Windows 8 开发 31 日

第 09 日

动态磁贴

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

第 09 日动态磁贴

1.0. 介绍

今天，我们来学习一个非常重要的东西：动态磁贴。在之前的文章中，我提到过，磁贴是程序成功的重要部分，而磁贴也是最容易被忽略的一部分。

你的磁贴代表了你和用户之间的关系。用户将其 pin 到开始屏幕，并按照类似功能的程序进行分组，还可以在大磁贴或小磁贴之间切换。而你的任务就是完成用户的需求。我们还可以对磁贴进行更新，这是一个非常有用的功能。下面是我最喜欢的几个程序：

我是 Jetpack Joyride 的超级粉丝，它的磁贴做得非常优秀。下面是 5 个状态中的 4 个，可以通过翻转看到，有最高分的提醒，还有我尚未完成的任务。这些更新多次把我吸引进去。

下面是天气程序，其磁贴虽然没有翻转的功能，但是其提供的信息已足够了。可以知道今天最高和最低温度，以及当前温度。

最后，Photos 程序也是我最喜欢的一个，目前我还不知道它多久更新一次图片，甚至不知道它使用到那个 library，但是它总是能够让我看到一些非常棒的老照片，使我记得照片中的那些事，下面是两个截图：

我有成千上万的老照片存储在我的照片库中（它们都备份在网上的。），

1.1. 规则

在开始之前，我们需要知道一些规则：

- 首先，阅读 Guidelines for Live Tiles。它们内容比较广泛，但是很有趣。
- 你的程序中必须有一个 small tile。Wide tile 值得拥有，不过 small tile 是必须有的。

Small tiles 有时候没有 wide tiles 有趣，不过 small tile 仍然可以显示一些有用的信息。下面是上面介绍过的 3 个程序的 small tiles：

可能由于空间限制的原因，Jetpack tile 没有更新。不过天气和 Photos 程序的 small tile 展现的内容跟 wide tile 是一样的，只不过更小一点。

我想把所有不同的 tile 都展现出来，不过我担心每个 tile template 都展现的话，本文会变得超级长。好在，微软提供了每个 tile template，总的有 45 个。你可以通过下面这个连接查看，并选择适当的 tile template，并且文章中还提供了 ml 标记。

必读：The Tile Template Catalog

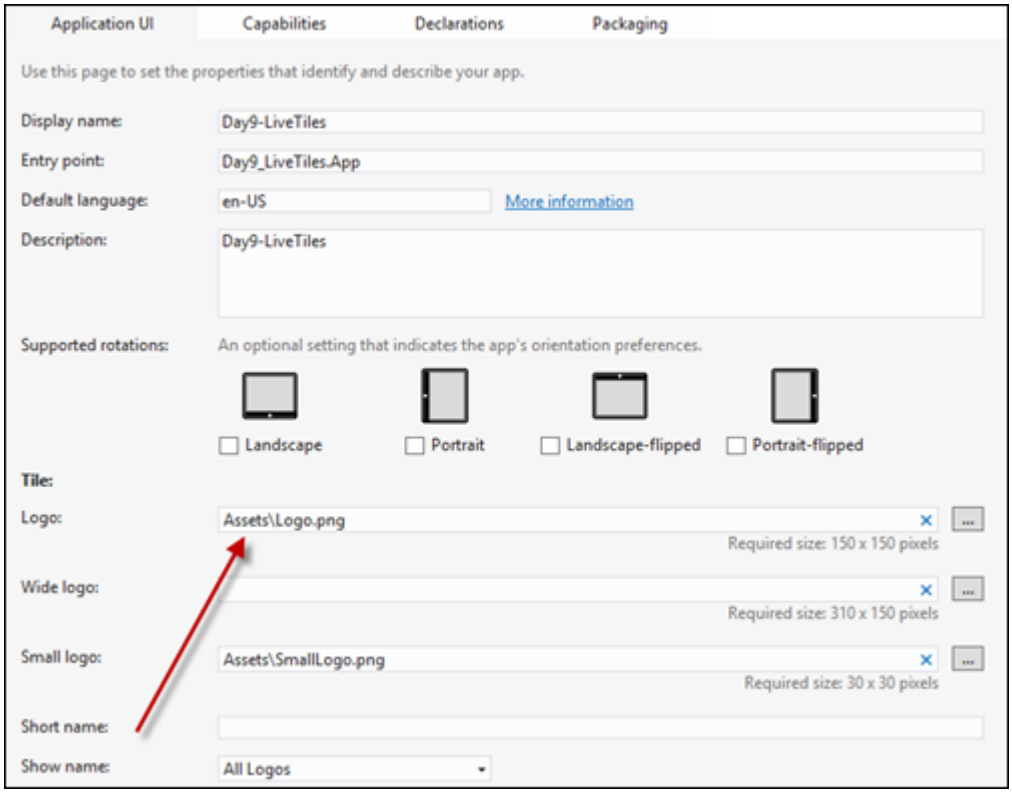
如何让我们的磁贴动起来呢？下面我们开始喽。

1.2. 更改默认静态磁贴

默认情况下，程序的中有如下一个图片：

如果你只想使用自己的 150 x 150 pixel 大小的图片当作静态磁贴，可以使用下面两种方法之一：

- 替换工程中 Assets/Logo.png 图片.
- 更新 Packge.appxmanifest 文件，如下图，修改 Logo 为别的图片即可：



OK，这有点无聊，下面我们开始写一些代码！

1.3. 发送一个通知给 small tile

我们就从发送给 small tile 通知开始 tile（wide 有点复杂，稍后介绍）。首先，如之前讨论的，你需要决定使用哪种类型的 Template。（这里我创建了一个 PDF 文件，里面是 tile 模板）

