



31 Days of Windows 8

Windows 8 开发 31 日

第 11 日

锁屏程序

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 11 日锁屏程序 5

 1.0. 介绍5

 1.1. 配置 Manifest 文件 6

 1.2. 请求允许 8

 1.3. 确保拥有权限 10

 1.4. 更新锁屏 10

 1.5. 总结 13

第 11 日锁屏程序



1.0. 介绍

今天我们来学习 Windows 8 中的最后一种通知：锁屏 (TheLock Screen)。如果你在设备上运行了 Window 8，应该已经很熟悉漂亮的锁屏了。锁屏显示了你收到了多少封 email，Facebook 消息，闹钟，以及连接状态。下面是我的锁屏截图：



对的，这是我屏幕上的一张照片。今天早晨我花了 30 分钟来获取锁屏截图，可是没有成功 (yes，我知道我可以开启一个 VM，或者远程进入我的机器，但是

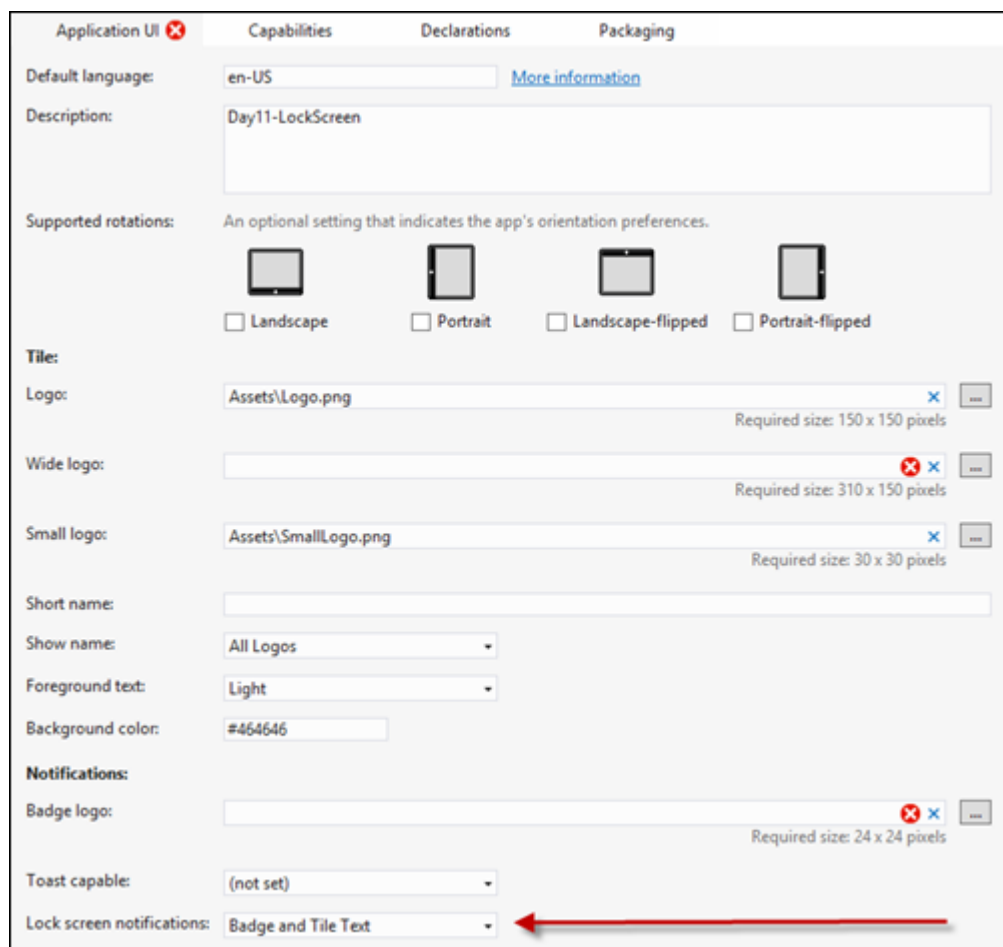


现在没有任何条件。)

