

# Windows

## 程序设计 (第6版 英文版)

[美] Charles Petzold 著

上册

Programming Windows  
Sixth Edition

- 使用C#和XAML  
编写Windows 8应用程序



Professional



Microsoft

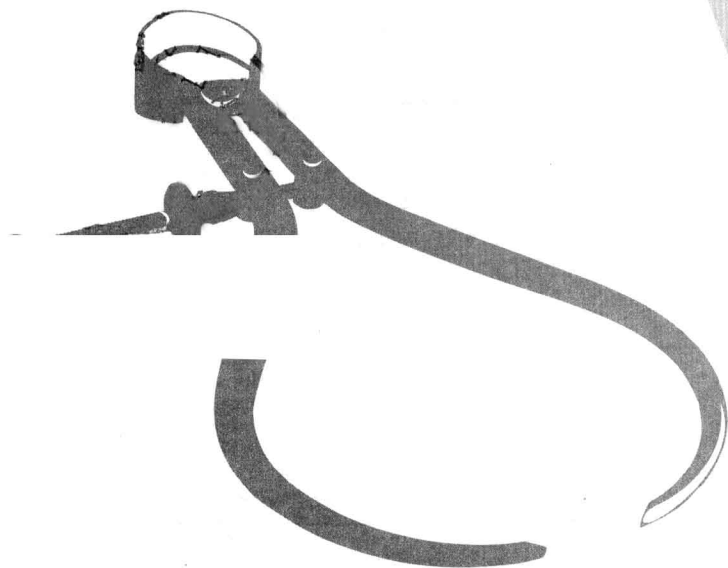
# Windows

## 程序设计(第6版 英文版)

[美] Charles Petzold 著

上册

Programming Windows  
Sixth Edition



人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

Windows程序设计 : 第6版 : 英文 / (美) 佩措尔德  
(Petzold, C.) 著. -- 北京 : 人民邮电出版社, 2013.10  
ISBN 978-7-115-32976-9

I. ①W… II. ①佩… III. ①Windows操作系统—程序  
设计—英文 IV. ①TP316.7

中国版本图书馆CIP数据核字 (2013) 第201789号

## 版 权 声 明

Charles Petzold, Programming Windows, Sixth Edition (ISBN: 9780735671768)  
Copyright 2013 by Posts & Telecom Press.

Original English language edition copyright ©2013 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A. All rights reserved.

本书中文简体字版由美国微软出版社授权人民邮电出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

- 
- ◆ 著 [美] Charles Petzold
  - 责任编辑 汪 振
  - 责任印制 程彦红 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京鑫正大印刷有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 71  
字数: 1684 千字 2013 年10月第 1 版  
印数: 1-2 500 册 2013 年10月北京第 1 次印刷  
著作权合同登记号 图字: 01-2013-2652 号
- 

定价: 148.00 元 (上、下册)

读者服务热线: (010)67132692 印装质量热线: (010)67129223  
反盗版热线: (010)67171154

# 内容提要

本书系经典图书《Windows 程序设计》的第 6 版，全书为最新的 Windows 8 操作系统进行了改写。作者在本书中细致地介绍了如何使用现有的 C#技能和 XAML 以及 Windows Runtime，创建全屏幕的 Windows 8 触摸应用程序，内容包括创建和初始化对象、管理和试验动态布局、创建 Windows Runtime 库、使用 MVVM 进行数据绑定和控制、使用及编写异步方法、动画过渡和创建 2D 和 3D 特效等，此外书中还介绍了操纵位图和富文本、使用 GPS 和方向传感器、调用 Win32 和 DirectX 函数、启用共享、搜索和打印等应用开发技能。

