



31 Days of Windows 8

# Windows 8 开发 31 日

## 第 01 日

### 一个空的应用程序

译者：BeyondVincent(破船)

时间：2013.4.22

版本： 2.0

## 关于破船

程序猿砌墙于云南昆明!

长期扎根移动软件开发!

爱跑步爱打篮球爱运动!

命中无大富大贵之面相!

愿健康与平淡相随一生!

你可以发邮件与破船取得联系: [BeyondVincent@gmail.com](mailto:BeyondVincent@gmail.com)

还可以关注破船的微博: [腾讯微博](#)和[新浪微博](#)。

这里是破船的个人博客, 欢迎光临: [破船之家](#)



## 关于 Windows 8 开发 31 日翻译



Windows 8 开发 31 日是由 Jeff Blankenburg 和 Clark Sell 原创的。

官方站点：<http://31daysofwindows8.com/>

涉及到两个版本：

XAML/C# ( 由 Jeff Blankenburg 撰写 )

HTML5/JS ( 由 Clark Sell 撰写 )

其中涉及到的资源和相关代码请到这里下载：

<https://github.com/csell5/31DaysOfWindows8>

在这里，由于破船对 HTML5/JS 不熟悉，所以只翻译 XAML/C# 相关主题。

建议大家前往看原创内容，如果看不明白，再来这里看我翻译的相关内容。

如果翻译不正确的地方，可以通过上面的联系方式告诉破船。

破船祝你阅读愉快！



目录

关于破船 ..... 2

关于 Windows 8 开发 31 日翻译 ..... 3

目录 4

第 01 日 一个空的应用程序 ..... 5

    1.0.    介绍 .....5

    1.1.    图片 .....6

        1.1.1.    Logo.png..... 7

        1.1.2.    SmallLogo.png ..... 7

        1.1.3.    SplashScreen.png..... 8

        1.1.4.    StoreLogo.png ..... 8

        1.1.5.    WideLogo.png ..... 9

    1.2.    AssemblyInfo.cs .....9

    1.3.    Package.appxmanifest..... 9

    1.4.    StandardStyles.xaml..... 10

    1.5.    App.xaml ..... 10

    1.6.    App.xaml.cs ..... 10

    1.7.    MainPage.xaml ..... 11

    1.8.    MainPage.xaml.cs ..... 11

    1.9.    Day1-TheBlanApp\_TemporaryKey.pfx 11

    1.10.    总结 ..... 12

## 第 01 日 一个空的应用程序



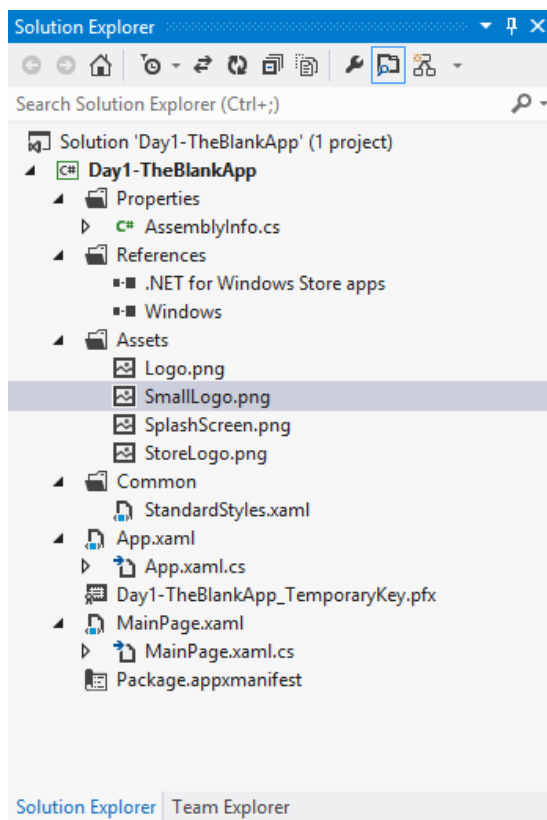
### 1.0. 介绍

在这系列的第一篇文章中，Clark 和我认为介绍 Visual Studio 2012 中的 Windows Store Blank App template 非常重要。我们将会介绍里面的每一个文件，它们非常的重要，因为许多设置都有可能隐藏在其中的每个文件。通过本文学习每一个文件，将会为后续的系列学习打下牢固的基础。

