



图灵程序设计丛书

Apress®

Pro Windows 8.1 Development with XAML and C#

精通 Windows应用开发

微软MVP引领精通Windows开发之路

探索在微软平台上应用XAML和C#的未来方向

【美】Jesse Liberty Philip Japikse Jon Galloway 著
苏宝龙 译



人民邮电出版社
POSTS & TELECOM PRESS



图灵程序设计丛书

Pro Windows 8.1 Development with XAML and C#

精通 Windows应用开发

【美】Jesse Liberty Philip Japikse Jon Galloway 著
苏宝龙 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

精通Windows应用开发 / (美) 利伯蒂 (Liberty, J.),
(美) 亚皮克瑟 (Japikse, P.), (美) 加洛韦
(Galloway, J.) 著; 苏宝龙译. — 北京: 人民邮电出
版社, 2015. 1

(图灵程序设计丛书)

ISBN 978-7-115-37657-2

I. ①精… II. ①利… ②亚… ③加… ④苏… III.
①Windows操作系统—程序设计 IV. ①TP316.7

中国版本图书馆CIP数据核字(2014)第269441号

内 容 提 要

Windows 8 的出现不仅提供了跨设备的用户体验, 也提供了跨设备的开发体验。本书着眼于实际项目中所需要的特性, 以及现有 C# 编程知识的运用, 对如何最大限度地利用 Metro、WinRT 和 Windows 8 进行了讲解, 内容详尽, 注重理论学习与实践开发的配合。

本书既适合 Windows 开发的入门读者, 对于有经验的开发人员也极具参考价值。

◆ 著 [美] Jesse Liberty Philip Japikse Jon Galloway

译 苏宝龙

责任编辑 朱 巍

执行编辑 杨 琳

责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京天宇星印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 17

字数: 402千字

2015年1月第1版

印数: 1—2 500册

2015年1月北京第1次印刷

著作权合同登记号 图字: 01-2014-5093号



定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

Original English language edition, entitled *Pro Windows 8.1 Development with XAML and C#* by Jesse Liberty, Philip Japikse and Jon Galloway, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2014 by Jesse Liberty, Philip Japikse and Jon Galloway. Simplified Chinese-language edition copyright © 2015 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P. 授权人民邮电出版社独家出版。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。

致 谢

撰写这样的书籍需要投入巨大的精力，但作者完成的工作仅是整个工作流程中的一部分。在不知不觉间我就参与了本书的许多编撰工作，但如果没有 Apress 出版社中优秀团队的帮助，我肯定也无法取得今天的成绩。在此我衷心地感谢他们。Dan Maharry 和 Todd Meister 不厌其烦地测试代码，确保它们的简洁性和正确性，以及更为重要的实效性！这些技术复审对本书尽心竭力。感谢你们使我保持敏锐的思维，编写出正确的代码，避免遗漏而使读者感到困惑或泄气。Ewan Buckingham 和 Jim DeWolf 与我一起开始本书的创作工作，并使我不断前进。Chris Nelson 宝贵的修改意见使本书一直保持正确的方向。感谢你一直不断提醒我要将自己的想法向读者解释清楚。Chris Nelson 使我们这些合著者在撰写本书时保持一致，并向读者传达清晰的正确讯息。Jana Weinstein 担负了本书的复审和统稿工作，帮助我提高了本书的质量。最后，感谢 Christine Ricketts 在必要时督促我前进。撰写书籍的工作并不轻松，但我能确定这是最具价值的一种工作！没有这支优秀团队的帮助，我们无法撰写出这本书，无法使其与读者见面。

我还要感谢 Jesse Liberty 和 Jon Galloway 给予我信任，使我能够为他们的工作锦上添花并与 Apress 出版社的优秀团队合作。

当然，我亲爱的家人们也给予了我很多鼓励。为了使我保持不断前进的动力，Conner 会不断问我：“你写到哪里了？”这时我就会重新回到电脑前继续撰写。感谢 Logan 和 Skylar 给予我的耐心。这本书已经写完了，我们可以去玩电子游戏了！