本书适合各个层级的 Windows 开发爱好者阅读使用。

# 作者简介

Charles Petzold 于 28 年前开始为 Windows 编程，当时他使用的是 Windows 1 Beta 版。他在杂志中撰写关于 Windows 编程的最早的一些文章，并在 1988 年出版了《Windows 程序设计》的第 1 版。过去 10 年间，他已经编写了 7 本关于 .NET 编程的书，包括最近出版的《*Programming Windows Phone 7*》（Microsoft 出版社，2010 年），目前正为《MSDN Magazine》撰写有关 Windows 8 DirectX 编程的专栏 DirectX Factor。Petzold 的书还包括《*Code: The Hidden Language of Computer Hardware and Software*》（Microsoft 出版社，1999），主要内容是对数字技术的独特探索，以及《*The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*》（Wiley 出版社，2008 年）。他的个人网站地址为 [www.charlespetzold.com](http://www.charlespetzold.com)。

# 前言

本书是《Windows 程序设计》的第 6 版，是一本介绍如何在 Microsoft Windows 8 操作系统上编写应用程序的指南。

要使用本书，读者需要一台安装了 Windows 8 操作系统的计算机，并在其中安装 Windows 8 开发工具和软件开发包 (SDK)，最方便的方式是从 Windows 8 开发者门户免费下载 Microsoft Visual Studio Express 2012 for Windows 8:

<http://msdn.microsoft.com/windows/apps>

选择页面中“开发”一栏下的“下载”链接，并接下来的网页中下载各种开发工具和 SDK。这个页面还提供了获得 Windows 8 开发者账户的相关信息，通过这个账户可以将新的应用上传到 Windows Store (Windows 应用商店)。

## Windows 8 的版本

---

对于大部分用户来说，会将 Windows 8 操作系统安装在与 Windows 7 操作系统同级的个人电脑上，这些机器基于 32 位或者 64 位的 Intel x86 微处理器。Windows 8 操作系统有一个直接叫做“Windows 8”的常规版本，以及一个具有吸引技术爱好者和专业人士的附加特性的“Windows 8 Pro”版。

“Windows 8”和“Windows 8 Pro”都能运行两类程序：

- 桌面应用程序；
- 新的 Windows 8 应用程序，常被称作“Windows Store 应用程序”。

桌面应用程序即是目前可以运行于 Windows 7 操作系统下的常规 Windows 程序，通过 Windows 应用编程接口（即 Win32 API）与操作系统交互。为了支持这些桌面应用，Windows 8 依旧包含了熟悉的 Windows 桌面系统。

新的 Windows Store 应用程序代表着与传统 Windows 的决裂。这些程序一般运行于全屏模式，也可以由两个程序通过分屏模式共享屏幕，其中许多程序为了方便触摸屏和平板电脑的使用作了优化。这些应用程序只能从 Microsoft 的官方应用商店中购买和安装（开发人员则直接可以在 Visual Studio 中部署和测试应用）。

除了在 x86 处理器上运行的 Windows 8 版本之外，还有一个运行于 ARM 处理器的 Windows 8 版本，这种处理器常见于价格低廉的平板电脑和其他移动设备中。Windows 8 操作系统的这个版本叫做“Windows RT”，它只能安装在上述这类机器上面。第一批运行 Windows RT 操作系统的机器是 Microsoft Surface 的首发版本。

除了某些预装的桌面应用之外，Windows RT 只能运行 Windows Store 应用程序，

用户不能在 Windows RT 操作系统中运行现有的 Windows 7 应用程序，不能在 Windows RT 上运行 Visual Studio，也不能在 Windows RT 上开发 Windows 8 应用。

Windows 8 的用户界面采用了新的设计风格，这些风格将会在具体的 Windows Store 应用程序中得到体现。其设计灵感来源于城市环境中的各种标志，强调内容重于程序，特征是使用朴素的字体、清爽的风格、基于磁贴（Tile）的界面和过渡动画。

许多开发人员已经在 Windows Phone 7 操作系统中见过了这种设计风格，因此他们都很好奇，想看看微软公司将会如何把这种风格加入到大小不同的各种终端中去。在过去的很长一段时间里，微软公司一直试图将传统的 Windows 桌面设计应用到小屏幕的终端设备之中，如手持电脑和智能手机，而现在则恰恰相反，一款原本是为手机开发的用户界面已经移植到了平板电脑和桌面电脑上。

