



31 Days of Windows 8

Windows 8 开发 31 日

第 21 日

摄像头

译者：BeyondVincent(破船)

时间：2013.4.25

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 21 日摄像头 5

 1.0. 介绍5

 1.1. 更新 Manifest5

 1.2. 获取相片6

 1.3. 获取视频 11

 1.4. 总结 13

第 21 日摄像头

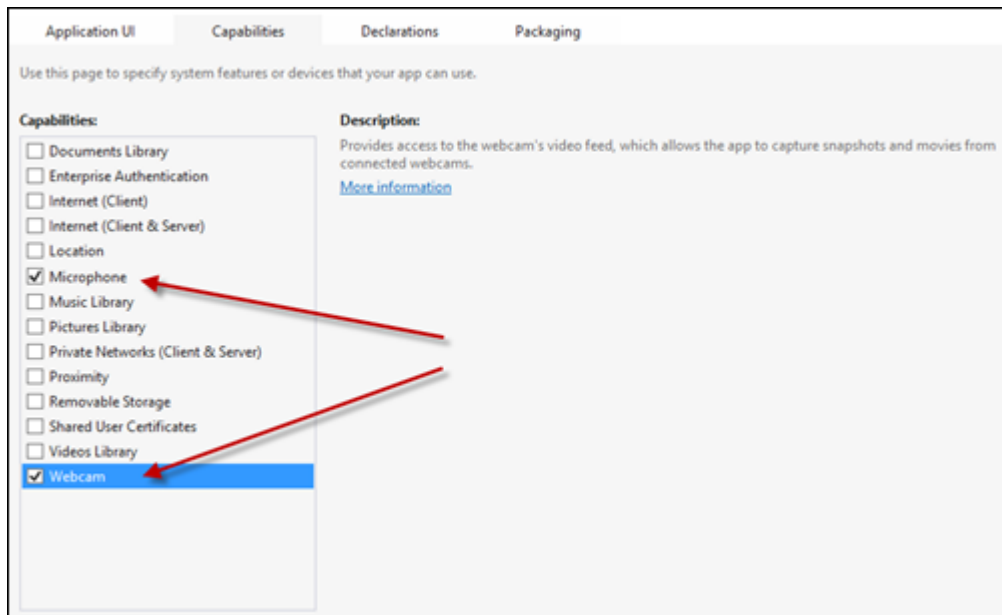


1.0. 介绍

今天,我们将开始学习 Windows 8 中的传感器,在之后的好几篇文章,都将涉及到传感器。首先我将从摄像头开始介绍。摄像头可以照相也可以录制视频,本文我将介绍摄像头的这两个功能。

1.1. 更新 Manifest

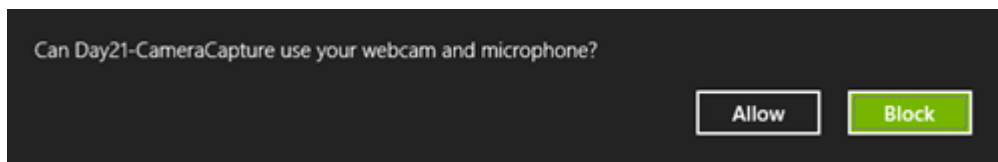
本系列的主题中,我们必须从更新 `package.appmanifest` 文件开始,本文让 webcam 和 microphone 可用。只需要在 Capabilities 中勾选对应的框即可。如下:



