Broadview
www.broadview.com.cn

葵花宝典

葵花宝典——WPF自学手册

李响 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

葵花宝典——
WPF
自学手册

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

这本书最大的作用是让从未接触过 Microsoft Windows Presentation Foundation 的读者能够从初学到精通掌握，运用 WPF 来进行桌面开发，而且这本书的叙事风格和手法使得读者在经历掌握 Microsoft WPF 开发的整个过程是如此轻松快乐，在作者风趣调侃的语言当中不知不觉地学会 WPF 开发。本书从 WPF 的相关工具开始讲起，从 WPF 的体系结构、XAML、依赖属性、路由事件、命令等方面为读者奠定了一个坚实的学习基础。之后就切入了应用程序窗口、页面导航、布局等起步应用，能让读者及时地体会到学习的成就感和乐趣。接下来的控件、样式、数据绑定、二维图形、动画等相关内容则能够为读者的 WPF 技术提升到一个比较高的层次，如同插上翅膀，自由翱翔。

这本书对于 WPF 核心技术的原理、概念、技术、技巧与开发实践的讲述，是基于一位完全不懂 WPF 的菜鸟学习经历的，非常符合国内程序员 WPF 技术的初学路线，如果您想学习 Microsoft WPF 技术的话，那么这本书将是您的不二选择。

图书在版编目（CIP）数据

葵花宝典：WPF 自学手册 / 李响著. — 北京：电子工业出版社，2010.9
ISBN 978-7-121-11405-2

Ⅰ. ①葵… Ⅱ. ①李… Ⅲ. ①窗口软件，Windows Vista—用户界面—程序设计—技术手册 Ⅳ. ①TP316.7-62

中国版本图书馆 CIP 数据核字（2010）第 138130 号

责任编辑：孙学瑛
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
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：860×1092　1/16　印张：39.25　字数：897 千字
印　　次：2010 年 9 月第 1 次印刷
印　　数：4000 册　　定价：79.00 元（含 DVD 光盘 1 张）

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。

楔　子

江湖中盛传一部奇书，谁要依此书修炼武功就能“号令群雄，莫敢不从”，此书正是天下第一奇书——《葵花宝典》。

天下武功都是循序渐进，越到后来越难。但此书却违背常理，最难的是第一关。只要打通这一关，以后就遇山开山，遇水架桥，佛来斩佛，魔来斩魔。而这第一关则是“欲练神功，引刀自宫”，据说有人挥泪自宫之后成为了传说中的“东方不败”。但是也有不幸的人是在地摊上花上五毛钱买的盗版《葵花宝典》，翻开第 1 页是“欲练神功，必先自宫”，于是依言挥刀自宫。血还未止，忍疼翻到第 2 页，又是 8 个大字“即使自宫，未必成功”映入眼帘。此时后悔已经来不及了，只好继续翻到第 3 页，还是 8 个大字“如不自宫，也能成功”，偶的神啊！

废话不能再多说了，本书的头号主角就要闪亮登场。此人姓木，单名一个木，全名木木，呆若木鸡的木。大学二年级，好金庸，喜武侠；除此之外，对计算机编程有几分兴趣，也有几分傻气。学了一点 C++和 C#语言，有过一些 MFC 及 WinForm 编程的经验。闲时没事会在旧书市场闲逛，妄想淘得一本程序上的武功秘籍，一举成为 Bill Gates 那样的首富。“到那时候，嘿，嘿……”木木干笑了几声：“一定要开宝马车去校园门口的鸡蛋饼摊买鸡蛋饼，或者在街头坐着一个小马扎吃热干面，不过是百元大钞摞起来的小马扎而已……”

这一次还是在旧书市场闲逛，木木在众多的“贪官和他的情妇们”、“二十位都市男女情感口述”等此类杂志中发现了一本泛黄的书，封面上写着“葵花”两字。抽出来一看，才发现下面还有“宝典”二字，完整起来就是传说中的“葵花宝典”。木木不禁心扑腾扑腾乱跳，再一细看，下面还有一行小字“WPF 自学手册”。以前隐约听说过 WPF，应该是.NET 里面的新技术，看来这本书就是传说中程序的“武功秘笈”了。旧书都用不了多少钱的，木木丢给了老板五元钱后，就像做贼一样将书揣在怀里回去了。