作为一个开发者，可以给这个屏幕添加一个徽章 (badge)，不过单独去做的时候，会有危险。看看这个链接的内容吧：[Microsoft's guidance on the best ways to utilize the Lock Screen notification area](#)。

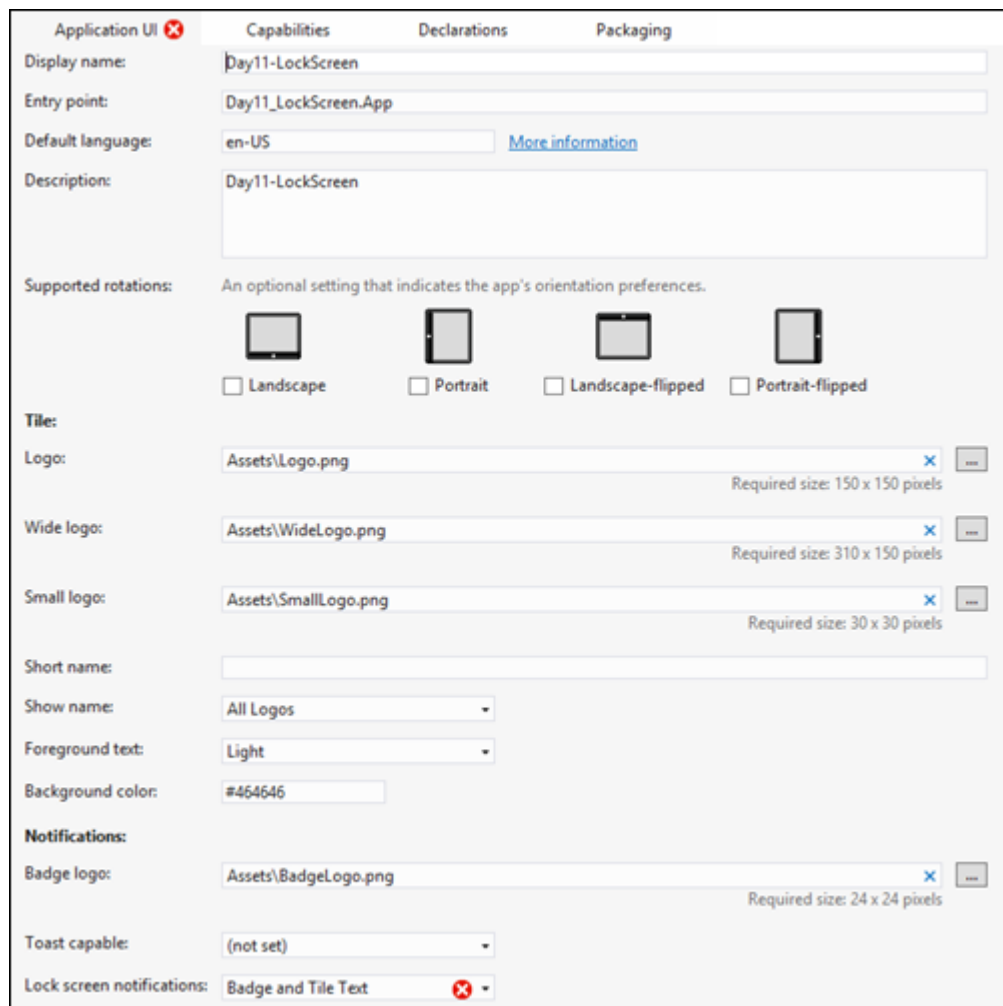
1.1. 配置 Manifest 文件

首先，我们需要设置 package.appxmanifest 文件，让程序支持锁屏通知。



当你设置 Lock screen notification 时，会看到有红色 X 提示。当 lock screen

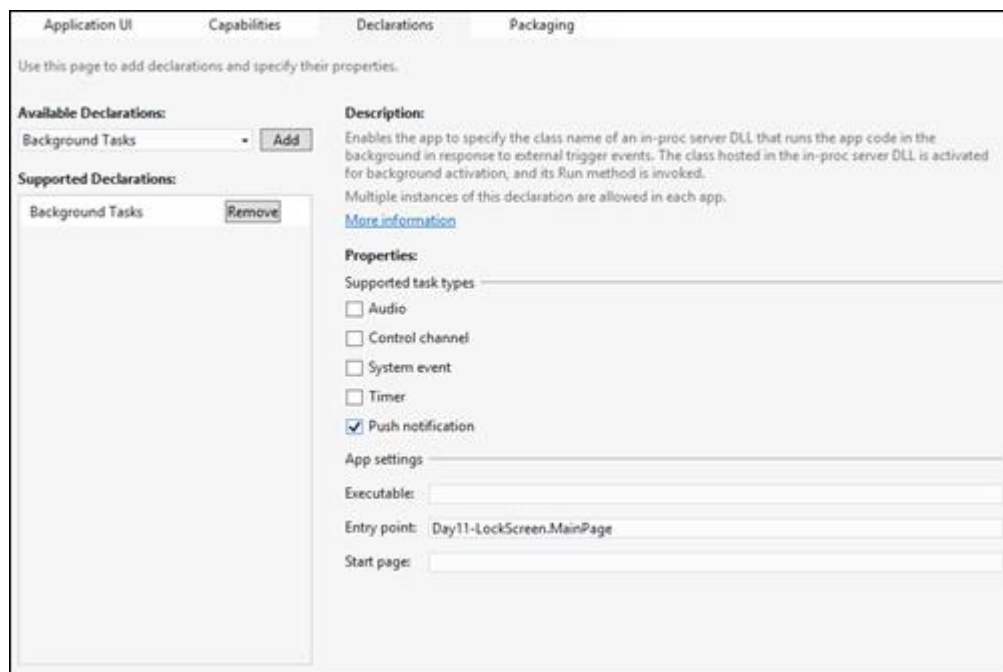
notification 可用时, 我们需要提供一个 Badge Logo, 另外还需要有一个 Wide Logo 图片。下图是我已经设置了这两个图片后的截图:



