

## WPF及Silverlight版Tiles概述

通过Tiles您可以创建**WPF及Silverlight版**拼接显示和导航中心，它可以帮助您在app中实现同windows8开始界面一样的视觉体验，同时支持采用两种瓷贴控制方式实现平滑的动画和实时更新。帮助您通过连接不同的瓷贴容器实现个性化的用户布局。

## WPS及Silverlight版帮助文档

### 入门

- 关于 **ComponentOne Studio WPF Edition** 安装，授权，技术支持，命名空间和控制工程的创建，请参阅 [Getting Started with WPF Edition](#)。
- 关于 **ComponentOne Studio Silverlight Edition** 安装，授权，技术支持，命名空间和控制工程的创建，请参阅 [Getting Started with Silverlight Edition](#)。

## 快速入门

快速入门可以帮助您顺利运行WPF及Silverlight版Tiles，在快速入门中您需要在Visual Studio中创建一个新的工程，添加WPF及Silverlight版Tiles控件到您的应用程序中并个性化设置控件的外观和活动。

您可以在您的应用程序中包含C1Tile和C1SlideTile控件，获取更多的信息，请参阅C1Tile和C1SlideTile。

- 第一步：建立瓷贴应用程序
- 第二步：添加Tiles到应用程序
- 第三步：运行Tiles应用程序

## 主要特性

WPF及Silverlight版Tile包含以下特有功能：

- 创建翻转及滑动瓷贴

通过C1FlipTile(仅适用Silverlight)和C1SlideTile控制，您可以利用滑动或者翻转动画（仅适用Silverlight）创建瓷贴并以此来选择需要的内容，以此简化您的模板设计并提供内容控制，这些更新和动画均可以自动处理。

- 熟悉的Windows 8 实时瓷贴特性

WPF及Silverlight版瓷贴控件的设计灵感来源于Windows Store及Windows Phone apps中广泛使用的Modern-UI风格，每个瓷贴控件体现了同Windows8及Windows Phones apps开始界面一样的实时交互风格，这就意味着只需要动动手指选择瓷贴，您的app就可以提供给用户熟悉的界面风格。

- 实时更新

C1Tiles可以提供翻转，滑动及显示“实时”更新的内容，您可以通过使用C1TileService静态类和UpdateInterval属性轻松的控制更新时间的间隔。

- 多样的容器嵌入

C1Tiles可以在任何的ItemsControl容器中嵌入，如C1TileListBox，标准WrapPanel及ListBox控件。因此您可以在数据绑定的相关场景中使用C1Tiles，每个容器可以采取不同的方法来组织多个C1Tiles以帮助您设计出丰富多样的界面。

- 提供不同的尺寸

瓷贴的大小可以不同，不同类型和大小的瓷贴可以组合构造出个性化的界面呈现，满足您的应用程序个性化需求。

## 第一步：建立瓷贴应用程序

这一步中您可以在Visual Studio中使用WPF及Silverlight版Tiles创建一个应用，在应用中添加一个包含C1Tiles控件的C1TileListBox，具体参阅以下步骤：

1. 在Visual Studio中，选择文件|新建|项目。
2. 新建项目对话框中，在左侧控制板中的语言下选择Windows，再从模板列表中选择WPF和Silverlight应用。通过输入名称并点击确认完成项目创建，MainPage.xaml页面会被打开。
3. 在解决方案管理器窗口中右键点击项目，选择添加引用。
4. 在引用管理对话框中，定位并选择C1.WPF.dll或C1.Silverlight，C1.WPF.Tile.dll或C1.Silverlight.Tile.dll，点击确定添加引用到您的项目中。
5. 添加XAML命名空间到Window（WPF）或UserControl（Silverlight），通过以下的标记：

xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"

这是统一的命名空间，可以帮助您应用大多数的**ComponentOne WPF**和**Silverlight**控件，而不用添加多用命名空间。Window或UserControl标记现在可以简化如下：

## WPF XAML

```
<Window xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
x:Class="C1WPFTileCS022813.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="MainWindow"
Height="350" Width="525">
```

## Silverlight XAML

```
<UserControl xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
x:Class="C1WPFTileCS022813.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="MainWindow"
Height="350" Width="525">
```

6. 在<Window>标签与<Grid>标签之间添加下列标记:

## WPF XAML

```
<Window.Resources>
    <Style x:Key="listBoxItemStyle" TargetType="ListBoxItem">
        <Setter Property="c1:C1TileService.PointerDownAnimation" Value="True"/>
        <Setter Property="Padding" Value="5" />
    </Style>
    <Style TargetType="c1:C1Tile" x:Key="baseTileStyle">
        <Setter Property="Background" Value="#FFC410" />
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontSize" Value="80"/>
        <Setter Property="HeaderForeground" Value="White"/>
        <Setter Property="HeaderFontSize" Value="12"/>
        <Setter Property="BorderThickness" Value="0" />
        <Setter Property="Width" Value="280" />
        <Setter Property="Height" Value="200" />
    </Style>
    <Style TargetType="c1:C1Tile" BasedOn="{StaticResource baseTileStyle}" />
    <Style TargetType="c1:C1SlideTile" BasedOn="{StaticResource baseTileStyle}" />
</Window.Resources>
```

## Silverlight XAML

```
<UserControl.Resources>
    <Style x:Key="listBoxItemStyle" TargetType="c1:C1ListBoxItem">
        <Setter Property="c1:C1TileService.PointerDownAnimation"
Value="True"/>
    </Style>
```

```
</Style>
<Style TargetType="cl:C1Tile" x:Key="baseTileStyle">
    <Setter Property="Background" Value="#FFC410" />
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="FontSize" Value="80"/>
    <Setter Property="HeaderForeground" Value="White"/>
    <Setter Property="HeaderFontSize" Value="12"/>
    <Setter Property="BorderThickness" Value="0" />
    <Setter Property="Width" Value="280" />
    <Setter Property="Height" Value="200" />
</Style>
<Style TargetType="cl:C1Tile" BasedOn="{StaticResource baseTileStyle}">
</Style>
<Style TargetType="cl:C1SlideTile" BasedOn="{StaticResource
baseTileStyle}">
</Style>
<Style TargetType="cl:C1FlipTile" BasedOn="{StaticResource
baseTileStyle}">
</Style>
</UserControl.Resources>
```

将以上标签添加到程序的外观风格中。

- 将光标放置于<Grid>与</Grid>标签之间，导航到Toolbox并双击**Listbox**控件添加到页面中。
- 更新Listbox标记，显示如下：

```
XAML
<ListBox ItemContainerStyle="{StaticResource listBoxItemStyle}"
ScrollViewer.HorizontalScrollBarVisibility="Disabled" >
    <ListBox.ItemsPanel>
        <ItemsPanelTemplate>
            <cl:C1WrapPanel/>
        </ItemsPanelTemplate>
    </ListBox.ItemsPanel>
    <ListBox.Items>
</ListBox>
```

本步中您可以创建一个WPF应用程序，在下一步您需要添加**WPF及Silverlight**控件到应用中。

## 第二步：添加Tiles到应用程序

在上一步中您创建了一个WPF应用，在这一步您将添加**WPF和Silverlight**控件。

完成以下步骤：

- 将光标放置于<ListBox.Items> </ListBox.Items>之间，导航到工具箱并双击**C1Tile**控件将它添加到页面中。
- 更新**C1Tile**标记，显示如下：

```
XAML
<cl:C1Tile Content="1" Header="C1Tile" HeaderPadding="12" Padding="0"
HeaderBackground="#22000000" HorizontalHeaderAlignment="Stretch" />
```

此处标签添加了标题和内容文本，间距，标题背景颜色和标题对齐。

3. 在**C1Tile**下面添加以下标记：

#### XAML

```
<c1:C1SlideTile Content="2" HeaderPadding="12"
HorizontalContentAlignment="Stretch" VerticalContentAlignment="Stretch"
Padding="0" Header="C1SlideTile">
    <c1:C1Tile.ContentTemplate>
        <DataTemplate>
            <Border Background="#FFBC1C48" >
                <TextBlock Text="{Binding}" VerticalAlignment="Center"
HorizontalAlignment="Center" />
            </Border>
        </DataTemplate>
    </c1:C1Tile.ContentTemplate>
    <c1:C1Tile.BackContentTemplate>
        <DataTemplate>
            <Border Background="#FF028541" >
                <StackPanel VerticalAlignment="Center"
HorizontalAlignment="Center">
                    <TextBlock Text="{Binding}" Foreground="White"
HorizontalAlignment="Center"/>
                    <TextBlock Text="Back Content Template" Margin="0 -10 0 0"
FontSize="12" Foreground="White" HorizontalAlignment="Center"/>
                </StackPanel>
            </Border>
        </DataTemplate>
    </c1:C1Tile.BackContentTemplate>
</c1:C1SlideTile>
```