回到家里，木木备好了剪刀（自宫而用），又放了一张女友靓照（看看也许会后悔）。闭上眼，颤抖地翻开了第 1 页。慢慢睁开眼，上面赫然写着几个大字“定价：3.9 元”。不禁大骂老板黑心，新书才 3.9 元，旧书居然卖给我 5 元。气愤之余不忘翻开书的第 2 页，果然有 8 个大字“欲练神功，必先自功”。木木早就听说过盗版葵花宝典的故事，于是继续往下翻，直到最后一页也没有“未必

成功、无需自宫”这样的字眼。宫还是不宫？木木左手拿剪刀，右手拿女友靓照，还真是难以抉择啊。

等等，木木又翻回第 2 页，揉揉眼再看了看，是成功的功，而不是自宫的宫。“唉，又是盗版书，‘宫’字还印错了。依法修炼，肯定不能成功了？”木木心想。于是将剪刀和靓照都抛去，拿着书躺在床上权当小杂志读读。谁知翻到第 3 页，该书却写到：“你宫了么？若挥刀自宫，请立刻拨打 120，兴许有救。此功非彼宫，学习 WPF 需要用功、用功再用功。因此欲练神功，必先自功。”木木腾地一下从床上跳起来，拿着书一屁股坐到书桌边，看来这的确是一本程序的武功秘笈，而且自功不自宫，相当划算喔。

诸位看到这儿，一定也获得这本武功秘籍了吧。那么请你们和木木一起走进 WPF 的世界，一定要“自功”才能学到真正的 WPF 喔^_^。

目 录

第一卷 程序江湖

第二卷 心 法

第三卷 小有所成

第四卷　峰回路转 夯实基础

第 13 章 样式和控件模板——听香水榭，千变阿朱

第 14 章 数据绑定——桃花岛软件公司人员管理系统之始末

第五卷 紫杉红烛

第 15 章 奇妙的二维图形世界——面壁

第六卷 华山之巅

第 19 章 互操作——“小无相功”

第 20 章 自定义控件——出手无招，何招可破

第七卷 志向无限大

第 21 章 木木能行，我也行

附录 A WPF 类图

第一卷

程序江湖

第 1 章

上路吧，WPF

沧海一声笑，滔滔两岸潮，浮沉随浪只记今朝。苍天笑，纷纷世上潮，谁负谁胜出天知晓。江山笑，烟雨遥，涛浪淘尽红尘俗事几多骄。清风笑，竟惹寂寥，豪情还剩了，一襟晚照……

——《笑傲江湖，词 曲：黄霑编》

程序也有江湖，回首且看有多少公司的兴衰，多少语言的纷争，多少平台的没落。

木木即将要接触的 Windows Presentation Foundation（WPF）是微软新一代的桌面平台技术，它是程序江湖的“锦带吴钩，把酒临风，潇湘之水，盈盈红烛的三生泪”。

本章内容如下。

（1）江湖前传。

（2）WPF 来了。

（3）接下来做什么。

1.1 江湖前传

1.1.1 微软的四重门

了解 WPF，先要了解其之前的江湖。了解江湖，先要了解在软件领域里横行 20 余年的江湖盟主——微软。在这 20 多年里，微软力主推行的编程语言和平台 API 大致经历了四重门[1]，如图 1-1 所示。

（1）第一重门（1985~1991）：C 语言和 Windows API。新生代的.Net 开发人员可能从未接触过注册一个窗口类，创建一个窗口或者在窗口过程函数中不断地 switch...case...case...。如果有时间的话，还是有必要了解 Windows 编程模型原理。为此笔者慎重推荐 Charles Peztold 所著的《Windows 程序设计》，无人能出其右[1]。