上图中, 你会发现这里有一个新的红色 X, 这是让你知道“如果要想“锁屏通知可用, 你必须指定一个或者多个后台任务, 后台任务可以声明为 Timer, Control channel 或 Push notification”

这以为着锁屏程序一般是希望通过后台 Agent 进行更新。切换到 Declaration 选项卡, 然后添加后台任务, 如下:





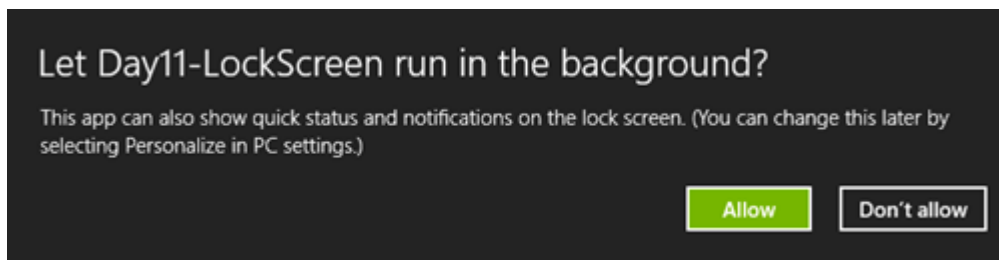
在我的示例程序中，我并不实际的创建后台任务。后台任务我将在明天进行介绍。相反，我将关注与允许我们更新锁屏信息的流程和代码。

1.2. 请求允许

在开始写相关更新锁屏信息之前，你需要获得相关权限。现在，你也可以不必进行权限请求，但是不主动进行权限请求，那么更新锁屏通知的路将会是冒险加危险的，原因如下：

你只有一次机会来询问用户获得访问锁屏的权限。如果你进行请求，但是用户选择“Don't Allow”，那么你将在也没有能力进行权限请求了。下面是请求权限的一个截图：