如果不按照上面的设置，那么这些功能是不可以使用的，所以，确保首先设置好了。

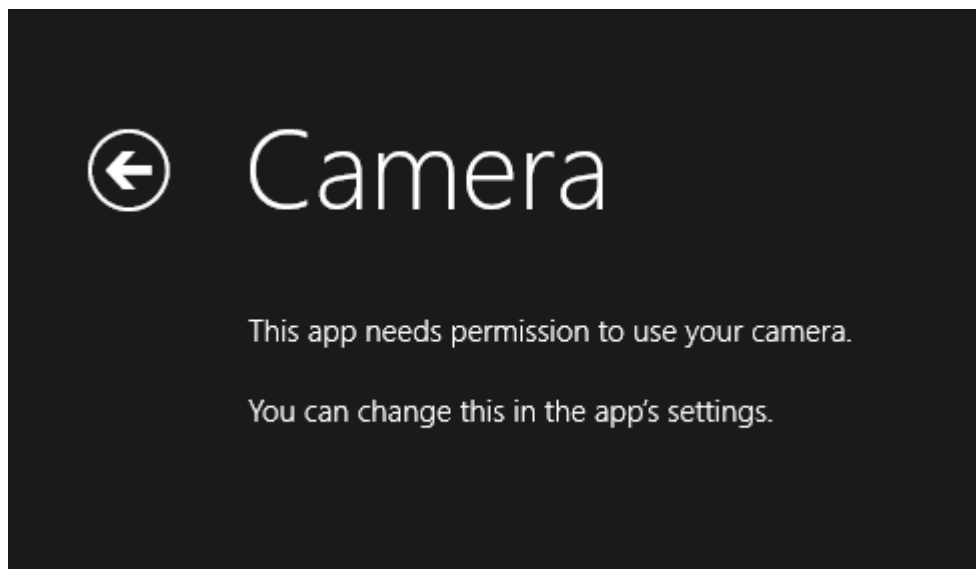
1. 2. 获取相片

在 Windows 8 中，非常好的是摄像头的所有事情都由你来决定。也就是说，有些事情，你不得不考虑好了。首先，当你试图访问摄像头时，程序将会提示用户将使用摄像头和麦克风，由用户决定是否授权。