目 录

第1章 入门	1	2.2.2 App.xaml 文件	27
1.1 背景	1	2.2.3 MainPage.xaml 文件	27
1.1.1 两种用户界面	1	2.3 Model View ViewModel	27
1.1.2 支持 ARM 设备	1	2.3.1 模式	27
1.1.3 接受度	2	2.3.2 创建模型	28
1.1.4 快速的发布周期	2	2.3.3 创建视图模型	31
1.1.5 微软应用商店	2	2.3.4 更新隐藏代码	32
1.1.6 Windows 8.1 的新增功能	2	2.3.5 创建视图	32
1.2 Windows 的设计原则	2	2.3.6 测试应用	33
1.2.1 快速流畅	3	2.3.7 指导原则	34
1.2.2 出色的尺寸调节功能	3	2.4 导航	34
1.2.3 使用正确的合约	3	2.4.1 创建新页面	35
1.2.4 打造精致的磁贴	3	2.4.2 向主页面添加导航功能	36
1.2.5 网络连接和实时性	3	2.4.3 处理 NavigatedTo 事件	37
1.2.6 可利用云平台	3	2.4.4 后退按钮	37
1.2.7 遵守现代应用程序的设计原则	4	2.5 小结	38
1.2.8 UX 设计指导原则	4	第3章 主题、面板和控件	39
1.3 工具	4	3.1 选择一种主题	39
1.3.1 Visual Studio 2013	5	3.2 使用面板	41
1.3.2 Blend for Visual Studio 2013	10	3.2.1 画布	41
1.4 Git	12	3.2.2 网格	43
1.5 NuGet	16	3.2.3 堆积面板	47
1.5.1 安装 NuGet	16	3.2.4 边框控件	47
1.5.2 启用程序包恢复功能	17	3.3 使用控件	48
1.5.3 安装第一个程序包	19	3.3.1 文本块和文本框	49
1.6 小结	20	3.3.2 拼写检查	51
第2章 创建第一个 Windows 8 应用	21	3.3.3 标题和水印	51
2.1 Windows 的体系结构 (面向开发者)	21	3.3.4 密码框	53
2.2 创建第一个应用	22	3.3.5 按钮和事件处理程序	54
2.2.1 应用项目概述	22	3.3.6 复选框、切换开关和单选按钮	56

3.3.7 列表框、列表视图和组合框	58	5.1.6 处理调整尺寸事件	100
3.3.8 图像控件	60	5.2 网格应用程序模板	101
3.3.9 滑块	61	5.2.1 网格应用程序模板的组成部 分	101
3.3.10 进度条	62	5.2.2 通过多种方式将数据整合到 网格应用程序项目中	104
3.3.11 工具提示	62	5.2.3 将自定义数据整合到 SampleDataSource 类中	104
3.3.12 日期选取器和时间选取器	63	5.3 拆分布局应用程序模板	110
3.3.13 弹出菜单	63	5.4 小结	111
3.4 了解依赖属性	65	第 6 章 本地数据	112
3.4.1 数据隐藏	65	6.1 应用程序数据	112
3.4.2 依赖属性	66	6.1.1 设置容器	112
3.5 小结	68	6.1.2 保存、读取和删除本地数据	113
第 4 章 绑定	69	6.2 创建数据层次	115
4.1 数据上下文	69	6.2.1 创建存储库接口	115
4.2 创建简单的绑定	70	6.2.2 创建数据模型	116
4.3 数据绑定语句	71	6.2.3 创建视图模型	116
4.4 绑定错误	72	6.3 本地数据	119
4.4.1 FallbackValue	73	6.3.1 使用 JSON 格式化数据	119
4.4.2 TargetNullValue	73	6.3.2 本地数据容器	120
4.5 绑定元素	74	6.3.3 创建文件存储库	120
4.6 绑定模式	75	6.3.4 创建视图	123
4.7 UpdateSourceTrigger	76	6.3.5 漫游	127
4.8 INotifyPropertyChanged	76	6.4 用户指定的存储位置	127
4.9 绑定集合对象	78	6.4.1 创建 FileOperations 类	127
4.9.1 创建集合	79	6.4.2 为 JSON 文件添加文件关联	129
4.9.2 创建已绑定数据的 ListView 控件	81	6.5 SQLite	131
4.9.3 创建已绑定数据的 GridView 控件	82	6.6 小结	136
4.9.4 INotifyCollectionChanged	83	第 7 章 远程数据和服务	137
4.10 数据转换器	84	7.1 远程数据服务	137
4.11 小结	86	7.2 REST	138
第 5 章 视图	87	7.3 常见的 API 格式	138
5.1 GridView 控件和 ListView 控件	87	7.3.1 XML	139
5.1.1 创建样本	87	7.3.2 JSON	139
5.1.2 支持代码	93	7.4 ASP.NET Web API	140
5.1.3 调整应用的宽度	94	7.4.1 使用 ASP.NET Web API 创建 服务	140
5.1.4 创建 ListView 控件	95		
5.1.5 切换视图	96		