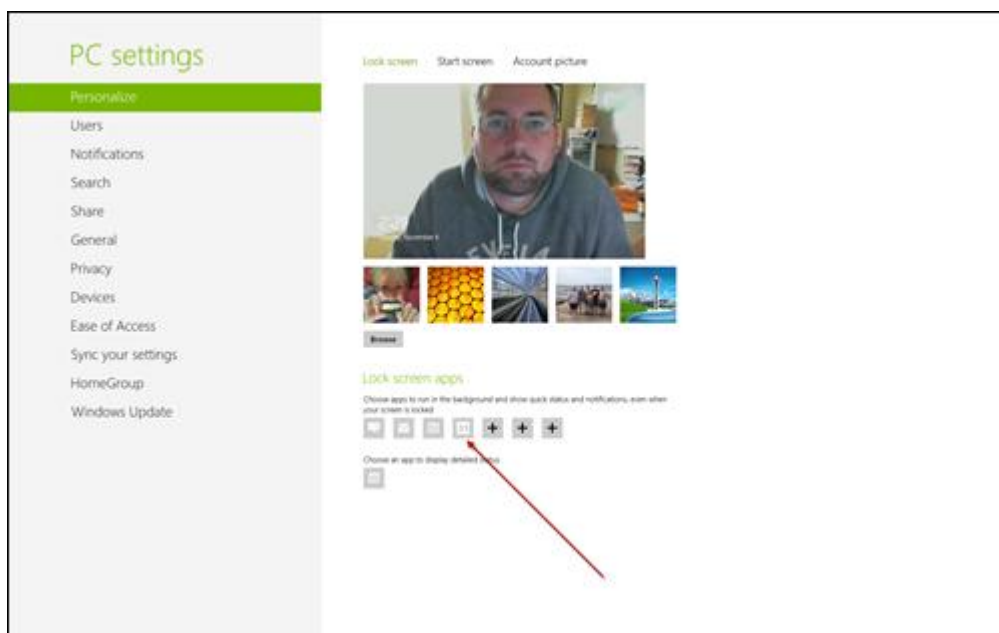




下面是如何请求的相关代码：

```
protected async override void OnNavigatedTo(NavigationEventArgs e)
{
    BackgroundAccessStatus status = await BackgroundExecutionManager.RequestAccessAsync();
}
```

如果你永远都不进行请求，唯一获得程序显示在锁屏上的方法就是通过用户打开他们的 PC 设置，然后明确的将你的程序添加进去，如下：



用户可以最多选择 7 个程序显示在锁屏上。

1. 3. 确保拥有权限

这不是一个必须的步骤。你可以让你的程序或者后台任务（明天将介绍）持续发送更新信息，即使没有权限。不过有可能你的更新用户永远都看不到，因为没有权限的话，你的更新不会显示在锁屏中。

下面是如何检查用户是否授权可以更新锁屏：

```
BackgroundAccessStatus status = BackgroundExecutionManager.GetAccessStatus();  
  
if ((status == BackgroundAccessStatus.AllowedWithAlwaysOnRealTimeConnectivity) ||  
    (status == BackgroundAccessStatus.AllowedMayUseActiveRealTimeConnectivity))  
{  
    // send you update here.  
}
```

到这里，我们已经向用户发出了权限请求，然后也检查了用户是否已经授予相关权限。下一步将是实际的发送更新。

1. 4. 更新锁屏

发送一个锁屏通知与之前的两篇文章十分类似（第 09 日——动态磁贴，第 10 日——Toast 通知）。跟前面的示例一样，我将获得一个有 xml 构成的模板（在本示例中，非常简单）：

```
<badgevalue="badge_value"/>
```

