资深移动平台开发技术专家林政原创经典！国内首本系统论述Windows Phone 8.1 UI控件编程的原创经典作品！

微软（中国）有限公司技术顾问总监夏鹏作序！

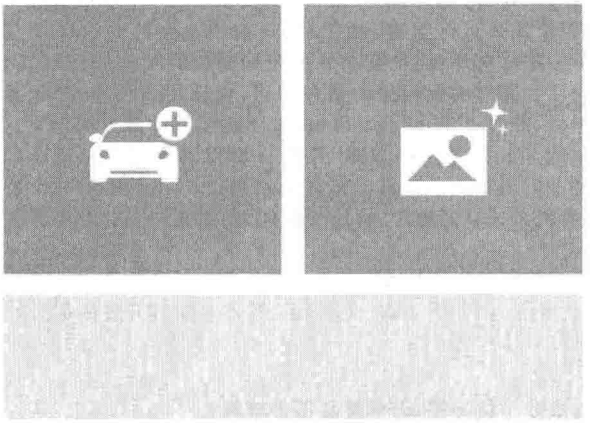
繁体中文版台湾地区同步发行！

深入理解 Windows Phone 8.1 UI控件编程

UI Programming of Windows Phone 8.1: a Rich Understanding

林政 著
Lin Zheng

清華大學出版社



深入理解
Windows Phone 8.1
UI控件编程

UI Programming of Windows Phone 8.1: a Rich Understanding

林政 著
Lin Zheng

清華大學出版社

内 容 简 介

本书深入地论述了 Windows Phone 8.1 的 UI 控件编程的相关技术知识。本书核心是引导读者掌握解决问题的思路，在介绍原理的同时，给出了大量应用实例来帮助理解和实践。本书从程序界面开始，解剖了 XAML 页面的生成原理及其运行原理，然后对 UI 编程常用的知识样式、模板、布局原理进行讲解。布局原理并不是讲解简单的控件布局，而是重点分析布局面板的工作原理以及如何去自定义实现自己的布局规则。书中介绍了与动画图形编程相关的诸多知识，包括图形绘图、图表编程、变换效果、三维效果、动画编程等。在动画编程里，分析了 Windows Phone 8.1 的所有动画解决方案，并介绍了如何选择最优的实现方案及如何编写复杂的动画效果。在掌握 UI 控件编程的原理的基础上，本书还介绍了 Expression Blend 工具的使用，尤其是如何借助这个工具去高效地实现绘图和制作动画。最后，本书介绍了控件和列表编程的相关知识，包括解剖系统空间原理、自定义控件、高效的列表的解决方案和 Toolkit 相关控件技术原理的研究等内容。

本书配套提供了书中实例源代码，最大限度满足读者高效学习和快速动手实践的需要。

本书内容针对性强、讲解深入、实例丰富、注重理论学习与实践开发的配合，非常适合想要在 Windows Phone 领域上进行更加深入学习的读者。

图书在版编目(CIP)数据

深入理解 Windows Phone 8.1 UI 控件编程/林政著. —北京：清华大学出版社，2014
(清华开发者书库)
ISBN 978-7-302-35875-6

Ⅰ. ①深… Ⅱ. ①林… Ⅲ. ①移动电话机－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2014)第 061764 号

责任编辑：盛东亮
封面设计：李召霞
责任校对：白　蕾
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：186mm×240mm　　**印　　张**：21.25　　**字　　数**：479 千字
版　　次：2014 年 5 月第 1 版　　**印　　次**：2014 年 5 月第1 次印刷
印　　数：1～3000
定　　价：79.00 元