1 也可以参见笔者的博文《永远的窗口》一文，放在脚注位置，实在是不敢和 Charles Peztold 大师比肩。

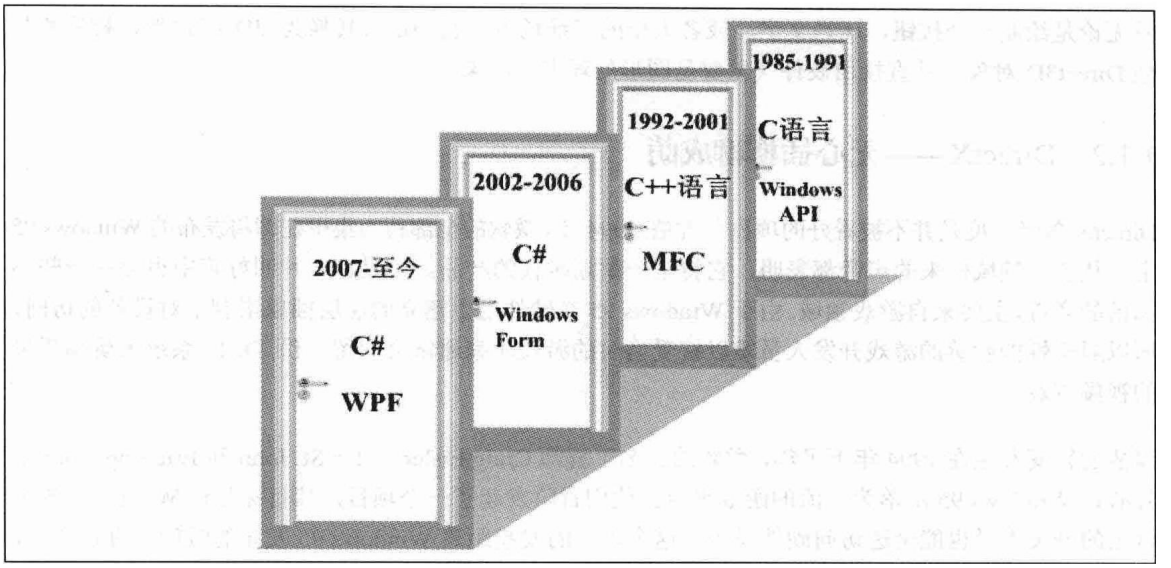

图 1-1　微软的四重门

（2）第二重门（1992~2001）：C++语言和 Microsoft Foundation Class（MFC）。尽管 MFC 一直以来被人诟病，但其至今却是历史上使用人数最多的 Windows 编程方法。如果想要掌握 MFC 的编程方法，那么一本大部头的《MFC 程序设计》可能是必需的；如果想要深入了解 MFC 的原理，那么 20 世纪风靡一时的侯大师的《深入浅出 MFC》可能作为案头书十分合适。当然还有一部《MFC Internals:Inside the Microsoft Foundation Class Architecture》（中文名是《深入解析 MFC》）是探求 MFC 源码的最佳书籍。

（3）第三重门（2002~2006）：C#和 Windows Form。C#是伴随微软的.NET 大战略产生新的语言，也是微软对抗 Java 的最有力武器。这个时期 IT 的技术焦点从桌面转向了网络，桌面开发显得有些冷清。Charles Peztold 此时推出了一本新书，即《C# Windows 程序设计》。但是这时已经不是“一家独大”，而是“城头变换大王旗，你方唱罢我登场”的百家争鸣时期。

（4）第四重门（2007 至今）：C#和 WPF。WPF 伴随着 Windows Vista 推出，虽然 Windows Vista 已经被逐渐证实只是一个过渡产品，但是 WPF 作为一个桌面开发平台并没有被摒弃。不过 Windows Vista 的失利，还是在一定程度上影响了 WPF 的推广。WPF 在 2006 年年底发布，但时至今日，仍未成燎原之势。如果希望在 Windows 平台上快速地打造够酷够眩的应用程序，则非 WPF 莫属。

无论是前三重门的哪一重门，Windows 开发人员从本质上都是在使用两种技术——User 和 GDI 子系统。一个 Windows 窗口实际上只是做两件事情，一是等待用户的输入，可能是鼠标、键盘、手写笔，以及单击菜单等，然后做出恰当的反应；二是在窗口上渲染图形。前者使用 User 子系统，后者使用 GDI 子系统，MFC 使用的正是这两个子系统。到了 Windows Form 阶段，微软在 GDI 的基础上封装成 GDI+，但是 GDI+绘制归根结底还是 GDI，因此 Windows Form 也是使用的这两个子系统。