下一步，需要设置想要显示的值。在这里有几个选项可选。不仅可以使使用 1-99 之间的任意数字，还可以从字形（glyphs）列表中选择一个。最方便的是模板比较

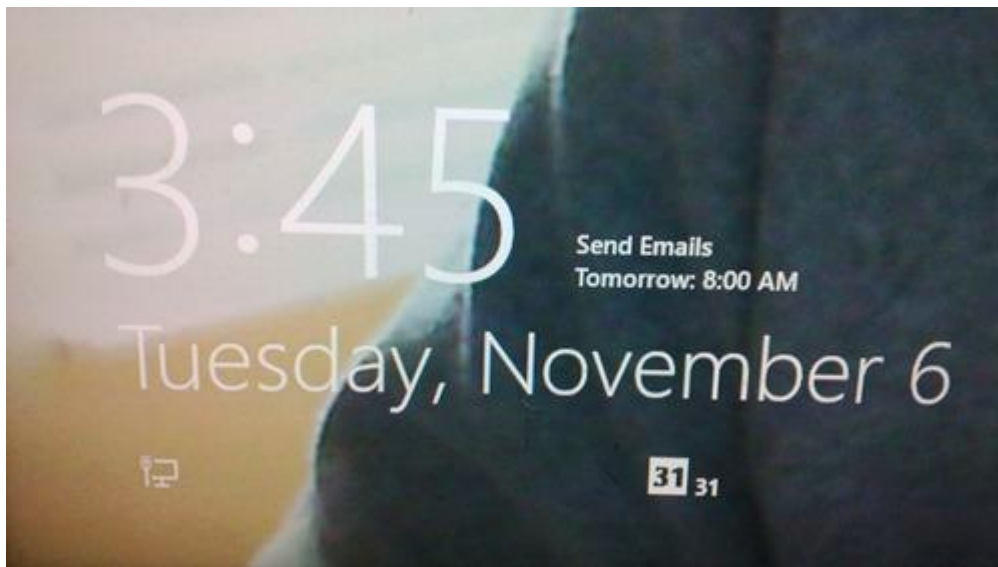


智能，能够识别出值的不同，因此我们只需要在代码中修改实际的值即可。下面是 badge 的更新：

```
XmlDocument badgeData = BadgeUpdateManager.GetTemplateContent(BadgeTemplateType.BadgeNumber);
XmlNodeList badgeXML = badgeData.GetElementsByTagName("badge");
((XmlElement)badgeXML[0]).SetAttribute("value", "31");












BadgeNotification badge = new BadgeNotification(badgeData);
BadgeUpdateManager.CreateBadgeUpdaterForApplication().Update(badge);
```

可以看到在 SetAttribute()方法的调用中，我传递了 31 这个值进去。下面是锁屏中的一个截图：



如果你想要使用字形 (glyphs)，下面的连接中是可以用的字形：[courtesy of the Badge overview page on MSDN](#)。



Status	Glyph	XML
none	No badge shown	<badge value="none"/>
activity		<badge value="activity"/>
alert		<badge value="alert"/>
available		<badge value="available"/>
away		<badge value="away"/>
busy		<badge value="busy"/>
newMessage		<badge value="newMessage"/>
paused		<badge value="paused"/>
playing		<badge value="playing"/>
unavailable		<badge value="unavailable"/>
error		<badge value="error"/>
attention		<badge value="attention"/>

为了指定字形，我们只需要修改 SetAttribute()方法的值即可，如下：

To specify a glyph, we only need to change the value in our SetAttribute() method, like this:

```
((XmlElement)badgeXML[0]).SetAttribute("value", "Playing");
```

这就是关于锁屏程序...希望今天你学到了点东西。稍等...这还有一个事情我忘记说了，当我在 appxmanifest 文件中选择“Badge and Tile Text”时，那么 Tile 也会显示创建的 Badge。（这就是为什么需要一个 WideTile.png）。下面是动态磁贴中看到的“Playing”徽章：



1.5. 总结

我们可以在任何时候从程序中更新我们的锁屏信息，但是棘手的是需要获得用户的许可。一旦你获得了许可，更新锁屏信息就非常的简单。

点击下图，可以下载到本文的示例代码：



明天，我将开始开涉及到过去 3 天所谈论到的后台任务。我将介绍如何更新从后台任务更新 Tile、Toast 和锁屏通知。明天见！



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