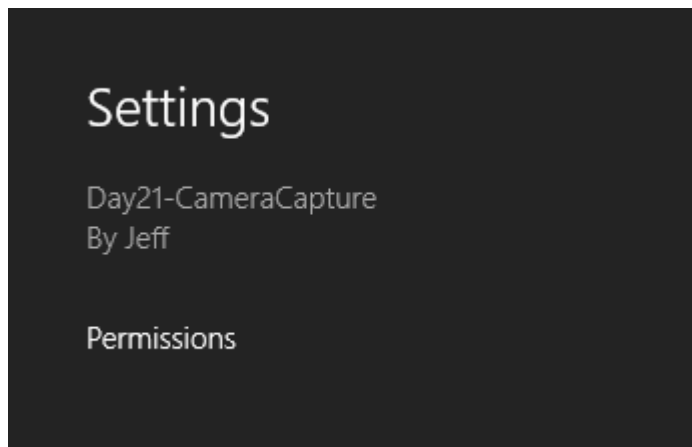


有点类似锁屏的访问，这个提示你只能获得一次。不过还好，不管什么原因，如果用户点击“Block”，摄像头显示的对话框会提示用户在设置中进行修改权限，

如下：

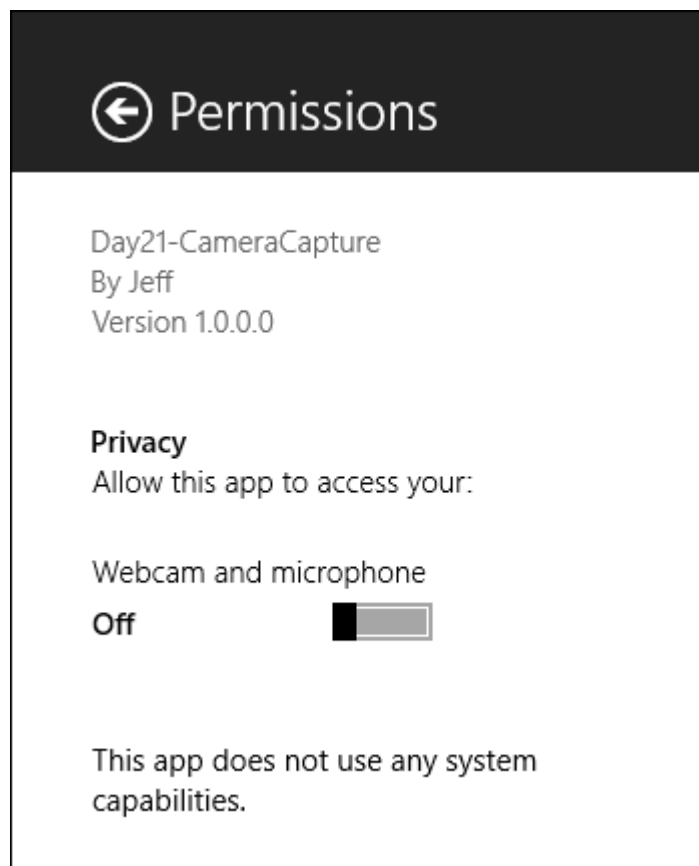


如果用户打开他们的设置面板，会发现一个“Permissions”选项，如下：



点击“Permissions”会打开如下菜单，即可设置相关权限：





用户总是有权限打开或者关闭这个选项。下面是从张相机获取一张图片的代码，摄像头返回的是一个 StorageFile 对象：

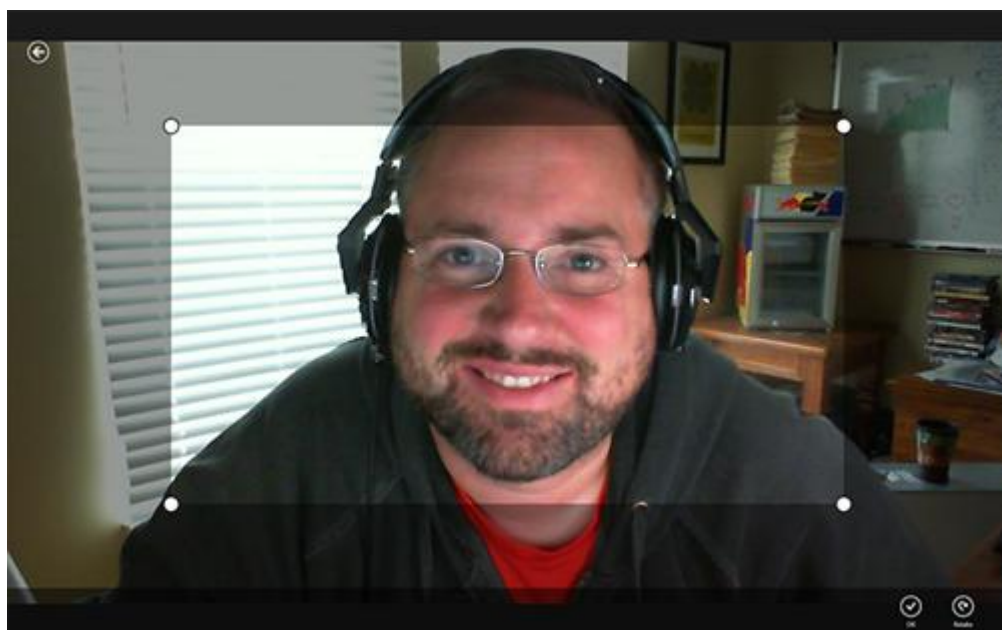
```
CameraCaptureUI camera = new CameraCaptureUI();  
camera.PhotoSettings.CroppedAspectRatio = new Size(16, 9);  
StorageFile photo = await camera.CaptureFileAsync(CameraCaptureUIMode.Photo);
```

从技术角度出发，上面只有 2 行代码是必须的。第 2 行代码—设置 CroppedAspectRatio 不是必须的，不过在这里我建议设置一下。当摄像头对话框打开时，提供了一个全屏画面，如下：

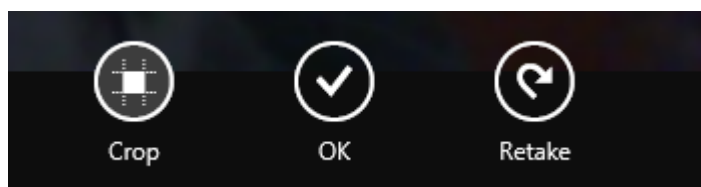




这里有几个选项可以设置（比如分辨率和计时器），然后通过 tap 或者单击屏幕，就可以触发照相。如果我们设置了一个 `CroppedAspectRatio`，那么将会看到如下画面：



