



Windows[®] Presentation Foundation 程序设计指南

Applications = Code + Markup: A Guide to the Microsoft Windows Presentation Foundation

[美] Charles Petzold 著

蔡学镛 译

胡志鹏 魏 颢 成 功 审校



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Windows Presentation Foundation

程序设计指南

Applications = Code + Markup:
A Guide to the Microsoft Windows Presentation Foundation

[美] Charles Petzold 著

蔡学镛 译

胡志鹏 魏颢 成功 审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要介绍了微软新一代操作系统平台上的 Microsoft Windows Presentation Foundation 核心技术的原理、概念、技术、技巧与开发实践。全书全面细致、深入浅出,主要内容包括 Windows Presentation Foundation 概述、基本 Bushes、Content 概念、Button 及其他控件、Stack、Wrap、Dock、Grid、Canvas (画布)、依赖性属性、Routed Input Event、定制元素等诸多内容,本书适合广大 Windows 平台开发人员阅读和参考。

Copyright © 2007 by Microsoft Corporation. All rights reserved.

Original English language edition ©2006 by Microsoft by Charles Petzold.

All rights reserved. Simplified Chinese edition published by arrangement with the original publisher, Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体版专有出版权由 Microsoft Corporation 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2007-5881

图书在版编目 (CIP) 数据

Windows Presentation Foundation 程序设计指南 / (美) 佩佐德 (Petzold,C.) 著; 蔡学镛译. —北京: 电子工业出版社, 2008.1

书名原文: Applications = Code + Markup: A Guide to the Microsoft Windows Presentation Foundation

ISBN 978-7-121-05115-9

I. W… II. ①佩…②蔡… III. 窗口软件, Windows—程序设计 IV.TP316.7

中国版本图书馆 CIP 数据核字 (2007) 第 150736 号

责任编辑: 梁 晶

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 64 字数: 1160 千字

印 次: 2008 年 1 月第 1 次印刷

定 价: 118.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

审校者序

第一次接触 WPF (Windows Presentation Framework) 还是在 2006 年年底。

当时我还在微软亚洲研究院的技术转换组, 和另外两个同事一道, 刚刚做完了一个叫 EYAM 的演示软件, 用以测试视觉计算组最近的一些研究成果。

由于我们一开始仅仅把 EYAM 的目标定位于测试, 所以, 在基于 Windows Forms 开发它时, 并没有花太多时间在界面设计上。然而, 需求总在变化。因为测试效果不错, 组里决定要把 EYAM 拿到来年的微软全球研究院技术节上展示。这不但要求我们为 EYAM 添加更多的功能, 也意味着我们必须重新设计它的界面。的确, 若要从展会上数以百计的演示软件中脱颖而出, 吸引与会者的眼球, EYAM 就不但需要有很坚实的技术支撑, 还要有足够动人的绚丽外表。

新增的功能, 我们自付还能应付, 但界面调整呢? 我们准备找一个专业设计师来帮我们, 但一时找不到合适的人选, 留给我们的时间却只有短短两个月了。

我们开始考虑其他的 Windows 程序开发框架。首先, 该框架要尽可能保证界面与后台处理逻辑分离, 以便在开发后期, 设计师能顺利介入调整界面, 而不影响程序内在逻辑; 其次, 该框架要有足够强大的界面设计能力, 以保证我们在设计师最终无法到位的情况下, 也有机会做出令人满意的用户界面。显然, WPF 成为了我们最合适的选择。虽然我们需要为此付出一定的学习时间, 但 WPF 的优良特性和对新技术的尝试欲望, 让我们下定决心, 在接下里的两个多月里, 用 WPF 来重构整个 EYAM。

开始工作后, 我们才发现 WPF 的确太新了, 以至于没有多少资料可供我们参考。浩如烟海的 MSDN 虽然全面, 但毕竟不是用来系统学习的, 而且在很多细节上, 也是语焉不详; WPF 核心开发组的 blog 的确解决了不少疑难, 也是个不错的补充, 但同样缺乏系统性。

