



31 Days of Windows 8

Windows 8 开发 31 日

第 03 日

启动画面

译者：BeyondVincent(破船)

时间：2013.4.23

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

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 03 日启动画面 5

 1.0. 介绍5

 1.1. 开始5

 1.2. 创建一个启动画面页面..... 7

 1.3. 修改 App.xaml.cs 8

 1.4. 更难、更完善的方法..... 9

 1.5. 订阅启动画面的消失事件..... 10

 1.6. 确定启动画面图片的位置和尺寸.. 11

 1.7. 总结 14

第 03 日启动画面



1.0. 介绍

今天，我们来学习一个很小的知识点，但它是在应用程序生命周期中非常重要的一部分：启动画面。可能你会认为这是一个差劲的主题，但是实际上，站在用户体验的角度，它非常重要。

比如，一个天气应用程序会连接网络，并获取天气数据。每次用户启动这个程序时，都希望获得最新的数据。你也知道，获取数据、正确的图片等信息大概需要 2-3 秒，或者更多。

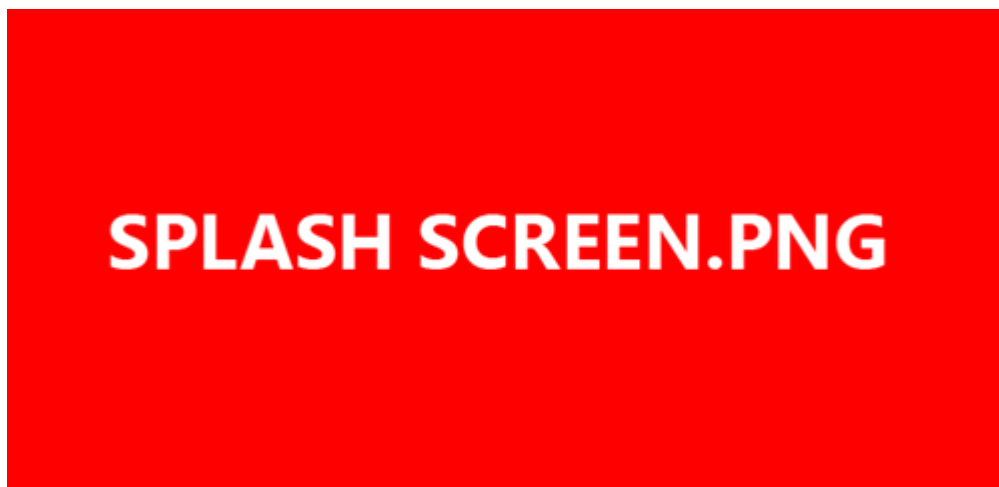
通过使用 splash screen 和它的事件，我们可以“加强”启动画面的体验，这样用户在程序没准备好的时候，不会看到程序的画面。