这种全新系统的一个重要特征是强调多点触摸，这种技术已经显著地改变了人和电脑之间的关系。实际上，“多点触摸”一词现在已经过时了，因为所有新的触摸设备都能够响应多个手指的触摸，简单的“触摸”一词就已经足够。Windows 8 应用程序所使用的新编程接口中，很多都采用统一的风格处理触摸、鼠标和手写笔输入，使得应用程序可以自动接受 3 种输入设备。

## 本书重点

---

本书只关注 Windows Store 应用程序的编写，关于 Win32 桌面应用程序的编写已经有大量的相关书籍，包括我偶尔作为 Win32 API 和桌面应用参考书的《Windows 程序设计》的第 5 版。本书完全是关于编写新的 Windows 8 应用程序的。

为了编写这些应用，微软公司推出了一种新的面向对象 API——Windows Runtime，即 WinRT（不要和运行于 ARM 处理器上的 Windows RT 操作系统混淆）。在内部，Windows Runtime 基于 COM（Component Object Model，组件对象模型），其接口通过/Windows/System32/WinMetadata 目录中扩展名为.winmd 的元数据文件输出；从外部看，Windows Runtime 是完全面向对象的。

从应用程序的编程人员的角度看，Windows Runtime 和 Silverlight 很类似，但是从内部看，它不是用于 Silverlight 编程人员的托管 API，两者之间的最直接差异包括命名空间的名称，以 System.Windows 开始的 Silverlight 命名空间被替换为以 Windows.UI.Xaml 开始的命名空间。

大部分的 Windows 8 应用程序不仅可以通过代码构建，还可以使用标记语言，既可以使用具备工业标准的超文本标记语言（HTML），也可以使用微软公司的可扩展应用程序标记语言（eXtensible Application Markup Language，XAML）。将应用的代码和标记分离的一个好处是，将应用开发中编程人员和设计人员的工作分离。

目前，编写 Windows 8 应用程序有 3 种选择，每种都包括一种编程语言和一种标记语言：

- C++ 和 XAML;
- C# 或 Visual Basic 和 XAML;
- JavaScript 和 HTML5。

对于上述所有选择, Windows Runtime 都是相同的, 但是 Windows Runtime 对于每种特定的语言都会补充一个不同的编程接口。虽然你不能在单个应用程序中混合使用不同的语言, 但是你可以创建具有可供任何其他 Windows 8 语言访问的.winmd 文件的库(称作 Windows 运行时组件)。

C++编程人员使用它的一种变种——C++ with Component Extensions (C++组件扩展, 或称 C++/CX), 这使得该语言能够更好地使用 WinRT。C++编程人员还能够直接访问 Win32、COM API 和 DirectX 的某个子集, 从而将 C++程序编译为本机代码。

使用托管语言(如 C#或者 Visual Basic .NET)的编程人员会发现 WinRT 是非常熟悉的领域。以这些语言编写的 Windows 8 应用程序无法像 C++编程人员那样轻松地访问 Win32、COM 或者 DirectX API, 但这并非完全不可能, 第 15 章中的一些示例程序展示了.NET 用于执行底层任务的原理。

对于 JavaScript, Windows Runtime 得到了 Windows Library for JavaScript (即 WinJS) 的补充, 这个库为 Windows 8 应用提供了一些系统级特性。

在深思熟虑(和思想斗争)之后, 我决定将本书的讲解重点放在 C# 和 XAML 这一选项之上。10 年以来, 我已经认识到托管语言在开发和调试上的优势, 而且在我看来, C#是最适合 Windows Runtime 的语言。我希望 C++编程人员能够觉得 C#代码容易理解, 并从本书中获益。

我也相信, 一本书专注于介绍某一种语言, 会比面面俱到地介绍多种语言要更有价值。毫无疑问的是, 还会有很多其他的 Windows 8 书籍向读者介绍如何使用其他语言编写 Windows 8 应用程序。