在 Visual Studio 2012 中有许多其它工程 templates ,但是我们建议总是从 Blank App template 开始创建一个真实的 app。因为那些 templates 是针对特定类型的程序，它可能会限制你的思维模式。

下面这个图是 Solution Explorer 中用 Blank App template 创建的一个全新 App。





## 1.1. 图片

我们从简单的开始——Assets 文件夹中的图片。在每一个行的工程里面，都会有 4 个图片，它们用在不同的地方。在这里，我们已经重新创建了这些图片的不同内容(另外还多了一个)，这样当运行程序的时候，可以很明显的知道哪个图片是做什么用的。[你可以在这里下载这五个文件](#)。只要你用这 5 个图片替换了缺省的图片，然后运行工程，将会看到这些图片用在不同的地方。如下所述。



### 1.1.1. Logo.png



这个图片可能是程序中使用常使用的一个图片。在开始屏幕中，这个图片是应用程序 tile 默认的背景图片。它的尺寸是 150x150pixels。

### 1.1.2. SmallLogo.png



当应用程序列表显示的时候，就用到这个图片。例如，在 Windows 8 中搜索一个应用程序时，或者点击 Share 菜单时。SmallLogo.png 的尺寸是 30x30 pixels。这么小的尺寸，如果你想确保这个图标能够真实的表达某个意思给用户，我强烈建议只用一个 logo 或者一个有意义的图像。



## 1.1.3. SplashScreen.png



这是一个初始化图片，当程序启动的时候，会加载显示这个图片。关于该图片有一些使用技巧，在后续的文章中会进行介绍。现在，你只需要记住这个图片的大小总是为 620x 300 pixels，并且总是在屏幕上垂直和水平居中显示。

## 1.1.4. StoreLogo.png



在 store 中为了节省空间，这里需要提供一个稍微小的图片。这个图片的尺寸只有 50x 50 pixels，但是，在整个程序中，这 2500 pixels 可能是最重要的了。因为当用户在决定是否将新的 app 安装到设备时，会看到这个图标。一个差劲的图标给用户的暗示就是一个差劲的 app。所以，在这个图标上花点时间是值得的。



### 1.1.5. WideLogo.png



这个图片是我新加到工程中的。你需要手动将其添加到工程中：在 Assets 文件夹上右键单击，选择“Add Existing Item...”。这个图片的尺寸是 310x150 pixels，用于比 Logo.png 标准尺寸还要宽的 tile。

### 1.2. AssemblyInfo.cs

这个文件包含应用程序集信息。比如版本号、标题、描述等。不过请记住，这些值是为程序集准备的。程序中所有的官方名称等都是存储在程序的 manifest 文件中，也就是下一个文件。

### 1.3. Package.appxmanifest

这个文件包含了程序的所有配置、设置和声明信息。例如，在这里可以将搜索合约(search contract)设置为可用，或者设置不同情况下的图标。它同样定义了默认的背景色，方向和程序需要的特殊功能，如位置访问。

确保熟悉这个文件，以后会经常用到。



### 1. 4. StandardStyles.xaml

这个文件有 1830 行代码，提供了许多默认的 XAML style。这些 Styles 可用于 text boxes、radio buttons，application bar 图标等。强烈讲义慢慢的熟悉这些 styles，另外，可以根据需要修改这些 styles。

如果你在这个文件中添加自己的 styles，那么请记住，你添加的这些 styles，程序中的所有 page 都可以使用。这样归功于下一个文件：App.xaml。

### 1. 5. App.xaml

这个文件是程序运行的起点(万恶之源)。当程序启动时，首先加载的就是这个文件，它包含了应用程序 app-level 的资源 and 设置信息。默认情况下，这个文件包含了一行重要的代码：

```
<ResourceDictionarySource="Common/StandardStyles.xaml"/>
```

这行代码以 ResourceDictionary 的方式加载 StandardStyles.xaml 文件，这样可以确保程序中的每个 page 都可以使用 StandardStyles.xaml 中的 styles。App.xaml 真正的魅力是在 code-behind 文件中：App.xaml.cs。下面我们来看看这个文件。