产品编号：056809-01

序
PREFACE

微软公司1975年成立，微软的童年可谓光芒四射，BASIC语言、Dos、Windows 3.1等不断地惊艳当时高速发展的信息时代。在他成长到20岁时（也就是1995年），发布了Windows 95，随后的几年，他达到一个无人可及的顶峰，那些年他几乎统治了整个IT界和几乎每个人的生活。又过了19年之后，2014年他迎来了新的掌门人——纳德拉（Satya Nadella），面对世界的新技术、新公司、新生活方式的挑战，感受着来自各方面的压力，他为公司提出了全新的策略，简言之就是“移动为先，云为先”。他同时指出：“我坚信，在未来十年，计算将无处不在，智能将触手可及。软件的进化与新式硬件的普及会在其中起到媒介作用，目前我们在工作和生活中从事和体验的很多内容都将实现数字化，甚至整个世界也是如此。可联网设备的数量快速增长、云环境所能提供的海量计算资源，大数据的洞察力，以及机器学习所获得的智能，诸多因素让这一切变为可能。”

接近不惑之年的微软，正在不断地调整以改变自己——从内部人员到产品线，进而到产品设计理念。现在，微软的产品线不仅软件产品异常丰富，而且在硬件领域不断出击，从常用的键盘、鼠标到家用游戏机Xbox、业界最好的体感设备Kinect及随后推出的Surface RT/Surface Pro。2014年，微软更是完成了对著名移动厂商Nokia的收购，从而使公司变成了“软硬”兼备的公司。微软目前拥有数十个著名的产品品牌、数百个优秀的产品、数以千计的先进技术、数万名业界著名人才、数百万个行业技术解决方案以及数百亿美金的现金储备，这些资源在一个敢于面对变革的新CEO领导下，微软像一位围棋高手一样不断变换布局迎接全新的21世纪，这个布局的核心就是“移动为先，云为先”，换言之就是“服务＋设备”。

笔者从小就是一个非常“Geek”的人，从装收音机、电视机到给科技杂志投稿，整天畅想着如科幻小说般的未来，这一切伴随着我的少年时代。后来逐步学习各种计算机语言和各种IT技术，希望自己能够修炼成IT界的“绝世高手”。但是我天赋平凡，面对发展迅猛的IT产业，我依然像个无知的孩子，只有不断地学习新的知识。另一方面，一直以来，在我的内心深处都认为传道授业、教书育人是一件无上光荣的事情。1996年春天，当Windows 95中文版在中国发布后不久，我加入了微软公司，我那时的头衔是“布道师”（Evangelist），虽然不是“老师”，但是我找到了“装老师”的感觉。从主办TechEd、PDC（Build），到在微软研究院和最聪明的科学家一起工作……我在微软经历了人生最美好的时光。2000年，我加入了另外一家伟大的“水果”公司……直到2012年，当Windows 8.0即将发布时，我回到了微软

公司，我的职业生涯和这家伟大的公司重新绑定，我相信我选择的未来之路！

清华出版社是令人敬仰的出版社，选题精准，作风严谨。小时候，它就是我寻找计算机和技术“武功秘籍”的地方。随着移动互联网的飞速发展，人们的时间被无情的“碎片化”——微信、微博、短信、邮件、网页，等等；但是我认为要想在技术方面有所作为，踏踏实实地读书并积极地实践是最有效的方式。很荣幸受邀为此微软技术系列图书撰写序言，当我看到这些选题和主要内容时，我迫不及待地恳请编辑务必“赐予”我一套图书，我一定会仔细拜读，我也会推荐给我的业界好友。

北京的雾霾好像越来越严重了，而周末在一个安静的地方阅读一本好书，整个人的“小宇宙”会被提升到另一个维度，大有醍醐灌顶、大彻大悟的感觉。希望您也能和我一样在阅读这套图书时找到这样的美妙感觉……