7.4.2 通过 Web API 约定代替配置	144	9.5.1 创建带 Toast 通知功能的应用	194
7.4.3 Entity Framework	144	9.5.2 更新清单文件	195
7.5 通过 Windows 8.1 客户端访问 ASP.NET Web API 服务	148	9.5.3 手动创建 Toast 通知	196
7.5.1 通过 NuGet 添加 Web API 客户端程序包	148	9.5.4 通过 NotificationsExtensions 项目创建 Toast 通知	197
7.5.2 添加 Contact 类	150	9.5.5 回应 Toast 通知的启动	198
7.5.3 编辑 SampleDataSource 类	150	9.6 通过 Windows 推送通知服务和 Windows Azure 移动服务推送通知	200
7.5.4 调用 LoadDataAsync 方法	152	9.6.1 WNS 交互功能概述	200
7.5.5 运行客户端	153	9.6.2 为什么使用 WAMS	201
7.6 在 Windows 8.1 中使用 OData 数据	154	9.6.3 WAMS 日志	217
7.7 小结	159	9.7 小结	217
第 8 章 搜索和共享合约	160	第 10 章 应用的生命周期	218
8.1 搜索	161	10.1 三种应用状态	218
8.1.1 为应用添加基于超级按钮的 搜索功能	161	10.1.1 正在运行	219
8.1.2 创建样本数据来匹配搜索关 键词	164	10.1.2 已挂起	219
8.1.3 使用搜索框执行搜索操作	168	10.1.3 已终止	219
8.2 共享	171	10.2 状态转换	219
8.2.1 结构	172	10.2.1 启动	220
8.2.2 实现	172	10.2.2 激活	220
8.3 创建共享目标	175	10.2.3 挂起	220
8.4 小结	177	10.2.4 继续	220
第 9 章 通知	178	10.2.5 终止	220
9.1 通知的显示选项	178	10.2.6 杀掉进程	221
9.2 通知的传送选项	179	10.3 使用挂起管理器	221
9.3 动态磁贴	179	10.3.1 使用异步方法	221
9.3.1 创建样本数据	180	10.3.2 在空白应用程序中添加 SuspensionManager 类	221
9.3.2 手动添加动态磁贴	183	10.4 创建应用	223
9.3.3 通过 NotificationsExtensions 添加动态磁贴	186	10.4.1 添加导航参数类	224
9.3.4 重置磁贴	188	10.4.2 创建详细信息页面	224
9.4 辅助磁贴	188	10.4.3 创建主页面	225
9.4.1 创建项目	189	10.4.4 添加/编辑列表项	226
9.4.2 创建辅助磁贴	189	10.5 回应挂起操作	227
9.4.3 回应辅助磁贴的启动	192	10.5.1 注册根帧	227
9.5 Toast 通知	194	10.5.2 OnSuspending 事件	227
		10.5.3 保存会话数据	228
		10.5.4 保存会话状态	228
		10.6 回应激活操作	229

10.6.1	确定上一个状态	229	11.2.3	将试用版转换为正式版	240
10.6.2	对恢复导航状态进行测试	230	11.2.4	应用内广告	241
10.6.3	恢复会话信息	230	11.3	小结	247
10.6.4	对恢复会话信息进行测试	231	第 12 章	发布应用	248
10.7	小结	231	12.1	注册开发者账号	248
第 11 章	获取利润	232	12.2	提交应用清单	249
11.1	选择获利方案	232	12.2.1	选择应用名称	250
11.1.1	免费	232	12.2.2	设置销售详细信息	251
11.1.2	销售	232	12.2.3	年龄分级和分级认证	254
11.1.3	应用内购物支持	233	12.2.4	加密	254
11.1.4	应用内广告支持	233	12.2.5	程序包	254
11.2	实现获利方案	233	12.2.6	提要	258
11.2.1	通过程序方式检查许可 状态	233	12.2.7	测试人员说明	263
11.2.2	设置应用内购物功能	236	12.2.8	提交应用进行认证	263
			12.3	小结	264

第 1 章

入 门

Windows 8.1 的开发工作与 Windows Presentation Foundation (WPF) 应用程序的开发工作有许多相似之处。当然,最大的相同点是这两项工作都使用了 XAML 和 C#。你掌握的许多用户界面 (UI) 和程序代码技巧也都可以在 Windows 8.1 应用程序中使用。但是,二者还有许多不同之处。Windows 8.1 应用程序以触摸屏为核心 (也支持鼠标和键盘交互方式),其中的某些设计原则与 WPF 的设计原则截然相反。Windows 8.1 的应用程序隔离级别远高于 WPF,从而将安全性提高了一个层次,同时也遇到了处理数据的独特挑战。其应用程序是通过应用商店 (而非手动操作或微软安装程序包) 部署的。