第四重门的 WPF 几乎改变了原有的 Windows 技术，之所以是“几乎”，因为其仍然使用 User 子系统，但是绘制图形则交给了新的图形平台 DirectX，准确地说是 DirectX 的一部分 Direct3D。这意味

着无论是绘制一个按钮，还是文字，或者大型的三维场景，它们都会转换为 3D 三角形、材质和其他 Direct3D 对象，并直接由硬件（主要是图形处理卡）渲染。

1.1.2 DirectX——无心插柳柳成荫

DirectX 曾经一度是并不被看好的项目[3]，早在 1994 年，微软的全部精力集中在即将发布的 Windows 95 上。从各个领域传来的声音都表明，它将是一款划时代的产品。但是在一片叫好声中也存在一些不和谐的音符，主要来自游戏领域。由于 Windows 95 在硬件之上建立的层层抽象限制了对设备的访问，所以对于性能敏感的游戏开发人员难以接受当时的游戏玩家通常会保留一份 DOS 系统来玩高质量的视频游戏。

事情的转变发生在 1994 年下半年，微软的 3 名工程师 Craig Eisler、Alex St. John 和 Eric Engstrom 不甘心让 Windows 95 沦落为二流的游戏平台。他们冒险发起了一个项目，其目标是让 Windows 95 平台上的开发人员也能全速访问硬件设备。这个项目的发起距离 Windows 95 发布的时间只有几个月，因此被视为一个可有可无的项目。就在这样的窘况之下，DirectX 在 Windows 95 发布仅一个月后就以“Windows Games SDK”名义发布了。

最初的 DirectX 并没有和当时大名鼎鼎的 OpenGL 构成明显的竞争关系，二者分工明确。DirectX 对硬件要求低，对应普通家用 PC 配置，长于开发 2D 游戏；OpenGL 对硬件要求高，对应高端图形工作站，长于 3D 图形。从技术上来说，当时的微软正处在 COM 的热潮中，所有的组件都要 COM 化，DirectX 也不例外。DirectX 以 COM 组件的方式暴露 API，与 OpenGL 这种平民化的 C 风格 API 相比显得格外烦琐。

DirectX 3.0 发布不久，微软发布了一个更新的版本，内部版本号从 4.04.00.0068 增加到 4.04.00.0069。这样微小的更新一般人是无法察觉到的，但是非常具有讽刺意义。这次更新实际上是一次对整个产业具有决定影响的重大事件，因为在更新版本中出现了一个称为“Direct3D”的组件，正是这个组件揭开了 DirectX 与 OpenGL 正式对抗的序幕。在随后不到 5 年的时间里，Direct3D 风卷残云，成为主宰整个 PC 游戏图形工业的标准，彻底打破了 OpenGL 一家独大的局面。

DirectX 升级到 9 时，COM 渐渐落幕，时代已经进入到.Net。微软用托管代码包装了 DirectX（称为托管“DirectX”，与之对应的则是非托管 DirectX），以使其可以应用到基于.Net 架构的程序开发中。这时微软已经认识到可以将 DirectX 作为底层实现传统的界面设计，从而跳出过去 GDI/GDI+的限制。于是，WPF 来了……

1.2 WPF 来了

2001 年微软成立了一个新的团队，它有一个听起来简单，却很宏伟的使命。即创建一个统一的界面呈现平台，最终替代微软现有的 UI 平台，并能提供新的开发方案以满足用户日益增长的用户体验需求。起初这个团队将团队和项目都命名为“Aralon”，此即后来的 WPF。

1.2.1 七十二变

“嘘……可是我还是什么都没看到。”木木已经有些不耐烦了，随手把书扔在一边。谁知叮咣一声，从书的中间掉下来了一张光盘。“葵花宝典还配光盘啊。”木木知道，按照以往的常识，这个光盘应该是随书附带的源码。但是毕竟这是一本葵花宝典，说不定会有一些蹊跷。忍不住好奇，将光盘插入到电脑的光驱中。