所以，在本文中，我们要做的是获得启动画面图片的位置，并用我们自己的图片替换了它，直到程序准备完毕。

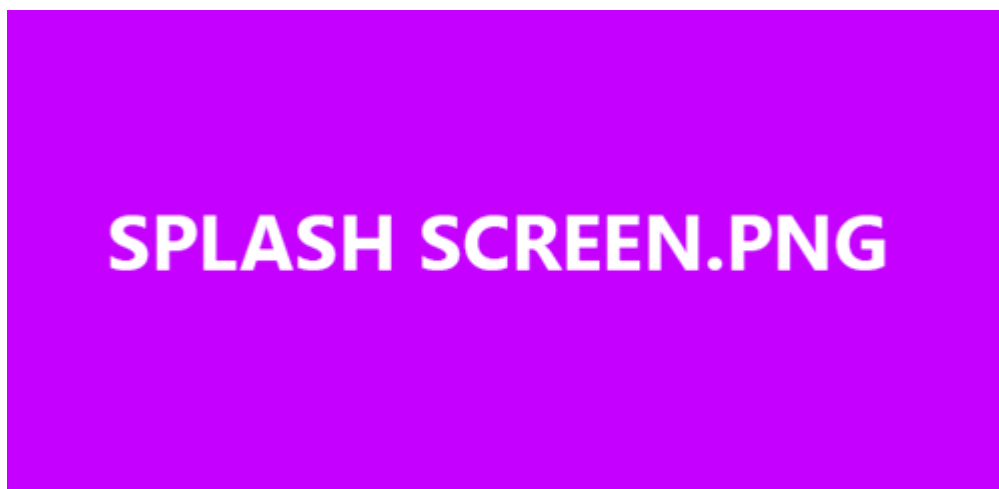
1.1. 开始

回想一下第 01 日讲的内容，splash screen 图片尺寸总是为 620 x 300 pixels。

在这这里的示例中，我准备了两个相关图片，第一个是红色背景的图片：



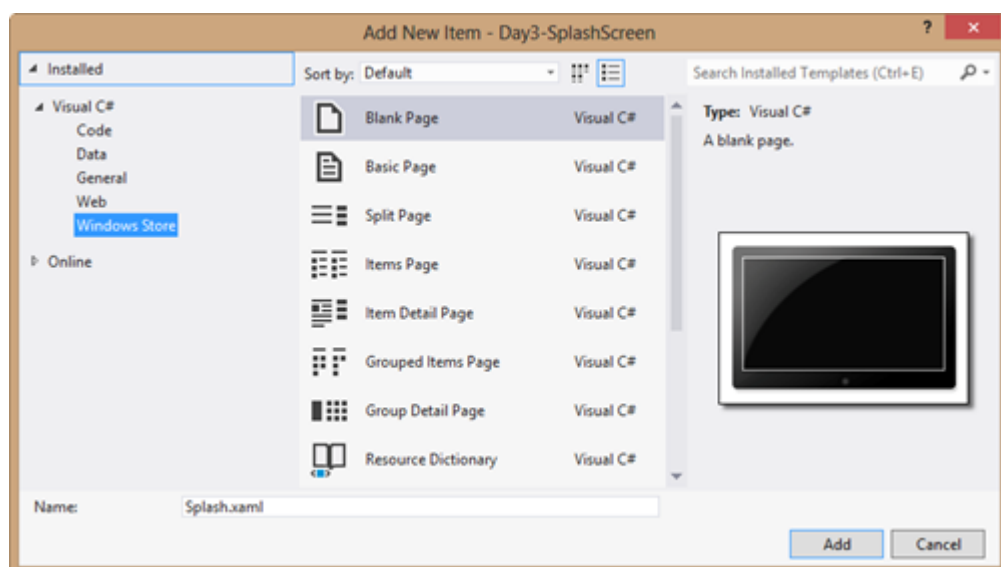
另外，还有紫色背景的一个图。这样在切换的时候，我们可以区分两个图。



新建一个“blank”工程，并将这两个图添加到“Assets”文件夹中。替换图片之后，如果你运行工程，会看到红色的那个图片显示 1-2 秒钟，然后程序会加载一个黑色的屏幕。

1.2. 创建一个启动画面页面

为了完成 Splash Screen 介绍，下一步是创建一个单独的 page。如下图，添加一个新的 Blank page 到工程中：



将 page 命名为 Splash.xaml，当然，你可以命名为你希望的名称，这个没有限制。然后将红色的图片添加到 page 中，如下代码：

```
<Page
  x:Class="Day3_SplashScreen.Splash"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:Day3_SplashScreen"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="#464646">
```



```
<Image x:Name="SplashImage" Source="Assets/SplashScreenPlaceholder.png" Width="620" Height="300" />
</Grid>
</Page>
```

如上，看起来很简单。只是在 page 中添加了一个图片。下一步是修改 App.xaml.cs 文件：首先导航到这个 page。

1.3. 修改 App.xaml.cs

一般，开发者在创建一个简单的程序是，基本不会打开这个文件，但是在这里面经常会有奇迹发生。App.xaml.cs 包含我们的启动方法以及系统级别的功能，例如管理不同程序的 session。在这里，我们定位到 OnLaunched()方法。我已经把默认添加的注释移除了。在这里，你可以看到程序这样的代码：Navigate(typeof(MainPage), args.Arguments)。

```
protected override void OnLaunched(LaunchActivatedEventArgs args)
{
    Frame rootFrame = Window.Current.Content as Frame;

    if (rootFrame == null)
    {
        rootFrame = new Frame();

        if (args.PreviousExecutionState == ApplicationExecutionState.Terminated)
        {
            //TODO: Load state from previously suspended application
        }

        Window.Current.Content = rootFrame;
    }

    if (rootFrame.Content == null)
    {
        if (!rootFrame.Navigate(typeof(Splash), args.Arguments))
        {

```




```
throw new Exception("Failed to create initial page");
    }
}
Window.Current.Activate();
}
```

