



31 Days of Windows 8

Windows 8 开发 31 日

第 10 日

Toast 通知

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 10 日 Toast 通知 5

 1.0. 介绍5

 1.1. 让 Toast 通知在程序中可用6

 1.2. 在程序中发送一个 Toast 通知7

 1.3. 设置 Toast 通知的声音 (以及更多 !) 8

 1.4. 总结 10

第 10 日 Toast 通知



1.0. 介绍

昨天我们学习了动态磁贴，它使你的程序如何让用户知道重要的相关信息。

今天，我们来看另外一种不同的用户通知：Toast 通知。

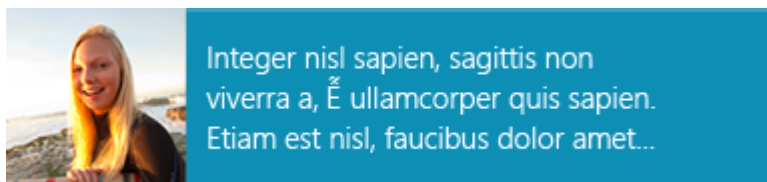
如果你之前没有用过 Toast 通知，可能会觉得 Toast 这个词有点傻。在程序中，当有一些事情发生时，可以将信息弹出在用户前端，让用户知晓。

Toast 通知与动态磁贴的主要不同点就是无论用户当前运行的是哪个程序，Toast 消息都会出现在用户的屏幕上，而动态磁贴必须在开始屏幕才能看到。

Toast 消息更多的设计目的是：你现在需要知道这个消息！，而动态磁贴则是属于被动发现的消息。当你发送 toast 消息时，你需要考虑它的设计目的。如果你发送的是类似垃圾邮件的消息，那么用户是可以关闭你发送的消息的，更糟糕的是，用户会卸载你的程序。在发送 Toast 消息是，你最好花点时间阅读一下：[Microsoft' guidance on toast notifications](#)。

下面我给出两个 Toast 通知，供参考：

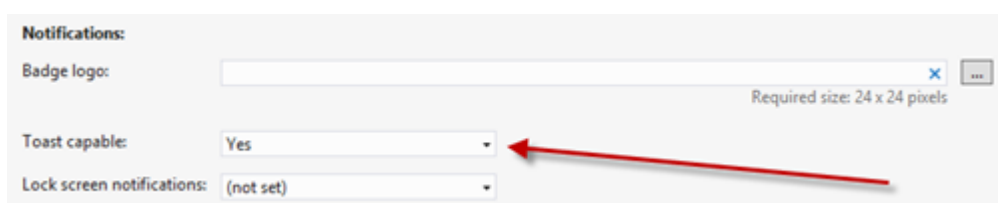
Integer nisl sapien, sagittis non viverra a, ^zē ullamcorper quis sapien. ^zEtiam est nisl, faucibus lorem ipsum dolor sit amet consectetur lorem...



跟动态磁贴一样，[这里同样有供使用的 Toast 通知类型](#)。

1.1. 让 Toast 通知在程序中可用

让 Toast 通知程序中可用，需要修改程序的 manifest 文件。打开 package.appxmanifest 文件，选择 UI 选项，然后在 Toast capable 下拉列表中选择 Yes。



如果创建一个 Badge logo file 并添加到工程中，将会很酷，不过在本文中，不需要，原因如下：

如果你添加一个 Badge Logo，那么必须要让 Lock Screen notifications 可用。

然而，要想 Lock Screen notifications 可用，必须要有一个 Background 任务。在这

两天，我不会涉及到 Background 任务，不过明天会介绍锁屏(Lock Screen)更新。

耐心等待吧。😊

今天主要精力放在同构 Toast 消息通知用户。下面开始吧。

1.2. 在程序中发送一个 Toast 通知

如果你阅读了昨天的内容：第 09 日主要介绍了动态磁贴，你会发现发送一个 Toast 通知与动态磁贴非常相似。下面是相关的代码：

```
private void ToastNotification_Click(object sender, RoutedEventArgs e)
{
    ToastTemplateType toastType = ToastTemplateType.ToastImageAndText02;
    XmlDocument toastXML = ToastNotificationManager.GetTemplateContent(toastType);
    XmlNodeList toastText = toastXML.GetElementsByTagName("text");
    XmlNodeList toastImages = toastXML.GetElementsByTagName("image");
    toastText[0].InnerText = "Funny cat";
    toastText[1].InnerText = "This cat looks like it's trying to eat your face.";
    ((XmlElement)toastImages[0]).SetAttribute("src", "ms-appx:///Assets/10-XAML-CatImageSmall.png");
    ((XmlElement)toastImages[0]).SetAttribute("alt", "Scary Cat Face");

    ToastNotification toast = new ToastNotification(toastXML);
    ToastNotificationManager.CreateToastNotifier().Show(toast);
}
```

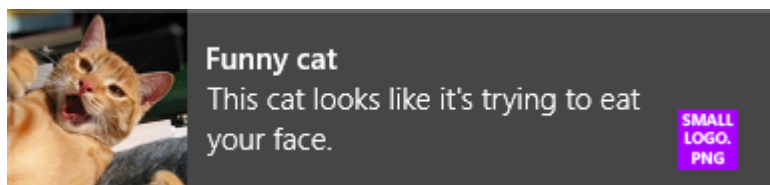
如上代码，第一行代码是选择 Toast 模板——ToastImageAndText02。然后获取选定模板的 xml，并将其存储在 toastXML 中，下面是 ToastImageAndText02 模板的 xml 内容：

```
<toast>
<visual>
<binding template="ToastImageAndText02">
<image id="1" src="image1" alt="image1"/>
<text id="1">headlineText</text>
<text id="2">bodyText</text>
</binding>
</visual>
</toast>
```