如此雾里看花般忙碌了两个多月, 我们最终没有等到传说中的设计师, 但也总算借助 WPF 完成了一个还算不错的新版 EYAM。回头看看这段经历, 的确走了不少弯路, 而缺乏可供系统学习的书籍资料是一大原因。

WPF 不同于以往的任何开发框架, 是一次全面的革新。对于初学者来说, 上手也许不难, 但如果系统地学习它, 不久之后你就会发现自己陷入一种知其然, 不知其所以然的尴尬境地。问题是解决了, 但为什么这么解决, 原理如何, 却不甚清楚。在开发过程中, 我们遇到的最棘手的问题就是 WPF 的效率。当我们试图用一个 WrapPanel 来加载并显示成千上万张图片

张图片的时候，才发现其低下的效率令人无法接受。在 WPF 的 WrapPanel 之下，到底隐藏着怎样的玄机？我们没有时间通过 Reflection 去分析。虽然最终在开发组的 Blog 上找到了解决的办法，但也仍是一知半解。相信做技术工作的读者都会有这种感受，如果你对某种技术的掌握程度只是半生不熟，那么开发过程会变得相当痛苦。不了解原理，就如同被一个无形的枷锁给钳制着，无法挥洒自如。

2007 年 3 月份，我参加完微软技术节回国，博文视点的周筠老师问我是否愿意审校 *Application = Code + Markup* 的中文译本。想起开发小组曾经缺少一本好书来系统学习 WPF，以及由此产生的不便，更考虑到本书作者 Charles Petzold 的鼎鼎大名，我便应承下来。

毕竟是 *Programming Windows* 作者的力作。通读全文，用一句话形容我的感受，便是“夫子言之，于我心有戚戚焉”。

根据笔者的经历，对于初学者来说，在没有正确引导的情况下，WPF 的学习曲线是比较陡峭的。“Visual Tree”，“Logical Tree”，“Data Binding”，“Template”，“Dependency Property”，……，当所有这些新概念一齐涌入你的大脑时，你是很难在短时间内把它们系统地组织起来并消化掉的。如何从读者已掌握的知识出发，引导读者循序渐进地接受新知识，这正是 Petzold 所擅长的。Petzold 从不孤立地介绍某个概念，而总是愿意把它们放在一个读者更为熟悉的上下文环境中，从背景说起，前因后果，娓娓道来，尽可能让这些新的概念变得不那么生涩难懂。

正是为了以一种合理的方式来引导读者学习 WPF，本书在内容上做了别具匠心的安排。前半篇幅从读者最熟悉的 Code 形式出发，详细讲解了如何单纯使用 WPF API 来开发程序。看完这部分内容，读者应该会对 WPF 的工作原理有一个全面的认识。接下来，作者才笔锋一转，用后半篇幅详细讲述 WPF 的最大特色——XAML。

这样的谋篇布局，对于那些希望从以前的 Windows 编程接口，如 Windows API，MFC，Windows Forms，转到 WPF 上来的程序员来说，是再恰当不过的，因为它提供了一条相对平坦的学习路线。这正如想深入学习 Windows 编程的人，应该从 Windows API 开始学起而不是一上来就用 MFC 一样，打好了基础，才能更进一步，正所谓磨刀不误砍柴工。

本书也非常适合有过一些 WPF 编程经验的人阅读。或许你会跟我和我的同事一样，读到某些精彩部分时，忍不住惊叹一声：“原来如此！”那是一种山重水复时，豁然开朗的畅快。“DataBinding 怎么使用？”“VirtualizingStackPanel 究竟有什么作用？”“ListBox 和 TreeView 控件是否还能做很多你意想不到的事？”很多困扰自己多时的问题，都在 Petzold 轻松的行文里得到了解答。浏览全文，你会发现章节之间的过渡不但非常自然，更有一些有趣的悬念设置，让读者在读完一章后就急着去翻阅下一章的内容，而这通常是读小说时才会有的那种不忍释卷的急迫感。