默认情况下，程序会导航到 MainPage.xaml，在这里，我将其修改为 Splash。这将会首先导航只 Splash 画面，然后在 Splash 画面中可以加载我们想要的内容。

继续运行程序。会看到当红色画面过后，会显示紫色的画面。

到此，我们的启动画面做完了吗？当然还没有。现在我们已经做了一个非常简单的延长启动画面。如果你感兴趣的话，这里有一个稍微复杂的 model 供我们使用。

1. 4. 更难、更完善的方法

可能你想知道 splash screen 什么时候消失，或定位启动画面在屏幕中的位置。

通过 [SplashScreen](#) 类，这是可以做到的。该类在名称空间 [Windows.ApplicationModel.Activation](#) 里面。

首先，我们再来看看 App.xaml.cs 文件。我将修改 Splash.xaml 页面的调用方式，并将 SplashScreen 对象传递到 page 的构造函数中。为了简单，我重写了 OnLaunched 方法的所有代码。如下：

```
protected override void OnLaunched(LaunchActivatedEventArgs args)
{
    //First we want to check to see if the application was already running.
    if (args.PreviousExecutionState != ApplicationExecutionState.Running)
    {
        //If it wasn't, then we want to check to see if it was terminated the last time it was run,
        //which we will pass on to our Splash page.
        bool loadState = (args.PreviousExecutionState == ApplicationExecutionState.Terminated);
        //Create a new Splash page object passing the SplashScreen object to the constructor.
        Splash splashPage = new Splash(args.SplashScreen, loadState);
        //Set our current app's content = the new Splash page.
        Window.Current.Content = splashPage;
    }
}
```



```
}  
Window.Current.Activate();  
}
```

现在已经将数据传递给 page 了,我们还需要修改一下 Splash.xaml.cs 的构造函数,这样才能获取到传递过来的数据。打开 Splash.xaml.cs 文件,并重写 public Splash()方法,如下代码:

```
public SplashScreen splashScreen;  
public Rect splashImage;  
  
public Splash(SplashScreen splashscreen, bool loadState)  
{  
    this.InitializeComponent();  
  
    splashScreen = splashscreen;  
    splashImage = splashScreen.ImageLocation;  
}
```

如上代码。因为我在 App.xamlcs 文件中,修改了 Splash page 的调用方法,因此在 Splash.xaml.cs 文件中,也需要修改一下构造函数的签名,以获取到 SplashScreen 对象和一个 boolean 值。LoadState 值传递给 page 的目的是让 page 能够判断出程序的启动状况。

1.5. 订阅启动画面的消失事件

下一步是订阅 SplashScreen 消失的事件。这一步跟其它 event handler 类似,将如下代码添加到构造函数的最后:

```
splashScreen.Dismissed += new TypedEventHandler<SplashScreen, Object>(splashScreen_Dismissed);
```



因为定义了一个新的 event handler，同样需要创建一个方法来捕获事件，命名为 `splashScreen_dismissed`。在这个例子中，该方法是空的，不过，当你想要在 splash screen 消失时，执行一些代码，那么就写在这里面。

```
void SplashScreen_Dismissed(SplashScreen sender, object args)
{
}
```

1.6. 确定启动画面图片的位置和尺寸

最后，需要确定 splash screen 图片在屏幕中的位置。在这里，会根据启动画面的位置，来布局我的广告图片。

首先，下面广告图使用的是下面这个图片。



将其添加到工程中，然后更新一下 `Splash.xaml` 文件，如下代码：

Add this to your project, and then update your `Splash.xaml` file to look like this:

```
<Page
  x:Class="Day3_SplashScreen.Splash"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:Day3_SplashScreen"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">
```



```

<Canvas Background="#464646">
<Image x:Name="SplashScreenImage"
Source="Assets/SplashScreenPlaceholder.png"
Width="620"
Height="300"
Visibility="Collapsed"/>

<Image x:Name="AdvertisementImage"
Source="Assets/advertisementsample.png"
Width="620"
Height="100"
Visibility="Collapsed"/>
</Canvas>
</Page>

```

在上面的 xaml 中，我移除了 Grid 控件，并用 Canvas 替代。这样我可以对图片自行定位，不再使用 Grid 的默认居中属性。

在 Splash.xaml.cs 文件中，添加一行代码到构造函数中（调用 PositionAdvertisement），至此，最终版看起来如下：

```

public SplashScreen splashScreen;
public Rect splashImage;

public Splash(SplashScreen splashscreen, bool loadState)
{
    this.InitializeComponent();

    splashScreen = splashscreen;
    splashImage = splashScreen.ImageLocation;
    splashScreen.Dismissed += new TypedEventHandler<SplashScreen, Object>(splashScreen_Dismissed);
    PositionAdvertisement();
}

void splashScreen_Dismissed(SplashScreen sender, object args)
{
}