如果你不是 WPF 开发者也无需担心。本书涵盖了开发 Windows 8.1 应用程序所需的全部知识。

1.1 背景

微软于 2012 年 10 月 26 日发布了 Windows 8.0 操作系统。这个版本在多个方面取得了突破,但它取得的最大进步莫过于 Windows 8 的两种界面和对 ARM 设备的支持。

1.1.1 两种用户界面

Windows 8.0 的初始版本引入了两种 UI: 支持 Intel 硬件的传统桌面,以及同时支持 Intel 硬件和 ARM 硬件的磁贴式界面 (以前称 Metro 式界面),这对于 Windows 操作系统来说是破天荒的创举。

简而言之,与 Windows 7 相比,Windows 8.0 的桌面 UI 更加优秀,其安装文件小得多 (升级后,我的硬盘还多出了 6GB 可用空间),而且更加安全,在相同硬件条件下,运行速度也比 Windows 7 更快。

磁贴式 UI 是 Windows 8.0 引入的全新界面。尽管相同类型的磁贴式界面已经被应用于 Windows Phone 操作系统,但这是 Windows 非手机系统版本多年来的第一次重大改变。

1.1.2 支持 ARM 设备

基于 ARM 架构的处理器 (由英国 ARM Holdings 公司制造) 已经称霸平板电脑市场。这个市场最明显的缺憾就是没有微软的产品,至少在微软看来是这样。Windows 用户也可能会有相同的观点。微软通过 Surface RT 将基于 Windows 的产品打入了平板电脑领域,使用户能够同步台式机、笔记本电脑、智能手机和平板电脑,而且 Microsoft Office 也能在 Surface RT 上运行!

1.1.3 接受度

这些巨大的改变在第一年的评价如何？单从媒体和专家的反映来看，情况很糟。这与操作系统本身无关，而与大众的呼声有关。从“开始菜单在哪里”到“怎样打印资料”，人们不断抱怨这些改变。这些意见都是正确的吗？观点不同，得出的结论也不同。很快，后续发布的 Windows 8.1 带回了开始按钮，并且解决了最常见的一些问题，使牢骚满腹的人们看到了新版本的优点。

有趣的是，多家网站的统计结果表明，Windows 8 发布第一年的接受度能与 XP 发布第一年的接受度匹敌。如果你经历过 Windows 95 的时代，就能回忆起似曾相识的场景。事物总在不断变化，经过最初的震惊，大家都平静下来，开始使用新软件。

1.1.4 快速的发布周期

发布 Windows 8 后不到一年，微软就在 2013 年 10 月 17 日发布了 Windows 8.1。该版本并非简单的补丁包（sp），它解决了人们抱怨的许多问题，如恢复了“开始”按钮，能够直接启动到桌面模式，等等。人们可以自由地通过微软应用商店（Microsoft Store）来安装 Windows 8.1，因此该版本的安装率极高。

1.1.5 微软应用商店

你一定遇到过因家人点击网页中的弹出窗口或者安装了别人推荐的软件，而不得不为他们解决问题的情况吧？在 Windows 8/8.1 中，应用程序的主要来源是微软应用商店（Microsoft Store，下文简称为“应用商店”）。通过这个核心站点，获取应用程序，有助于防止安装恶意软件，提高设备的安全性和可靠性。应用商店还为开发者提供了一个向其他人展示应用程序的平台。详细了解向应用商店提交应用程序的方法，请参阅第 12 章。

1.1.6 Windows 8.1 的新增功能

与 Windows 8.0 相比，Windows 8.1 做出了许多改变。本书的每一章开头都有“Windows 8.1 的新增功能”栏，概述与该章主题有关的 Windows 8.1 新增功能。这些章节详细介绍了使用 Visual Studio 2013 和 Windows 8.1 开发应用程序的方法。

1.2 Windows 的设计原则

如果你希望开发的应用程序顺利进入应用商店，那么它们必须拥有优秀应用程序的 7 种特质，而且遵守微软的 5 条设计准则。第 11 章还将详细介绍一些其他的设计要求。

下面是优秀应用程序的 7 种特质：

- ☐ 快速流畅；
- ☐ 出色的尺寸调节功能；
- ☐ 使用正确的合约；
- ☐ 打造精致的磁贴；
- ☐ 网络连接和实时性；

- ❑ 可利用云平台；
- ❑ 遵守现代应用程序的设计准则。

1.2.1 快速流畅

现代应用程序可以在各种设备上运行并提供许多功能。尽管微软为 Windows 8 设置了各种硬件的最低标准，但对于应用（以及 Windows 8）是否能够取得成功而言，关键还是要看应用程序的性能。当你学习本书关于开发 Windows 8.1 应用程序的知识时，就能体会到必须使用异步编程方法确保 UI 的高响应性。此外，Windows 8.1 进程生命周期管理的精巧设计，确保了后台应用程序不会很快耗尽电池电量和宝贵的系统资源。