可以看到画面中有 4 个白色的圆圈，表示裁减的区域。用户可以改变这个 box 的 size，不过指定一个 `CroppedAspectRatio`，我们可以强制用户选这个具有一定比例的图片。在示例中，我使用 16:9，如果 1:1 这是一个正方形图片。没有这个选项的话，用户就不必裁减图片，不过在 bottom 中的菜单选项中，有一个任意裁剪的选项：



如果用户选择 Crop，那么用户就可以裁减图片的任意内容：



最后，当用户点击 OK 按钮，将会返回一个图片到我们程序中，这个图片我们可以随意使用。

下面是我示例中所有相关代码：

```
CameraCaptureUI camera = new CameraCaptureUI();
camera.PhotoSettings.CroppedAspectRatio = new Size(16, 9);
StorageFile photo = await camera.CaptureFileAsync(CameraCaptureUIMode.Photo);

if (photo != null)
{
    BitmapImage bmp = new BitmapImage();
    IRandomAccessStream stream = await photo.OpenAsync(FileAccessMode.Read);
    bmp.SetSource(stream);
    ImageSource.Source = bmp;
    ImageSource.Visibility = Visibility.Visible;
}
```

如上所示，前三行是获取相片，之后我检查是否真的获取到数据了。如果返回了数据，我将数据读取到一个 `BitmapImage` 对象中，然后将这个 `BitmapImage` 对象设置给我页面中的 `Image` 控件（`ImageSource`）。最后，我将 `Image` 控件显示出来，这样用户就可以看到图片了。上面就是在 Windows 中使用 C# 和 XAML 获取相片所需做的事情。下一步，我将用相同的过处获取视频。

1.3. 获取视频

很好的时，视频的获取过程与图片类似。有点不同的是当数据返回的时候，如何使用。

在之前介绍获取相片时，由于相片格式有限（JPG 或 PNG），所以我没有介绍。而在获取视频的时候，我指定了具体的视频类型。如下是获取视频的代码：

```
CameraCaptureUI videocamera = new CameraCaptureUI();
videocamera.VideoSettings.Format = CameraCaptureUIVideoFormat.Mp4;
videocamera.VideoSettings.AllowTrimming = true;
videocamera.VideoSettings.MaxDurationInSeconds = 30;
```



```
videocamera.VideoSettings.MaxResolution = CameraCaptureUIMaxVideoResolution.HighestAvailable;  
  
StorageFile video = await videocamera.CaptureFileAsync(CameraCaptureUIMode.Video);  
  
if (video != null)  
{  
    IRandomAccessStream stream = await video.OpenAsync(FileAccessMode.Read);  
    VideoSource.SetSource(stream, "video/mp4");  
    VideoSource.Visibility = Visibility.Visible;  
}
```

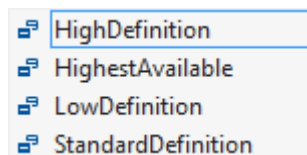
下面来说一下上面的代码。首先可以指定视频的格式：MP4 或 WMV。一般你是看不出这些格式的区别的。MP4 要比 WMV 使用更流行，但是 MP4 文件会很大，所以值得考虑一下。

其次，我将 AllowTrimming 设置为 true，这样用户可以选择根据喜好，选择录制视频的片段。如下是相关截图，用户可以拖动白色的裁减圈来裁减视频：



第三，我指定了一个 MaxDurationInSeconds，这个值显示用户录制视频的最长时间。如果用户录制的视频比这个时间长，那么用户只能选择 30 秒（或者是你指定的一个长度）的视频。

最后，我还指定了一个 MaxResolution，这是管理用户可以录制的分辨率，这也决定了最终文件的大小，有下面四个可选项：



获取到视频之后，我设置了我的 MediaElement 的 source，将新的数据流设置



给 VideoSource，并将其显示出来。

1. 4. 总结

简单的来说，从用户摄像头获取相片和视频的很容易的，并且可以定制。大多数复杂的 UI 都是有 Windows 8 直接处理了，所以，我们不用担心与用户的交互问题。

点击下图，下载本文示例代码：



明天，我将介绍非常酷的一个技术 Play To。利用这个技术，我们可以将设备中的流媒体传递给电视机，Xbox 设备等。到时候见！



感谢你的阅读！

如果对这篇文章有什么想法，可以与破船联系，破船的联系方式在文章开头。

破船