“咔，暂停！”由于全书的示例均是在木木的笔记本电脑上编写的，因此不得不介绍这个头号道具，木木的笔记本电脑为 IBM T400，配置如下。

（1）CPU 为双核，主频 2.26 GHz。

（2）内存为 2 GB。

（3）显卡为 ATI Mobility Radeon HD 3400 Series。

（4）操作系统为 Windows XP SP3。

（5）编程工具为 VS2010、Reflector 和 XAMLPad。

木木满怀期待地以为有什么奇迹发生，光驱在“咔、咔、咔”地读盘，窗口弹了出来。唉，还是随书附带的源码，葵花宝典不过如此。既然打开了，那么还是看看吧。

源码第 1 章中的其他名称都很普通，唯有一个“看我七十二变”的文件夹听名字稍稍吸引人一些。于是木木点开这个文件夹，这是一个在 VS 2010 下运行的一个项目。运行后的界面非常简单，右侧有 6 个按钮，左侧则为空空荡荡，如图 1-2 所示。

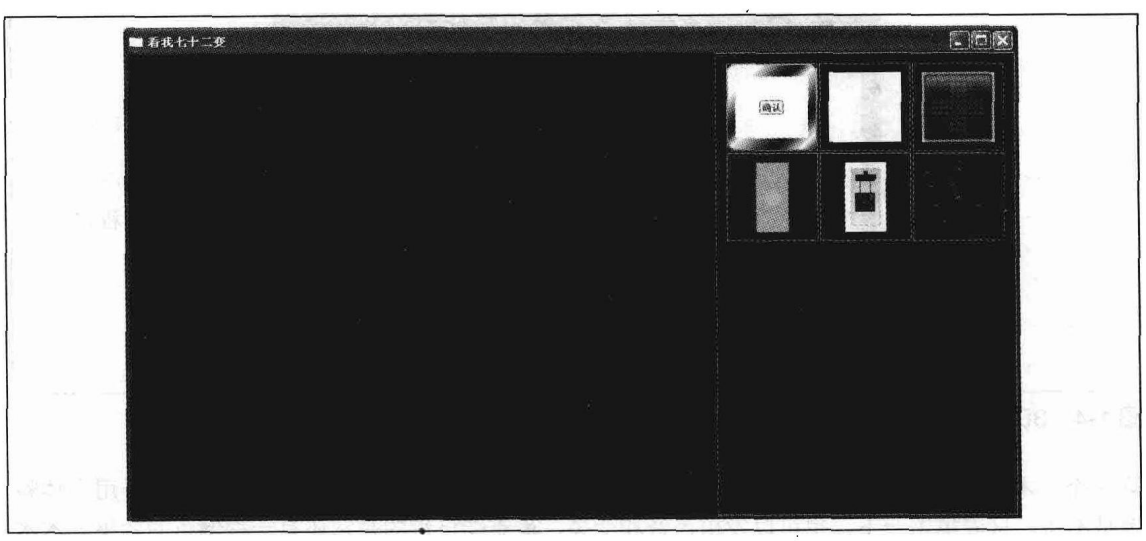

图 1-2 “看我七十二变”运行界面