最令人称道的是，Petzold 结合讲解内容，为读者精心准备了大量的实例代码。这些实例不但紧扣讲述内容，而且非常实用，绝无半点敷衍。和任何一本介绍编程语言的书一样，

Petzold 从最简单的“Hello World”程序开始，引领读者进入了 WPF 的奇幻旅程。在随后的阅读过程中，读者可以看到更多精彩例程，从而逐步体会到 WPF 的优雅与强大。这当中有不少小巧精致的例程，比如第 4 章的仿 Windows 记事本和第 5 章的仿 Windows Explorer，都只不过区区百行代码，却实现了足够多的功能。所有的代码都非常简洁清晰，十分便于读者阅读和仿效。

比较另类的是，作为一本主讲界面编程的书，全书没有一处程序截图，想要观看书中的例程的效果，唯一的途径就是下载（或手工输入）源代码，编译并运行它们。“纸上得来终觉浅，绝知此事要躬行”，上手尝试，始终是学习一门语言的必经之路。

在 Amazon 上，有不少读者留下了对该书英文版的评论。有趣的是，评论出现明显的两个极端，要么说很好，要么说很糟。批评这本书的读者，认为 Petzold 花了太多篇幅在 WPF 的原理上，而没有直奔主题的去讲解如何用 WPF 来做一个完整的应用程序。或许是个人口味不同吧，我还是很喜欢 Petzold 这种抽丝剥茧的讲解风格，不但告诉你 How，还告诉你 Why。

如果你想了解隐藏在 WPF 华丽外表之下的奇妙世界，如果你有“千淘万沥虽辛苦，吹尽黄沙始到金”的那份执着和耐心，那么就请泡上一杯茶，打开你的计算机，准备好 WPF 的编程环境，开始细细品味这本充满乐趣的书吧。

胡志鹏

2007 年 11 月于广州

联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿信箱: bvtougao@gmail.com

北京博文视点资讯有限公司 (武汉分部)

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室

邮政编码: 430074

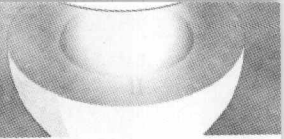
电话: (027) 87690813 传真: (027) 87690813 转 817

若您希望参加博文视点的有奖读者调查, 或对写作和翻译感兴趣, 欢迎您访问:

<http://bv.csdn.net>

关于本书的勘误、资源下载及博文视点的最新书讯, 欢迎您访问博文视点官方博客:

<http://blog.csdn.net/bvbook>



介绍

本书介绍如何使用 Windows Presentation Foundation (WPF) 编写微软 Windows 平台的程序。WPF 的程序可以是一般的独立 Windows 应用程序（这样的程序现在常称为 client 应用程序），或者是分布式（distributed）应用程序的前端。WPF 被认为是 Windows Vista 的主要 API，也可以用在安装有 .NET 3.0 的 Windows XP/SP2 或 Windows 2003。

虽然你使用 WPF 来编写“一般的窗口应用程序”，但是 WPF 应用程序绝对不同于以往的窗口应用程序。WPF 具有崭新的外观、新的控件设计哲学、新的图形效果（包含动画和 3D）、和新的 API（应用程序接口）。

WPF 其实包含“两个”相互关联的编程接口。你可以利用 C#（或其他 .NET 语言）编写 WPF 程序。WPF 也具备一个让人兴奋的 XML 语言，称为 Extensible Application Markup Language（XAML 发音为“zammel”），甚至可以只用 XAML 开发某些应用程序。但是一般来说，你会同时利用代码（code）和标记（markup）一同开发出你的应用程序（这也是本书英文书名的意义所在）。你会使用 XAML 定义用户界面的视觉部分（包括图形和动画），然后写程序处理使用者的输入。

你的背景

在写这本书的时候，我假定读者熟悉 C# 编程语言以及具备之前版本的 .NET Framework 经验。如果你不具备这样的能力，请阅读我写的另一本书 *.NET Book Zero: What the C or C++ Programmer Needs to Know about C# and the .NET Framework*，你可以从我的网站的下列网址中免费获得这本书：

