

# Tiles for WPF/Silverlight

2018.04.11 更新


グレースィティ株式会社

## 目次

<a href="#">製品の概要</a>	2
<a href="#">ComponentOne for WPF/Silverlight のヘルプ</a>	2
<a href="#">主な特長</a>	3
<a href="#">クイックスタート</a>	4
<a href="#">手順1: タイルアプリケーションの設定</a>	4-6
<a href="#">手順2: アプリケーションへのタイルの追加</a>	6-7
<a href="#">手順3: タイルアプリケーションの実行</a>	7-8
<a href="#">Tiles の使い方</a>	9
<a href="#">C1Tile コントロール</a>	9-10
<a href="#">C1SlideTile コントロール</a>	10
<a href="#">ContentTemplate と BackContentTemplate</a>	10-12
<a href="#">タスク別ヘルプ</a>	13
<a href="#">タイルヘッダーに XAML コンテンツを追加する</a>	13
<a href="#">タイルコンテンツを更新する</a>	13-14
<a href="#">項目のコレクションに連結する</a>	14
<a href="#">連結コントロールでのタイルの使用</a>	14-15

## 製品の概要

**Tiles for WPF/Silverlight** を使用して、さまざまなタイル表示やナビゲーションハブを作成できます。タイルを使用すると、Windows 8 のスタート画面の操作性を独自のアプリケーションに容易に採り入れることができます。スライドアニメーションと Live 更新をサポートする2種類のタイルコントロールを利用できます。タイルをさまざまなコンテナと組み合わせることで、多種多様なレイアウトを作成することが可能です。

 **メモ:** 説明内に含まれるクラスおよびメンバーに対するリファレンスへのリンクは、原則としてWPF版のリファレンスページを参照します。Silverlight版については、目次から同名のメンバーを参照してください。

## ComponentOne for WPF/Silverlight のヘルプ

ComponentOne for WPF/Silverlight のインストール、ライセンス、テクニカルサポート、名前空間、およびこのコントロールを含むプロジェクトの作成方法については、「[ComponentOne for WPF/Silverlight ユーザーガイド](#)」を参照してください。

## 主な特長

Tiles for WPF/Silverlight には、次の主要な機能があります。

- **反転およびスライドタイトルの作成**  
C1FlipTile コントロールと C1SlideTile コントロールを使用して、コンテンツをスライドアニメーションで交互に表示するタイトルを作成できます。テンプレートを設計し、コントロールにコンテンツを提供するだけです。更新とアニメーションは自動的に処理されます。
- **おなじみの Windows 8 ライブタイトルの動作**  
Tile のコントロールは、Windows ストアや Windows Phone 用のアプリでよく使用されているモダン UI スタイルからヒントを得ています。どのタイトルコントロールも、Windows 8 や Windows Phone のスタート画面にあるライブタイトルと同じ形式のインタラクティブな動作を示します。したがって、タイトルを選択するとき以外は指を上げる必要がないという、ユーザーにとってはおなじみの動作でアプリを操作できます。
- **Live 更新**  
Tiles では、最新の「ライブ」コンテンツをフリップ、スライド、および表示できます。静的 C1TileService クラスと UpdateInterval プロパティを使用して、更新間隔を簡単に制御できます。詳細については、「[タイトルコンテンツを更新する](#)」を参照してください。
- **任意のコンテナでホスト可能**  
Tiles は、C1TileListBox、C1WrapPanel、標準の StackPanel や ListBox コントロールなど、任意の ItemsControl コンテナでホストできます。したがって、C1Tiles をデータ連結シナリオで使用できます。コンテナごとに異なる方法で複数の C1Tiles を配置できるため、それらの組み合わせと可能性は無限です。
- **異なるサイズのサポート**  
すべてのタイトルを同じ大ききで作成する必要はありません。さまざまな種類やサイズのタイトルを組み合わせ、それぞれのアプリケーションで個性的な表示方法を作成できます。

## クイックスタート

このクイックスタートガイドは、**Tiles for WPF/Silverlight** を初めて使用するユーザーのために用意されています。このクイックスタートでは、Visual Studio で新しいプロジェクトを作成し、アプリケーションに **Tiles for WPF/Silverlight** コントロールを追加して、コントロールの外観と動作をカスタマイズします。

**C1Tile** および **C1SlideTile** コントロールを含むアプリケーションを作成します。各コントロールの詳細については、「[C1Tile コントロール](#)」および「[C1SlideTile コントロール](#)」を参照してください。