在这里的代码中，我选择使用 TileSquareBlock 模板，该模板看起来如下：

是用下面的 XML 代表的:

```
<tile>
<visual>
<bindingtemplate="TileSquareText04">
<textid="1">Text Field 1</text>
<textid="2">Text Field 2</text>
</binding>
</visual>
</tile>
```

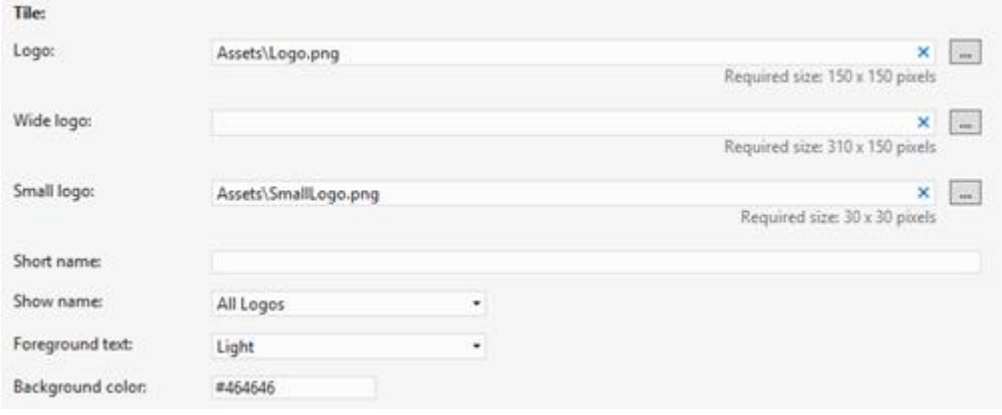
现在我只需要创建一个基于上面这个 xml 的 tile 更新,并推送给 TileUpdateManager 即可。下面是更新的代码：

```
privatevoid TextUpdateButton_Click(object sender, RoutedEventArgs e)
{
//First, we grab the specific template we want to use.
XmlDocument tileData = TileUpdateManager.GetTemplateContent(TileTemplateType.TileSquareBlock);

//Then we grab a reference to the node we want to update.
XmlNodeList textData = tileData.GetElementsByTagName("text");

//Then we set the value of that node.
      textData[0].InnerText = "31";
      textData[1].InnerText = "Days of Windows 8";

//Then we create a TileNotification object with that data.
TileNotification notification = newTileNotification(tileData);

//We can optionally set an expiration date on the notification.
      notification.ExpirationTime = DateTimeOffset.UtcNow.AddSeconds(30);

//Finally, we push the update to the tile.
TileUpdateManager.CreateTileUpdaterForApplication().Update(notification);
}
```

从上面的注释来看，这非常简单。可以看到，因为 text 节点可以有多个，所以我使用了一个数组，在里面填充 text 值。

这个模板需要注意一点的是你的 SmallLogo.png 会被显示在 tile 中。下图是该示例的一个截图：

在工程的 Package.appxmanifest 文件中，不仅仅可以修改 logo 图片，还可以设置背景属性，small logo 文件，以及其它一些文字和颜色值。



在 manifest 中简单的设置一个 wide logo 就可以支持 wide tile 了。实际上，这是设置支持 wide tile 的唯一方法。下面我们就来看看如何给 wide tile 发送通知。

1.4. 发送一个通知给 large tile

这次，我们需要选择 Large Tile 模板，不过这次会包含一个图片。稍后你会看到相关的语法。

记住：显示 wide tile 还是 small tile 是由用户控制的，开发者不能控制。开发者只能控制知否支持通知。

正是这样的原因，当我们在发送更新时，可以同时创建两个通知，一个是 small tile，另外一个是 wide tile，然后通过 xml，将两个通知合并到一起，当做一个通知发送。下面是相关代码：