<http://www.charlespetzold.com/dotnet>

如果你是编程初学者，我建议你利用写 console 程序也就是文字模式的程序来学习 C#。我的书 *Programming in the Key of C#: A Primer for Aspiring Programmers*（Microsoft Press, 2003）正是采用这样的方式。

本书

早在 1985 年, 我就开始写 Windows 程序了, WPF 是我体验过的 Windows 技术中最让人感到兴奋的。但是由于 WPF 支持两种非常不同的编程接口, 所以写这本 WPF 的书对我来说也是一大挑战。在深思熟虑之后, 我认为每个程序员都应该要对 WPF 的代码部分建立稳固的根基, 因此, 本书第 1 部分引导你如何利用 C# 写出整个 WPF 程序。

WPF 有数个特色, 需要加强 .NET 的 property 和事件, 了解这些“加强的方式”相当重要, 因此, 本章第 1 部分 (Part I) 将焦点放在 dependency 属性 (property) 和 routed 输入事件上。

本书第 2 部分 (Part II) 将焦点转移到 XAML。我会展示如何只用 XAML 建立小型的应用程序, 以及如何结合 XAML 和 C# 以建立更大型、更复杂的应用程序。第 2 部分一开始, 我所做的第一件事就是建立一个名为 XAML Cruncher 的编程工具, 这个工具对我学习 XAML 的帮助很大, 我希望对你有相同的效果。因为 XAML 是用来建立应用程序的可视化部分的, 所以本书的大部分和图形 (graphics) 相关的内容, 会放在第 2 部分。

从长远看来, XAML 的文件大多数是用交互式工具和编程工具产生出来的, 我知道你以后也会使用这些设计工具来加速应用程序的开发。然而, 我认为每个程序员还是有必要具备“亲自动手编写 XAML”的能力, 本书试图要教你这么做。如果你和我一样, 你会发现用 XAML 来编写 (特别是搭配 XAML Cruncher 这样的工具) 本身就是一件很有趣的事。

我写的书向来比较偏向“学习用书”而非“参考用书”。读过我以前书籍的读者, 认为从第 1 章开始, 就能够让他们以循序渐进的方式阅读后继章节。这样的书最有用, 让你的手指习惯键入 C# 和 XAML, 这是很重要的。就像学习杂技或演奏双簧管一样, 编程可不是光靠读书就可以学来的。一般来说, 实际输入代码的人, 学习的效果会比较快速, 比较深入。特别是在前几章, 最好要试着这么做, 而且勇于改变程序, 自己做一些实验。在前面的几章中, 我会提到许多实验, 你可以动手尝试一下。

Windows 操作系统与编程

1985 年秋天, 微软发布了 Windows 的第一个版本。从此以后, Windows 一直在进步, 在扩展, 特别是 1993 年的 Windows NT 和 1995 年的 Windows 95, 变化相当大, 从 16 位进入 32 位的构架。

当 Windows 刚发布时，编写 Windows 程序的方法只有一种，就是利用 C 语言和 Windows API。过去这些年来，越来越多的语言被采用在 Windows 编程上，包括微软的 Visual Basic 和 C++。对于“以 C 为基础”的语言来说，微软目前提供 4 种 Windows 应用程序编程方法，如下面表格所示：

如何通过“以 C 为基础”的语言，编写 Windows 应用程序（从微软的角度而言）。

开始年份	语言	接口
1985	C	Windows API
1992	C++	微软 MFC 链接库
2001	C#或 C++	Windows Form (.NET Framework 1.0 的一部分)
2006	C#或 C++	Windows Presentation Foundation (WPF)

告诉你应该用什么语言和接口来编写 Windows 应用程序，这不是我该做的事。只有你自己才能根据任务和资源，作出适当的决策。

如果你想学习关于 Windows API，你可以参考我写的 *Programming Windows*（第五版，Microsoft Press，1998），许多人觉得这本书对他们很有价值。虽然我从来不在乎 MFC，但是它是历史上被最多人使用的 Windows 编程方法。Jeff Prosisxie 写了一本不错的 MFC 学习书 *Programming Windows with MFC*（第二版，Microsoft Press，1999），Jeffrey Richter 写了一本进阶书 *Programming Applications for Microsoft Windows*（Microsoft Press，1999）。