## 手順1: タイルアプリケーションの設定

この手順では、最初に Visual Studio で **Tiles for WPF/Silverlight** を使用するアプリケーションを作成します。この手順では、WPF/Silverlight ストアアプリケーションを作成し、**C1TileListBox** を追加し、**Tiles for WPF/Silverlight** コントロールを挿入します。それには、次の手順に従います。

1. Visual Studio で、**[ファイル]**→**[新規作成]**→**[プロジェクト]**を選択します。
2. **[新しいプロジェクト]**ダイアログボックスで、左ペインの言語を展開し、言語の下で**[Windows]**を選択し、テンプレートリストで**[WPF アプリケーション]**を選択します。名前を入力し、**[OK]**をクリックしてプロジェクトを作成します。MainPage.xaml ページが表示されます。
3. ソリューションエクスプローラウィンドウでプロジェクトを右クリックし、**[参照の追加]**を選択します。
4. **[参照の追加]**ダイアログボックスで、**C1.WPF.dll** または **C1.Silverlight.dll** および **C1.WPF.Tile.dll** または **C1.Silverlight.Tile.dll** アセンブリを見つけて選択し、**[OK]**をクリックしてプロジェクトに参照を追加します。
5. 次のマークアップを使用して、Window タグに XAML 名前空間を追加します。

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

これは、複数の名前空間を追加しなくても、ほとんどの **WPF** コントロールを使用できるようにするための統合名前空間です。UserControl タグは次のように表示されます。

## 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 x:Class="TileSamples.TileTypes"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml" mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="400">
```

# Tiles for WPF/Silverlight

6. <Window> タグの直後で <Grid> タグの前に、次のマークアップを追加します。

## 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>
  <Style TargetType="c1:C1SlideTile" BasedOn="{StaticResource baseTileStyle}">
  </Style>
</Window.Resources>
```

## Silverlight

### XAML

```
<UserControl.Resources>
  <Style x:Key="listBoxItemStyle" TargetType="c1:C1ListBoxItem">
    <Setter Property="c1:C1TileService.PointerDownAnimation"
Value="True"/>
  </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>
  <Style TargetType="c1:C1SlideTile" BasedOn="{StaticResource
baseTileStyle}">
  </Style>
```

```

        </Style>
        <Style TargetType="c1:C1FlipTile" BasedOn="{StaticResource
baseTileStyle}">
        </Style>
    </UserControl.Resources>

```

このマークアップは、アプリケーションの外観をスタイル設定するためのリソースを追加します。

- <Grid> タグと </Grid> タグの間にカーソルを置き、ツールボックスに移動し、**Listbox** コントロールをダブルクリックしてページに追加します。
- Listbox** マークアップを更新します。次のようになります。

```

XAML
<ListBox ItemContainerStyle="{StaticResource listBoxItemStyle}"
ScrollViewer.HorizontalScrollBarVisibility="Disabled" >
    <ListBox.ItemsPanel>
        <ItemsPanelTemplate>
            <c1:C1WrapPanel/>
        </ItemsPanelTemplate>
    </ListBox.ItemsPanel>
    <ListBox.Items>
    </ListBox.Items>
</ListBox>

```

この手順では、**Tiles for WPF/Silverlight** アプリケーションを作成しました。次の手順では、このアプリケーションに WPF/Silverlight コントロールを追加します。

## 手順2: アプリケーションへのタイルの追加

前の手順では、WPF アプリケーションを作成しました。この手順では、**Tiles for WPF/Silverlight** コントロールを追加します。次の手順に従います。

- <ListBox.Items></ListBox.Items> タグの間にカーソルを置き、ツールボックスに移動し、**C1Tile** コントロールをダブルクリックしてページに追加します。
- C1Tile** マークアップを更新します。次のようになります。

```

XAML
<c1:C1Tile Content="1" Header="C1Tile" HeaderPadding="12" Padding="0"
HeaderBackground="#22000000" HorizontalHeaderAlignment="Stretch" />

```

- C1Tile** マークアップの直後に次のマークアップを追加します。

```

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>
```

このマークアップは、**ContentTemplate** および **BackContentTemplate** とともに **C1SlideTile** コントロールを追加します。**ContentTemplate** は、コントロールの初期コンテンツと外観を決定します。**BackContentTemplate** は、コントロールが変化するとき( **C1SlideTile** コントロールの場合は、1つのテンプレートから別のテンプレートにスライドするとき)、コントロールのコンテンツと外観を決定します。