尽管如此, 我还是很高兴见到新近出现的有关 C++和原生代码在编写高性能应用时孰优孰劣的辩论。没有一种工具能够在所有问题上都有最好的表现, 在未来, 我将在自己的博客和 MSDN 杂志的页面上更多地探讨 Windows 8 的 C++和 DirectX 开发。作为一个开始, 本书附带的内容中包含了所有转换为 C++的程序示例。

## 本书使用指南

---

编写本书时, 我假定读者熟悉 C#语言, 如果你不熟悉, 可以用 C#教程作为本书的补充, 如果你有 C 或 C++的背景, 阅读我的免费在线书籍《.NET Book Zero: What the C or C++ Programmer Needs to Know About C# and the .NET Framework》可能就足够了, 这本书的 PDF 或者 XPS 格式版本可以在 [www.charlespetzold.com/dotnet](http://www.charlespetzold.com/dotnet) 找到。

我还假定读者了解 XML (eXtensible Markup Language, 可扩展标记语言) 的基本语法, 因为 XAML 是基于 XML 构建的。但是我假定读者并不熟悉 XAML 或者任

何基于 XAML 的编程接口。

这是一本 API 书而不是一本工具书，在这本书中我使用的唯一编程工具就是 Microsoft Visual Studio Express 2012 for Windows 8（我通常直接称其为 Visual Studio）。

标记语言通常比编程代码更容易加工，有些编程人员甚至认为 XAML 等标记应该完全由机器生成。Visual Studio 有一个内建的交互式 XAML 设计器，可以将控件拖放到页面上，许多编程人员已经了解并喜欢使用 Microsoft Expression Blend 为他们的应用程序生成复杂的 XAML（Expression Blend 包含在前面所提到的开发工具和 SDK 免费下载中）。

这些设计工具对于有经验的编程人员来说很棒，但是我认为刚进入这一环境的编程人员最好学习手工编写 XAML 的方法。这也是我处理本书中 XAML 的方式，第 8 章中的 XAML Cruncher 工具与这种思想非常一致，它让读者输入 XAML，并且交互式地查看生成的对象，但是不会试图为读者编写 XAML。

另一方面，有些编程人员已经很熟悉 XAML 的使用，以至于忘了如何用代码创建和初始化某些对象。我认为这两种技能都很重要，在后续的章节中，我常常会说明如何用代码和标记进行类似的工作。

在我开始编写本书的时候，我曾经设想过好几种不同的 Windows Runtime 教程写法，其中之一是从相当低级的图形和用户输入开始，演示构建控件的方法，然后描述已经构建好的控件。

现在我已经改变了选择，本书将从我认为对于主流编程人员最重要的技巧开始，在一个应用程序中汇编预定义控件，并将它们与代码和数据链接。这是本书第一部分 12 章的焦点，第一部分的目标是帮助读者理解 Visual Studio 在其支持的各种项目模板中生成的代码和标记。

第二部分介绍更底层和难解的任务，如触摸、位图、富文本、打印以及方向传感器及 GPS 传感器的使用。

---

## 源代码

学习新的 API 和学习如何打篮球、演奏双簧管是一样的，仅仅观看别人是无法完全理解的，必须亲身参与。本书中的源代码可以通过以下链接下载：

<http://shop.oreilly.com/product/0790145369079.do>

但是，自己输入这些代码学习效果可能更好。

---

## 我的配置

为了编写本书，我使用了 2011 年 9 月 Microsoft Build 会议上发放给参会者的三星 700T 特别版平板电脑（它有时被称作 Build Tablet）。这台机器使用一个 1.6GHz

的 Intel Core i5 处理器，具有 4GB RAM 和 64GB 硬盘，其屏幕（本书中的大部分屏幕截图来自于它）支持 8 点触摸，分辨率为 1366 像素×768 像素，这是分屏视图支持的最低分辨率。

尽管 Build Tablet 发放时已经安装了 Windows 8 开发人员预览版，我在 2012 年 3 月还是将其替换为消费者预览版（Build 8250），同年 6 月又替换为发行预览版（Build 8400），最终安装了 Windows 8 专业正式版。除了测试方向传感器之外，其他时间我一般将平板电脑通过扩展坞与一台配有 1920 像素×1080 像素的 HDMI 显示器和外接键盘及鼠标一起使用。