大量使用异步模式能够确保应用程序的高反应性。如果加载与启动应用都需要很长时间，那么用户就会卸载它。更糟糕的是，他们可能还会先写一条措辞严厉的差评，然后立刻将它卸载。

1.2.2 出色的尺寸调节功能

运行 Windows 8 的设备拥有各种尺寸和屏幕分辨率。你的应用程序可能会全屏运行，可能使用纵向视图，也可能需要调整大小与其他应用程序共用一个屏幕。因此，它必须能够适应不同的布局和尺寸，这不仅应体现在外观上，还应体现在可用性上。例如，如果你的应用程序使用网格模式显示大量数据，那么当其显示模式被切换为停靠到屏幕的一侧时，就应该自动将网格模式切换为列表模式。

1.2.3 使用正确的合约

Windows 8 引入了一种全新的方式来实现与操作系统和其他应用程序的交互——合约（contract）。合约涵盖了搜索、共享和设置等方面。通过这些合约，你可以使用用户熟悉的方式展示应用程序的功能。

1.2.4 打造精致的磁贴

磁贴是应用程序的门面。动态磁贴可以吸引用户使用应用程序，而且会增加用户使用的趣味和时段。过多的更新会导致用户关闭更新功能，甚至卸载应用程序。

采用辅助磁贴非常不错，用户能够在开始界面保留一些信息，从而快速访问他们感兴趣的项目。

1.2.5 网络连接和实时性

用户是 Windows 8 应用程序中最重要的考虑因素。确保应用能够连接互联网、接收实时信息非常重要。不论是股价还是公司的销售曲线图，陈旧的数据会使用户不再理会你的应用程序。通报昨天的天气情况毫无意义，人们只关心未来几天的天气预报。

1.2.6 可利用云平台

用户使用 Windows 8 可以在各种设备之间共享数据。这不仅可以同步应用程序的设置，还可以同步应用程序的数据。设想一下，用户在办公室的计算机上将数据输入你的应用程序，下班回到家后拿

起另一台运行 Windows 8 的设备并启动同一应用程序, 上班时使用的数据都在里面, 这是多么方便啊! 从一个设备无缝跃迁到另一个设备的要点是利用云平台。

1.2.7 遵守现代应用程序的设计原则

除了前面介绍的特质, 开发的应用程序还必须符合现代应用程序的设计原则。微软详细介绍了这些准则: <http://go.microsoft.com/fwlink/p/?linkid=258743>。下面概括了 5 条当今应用程序设计准则。

❑ 展示技术自豪感 要做到这一点非常简单。恕我直言, 别开发没有价值的软件。开发那些你愿意在简历上列出来的应用程序。如果你想通过开发应用程序获利, 就需要让别人使用你开发的应用程序, 在社交媒体上推荐它, 在朋友面前炫耀它。除非你有大笔的营销预算, 否则这一点非常重要。但是即使你做到了这一点, 也可能只能得到首批用户。一旦人们对该应用程序有了负面评论 (记住, 应用商店有一个评分系统), 你的应用程序就没前途了。

因此, 确保你的应用程序展示了技术自豪感。如果你不擅长艺术设计, 可考虑雇用平面设计师。请用户 (如你的朋友和家人) 测试你的应用程序, 获取反馈信息, 解决所有问题, 包括小缺陷和用户体验问题。

❑ 快速流畅 微软非常重视这一点, 将其作为评判应用程序的重要标准。除了前面介绍的特质, 触控式和直观的交互设计也很重要。应重视用户的交互感受, 开发具有沉浸感和吸引力的 UI。

❑ 实现真正的数据化 彻底发掘设备的潜能。使用醒目的颜色、漂亮的字体和动画。当然, 所有这些效果 (尤其是动画) 都应有的放矢。就像当今的电影特效一样, 即使已经到了无所不能的程度, 也不应在讲述日常生活的影片中添加魔幻镜头!

