话框

从图中可以看出 VS2008 有关 WPF 的项目模板(Project Template)包括:

- WPF Application: 使用 WPF 控件(WPF Control)和事件处理设计用户接口(User Interface)。
- WPF Browser Application: 创建一个能在浏览器中运行的 WPF 应用程序。
- WPF Custom Control Library: 用于创建自定义控件, 自定义控件继承 Control 类, 此模板中不能可视化设计。
- WPF User Control Library: 用于创建用户控件, 用户控件继承 UserControl 类, 能可视化设计控件。