注意此处标记添加了带有**ContentTemplate**和**BackContentTemplate**的**C1SlideTile**控件，**ContentTemplate**决定了初始内容和控件的外观，**BackContentTemplate**决定了控件改变时控件的内容和外观。（在**C1SlideTile**情况下，它从一个模板滑向另一个模板）。

4. 在**C1SlideTile**下面添加以下标记：

#### XAML

```
<c1:C1SlideTile Content="3" BackContent="Back Content 3" FontSize="36"
Header="C1SlideTile" HeaderPadding="12" Padding="0"/>
<c1:C1SlideTile Content="4" Header="C1SlideTile, SlideDirection = Right"
HeaderPadding="12" Padding="0" SlideDirection="Right"
HorizontalHeaderAlignment="Right"/>
<c1:C1SlideTile Content="5" Header="C1SlideTile" />
```

此标记添加了额外的C1Slide控件。

在此步中您添加**WPF及Silverlight版Tiles**控件到您的应用中。在下一步中，您将运行应用程序并看到它的实时呈现。

## 第三步：运行Tiles应用程序

在上一步中，您创建了一个WPF应用并添加和定制了**WPF及Silverlight版Tile**控件，接下来您将运行您的应用程序。

1. 选择调试|启动调试来运行您的应用程序，运行结果会与以下情形相似：



注意在上图显示的红色的**C1SlideTile**采用的是**ContentTemplate**,当瓷贴转变为**BackContentTempate**时，它会变为绿色。

2. 观察每个控件的行为，**C1SlideTiles**在内容选择时会出现滑动效果。
3. 观察到一些**C1SlideTile**控件的滑动会有不同的方向，您可以通过更改**SlildeDirection**属性来设置方向。

恭喜您！您已经完成了**C1Tiles**控件的快速入门，在接下来的文档中，您将了解更多如何使用**WPF及Silverlight版 Tiles**控件的相关信息。

## C1Tiles的使用

以下主题给出了WPF及Silverlight版Tiles的使用概略：

- C1Tile控件
- C1SlideTile控件
- ContentTemplates及BackContentTemplates

### ContentTemplates及BackContentTemplates

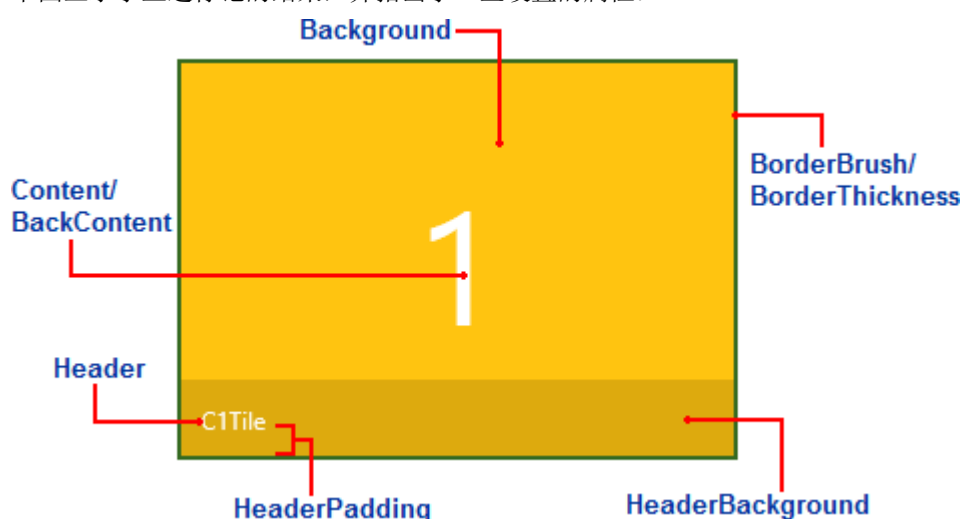
## C1Tile控件

**C1Tile**控件是一个带内容标题的动画控件，并且它模仿了Windows8的实时瓷贴活动，比如，创建一个基础**C1Tile**控件下加入几个属性标签：

### XAML

```
<c1:C1Tile Content="1" BackContent="1" Header="C1Tile" HeaderBackground="#22000000"
HeaderPadding="12" Padding="0" BorderBrush="#FF356A21" BorderThickness="2" />
```