在 Microsoft Surface 刚刚推出时，我订购了一台来测试我的应用程序，为了在 Surface 上部署和调试应用程序，我使用了 Tim Heuer 在其博客中讨论的技术：

<http://timheuer.com/blog/archive/2012/10/26/remote-debugging-windows-store-apps-on-surface-arm-devices.aspx>

这一技术在主题为“Running Windows Store apps on a remote machine”的文档中有更加正式的描述：

<http://msdn.microsoft.com/en-us/library/hh441469.aspx>

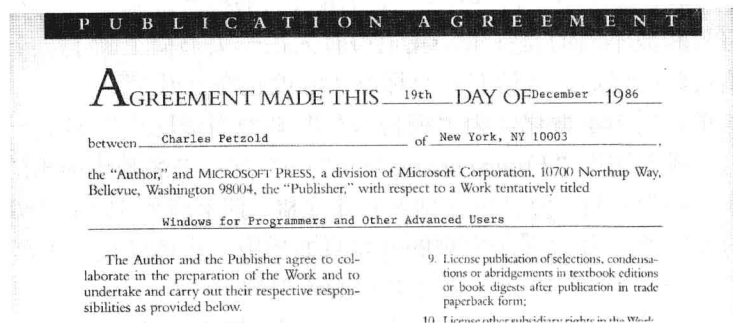
在测试需要访问方向传感器的程序时，Surface 非常关键。

然而，对于大部分情况，我仍然使用扩展坞中的 Build Tablet。外接的键盘、鼠标和显示器使我可以按照自己的习惯使用 Visual Studio 和 Microsoft Word，同时在平板电脑的触摸屏上运行 Windows 8 程序。这是一个很好的开发环境，特别是与我在编写本书第 1 版时使用的配置相比。

不过，编写本书第 1 版已经是 25 年前的事了。

## 过去版本留下的遗产

这是《Windows 程序设计》的第 6 版，它的最初构想是由 Microsoft 出版社在 1986 年秋季提出的。我参与这一项目是因为当时我正在为《Microsoft Systems Journal》（《MSDN Magazine》的前身）撰写文章。我仍然能够回想起看到第一份合同时自己的兴奋心情。



这份合同最有趣的部分应该是它的第一页。

## II MANUSCRIPT

The Author agrees to prepare and submit one (1) clean copy and one (1) ASCII readable diskette of the final manuscript of the Work, equivalent to approximately 100,000 words, not later than April 30, 1987

the due date. (A full manuscript page of text consists of approximately 250 words.) The Author's final manuscript shall be in double-spaced typescript or its equivalent, satisfactory to the Publisher in organization, form, content, and style and accompanied by appropriate illustrative material, table of contents, tables, bibliography, and instructional aids ready for reproduction.

合同中提到的“Typescript”（打字文件）意味着，书稿至少应该类似于打字机的输出。说明中提到，使用等宽字体的双倍行距手稿每页大约为 250 个单词。出版图书的每页大约为 400 个单词，因此 Microsoft 出版社显然不希望得到一本冗长的书籍。

为了这本书，我使用了一部 IBM PC/AT，配置为一个 8MHz 的 80286 处理器、512KB 内存和 2 个 30MB 硬盘。显示器使用了 IBM 增强型图形适配器，最大分辨率为 640×350/16 色，我用 Windows 1（1985 年 11 月推出）编写了初期的几章，但是很快就有了 Window 2 的 Beta 版本。

那些年，Windows 程序的编写和编译需要在 Windows 之外的 MS-DOS 中进行，我用 WordStar 3.3 编辑源代码，同时也用它编写文章。在 MS-DOS 命令行中，用户可以运行 Microsoft C 编译器，然后通过自己编写的程序启动 Windows 进行测试，下一个编辑-编译-运行周期必须退出 Windows 返回 MS-DOS。