### 1. 6. App.xaml.cs

这个文件跟 Windows Phone 中的一样，是程序运行的起点。应用程序启动的方法就在这里面，比如 OnLaunched()，OnSuspending()等。这个文件同样是设置启



动页面的地方。在 OnLaunched()方法中，你可以看到类似如下的几行代码：

```
if (!rootFrame.Navigate(typeof(MainPage), args.Arguments))
{
    throw new Exception("Failed to create initial page");
}
```

“MainPage”这个参数涉及到下一个文件：MainPage.xaml。

### 1.7. MainPage.xaml

这个文件是应用程序的默认“主页”。当 splash screen 加载完毕之后，用户第一眼看到的界面就是 MainPage.xaml。在工程中，它除了包含一个使用了 style 的 Grid，别的什么都没有。你可能希望立即给这个 page 添加更多的内容。这里提供了许多很棒的控件，建议你以 [GridView](#) 或 [FlipView](#) 控件作为开端，但是具体用来做什么取决于你的程序。

### 1.8. MainPage.xaml.cs

跟 MainPage.xaml 类似，这个文件实际上也是空的。只有一个 page 的构造函数，和一个 OnNavigatedTo()事件处理函数，在 Windows Phone 中，这个方法总是使用这个方法(强烈建议)，但是现在，它是默认就有了，意味着许多开发者将用其取代 page 的构造函数。

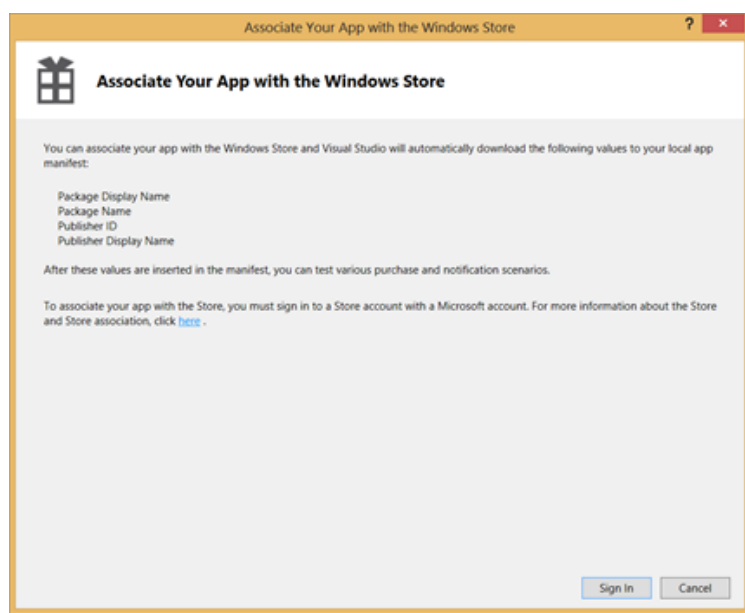
### 1.9. Day1-TheBlankApp\_TemporaryKey.pfx



每一个 Windows Store app 都使用证书进行签名。在 Visual Studio 中第一次创建新的工程时，将会自动的创建一个测试证书。在这里，测试证书是 Day1-TheBlankApp\_TemporaryKey.pfx ,其中 Day1-TheBlankApp 是我的程序名称。在 package.appxmanifest 文件的打包(Packaging) tab 中，如果需要，你还可以创建一个新的证书。

当你准备提交应用到商店时，你需要将应用程序与微软商店和你的开发者账号关联起来。可以通过下面的步骤进行：

Projects->Store->Associate App with the Store



在这里[签名一个应用程序包](#)，你会找到更多信息。

### 1. 10. 总结



第一日，至此结束！这里，我们只是开启了本系列学习，但是，现在你已经对 Blank App template for XAML 创建的 app，有了很好的认识。如果你还没有安装 Visual Studio 2012，但是仍然想看一下相关代码，你可以来这里下载：



明天，我们主要学习 orientation 和 snapping。在程序中，经常遇到的两个重要内容。如果你还没有准备好，那么可以从下面的链接中下载到免费的开发工具。



感谢你的阅读！

如果对这篇文章有什么想法想法，可以与破船联系，破船的联系方式在文章开头。

破船