Because of this, when we send updates, we can actually create two. One for the small tile, and one for the large one. Then, through the magic of XML, we'll merge them together, and send them as one notification. Here's what your code might look like:

```
privatevoid LargeImageTileUpdateButton_Click(object sender, RoutedEventArgs e)
{
//Create the Large Tile exactly the same way.
XmlDocument largeTileData = TileUpdateManager.GetTemplateContent(TileTemplateType.TileWidePeekImage01);
XmlNodeList largeTextData = largeTileData.GetElementsByTagName("text");
XmlNodeList imageData = largeTileData.GetElementsByTagName("image");
      largeTextData[0].InnerText = "Funny cat";
      largeTextData[1].InnerText = "This cat looks like it's trying to eat your face.";
      ((XmlElement)imageData[0]).SetAttribute("src", "ms-appx:///Assets/9-XAML-CatImage.png");
//((XmlElement)imageData[0]).SetAttribute("src", "http://jeffblankenburg.com/downloads/9-XAML-CatImage.png");

//Create a Small Tile notification also (not required, but recommended.)
XmlDocument smallTileData =
TileUpdateManager.GetTemplateContent(TileTemplateType.TileSquarePeekImageAndText02);
XmlNodeList smallTileText = smallTileData.GetElementsByTagName("text");
XmlNodeList smallTileImage = smallTileData.GetElementsByTagName("image");
      smallTileText[0].InnerText = "Funny cat";
      smallTileText[1].InnerText = "This cat looks like it's trying to eat your face.";
      ((XmlElement)smallTileImage[0]).SetAttribute("src", "ms-appx:///Assets/9-XAML-CatImageSmall.png");
```

```
//Merge the two updates into one <visual> XML node
IXmlNode newNode = largeTileData.ImportNode(smallTileData.GetElementsByTagName("binding").Item(0), true);
      largeTileData.GetElementsByTagName("visual").Item(0).AppendChild(newNode);

//Create the notification the same way.
TileNotification notification = newTileNotification(largeTileData);
      notification.ExpirationTime = DateTimeOffset.UtcNow.AddSeconds(30);

//Push the update to the tile.
TileUpdateManager.CreateTileUpdaterForApplication().Update(notification);

}
```

如上代码，我并没有详细的做注释，不过你可以看到我从本地工程中添加了一个图片，当然，这个图片也可以来自互联网，只不过加载时间会长一点，所以考虑好了。

例如，上面代码我这是 30 秒后过期，如果使用网络中的图片，有可能 30 秒过去了，通知还没有更新，因为图片还没有加载完。当然，你的情况可能不同，不过，如果使用本地图片的话，通知立马就会显示出来。

上面我还提到，将两个不同 xml 合并到一块，每个 xml 代表一个 template。下面是 small template——TileSquarePeekImageAndText02：

Also, I mentioned that we're merging two different XML files, one for each template. Here's the small template, for the TileSquarePeekImageAndText02:

```
<tile>
<visual>
<bindingtemplate="TileSquarePeekImageAndText02">
<imageid="1"src="image1"alt="alt text"/>
<textid="1">Text Header Field 1</text>
<textid="2">Text Field 2</text>
</binding>
```

```
</visual>
</tile>
```

而下面的 xml 是 large tile template：

```
<tile>
<visual>
<bindingtemplate="TileWidePeekImage01">
<imageid="1"src="image1.png"alt="alt text"/>
<textid="1">Text Header Field 1</text>
<textid="2">Text Field 2</text>
</binding>
</visual>
</tile>
```

合并后的内容如下：

```
<tile>
<visual>
<bindingtemplate="TileWidePeekImage01">
<imageid="1"src="image1.png"alt="alt text"/>
<textid="1">Text Header Field 1</text>
<textid="2">Text Field 2</text>
</binding>
<bindingtemplate="TileSquarePeekImageAndText02">
<imageid="1"src="image1"alt="alt text"/>
<textid="1">Text Header Field 1</text>
<textid="2">Text Field 2</text>
</binding>
</visual>
</tile>
```

如上代码，visual 节点里面有多个 binding。要还想添加别的通知，可以用类似的处理过程，添加更多的合并代码即可。

1.5. 清除通知

有时候你还想清除已经更新的通知。可能有新的信息可用，或在通知过期之

前有有所改变，此时需要重新开始。那么调用如下代码就可以清除已有的通知：

```
privatevoid ClearNotificationsButton_Click(object sender, RoutedEventArgs e)
{
TileUpdateManager.CreateTileUpdaterForApplication().Clear();
}
```

好的，现在我已经介绍了 small 和 wide tile 了，包括合并。不过，你还想给用户提供多个 tile 吗？

1.6. 辅助磁贴(Secondary tile)

当用户想直接在开始屏幕显示一些信息时，辅助屏幕是一个很优秀的方法。例如一个天气程序，用户可以将多个城市 pin 到开始屏幕，这样就可以方便快速的查看对应城市的天气信息。如果是一个 People 程序，可以将重要的人 pin 到开始屏幕，用户就可以很容易的查看某个 People 的相关信息。

我原来计划要写一些本节的全部相关内容——如何将辅助磁贴 pin 到开始屏幕，但是后来发现微软已经有很详细的文档了，查看下面链接：

Quickstart：Pinning a Secondary Tile

上述链接不仅详细说明了如何创建一个 Secondary Tile ,可以判断是否已经 pin 到开始屏幕，另外还包含如何移除 tile。非常棒的教程，强烈建议阅读。

1.7. 总结

今天，介绍了 Windows 8 中非常重要的特征：Tile，以及 Tile 更新。从 tile 的创建到更新，都非常简单。我不建议你频繁的更新 tile。

点击下面的图标，可以下载到本文的示例代码：

明天，我将介绍另外一种通知：Toast 消息。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