❑ 使用更少的组件做更多的事情 铬合金色可以在摩托车上取得很好的效果, 但是在软件中使用这种颜色只会分散用户对数据的注意力。使用应用栏和导航栏代替按钮和选项卡。还可以使用超级按钮 (charm) 和合约机制进一步简化。

你应该将注意力集中到应用程序的功能上。不应尝试四面出击地解决问题, 只需解决一两个真正的问题即可。集中精力于问题的解决方案, 尽力使用户专注于他们的数据, 而不是应用程序本身。

❑ 共赢 在开发工作中利用 Windows 8 的范例模式。使用常用的触控手势和用户熟悉的键鼠主题, 这样用户就可以使用以前学过的操作知识。还应通过合约与其他应用程序协作, 这样你的应用程序就可以发挥整体大于局部简单相加之和的效果。

1.2.8 UX 设计指导原则

微软还提供了很多有关设计用户体验的指导原则。要详细了解这些原则, 请浏览 <http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx>。指导原则的 PDF 文件下载, 参见 <http://go.microsoft.com/fwlink/p/?linkid=258743>。

1.3 工具

尽管开发应用程序时, 你可以全程使用 Visual Studio, 但是同时利用其他工具可以获得最佳体验。

开发 Windows 8 应用程序，需要使用两种主要工具：Visual Studio 2013 和 Blend for Visual Studio 2013。

1.3.1 Visual Studio 2013

微软在 2013 年 11 月发布了 Visual Studio 2013。Visual Studio 的该最新版本新增了支持 Windows 8.1 应用程序开发的功能。这次的版本升级是强制性的，因为 Visual Studio 2012 不支持 Windows 8.1 应用程序的开发。

1. 版本

既然选择了本书，那可能说明你非常熟悉 Visual Studio。本节介绍 Visual Studio 2013 的不同版本和它们之间的差异。如果你不熟悉 Visual Studio 也无需担心。本书后面的章节会详细地介绍相关功能。

● Visual Studio Express

Visual Studio Express 2013 为不同目标平台提供了多个版本。尽管没有见过任何官方说明解释过为何要将该工具分割为多个版本，但我认为拆分它的原因是该工具过大不利于免费下载，而且将所有功能都放在一起，也会使初学者感到迷惘。

下面是该工具的可用版本。

- Visual Studio 2013 Express for Web: 供 ASP.NET 开发者使用。
- Visual Studio 2013 Express for Windows: 供 Windows 8.1 应用程序开发者使用。
- Visual Studio 2013 Express for Windows Desktop: 供使用 Windows Presentation Foundation(WPF) 或 WinForms 的 Windows 客户端应用程序开发者使用。

由于本书的目的是创建 Windows 8.1 应用程序，你需要使用 Visual Studio 2013 Express for Windows。

● Visual Studio 的 MSDN 版

面向开发者的 Visual Studio 2013 付费版本主要有 3 个：专业版、高级版和旗舰版。所有这些版本都是 MSDN 版的组成部分，它们能够为 Windows 8.1 的开发工作提供所需的一切工具。注意，Visual Studio 2013 还有第 4 个版本(Visual Studio for Test Professional)，但是该版本不能用于创建 Windows 8.1 应用程序，因此本书不作介绍。详细了解这些版本，请参阅 Visual Studio 网站提供的文档：www.visualstudio.com/products/compare-visual-studio-products-vs。

2. Windows 8.1 模拟器

Visual Studio 的所有版本都有使用模拟器运行 Windows 8.1 应用程序的功能。这实际上是一种带有额外的功能的远程桌面会话，这些额外功能包括改变设备方向、外形因素、手势支持和模拟平板电脑的许多因素（即使你在非触控设备上进行开发工作）。这也是为什么必须在 Windows 8.1 环境中开发 Windows 8.1 应用程序的原因之一。

3. 创建你的第一个 Windows 8.1 应用程序

要创建 Windows 8.1 应用程序，首先在 Visual Studio 2013 中选择“文件”>“新建”>“项目”，创建一个新项目。在“新建项目”窗口的左侧，你可以看到当前使用的 Visual Studio 版本中已安装的模板（由于不同的版本和第三方产品，你看到的情况可能与图 1-1 略有不同）。选择“已安装”>“模板”>Visual C#>“Windows 应用商店”，就会出现图 1-1 所示的对话框。