木木单击右侧的第 1 个按钮，按钮旋转一下后弹出其形状。这是一个普通的按钮，不过分为两个页

面。即一个按钮及其代码实现，代码语言类似 XML，它的名称叫做 XAML，是微软为设计 UI 界面所提供的一套新语言，如图 1-3 所示。

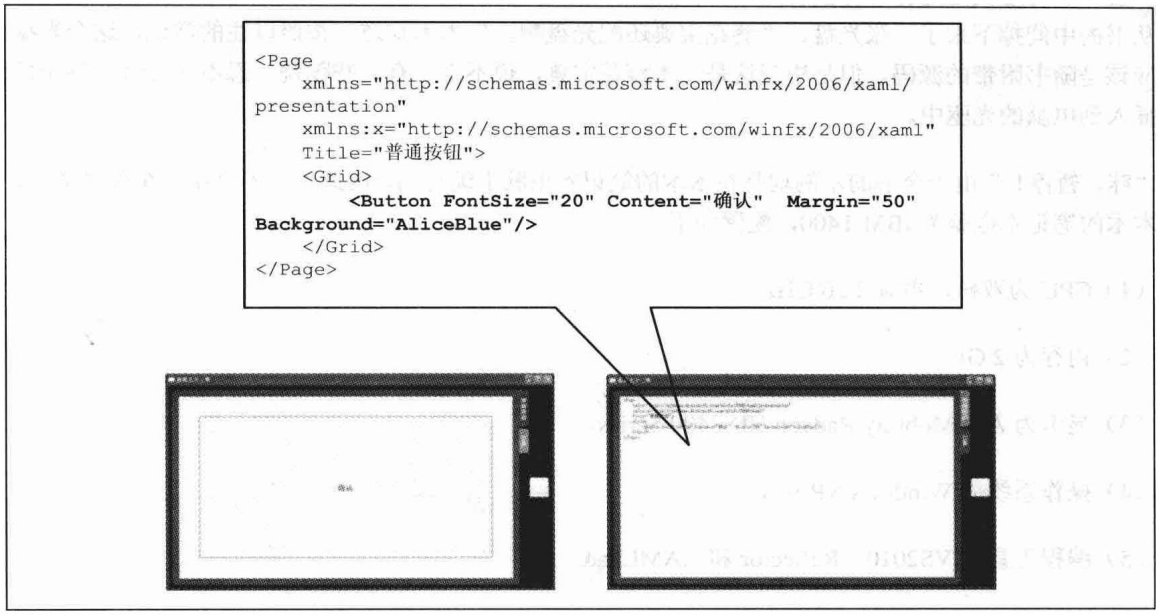

图 1-3　第 1 个按钮界面及其代码语言

“这个也太普通了吧”，木木顿时感到非常扫兴。原来听说 WPF 如何炫目，也不过如此。“反正既然点开了索性看完”木木心想，于是点开第 2 个按钮。没想到这是一个 3D 按钮，暗灰色调，如图 1-4 所示。单击后可以旋转，但是代码太长，木木也无心看下去。

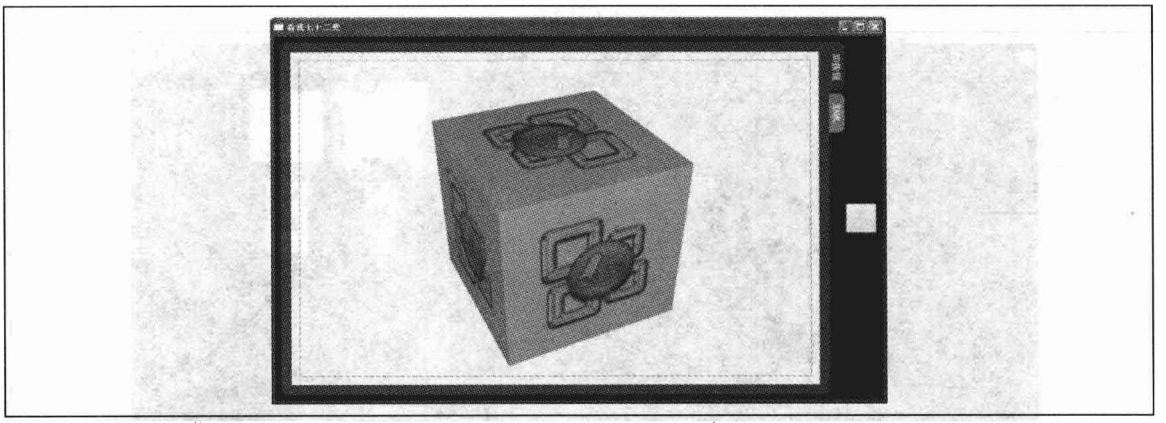

图 1-4　3D 按钮

第 3 个，木木开始有些期待了，是一个简单的十字形。但是鼠标放上后会突出放大，这样用户体验倒是不错。试着单击一下，也可以发出声音和变亮，最难能可贵的是它还有一种倒影的效果。木木以前接触过一些控件编程，知道在过去的 VC++程序中设计一个这样的按钮颇费工夫，心里开始隐隐觉得 WPF 还真是个好东东。