下图显示了上述标记的结果，并指出了一些设置的属性：



下述的属性可以在图/上述的标签中设置：

- **Content:** 此属性设置**C1Tile**控件的初始内容，在上图中，**Content**被设置为“1”。
- **BackContent:** 此属性设置了**C1Tile**控件可改变的内容，在上图中**BackContent**也被设置为“1”（与Content内容相同），但如果**BackContent**属性被设置为另一个值，当瓷贴内容状态改变时会显示这个值。
- **Header:** 此属性设置了**C1Tile**控件的标题内容，默认情况下标题会显示在瓷贴的底部，在上图中，**Header**被设置为“C1Tile”。
- **HeaderBackground:** 此属性设置标题背景的颜色和透明度，在上述例图中，**HeaderBackground**属性被设置为“22000000”，前两位数值表示颜色的透明度，后六位数值表示颜色为黑色。
- **HeaderPadding:** **HeaderPadding**属性设置Header值周围的边距值，在上述例图中**HeaderPadding**被设置为“12”，此处数值越大，**Header**属性内容离瓷贴的底边越远。
- **Padding:** **Padding**属性设置瓷贴中Content和Header周边的边距值，在上述例图中，**Padding**被设置为“0”，这表示标题域将与瓷贴的底部平齐，这个值越大，Content和Header属性内容与瓷贴边距越远。
- **BorderBrush:** **BorderBrush**属性表示C1Tile控件边界的颜色，在上述例图中此项值被设置为“#FF356A21”，显示为绿色。
- **BorderThickness:** **BorderThickness**表示C1Tile周围的厚度，在上述例图中此项值被设置为“2”。

## ContentTemplates及BackContentTemplates

您可以使用**ContentTemplates**及**BackContentTemplates**来制定**C1Tiles**控件外观。首先您需在**ContentTemplates**中设置需要显示的初始内容，其次你需在**BackContentTemplates**中设置当瓷片滑动时需要显示的内容。

例如，下述标签将一个**ContentTemplates**和**BackContentTemplates**加入到一个**C1SlideTile**控件中：

### XAML

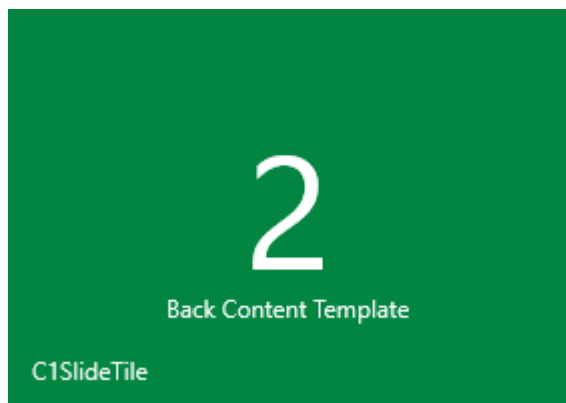
```
<c1:C1SlideTile Content="2" HeaderPadding="12" HorizontalContentAlignment="Stretch"
VerticalContentAlignment="Stretch" Padding="0" Header="C1SlideTile">
  <c1:C1Tile.ContentTemplate>
    <DataTemplate>
      <Border Background="#FFBC1C48" >
        <TextBlock Text="{Binding}" VerticalAlignment="Center"
HorizontalAlignment="Center" />
      </Border>
    </DataTemplate>
  </c1:C1Tile.ContentTemplate>
  <c1:C1Tile.BackContentTemplate>
    <DataTemplate>
      <Border Background="#FF028541" >
        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
          <TextBlock Text="{Binding}" Foreground="White"
HorizontalAlignment="Center"/>
          <TextBlock Text="Back Content Template" Margin="0 -10 0 0"
FontSize="12" Foreground="White" HorizontalAlignment="Center"/>
        </StackPanel>
      </Border>
    </DataTemplate>
  </c1:C1Tile.BackContentTemplate>
</c1:C1SlideTile>
```

在初始运行时，控件的初始外观如下图**ContentTemplate**显示：



当转换为内容替换模板后，控件外观将如下图**BackContentTemplate**显示：





## C1Tiles任务帮助

如果您熟悉Visual Studio编程并且了解在一般情况下如何创建和使用WPF应用程序，您可以直接参考任务帮助。如果您是初次使用**WPF及Silverlight版Tiles**产品，请先参考快速入门。

在使用**WPF及Silverlight版Tiles**产品时，每个主题都提供一个特定任务的解决方案，通过下列帮助中的步骤，您可创建一个项目来演示**WPF及Silverlight版Tiles**的多种特性。

注意以下步骤需要您先创建一个新的WPF工程。

- 添加XAML到Tile Header
- 更新Tile Content
- 绑定条目集
- 绑定Tiles控件

## 添加XAML到Tile Header

**WPF及Silverlight版Tiles**控件不支持Header属性中直接添加**XAML**内容，您需要为**Header**属性绑定CLR值，或者保留为空白，在内容模板中添加您想要的Header内容（例如一张图片）。

例如：

XAML

```
<c1:C1SlideTile Width="280" Height="200" Header="{Binding Name}" />
```