夏鹏（微软（中国）有限公司）

2014 年 4 月 25 日深夜

于春雨中的北京

前言

FOREWORD

伴随着移动互联网的快速发展，智能手机应用程序的开发已经是当前 IT 行业最火的一项工作。三大手机平台（iOS、Android 和 Windows Phone）各有千秋，都在不断地快速迭代发展，寻求占领更大的市场份额。iOS 和 Android 系统已经占领了手机市场大部分的份额，Windows Phone 作为一个后来者正在奋力追赶。目前，各大互联网公司都在 Windows Phone 这块新兴的市场上积极布局，推出了各自产品相应的客户端，争取尽早获得这个平台的用户。但是 Windows Phone 的软件工程师很缺乏，各大公司在招聘 Windows Phone 软件工程师的时候，普遍困难是招聘不到合适的软件工程师，招聘高级 Windows Phone 软件工程师更是难上加难。同时，Windows Phone 平台也成了众多个人开发者或者创业团队项目试水的平台，因为这个平台的应用程序数量相对较少，没有过于激烈的竞争，优秀的作品更加容易取得成功。

智能手机应用程序开发走向成熟化，对软件工程师能力的要求也越来越高，若想成为一名移动平台的高级软件工程师，就需要深入地去掌握这个平台的开发技术。衡量一个智能手机软件工程师的技术能力主要有三点：第一，对这个平台的特性的掌握程度，如在 Windows Phone 中 C# 编程的能力，推送通知、后台任务等的开发特性及其掌握程度；第二，对这个平台的 UI 编程的掌握程度，具体来说就是能否熟练地写出各种复杂的控件、精美的动画、复杂的 UI 交互效果等；第三，就是程序架构能力，程序架构能力会跟平台的特性相关，如 Windows Phone 的 MVVM 模式（也有很多通用的架构知识，如设计模式）。目前，手机应用程序精益求精，UI 控件的编程能力也要求越来越高，一个应用程序的 UI 交互效果是否实现到位，是否实现得更加高效流畅，会直接影响整个应用程序的好坏。目前，很多手机客户端的开发团队，甚至还用专门的开发人员负责 UI 方面的编程，既说明了 UI 部分的重要性，同时也说明了 UI 编程的复杂度越来越高。

本书专门对 Windows Phone 的 UI 控件编程技术进行了详细的探讨，旨在让 Windows Phone 的开发人员能够更加全面和深入地掌握 Windows Phone 的 UI 控件编程，开发出更加优秀的、精美的应用程序。

本书包含哪些内容

本书内容涵盖 Windows Phone 8.1 UI 控件编程的各方面的知识，比如 XAML 的原理与动态解析、样式模板、布局原理、图形绘图、图标编程、变换效果、三维效果、动画编程、控件编程等，讲解全面深入，并且提供了很多程序示例演示。

书中配套实例源代码

本书配套提供实例内容的所有源代码，包括各章涉及的所有实例源代码。源代码下载地址为 www.tup.com.cn。

如何高效阅读这本书

由于本书论述的内容是 UI 控件编程，属于 Windows Phone 8.1 开发中的一个特定模块技术，所以读者在阅读本书的时候需要具备最基本的 Windows Phone 8.1 的编程基础。本书并不是针对一个完全没有 Windows Phone 8.1 开发技术基础的读者，而是在假定读者具备了 Windows Phone 8.1 的入门的基础上进行设计内容安排的。另外，本书重点是贯穿一种解决问题的思路和方案，读者在阅读的过程中可以去思考这些思路，多动手去编程实践，这样就更能领会到其中的原理。本书的各章节之间有一定的知识关联，由浅至深地渐进式叙述，建议读者按照章节的顺序来阅读和学习。

如何快速动手实践

本书每个知识点都配有相应的实例，读者可以直接用 Microsoft Visual Studio 2012 Express for Windows Phone 开发工具打开工程文件进行调试和运行。本书的代码是基于 Windows Phone 8.1 开发环境开发的，由于微软的开发工具和 Windows Phone SDK 更新较频繁，所以不能保证最新的开发环境和本书中描述的内容完全一致，要获取最新的开发工具和 Windows Phone SDK，请关注微软的 Windows Phone 开发的中文网站(https://dev.windowsphone.com/zh-cn)的动态。