如果你比较想学习更现代化的 Windows 编程（毕竟都已经是 21 世纪了），.NET 才是你的王道。.NET 程序会被编译成中间代码（微软中间语言，MSIL），然后在执行时被编译成机器码，这种 managed code（受控制的代码）可以保护操作系统免于有问题的程序造成的危害，也让.NET 程序具有跨平台的潜力。

你可能觉得很奇怪，微软为何发布出两个不同的.NET API 用来开发 Windows Client application？我相信 Windows Forms 仍然会在 Windows 应用程序开发上扮演重要的角色，特别是.NET 2.0 中，Windows Forms 的控件（control）和标准对话框（dialog box）都比 WPF 更好，而且在许多方面，Windows Forms 程序比较容易开发。我写了两本关于 Windows Forms 的书：*Programming Windows with C#*（Microsoft Press 2005）很厚，很完整，详细涵盖第一版的 Windows Forms；*Programming Microsoft Windows Forms*（Microsoft Press，2005）比较薄，容易阅读，将旧的内容略过不提，重点放在 2.0 版特色上，而即使没有旧版 Windows Forms 经验的读者，也可以通过本书很快地学会 Windows Forms。

虽然 WPF 尚未像 Windows Forms 这样提供这么多控件和标准对话框，但是未来却很有发展潜力。特别是，如果你需要做很多自定义控件和绘图，WPF 会是你需要的。

系统需求

想使用这本书，你将需要：

- Windows Vista、Windows XP+SP 2、Windows Server 2003
- Microsoft Visual C# 2005 Standard Edition, Visual Studio 2005 Professional Edition、Microsoft Visual C# 2005 Express Edition，或者更新版的 Visual Studio。（Visual C# 2005 Express Edition 不具有本书会用到的图库。）
- .NET Framework 3.0 （Windows Vista 已经内建）
- .NET Framework 3.0 SDK¹
- 能够运行上述软件的电脑

上面的许多软件都可以从下面的微软网址获得：

<http://msdn.microsoft.com/netframework/downloads/updates>

理论上，你不需要 Visual Studio。免费的 SDK 具有 MSBuild 命令列工具可以用来编译 WPF 应用程序。但是 Visual Studio 确实可以让 WPF 的开发更容易。有些人认为，以现在来说，Visual Studio 是不可或缺的工具。

CTP 软件

本书的内容符合 2006 年 6 月 CTP（社区技术预览）版本的 .NET Framework 3.0。我也希望能符合最终版本的 .NET Framework 3.0 和 SDK，但是如果真的有所改变的话，你可以在“本书的支持”一节中找到一个网址，这个网页会集合相关的“微软知识库”文章，以便你参考。

代码范例

本书所有的代码范例都可以从本书的原文网站下载，网址在：

<http://www.microsoft.com/mspress/companion/0-7356-1957-3>

本书的支持

我们作了最大的努力，希望本书内容精确。本书内容若有任何的修正，都会被放进微软知

¹ 如果要阅读 WPF 文档，并与 VisualStudio 集成，可以直接安装 Vista Platform SDK (XP 上也可以安装)<http://www.microsoft.com/downloads/details.aspx?familyid=C2B1E300-F358-4523-B479-F53D234CDCCE>。——审校者注