4. **C1SlideTile** マークアップの直後に次のマークアップを追加します。

```
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" />
```

このマークアップは **C1SlideTile** コントロールを追加します。

この手順では、アプリケーションに **Tiles for WPF/Silverlight** コントロールを追加しました。次の手順では、アプリケーションを実行し、実行時にアプリケーションがどのように表示されるかを確認します。

## 手順3: タイルアプリケーションの実行

前の手順では、WPF ストアアプリケーションを作成し、**Tiles for WPF/Silverlight** コントロールを追加してカスタマイズしました。この手順ではアプリケーションを実行します。

1. **[デバッグ]** → **[デバッグ開始]** を選択して、アプリケーションを実行します。次のようになります。





上の図の赤色の **C1SlideTile** は、**ContentTemplate** を使用して表示されます。タイルが **BackContentTemplate** に移行すると、緑になります。

1. 各コントロールの動作を確認してください。**C1SlideTile** がコンテンツの選択によってスライドします。
2. いくつかの **C1SlideTile** コントロールは、異なる方向に移動するように表示されます。この動作は、**SlideDirection** プロパティを使用して設定できます。

おめでとうございます!

これで、**C1Tiles** クイックスタートは終了です。**Tiles for WPF/Silverlight** コントロールの使用法の詳細については、以降のトピックを参照してください。

## Tiles の使い方

以下のトピックでは、**Tiles for WPF/Silverlight** の要素および機能について説明します。

- [C1Tile コントロール](#)
- [C1SlideTile コントロール](#)
- [ContentTemplate と BackContentTemplate](#)

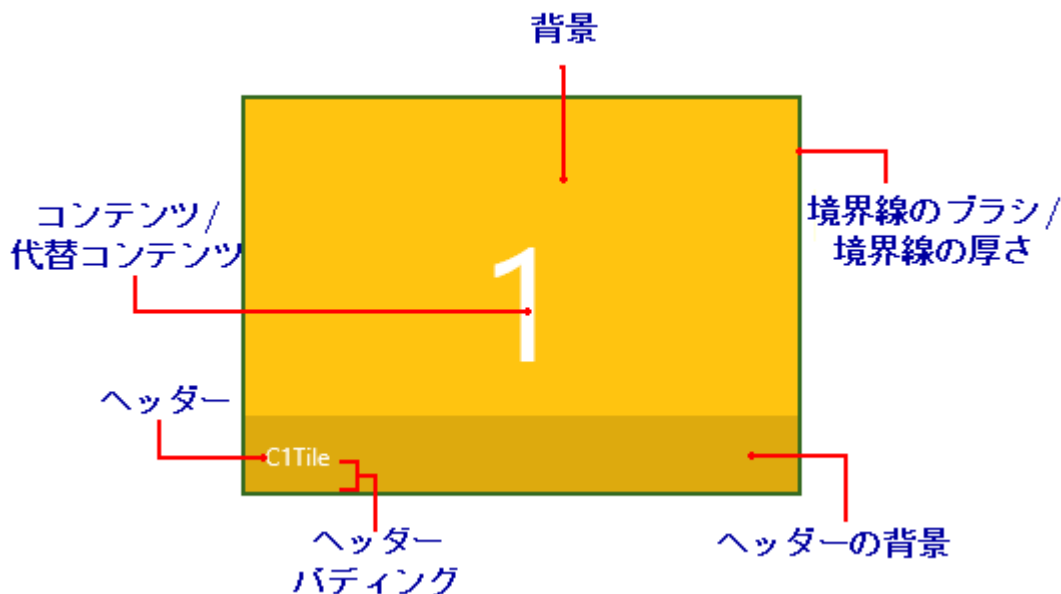
## C1Tile コントロール

**C1Tile** コントロールは、Windows 8 のライブタイトルの動作を真似たヘッダー付きのアニメーションコンテンツコントロールです。たとえば、以下のマークアップは、いくつかのプロパティセットを含む基本的な **C1Tile** コントロールを作成します。

XAML

```
<c1tile:C1Tile Content="1" BackContent="1" Header="C1Tile"  
HeaderBackground="#22000000" HeaderPadding="12"  
Padding="0" BorderBrush="#FF356A21" BorderThickness="2" />
```