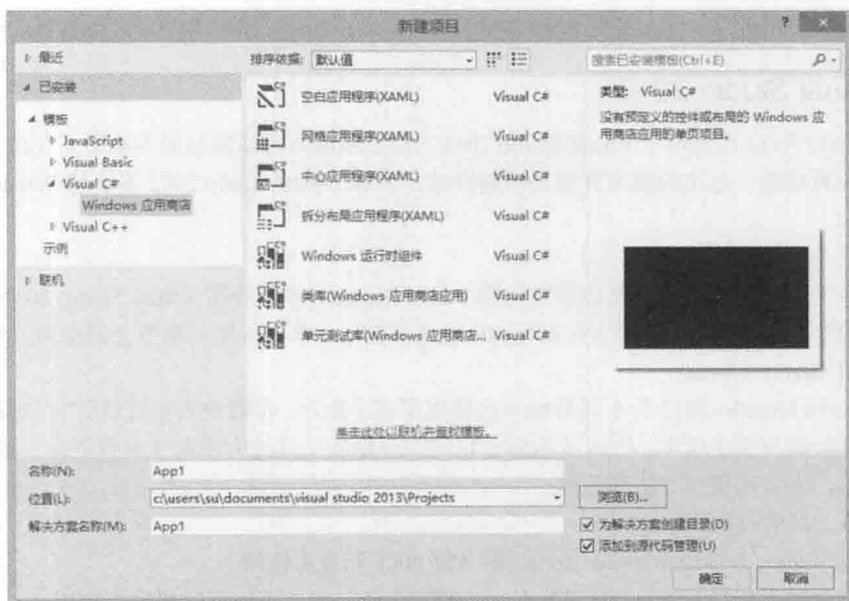


图 1-1 Windows 8.1 应用程序的新建项目模板

第4章将详细介绍所有项目模板，现在我们选择“空白应用程序”选项。实际上，这个模板是本书中大多数项目的模板，也是我开发 Windows 8.1 应用程序的常用模板。可以将项目名称保留为默认的 App1。

创建项目后，应查看默认解决方案文件夹（如图 1-2 所示）。空白应用程序模板实际上替我们做了许多工作。除了创建项目并添加合适的引用外，它还提供了多个资源，如 App.xaml 和 MainPage.xaml 文件。Assets 文件夹含有用于启动画面和默认磁贴（本书稍后会介绍）的图像文件，如果你熟悉 WPF，就会对 App.xaml 和 MainPage.xaml 文件非常熟悉。本书会详细介绍这些文件。

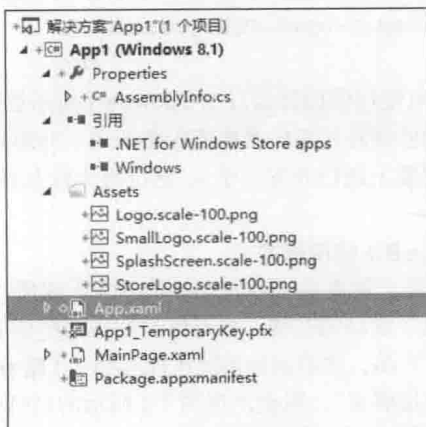


图 1-2 默认解决方案资源管理器文件

要运行该应用程序，可按 F5 键（带调试信息的启动模式）、Ctrl-F5 组合键（不带调试信息的启动模式）、单击 Debug 菜单（含有前面介绍的两个选项）中的选项或者单击工具栏中带绿色箭头的项目，如图 1-3 所示。

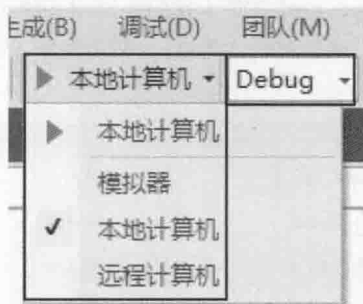


图 1-3 运行工具栏

默认情况下，Visual Studio 会在本地计算机上，使用调试配置运行你的应用程序。单击绿色箭头（或按 F5 键）运行应用程序。与预想中完全空白的画面不同，屏幕上方会显示一些不断变化的数字（当你在屏幕上移动鼠标时，这些数字就会变化），如图 1-4 所示。这些数字从左到右分别为 UI 帧速率（每秒的帧数）、应用程序 UI 线程的 CPU 使用情况、每秒的系统合成帧和系统 UI 线程的 CPU 使用情况。如果你在运行应用程序时没有开启调试模式，画面就不会显示这些数字。这是因为 Visual Studio 提供的所有模板都启用了帧速率计数功能，在使用调试模式运行程序时使用该功能可以显示这些信息。