1987 年，我更加投入这本书的编写工作，这占用了我的大部分时间。熬夜的情况越来越严重，当时我没有电视，总是不停地收听本地广播电台 WNYC-FM 的古典音乐节目和全国公共广播电台的节目。在那段时间里，我已经形成了生物钟，在《Morning Edition》节目之后上床休息，而在《All Things Considered》节目开始时醒来。

按照合同规定，我向 Microsoft 出版社寄出磁盘和纸质的书稿（当然，我们都有电子邮件，但是那时的电子邮件不支持附件）。经过编辑的书稿通过邮寄方式返回给我，上面满是校对符号和数不清的便签条。我记得有人在一页书稿上画了一个温度计，表示我提交手稿页数的增加，旁边有一个标题“温度正在上升！”。

就这样，该书的重点发生了变化。为“编程人员和其他高级用户”编写一本书被证明是一个伪命题。我不知道“*Programming Windows*”这一书名是谁提出的。

合同规定的完成日期为 4 月，但是我直到 8 月才完稿。该书直到 1988 年初才得以出版，最终有大约 850 页。如果采用常规的书籍页面排版（也就是说，不算程序清单或者框图），单词数量大约有 40 万，而非合同所约定的 10 万。

第 1 版的封面上将该书描述为“微软公司为 MS-DOS 表示层管理程序——Windows

2.0 及 Windows/386 提供的编程指南”，使用“表示层管理程序”(Presentation Manager)一词提醒我们，当时 Windows 和 OS/2 Presentation Manager 被认为应该作为两个不同操作系统的类似环境和平共处。

《Windows 程序设计》的第 1 版没有得到编程社区太多的注意。当 MS-DOS 编程人员逐渐意识到自己必须学习 Windows 新环境相关知识时，他们大部分开始从第 2 版（1990 年出版，重点关注 Windows 3）和第 3 版（1992 年出版，Windows 3.1）中寻求帮助。

当 Windows API 从 16 位转向 32 位时，《Windows 程序设计》以第 4 版（1996 年，Windows 95）和第 5 版（1998 年，Windows 98）做出了回应。尽管第 5 版还在印刷，但是从读者那里收到的电子邮件来看，该书在印度和中国最受欢迎。

从第 1 版到第 5 版，我使用 C 编程语言。在第 3 版和第 4 版之间，我的好朋友 Jeff Prosise 说他打算编写《MFC Windows 程序设计》一书，我对此无动于衷。我不太关心 Microsoft Foundation Class，对我来说这只是 Windows API 相当轻量级的包装器，我也没有深入钻研 C++。

随着时间的推移，《Windows 程序设计》被誉为编程人员跨越程序代码和操作系统之间障碍、进行硬件编程的宝典。

但是对我来说，《Windows 程序设计》的早期版本绝非如此。在那些日子里，硬件编程需要汇编语言编码，直接将字符写入视频显示内存，且文件 I/O 只能采用 MS-DOS。相反，Windows 编程涉及的是高级语言，图形完全未经加速，访问硬件只能通过笨重的 API 层和设备驱动程序。

从 MS-DOS 转向 Windows 意味着有意放弃速度和效率来换取其他的好处。但是有哪些好处呢？许多老练的编程人员无法抓住重点。图形？图像？颜色？有趣的字体？鼠标？那些都不是计算机的全部！怀疑论者将其称作 WIMP（窗口-图标-菜单-指针）接口，但这不是人们选择使用这种环境以及为其编程的意义所在。

如果等待足够长的时间，高级语言就会变成低级语言，多层接口似乎缩减（至少从行话上是如此）为一个原生 API。现在有些 C 和 C++ 编程人员拒绝接受 C# 等托管语言的效能，Windows 再次引起了激烈的辩论。Windows 8 操作系统是 Windows 操作系统自 1985 年第一次发行以来最具革命性的更新，但是许多过去的 Windows 用户对将为智能手机和平板电脑量身定做的触摸屏界面引入主流桌面的想法表示怀疑，当他们无法找到熟悉的功能时就会心生抱怨。