次の図に、上のマークアップの結果と設定されたプロパティの一部を示します。



上の図とマークアップでは、次のプロパティが設定されています。

- **Content**: このプロパティは、**C1Tile** コントロールの初期コンテンツを設定します。上の図では、**Content** が「1」に設定されています。
- **BackContent**: このプロパティは、**C1Tile** コントロールの代替コンテンツを設定します。上の図では、**BackContent** も「1」に設定されています (**Content** と同じ)。ただし、**BackContent** プロパティに別の値が設定された場合は、タイトルのコンテンツが変化したときにその値が表示されます。
- **Header**: このプロパティは、**C1Tile** コントロールのヘッダーのコンテンツを設定します。デフォルトでは、ヘッダーはタイトルの下部に表示されます。上の図では、**Header** は「C1Tile」に設定されています。
- **HeaderBackground**: このプロパティは、ヘッダーの背景の色と透過性を設定します。上の例では、**HeaderBackground** プロパティは 22000000 に設定されています。最初の2つの値は色の透過性を示し、後の6つの

値は色が黒であることを示しています。

- **HeaderPadding:HeaderPadding** プロパティは、**Header** 値の周囲のパディングを設定します。上の例では、**HeaderPadding** が「12」に設定されています。数値が大きいほど、**Header** プロパティで表される値がタイトルの端から離れて表示されます。
- **Padding:Padding** プロパティは、タイトル内の **Content** と **Header** の周囲のパディングを設定します。上の例では、**Padding** が「0」に設定されています。これは、ヘッダー領域がタイトルの下境界線にぴったりくっついて表示されることを示します。数値が大きいほど、**Content** プロパティと **Header** プロパティで表される値がタイトルの端から離れて表示されます。
- **BorderBrush:BorderBrush** プロパティは、**C1Tile** コントロールの周囲の境界線の色を示します。上の例では、これは「#FF356A21」(不透明な緑色)に設定されています。
- **BorderThickness:BorderThickness** は、**C1Tile** を囲む境界線の厚さを示します。上の例では、このプロパティは「2」に設定されています。

## C1SlideTile コントロール

**C1Tile** コントロールに基づく **C1SlideTile** コントロールには、テンプレート間をスライドするアニメーションが含まれます。このコンテンツのスライド方向は、**SlideDirection** プロパティによって決まります。**SlideDirection** プロパティは、次のいずれかの値に設定できます(デフォルト値は **All**)。

値	方向
Up	下から上。
Left	右から左。
Down	上から下。
Right	左から右。
All(デフォルト)	すべて。

たとえば、**SlideDirection** プロパティを **Right** に設定した **C1SlideTile** コントロールの例を以下に示します。

XAML

```
<c1tile:C1SlideTile Content="4" Header="C1SlideTile, SlideDirection = Right"
HeaderPadding="12"
Padding="0" SlideDirection="Right" HorizontalHeaderAlignment="Right"/>
```

上のマークアップは、次の **C1SlideTile** コントロールを作成します。



## ContentTemplate と BackContentTemplate

**ContentTemplate** と **BackContentTemplate** を使用すると、**Tiles for WPF/Silverlight** コントロールの外観をカスタマイズすることができます。タイルに最初に表示するコンテンツを **ContentTemplate** に置き、別のコンテンツにスライドまたはフリップされたときに表示するコンテンツを **BackContentTemplate** に置きます。

たとえば、次のマークアップは、**ContentTemplate** と **BackContentTemplate** を **C1SlideTile** コントロールに追加します。

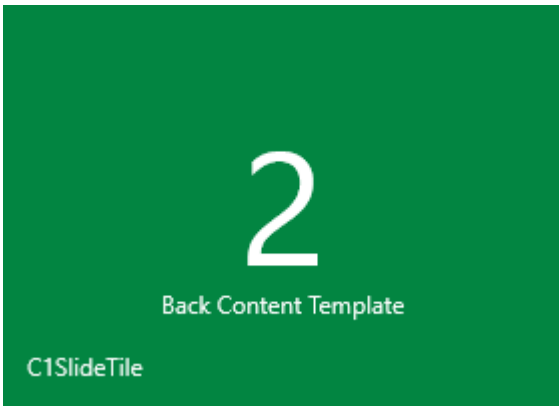
### XAML

```
<ctile:C1SlideTile Content="2" HeaderPadding="12"
HorizontalContentAlignment="Stretch"
VerticalContentAlignment="Stretch" Padding="0" Header="C1SlideTile">
  <ctile:C1Tile.ContentTemplate>
    <DataTemplate>
      <Border Background="#FFBC1C48" >
        <TextBlock Text="{Binding}" VerticalAlignment="Center"
HorizontalAlignment="Center" />
      </Border>
    </DataTemplate>
  </ctile:C1Tile.ContentTemplate>
  <ctile: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>
  </ctile:C1Tile.BackContentTemplate>
</ctile:C1SlideTile>
```