识库的文章中。

微软出版社对于其出版品和附属品内容提供相关的支持，网址在：

<http://www.microsoft.com/learning/support/books>

阅读与评论

关于本书，如果你有任何问题、评论、想法，或者你在上述网址找不到你的问题解答，你可以通过 e-mail 联系微软出版社：

mspinput@microsoft.com

或者通过传统邮件：

Microsoft Press

Attn: Applications = Code + Markup Editor

One Microsoft Way

Redmond, WA 98052-6399

请注意，关于“微软的软件产品”，不在这里的支持之列。

作者的网址

你也可以在我的网站，找到这本书的网址：

<http://www.charlespetzold.com/wpf>

关于我其他的书，以及博客、文章，都可以在我的网站上找到。

特别感谢

对于本书的成功出版，我想感谢我的代理人，Moore Literary Agency 的 Claudette Moore 和 Microsoft Press 的编辑们：Valerie Woolley（项目编辑），Kenn Scribner（技术编辑），以及其他列于本书版权页上的朋友们，感谢他们对本书的鼎力相助。

社区中的博客（大部分是微软员工）撰写的关于 WPF 开发的文章对本书作出了间接贡献，还有一些人通过 email 给予我指点，所有这些都使我受益匪浅。他们中给我帮助最大的是（按字母顺序排列）：Chris Anderson, Pete Blois, Aaron Cornelius@www.wiredprairie.us, Pable Fernicola, Jessica Fosler, Henry Hahn, Karsten Januszewski, Chuck Jazdzewski, Nick Kramer, Rahul Patil, Rob Relyea, Greg Schechter, Tim Sneath, David Teitlebaum, and Shawn Van Ness。

当然,即使拥有如此强大阵容的帮助,错误仍在所难免,我个人愿意对书中遗留的任何错误和纰漏负完全的责任。

最后,我要感谢 Deirdre, 我的字里行间,段落章节,乃至整本书里,都包含了她的智慧、耐心、宽容和爱。

Charles Petzold

New York City and Roscoe, New York

September 2005–June 2006

目 录

介绍	I
你的背景	I
本书	II
Windows 操作系统与编程	II
系统需求	IV
CTP 软件	IV
代码范例	IV
本书的支持	IV
阅读与评论	V
作者的网址	V
特别感谢	V
 第 1 部分 代码	1
第 1 章 应用程序与窗口	3
第 2 章 基本画刷	23
第 3 章 内容的概念	45
第 4 章 按钮与其他控件	65
第 5 章 Stack 与 Wrap	89
第 6 章 Dock 与 Grid	107
第 7 章 Canvas	131
第 8 章 Dependency Property	141
第 9 章 Routed 输入事件	157
第 10 章 自定义 Element	185

第 11 章 单一孩子的 Element.....	203
第 12 章 自定义面板.....	235
第 13 章 ListBox 选取.....	257
第 14 章 菜单层次结构.....	289
第 15 章 工具栏与状态栏.....	317
第 16 章 TreeView 与 ListView.....	341
第 17 章 打印与对话框.....	375
第 18 章 仿造记事本.....	413
 第 2 部分 Markup.....	455
第 19 章 XAML (和 Camel 押韵).....	457
第 20 章 Property 与 Attribute.....	487
第 21 章 资源.....	523
第 22 章 窗口、页面、导航.....	545
第 23 章 数据绑定.....	605
第 24 章 Style.....	639
第 25 章 模板.....	663
第 26 章 Data Entry、Data View.....	719
第 27 章 图形.....	759
第 28 章 几何和路径.....	783
第 29 章 图形变换.....	819
第 30 章 动画.....	859
第 31 章 位图、画刷、绘图.....	939
索引.....	977

第 1 部分

代码

第 1 章	应用程序与窗口	3
第 2 章	基本画刷	23
第 3 章	内容的概念	45
第 4 章	按钮与其他控件	65
第 5 章	Stack 与 Wrap	89
第 6 章	Dock 与 Grid	107
第 7 章	Canvas	131
第 8 章	Dependency Property	141
第 9 章	Routed 输入事件	157
第 10 章	自定义 Element	185
第 11 章	单一孩子的 Element	203
第 12 章	自定义面板	235
第 13 章	ListBox 选取	257
第 14 章	菜单层次结构	289
第 15 章	工具栏与状态栏	317
第 16 章	TreeView 与 ListView	341
第 17 章	打印与对话框	375
第 18 章	仿造记事本	413