我认为，只有激动人心和充满争议的新 Windows 用户界面和适合其现代化方向的 API 和编程语言，才能使《Windows 程序设计》从“半退休状态”中重生。

## 展望未来

---

我觉得 Windows 8 操作系统将在一段时间内主导我的编程生涯，这意味着，我

可能发表有关 Windows 8 编程各个方面的博客。读者可以在 [www.charlespetzold.com](http://www.charlespetzold.com) 访问我的博客、订阅 RSS 摘要。

我始终乐于解决棘手的编程问题，并发表相关的帖子，所以如果读者有需要我关注和解决的与 Windows 8 编程相关的问题，可以通过 [cp@charlespetzold.com](mailto:cp@charlespetzold.com) 联系我。

从《MSDN Magazine》2013 年 1 月号起，我将撰写一个每月专栏《DirectX Factor》，专门关注如何在 Windows 8 操作系统和 Windows Phone 8 应用程序中使用 DirectX。《MSDN Magazine》可以在 <http://msdn.microsoft.com/magazine> 免费阅读。

## 致谢

---

本书之所以能出版完全是因为 Microsoft 出版社的 Ben Ryan 和 Devon Musgrave，他们开发了一种有趣的方式，在编程社区中发布预览版，并销售纸质的最终版图书。

Devon 和我的技术审阅者 Marc Young 的部分工作是帮助我修改文字和代码中的错误，在此我要感谢他们。

我还要感谢 Andrew Whitechapel 为我提供的关于 C++ 样板代码的反馈；感谢 Brent Rector 发来的电子邮件为触摸相关问题以及 IBuffer 背景资料提供的关键解决方案；感谢 Robert Levy 对触摸的见解；感谢 Jeff Prosise 在我困惑时总是能给出正确的解答；感谢 Larry Smith 对我的文字进行勘误；感谢 Admiral 促使我竭尽全力使本书成为对 C++ 编程人员有用的书籍。

书中仍然存在的错误都是我自己的问题。后面有一个电子邮件地址，读者可以通过它向出版社报告错误，我也将在我的网站 ([www.charlespetzold.com/pw6](http://www.charlespetzold.com/pw6)) 上对严重的问题作出说明。

最后，我要感谢我的妻子 Deirdre Sinnott，感谢她的爱和支持，并为了我的写作对我们的生活进行必要的调整。

Charles Petzold

2012 年 12 月 31 日于纽约

## 勘误和图书支持

---

我们已经尽一切努力确保本书及其附带内容准确。从本书出版以来报告的任何错误都将列在 [oreilly.com](http://oreilly.com) 的 Microsoft 出版社网站上。在 <http://microsoftpress.oreilly.com> 上搜索本书，然后单击“View/Submit Errata”链接即可。如果你发现一个还没有列出的错误，可以通过同一个页面向我们报告。如果需要额外的支持，可以发送邮件至 Microsoft 出版社图书支持部门：[mspinput@microsoft.com](mailto:mspinput@microsoft.com)。