実行すると、最初は **ContentTemplate** が次のように表示されます。



もう一方のコンテンツテンプレートに変化すると、コントロールには次の **BackContentTemplate** が表示されます。



## タスク別ヘルプ

タスク別ヘルプは、ユーザーの皆様が Visual Studio のプログラミングに精通しており、Windows ストアアプリケーションを作成して使用する一般的な方法を理解していることを前提としています。**Tiles for WPF/Silverlight** 製品に精通していない場合は、まず「[クイックスタート](#)」を参照してください。

各トピックは、**Tiles for WPF/Silverlight** 製品を使用して特定のタスクを実行するためのソリューションを提供します。ヘルプに示される手順に従って作業を進めるだけで、さまざまな **Tiles for WPF/Silverlight** の機能を具体的に示すプロジェクトを作成できます。

次のトピックは、新しい Windows ストアプロジェクトが作成されていることを前提としています。

- [タイルヘッダーに XAML コンテンツを追加する](#)
- [タイルコンテンツを更新する](#)
- [項目のコレクションに連結する](#)
- [連結コントロールでのタイルの使用](#)

## タイルヘッダーに XAML コンテンツを追加する

**Tiles for WPF/Silverlight** コントロールは、**Header** プロパティに直接設定した XAML コンテンツをサポートしません。**Header** を何らかの CLR 値に連結するか、Header は空にしておき、Content テンプレートに目的の **Header** コンテンツ (画像など) を追加する必要があります。

次に例を示します。

XAML

```
<ctile:C1FlipTile Width="280" Height="200" Header="{Binding Name}" />
```

## タイルコンテンツを更新する

**C1SlideTile** コントロールと **C1FlipTile** コントロールは、コンテンツを自動的に一定の間隔で交互に表示します。この間隔は、静的 **C1TileService** クラスと **UpdateInterval** プロパティを使用して調整できます。次に例を示します。

## Visual Basic

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

## C#

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

更新間隔の値を大きくすると、更新の頻度が低くなります。

プログラムで、各タイルの **UpdateTile** メソッドを呼び出して、タイルを更新することもできます。たとえば、クリックされたときにタイルを更新するには、次のコードを使用します。

## 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();
}
```

## 項目のコレクションに連結する

**C1SlideTile** または **C1FlipTile** コントロールでは、2つ以上の項目を交互に表示することができます。**ContentSource** プロパティを使用して、フリップする項目をいくつでも指定することができます。**ContentTemplate** (およびオプションで **AlternateContentTemplate**) を定義して、連結コンテンツの外観を定義します。

### 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>
```

## 連結コントロールでのタイルの使用

**C1TileListBox** や **GridView** コントロールなどの任意の **ItemsControl** で **C1Tile** を使用できます。連結 **ListBox** コントロールで **C1FlipTile** を使用するマークアップの例を以下に示します。

### XAML

```
<ListBox ItemsSource="{Binding}" x:Name="listBox" >
    <ListBox.ItemsPanel>
        <ItemsPanelTemplate>
            <c1:C1WrapPanel Background="YellowGreen"/>
        </ItemsPanelTemplate>
    </ListBox.ItemsPanel>
</ListBox>
```

# Tiles for WPF/Silverlight

```
</ListBox.ItemsPanel>
<ListBox.ItemContainerStyle>
  <Style TargetType="ListBoxItem">
    <Setter Property="c1tile:C1TileService.PointerDownAnimation"
Value="True"/>
  </Style>
</ListBox.ItemContainerStyle>
<ListBox.ItemTemplate>
  <DataTemplate>
    <c1tile: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}}">
      <c1tile: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>
      </c1tile:C1Tile.ContentTemplate>
      <c1tile:C1Tile.AlternateContentTemplate>
        <DataTemplate>
          <Grid>
            <Image Source="{Binding Thumbnail}"
Stretch="UniformToFill"/>
          </Grid>
        </DataTemplate>
      </c1tile:C1Tile.AlternateContentTemplate>
    </c1tile:C1FlipTile>
  </DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
```

 **メモ:** **VirtualizingStackPanel** を **ItemsPanel** として使用し、**ListBox** で **C1Tile** コントロールを使用する場合は、**VirtualizingStackPanel.VirtualizationMode** を **Standard** に設定して、**ListBox** のスクロール中にタイルアニメーションが発生しないようにしてください。