本书适合哪些读者

本书适合于具有 Windows Phone 8.1 编程基础的读者，适合于想要深入学习 Windows Phone 8.1 的 UI 控件编程技术知识的读者，适合于想要往 Windows Phone 8.1 高级软件工程师发展的读者。

由于作者水平有限，Windows Phone 8.1 开发知识极其广泛，书中难免存在疏漏和不妥之处，敬请广大读者批评指正。

作者联系方式：zheng-lin@foxmail.com

编辑联系方式：shengdl@tup.tsinghua.edu.cn

林　政

2014 年 4 月

目录

CONTENTS

第1章 程序界面

程序界面是手机应用程序最基本、最直接的部分，它直接展现在用户面前，是所有底层技术结果的最终的呈现，是连接应用程序所有功能的最上层接口。掌握好程序界面的相关原理及编程技巧是深入理解和开发 Windows Phone 8.1 应用程序的重要部分。本章以程序界面为主题，介绍 Windows Phone 8.1 与程序界面相关的技术实现原理和编程技巧。本章是从 Windows Phone 应用程序界面的实现原理的角度去探讨程序界面编程，以便读者更深刻理解 Windows Phone 程序界面的奥妙。

1.1 XAML 的原理

XAML 是 Windows Phone 界面编程的编程语法，承担了如何去显示和布局 Windows Phone 程序界面的工作。本节将不会详细地讲解 XAML 的相关编程语法，而是从应用程序运行原理的角度去介绍 XAML 文件是如何被程序所使用的。

1.1.1 XAML 的概念

XAML 是一种 XML 的用户界面描述语言，有着 HTML 的外观，又揉合了 XML 语法的本质，Windows Phone 应用程序的界面就是由一个个的 XAML 文件构建而成的。XAML 本质上属于一种. NET Programming Language，属于通用语言运行时(Common Language Runtime)，同 C#、VB. NET 等同。

因为 XAML 仅仅是一种使用. NET API 的方式，想把它与 HTML、可伸缩向量图形(Scalable Vector Graphics，SVG)或者其他特定于域的格式或语言作比较，是完全错误的。XAML 由一些规则(告诉解析器和编译器如何处理 XML)和一些关键字组成，但是它自己没有任何有意义的元素。因此，如果在没有 WIndows Phone 或者 WPF 这样的框架的基础上讨论 XAML，就如同在没有. NET Framework 的基础上讨论 C#一样。

XAML 在 Windows Phone 中扮演的角色通常是令人困惑的，因此第一件要搞清楚的事情是 Windows Phone 和 XAML 可以独立使用，它们并不是互相依赖的，XAML 还可以应用于微软的其他技术里面(如 Windows 8，WPF，Silverlight 等)。由于 XAML 的通用性，

实际上可以把它应用于任何.NET 技术。

1.1.2 XAML 页面的编译

Windows Phone 的应用程序项目会通过 Visual Studio 完成 XAML 页面的编译，在程序运行时会通过直接链接操作加载和解析 XAML，将 XAML 和过程式代码自动连接起来。如果不在乎将 XAML 文件和过程式代码融合，那么只需把它添加到 Visual Studio 的 Windows Phone 项目中来，并用界面中的 Build 动作来完成编译即可，一般公共的样式资源的 XAML 文件都是采用这种方式。如果要编译一个 XAML 文件并将它与过程式代码混合，第一步要做的就是为 XAML 文件的根元素指定一个子类，可以用 XAML 语言命名空间中的 Class 关键字来完成，一般 Windows Phone 的程序页面是采用这种方式，通常在 Windows Phone 项目新增的 XAML 文件都会自动地生成一个对应的 XAML.CS 文件，并且默认地将两个文件关联起来，例如，添加的 XAML 文件如下：

```
< Page
    x:Class = "PhoneApp1.MainPage"
    ...>
    ...省略若干代码
</Page>
```

与 XAML 文件关联起来的 XAML.CS 文件如下：

```
namespace PhoneApp1
{
    public sealed partial class MainPage : Page
    {
    ...省略若干代码
    }
}
```