如上 xml 所示，这里有两个 text 属性，以及一个 image 属性。上面给出代码的下一步是给这些属性赋值。在设置 image 时，我添加了另外一行代码，涉及到“alt”属性。如果你做过 web 开发，你可能已经知道这是什么，在 Windows 8 中，这里提供的 text 代表了 image，用户在屏幕中是看不到这个 text 的。

最后，我创建了一个 ToastNotification 对象，并将其显示到屏幕上。下面是运行截图：



稍等，还有更多的内容！如果你创建一个工程，并发送 toast 消息，你可能注意到 toast 消息出现的时候，会播放一个音频片段，这是 Toast 通知默认的声音，不过这声音你是可以选择的。同样，在发送 toast 消息时，你还可以做其它一些很酷的事情。

1.3. 设置 Toast 通知的声音（以及更多！）

创建 Toast 通知的每一个事情都是类似的，不过我们可以在 XML 中添加一些额外的节点，以选择不同的声音，设置消息显示持续时间，以及当点击通知时，决定加载那个页面（通过查询参数！）

首先，关于声音，我要告诉你一个坏消息：你不能使用自定义的声音文件，



只能选择 9 个声音文件中的一个，其中 4 个是可循环的声音，可用于 ringtones 或 alarms。 [在这里你可以看到 Toast 可用的全部声音文件。](#)

为了设置声音，首先需要打开创建好的 xml。默认情况下，Toast 模板中没有定义 audio 节点，需要我们自己创建一个。首先，我们需要进入到模板的<toast> 节点，然后才可以进行设置。在下面的代码中，我在 Toast 通知中包含了额外的一些信息，其中我还做了一些注释：

```
private void ToastNotificationOptions_Click(object sender, RoutedEventArgs e)
{
    ToastTemplateType toastType = ToastTemplateType.ToastImageAndText02;
    XmlDocument toastXML = ToastNotificationManager.GetTemplateContent(toastType);
    XmlNodeList toastText = toastXML.GetElementsByTagName("text");
    XmlNodeList toastImages = toastXML.GetElementsByTagName("image");
    toastText[0].InnerText = "Funny cat";
    toastText[1].InnerText = "This cat looks like it's trying to eat your face.";
    ((XmlElement)toastImages[0]).SetAttribute("src", "ms-appx:///Assets/10-XAML-CatImageSmall.png");
    ((XmlElement)toastImages[0]).SetAttribute("alt", "Scary Cat Face");

    //This is the options code, which is all optional based on your needs.
    XmlNode toastNode = toastXML.SelectSingleNode("/toast");

    ((XmlElement)toastNode).SetAttribute("duration", "long");

    XmlElement audioNode = toastXML.CreateElement("audio");
    audioNode.SetAttribute("src", "ms-winsoundevent:Notification.Looping.Alarm");

    //Must be used when looping audio has been selected.
    audioNode.SetAttribute("loop", "true");
    toastNode.AppendChild(audioNode);

    //You can append any text data you would like to the optional
    //launch property, but clicking a Toast message should drive
    //the user to something contextually relevant.
    ((XmlElement)toastNode).SetAttribute("launch", "<cat state='angry'><facebite state='true' /></cat>");

    ToastNotification toast = new ToastNotification(toastXML);
    ToastNotificationManager.CreateToastNotifier().Show(toast);
}
```

记住，额外添加所有的选项都是可选的，正如你所看到第一个例子中所缺少



的内容。下面列出了一些相关内容，是你需要记住的：

- 长时间的 Toast 消息针对的程序是这样的：VOIP 呼叫和闹钟，而不是 email 通知。请谨慎使用。
- 只有将 duration 设置为 Long，循环音乐选项才可用。
- 循环音乐的可用还需要将 loop 属性设置为 true。

如果没有按照上面的规则来，那么将会播放 Notification.Default 声音。

另外，可能有时候你希望 toast 消息发送时，是静音的。当然，这也是可以做到的。仍然指定一个 audioNode 节点，但是设置一新属性 silent，值为 true，如下代码：

```
audioNode.SetAttribute("silent", "true");
```

上面的代码就可以实现静音，但是其它一般的 toast 通知是正常的。最后，这里还有一个“launch”属性。通过这个属性，我们可以传递任意我们希望的字符串，当用户点击 toast 消息时，这些字符串会当做页面参数传递到程序中（[OnLaunched 方法中获取](#)）。我们可以用这个参数决定加载那个 page。不要把任何内容都放到 launch 参数中进行传递。

1. 4. 总结

这就是关于 Windows 8 中的 Toast 通知。跟动态磁贴非常相似，toast 消息是让用户立即得到关注的好方法，无论用户正在做什么。

可能你在考虑如何通过后台任务做这些通知更新，请你再耐心点。在开始介绍后台任务之前，我希望对通知类型有一个完整的介绍，我承诺，再有两天时间，就开始介绍后台任务。



点击下面的图片，可以下载到本文的示例代码：



明天，我将介绍通知里面的最后一个主题：锁屏。到时候见！



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