## 更新Tile Content

**C1SlideTile**控件可以自动按照固定的时间间隔交替呈现需要的内容，您可以通过**C1TileService**静态类和**UpdateInterval**属性调节时间间隔，例如：

Visual Basic

```
C1TileService.UpdateInterval = TimeSpan.FromSeconds(20)
```

C#

```
C1TileService.UpdateInterval = TimeSpan.FromSeconds(20);
```

时间间隔值设置的越大，交替更新出现的频率越低。

您也可以编程的方式通过调用**UpdateTile**方法来更新每个特定的瓷贴，例如当瓷贴被点击时触发的更新操作，可以采用以下代码：

Visual Basic

```
Private Sub C1Tile_Click(sender As Object, e As System.EventArgs)
    Dim tile As C1Tile = TryCast(sender, C1Tile)
    If tile IsNot Nothing Then
        tile.UpdateTile()
    End If
End Sub
```

C#

```
private void C1Tile_Click(object sender, System.EventArgs e)
```

```
{
    C1Tile tile = sender as C1Tile;
    if(tile != null)
        tile.UpdateTile();
}
```

## 绑定条目集

您可以在**C1SlideTile**控件上呈现两个以上的可替换条目，通过使用**ContentSource**属性，您可以设置任意数量的翻转条目，通过定义**ContentTemplate**(和可选的**AlternateContentTemplate**)来定义绑定内容的外观。

### XAML

```
<c1:C1SlideTile Header="Photos" ContentSource="{Binding Items}">
    <c1:C1SlideTile.ContentTemplate>
        <DataTemplate>
            <Grid>
                <TextBlock Text="{Binding Author}" Foreground="White"
VerticalAlignment="Top" Margin="4,2,0,2"/>
                <Image Source="{Binding Thumbnail}" Stretch="UniformToFill"
Margin="24, 24, 1, 1"/>
            </Grid>
        </DataTemplate>
    </c1:C1SlideTile.ContentTemplate>
</c1:C1SlideTile>
```

## 绑定Tiles控件

您可以在任意的**ItemsControl**中使用**C1Tile**，如**C1TileListBox**或者**GridView**控件，这里给出在**ListBox**控件绑定**C1FlipTile**的标记示例：

### XAML

```
<ListBox ItemsSource="{Binding}" x:Name="listBox" >
    <ListBox.ItemsPanel>
        <ItemsPanelTemplate>
            <c1:C1WrapPanel Background="YellowGreen"/>
        </ItemsPanelTemplate>
    </ListBox.ItemsPanel>
    <ListBox.ItemContainerStyle>
        <Style TargetType="ListBoxItem">
            <Setter Property="c1:C1TileService.PointerDownAnimation" Value="True"/>
        </Style>
    </ListBox.ItemContainerStyle>
    <ListBox.ItemTemplate>
        <DataTemplate>
            <c1:C1FlipTile Height="200" Width="200" Header="{Binding Title}"
Content="{Binding}" Background="DarkGreen" HeaderBackground="#88000000"
HeaderFontSize="18" Command="{Binding TileCommand, ElementName=pageRoot}"
HeaderPadding="2" Padding="0" HeaderForeground="White" CommandParameter="{Binding
Content, RelativeSource={RelativeSource Self}}"/>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```

```
<c1:C1Tile.ContentTemplate>
    <DataTemplate>
        <Grid>
            <TextBlock Text="{Binding Author}" Foreground="White"
VerticalAlignment="Top" Margin="4,2,0,2"/>
            <Image Source="{Binding Thumbnail}"
Stretch="UniformToFill" Margin="1, 24, 24, 1"/>
        </Grid>
    </DataTemplate>
</c1:C1Tile.ContentTemplate>
<c1:C1Tile.AlternateContentTemplate>
    <DataTemplate>
        <Grid>
            <Image Source="{Binding Thumbnail}"
Stretch="UniformToFill"/>
        </Grid>
    </DataTemplate>
</c1:C1Tile.AlternateContentTemplate>
</c1:C1FlipTile>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
```

 **注意：** 如果您在**ListBox**的**VirtualizingStackPanel**使用**C1Tile**控件作为**ItemsPanel**，为了避免在滚动**ListBox**时动画效果，您需要将**VirtualizingStackPanel.VirtualizationMode**设置为**Standard**。