```

最后，需要创建 PositionAdvertisement 方法。在显示出图片之前，通过访问 SplashScreen.ImageLocationd X、Y、Height 和 Width 属性，将图片移动到合适的位置。



置。

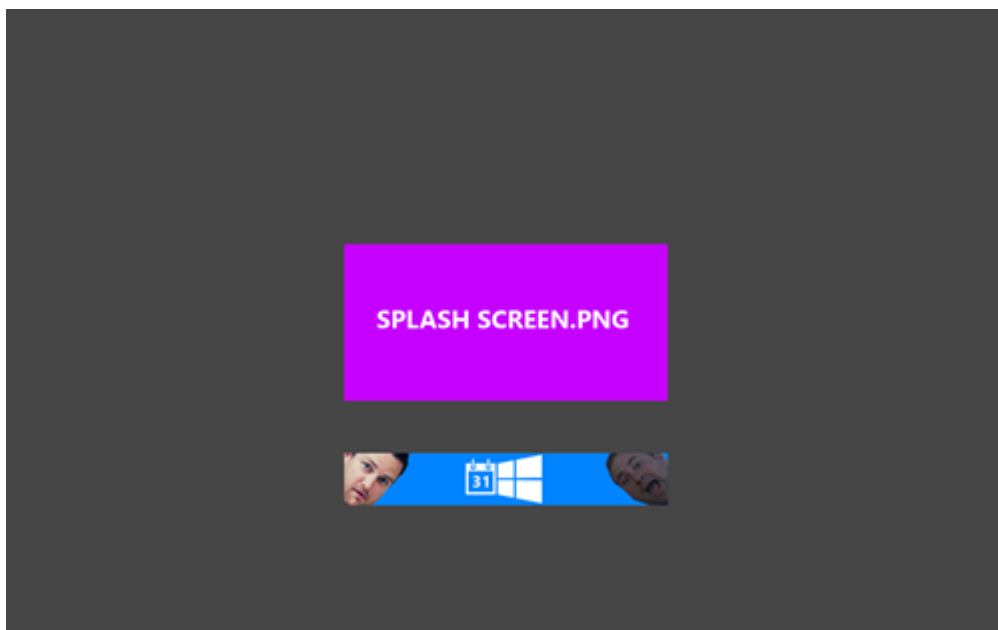
```
private void PositionAdvertisement()  
{  
    SplashScreenImage.SetValue(Canvas.TopProperty, splashImage.Y);  
    SplashScreenImage.SetValue(Canvas.LeftProperty, splashImage.X);  
    SplashScreenImage.Height = splashImage.Height;  
    SplashScreenImage.Width = splashImage.Width;  
    SplashScreenImage.Visibility = Visibility.Visible;  
  
    AdvertisementImage.SetValue(Canvas.TopProperty, (splashImage.Y + splashImage.Height + 100));  
    AdvertisementImage.SetValue(Canvas.LeftProperty, splashImage.X);  
    AdvertisementImage.Visibility = Visibility.Visible;  
}
```

从上面的代码，可以看出，SplashScreenImage 的位置与默认启动画面相同。

然后将

Advertisement 图片显示在 SplashScreenImage 下方 100pixels 的地方。所以，

Splash.xaml 的最终效果如下图：



就是这样的！



译者注：在启动画面显示时，如果完成了数据的加载等之后，需要切换到指定画面，可以这样做：

```
Frame frame = newFrame();  
frame.Navigate(typeof(MainPage));
```

上面代码的调用一般放在一个用户操作事件里面，比如触摸屏幕。不能将上面的代码放到 `splashScreen_Dismissed` 方法中处理，否则会抛出异常。

1.7. 总结

简而言之，启动画面可以作为很好的朋友。在程序准备时，当做一层临时的面纱，挡住程序，并且是房子广告的好地方。这是一个非常有价值的地方，值得考虑。

点击下面的图标，可以下载到本文的完整示例：



明天，我将介绍 Windows 8 XAML 开发中，一些常用的，新的控件。比如 GridView、FlipView 等。



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