请注意，上述地址中不提供 Microsoft 软件的产品支持。

## 期待反馈

---

在 Microsoft 出版社，你的满意是我们的头等大事，你的反馈是我们最宝贵的财富。请在如下地址告诉我们你对本书的意见：

<http://aka.ms/tellpress>

反馈表很简短，我们会阅读所有评论和想法。感谢你所提的意见。

## 联系方式

---

让我们持续交流下去！我们的 Twitter 地址为 <http://twitter.com/MicrosoftPress>。

# Contents at a glance

| <b>PART I</b>  | <b>ELEMENTALS</b>          |          |
|----------------|----------------------------|----------|
| CHAPTER 1      | Markup and Code            | 3        |
| CHAPTER 2      | XAML Syntax                | 31       |
| CHAPTER 3      | Basic Event Handling       | 69       |
| CHAPTER 4      | Presentation with Panels   | 97       |
| CHAPTER 5      | Control Interaction        | 139      |
| CHAPTER 6      | WinRT and MVVM             | 193      |
| CHAPTER 7      | Asynchronicity             | 221      |
| CHAPTER 8      | App Bars and Popups        | 261      |
| CHAPTER 9      | Animation                  | 329      |
| CHAPTER 10     | Transforms                 | 377      |
| CHAPTER 11     | The Three Templates        | 449      |
| CHAPTER 12     | Pages and Navigation       | 539      |
| <br>           |                            |          |
| <b>PART II</b> | <b>SPECIALTIES</b>         |          |
| CHAPTER 13     | Touch, Etc.                | 615      |
| CHAPTER 14     | Bitmaps                    | 683      |
| CHAPTER 15     | Going Native               | 779      |
| CHAPTER 16     | Rich Text                  | 845      |
| CHAPTER 17     | Share and Print            | 893      |
| CHAPTER 18     | Sensors and GPS            | 953      |
| CHAPTER 19     | Pen (Also Known as Stylus) | 1013     |
|                | <br><i>Index</i>           | <br>1057 |

# Table of Contents

## PART I      ELEMENTALS

---

|                  |                                             |           |
|------------------|---------------------------------------------|-----------|
| <b>Chapter 1</b> | <b>Markup and Code</b>                      | <b>3</b>  |
|                  | The First Project .....                     | 3         |
|                  | Graphical Greetings .....                   | 9         |
|                  | Variations in Text .....                    | 13        |
|                  | Media As Well .....                         | 22        |
|                  | The Code Alternatives .....                 | 23        |
|                  | Images in Code .....                        | 27        |
|                  | Not Even a Page .....                       | 29        |
| <br>             |                                             |           |
| <b>Chapter 2</b> | <b>XAML Syntax</b>                          | <b>31</b> |
|                  | The Gradient Brush in Code .....            | 31        |
|                  | Property Element Syntax .....               | 34        |
|                  | Content Properties .....                    | 37        |
|                  | The <i>TextBlock</i> Content Property ..... | 41        |
|                  | Sharing Brushes (and Other Resources) ..... | 43        |
|                  | Resources Are Shared .....                  | 47        |
|                  | Exploring Vector Graphics .....             | 48        |

---

**What do you think of this book? We want to hear from you!**

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

[microsoft.com/learning/booksurvey](http://microsoft.com/learning/booksurvey)

|                                                 |           |
|-------------------------------------------------|-----------|
| Stretching with <i>Viewbox</i> .....            | 58        |
| Styles .....                                    | 60        |
| A Taste of Data Binding .....                   | 66        |
| <b>Chapter 3 Basic Event Handling</b> .....     | <b>69</b> |
| The <i>Tapped</i> Event .....                   | 69        |
| Routed Event Handling .....                     | 72        |
| Overriding the <i>Handled</i> Setting .....     | 78        |
| Input, Alignment, and Backgrounds .....         | 80        |
| Size and Orientation Changes .....              | 83        |
| Bindings to <i>Run</i> ? .....                  | 87        |
| Timers and Animation .....                      | 89        |
| <b>Chapter 4 Presentation with Panels</b> ..... | <b>97</b> |
| The <i>Border</i> Element .....                 | 97        |
| Rectangle and Ellipse .....                     | 101       |
| The <i>StackPanel</i> .....                     | 103       |
| Horizontal Stacks .....                         | 106       |
| WhatSize with Bindings (and a Converter) .....  | 108       |
| The <i>ScrollViewer</i> Solution .....          | 112       |
| Layout Weirdness or Normalcy? .....             | 118       |
| Making an E-Book .....                          | 119       |
| Fancier <i>StackPanel</i> Items .....           | 122       |
| Deriving from <i>UserControl</i> .....          | 124       |
| Creating Windows Runtime Libraries .....        | 127       |
| The Wrap Alternative .....                      | 130       |
| The <i>Canvas</i> and Attached Properties ..... | 132       |
| The Z-Index .....                               | 136       |
| Canvas Weirdness .....                          | 137       |