通常把与 XAML 文件关联的 XAML.CS 文件叫作代码隐藏文件。如果引用 XAML 中的任何一个事件处理程序(通过事件特性，如 Button 的 Click 特性)，这里就是我们定义这些事件处理程序的地方。类定义中的 partial 关键字很重要，因为类的实现是分布在多个文件中的。可能你会觉得奇怪，因为在项目里面只看到了 MainPage.xaml.cs 文件定义了 MainPage 类，其实 MainPage 类还在另外一个地方定义了，只是在项目工程里面隐藏了而已。当我们编译完 Windows Phone 的项目时，你会在项目的 obj\Debug 文件夹下看到 Visual Studio 创建的以 g.cs 为扩展名的文件，对于每一个 XAML 文件，你会找到对应有一个 g.cs 文件。例如，如果项目中有一个 MainPage.xaml 文件，就会在 obj\Debug 文件夹下找到 MainPage.g.cs 文件。下面来看一下 MainPage.g.cs 文件的结构：

```
using System;
...
namespace PhoneApp1 {
```

```
    public partial class MainPage : global::Windows.UI.Xaml.Controls.Page {
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.Windows.UI.Xaml.Build.
Tasks","4.0.0.0")]
        (global::Windows.UI.Xaml.Controls.Grid LayoutRoot;
        ...
        private bool _contentLoaded;
        [System.Diagnostics.DebuggerNonUserCodeAttribute()]
        public void InitializeComponent() {
            if (_contentLoaded) {
                return;
            }
            _contentLoaded = true;
            global::Windows.UI.Xaml.Application.LoadComponent(this, new
global::System.Uri("ms-appx:///MainPage.xaml"),
global::Windows.UI.Xaml.Controls.Primitives.ComponentResourceLocation.Application);
            LayoutRoot = (global::Windows.UI.Xaml.Controls. Grid)this.FindName("LayoutRoot");
            ...
        }
    }
}
```

从 MainPage.g.cs 文件中可以看到，MainPage 类在这里还定义了一些控件和相关的方法，并且 InitializeComponent()方法里面加载和解析了 MainPage.xaml 文件，MainPage.cs 文件里面的 MainPage()中调用的 InitializeComponent()方法就是在 MainPage.g.cs 文件里面定义的。在 xaml 页面中声明的控件，通常会在.g.cs 中生成对应控件的内部字段。实际上这取决于控件是否有 x:Name 属性，只要有这个属性，都会自动调用 FindName 方法，用于把字段和页面控件关联。没有 x:Name 属性，则没有字段，这种关联会有一定的性能浪费，因为是在应用载入控件的时候，通过 LoadComponents 方法关联的，而 xaml 也是在这个时候动态解析的。

在项目的 obj\Debug 文件夹下，还可以找到以 g.i.cs 为扩展名的文件，对于每一个 XAML 文件，也会找到对应有一个 g.i.cs 文件，并且这些 g.i.cs 文件与对应的 g.cs 文件是基本一样的。那么这些 g.i.cs 文件又有怎样的含义呢？其实这些 g.i.cs 文件并不是在编译的时候生成的，而是当创建了 XAML 文件的时候就马上生成，或者你修改了 XAML 文件 g.i.cs 文件也会跟着改变，而 g.cs 文件则是必须要成功编译了项目之后才会生成的。文件后缀中的 g 表示 generated 产生的意思，i 表示 intellisense 智能感知的意思，g.i.cs 文件是 XAML 文件对应的智能感知文件，在 vs 中利用 go to definition 功能找 InitializeComponent 方法的实现的时候，进入的就是 g.i.cs 文件的 InitializeComponent 方法里面。

1.1.3 动态加载 XAML

动态加载 XAML 是指在程序运行时通过解析 XAML 格式的字符串或者文件来动态生成 UI 的效果。通常情况下，Windows Phone 的界面元素都是通过直接读取 XAML 文件的