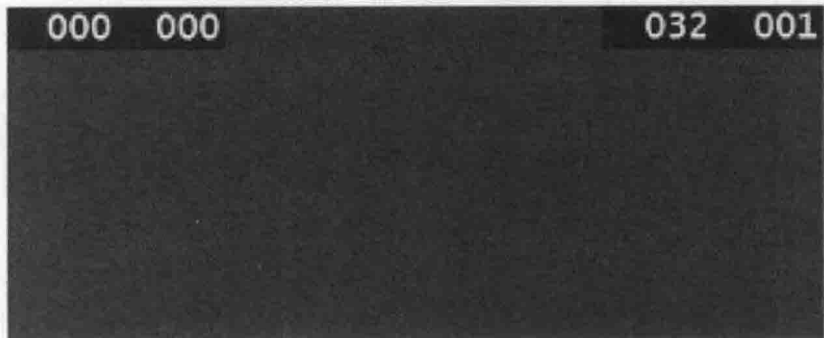


图 1-4 通过帧速率计数功能进行调试

关闭该方法非常简单，只需打开 App.xaml.cs 文件，并在 OnLaunched 事件处理程序中将下面的代码改写为注释：

```
this.DebugSettings.EnableFrameRateCounter = true;
```

结果为：

```
//this.DebugSettings.EnableFrameRateCounter = true;
```

现在使用调试模式运行应用程序，这些数字就不会再出现了。

● 添加一个基本页

尽管我经常使用空白应用程序模板开发应用程序，但是极少保留该模板中的 MainPage.xaml 文件（以及它的隐藏在 MainPage.xaml.cs 中的代码）。Visual Studio 提供了一个基本页文件模板，该模板中含有许多必要的功能。删除 MainPage.xaml 文件（我们将使用其他文件替代），然后右键单击你的项目并选择“添加”>“新建项”。在“添加新项—App1”对话框中，选择“基本页”选项并将该页面命名为 MainPage.xaml，如图 1-5 所示。

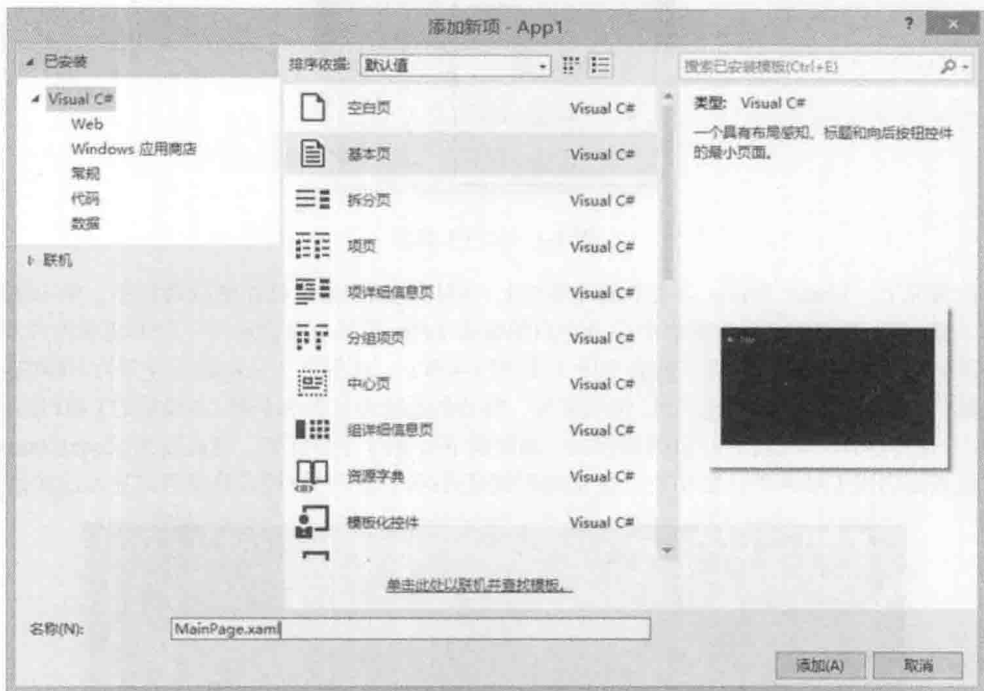


图 1-5 添加一个新的基本页

注意 我们将该页面命名为 MainPage.xaml，这样我们就无需修改 App.xaml.cs 文件。如果你想要使用其他名称（或者更改首次启动应用程序时加载的页面），可以打开 App.xaml.cs 文件，切换到 OnLaunched 事件处理程序的末尾，更改下面代码中的相应名称：

```
rootFrame.Navigate(typeof(MainPage), e.Arguments);
```

加载了新的基本页后，Visual Studio 会提示向你的项目中添加几个文件。应选择“是”按钮。这些文件极为有用，本书也会广泛地使用它们。因为现在我们只需显示一些文本，所以还无需介绍这些文件。为了在模拟器中运行应用程序，更改“调试位置”工具栏中的选项，然后按 F5 键（或者单击用于启动调试过程的绿色箭头）。屏幕会显示应用程序的磁贴（在类似平板电脑的窗口中运行）和一系列模拟器控件（窗口右侧），如图 1-6 所示。