

# TileView for UWP

2018.03.07 更新

グレースィティ株式会社

## 目次

<a href="#">TileView for UWP</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: C1TileView アプリケーションの作成</a>	4
<a href="#">手順2: C1TileView コントロールのカスタマイズ</a>	4-5
<a href="#">手順3: C1TileView アプリケーションの実行</a>	5-6
<a href="#">C1TileView の使い方</a>	7
<a href="#">TileViewItem の要素</a>	7
<a href="#">TileViewItem の状態</a>	7-8
<a href="#">列と行</a>	8
<a href="#">最小化項目の位置</a>	8
<a href="#">ドラッグ &amp; ドロップ操作</a>	8
<a href="#">基本的なプロパティ</a>	8-9
<a href="#">タスク別ヘルプ</a>	10
<a href="#">アプリケーションに C1TileView を追加する</a>	10
<a href="#">C1TileView に項目を追加する</a>	10-11
<a href="#">ドラッグ &amp; ドロップ機能を無効にする</a>	11-12
<a href="#">ヘッダーの外観をカスタマイズする</a>	12
<a href="#">最大化/最小化項目テンプレートを作成する</a>	12-13

## TileView for UWP

**TileView for UWP**を使用すると、データをインタラクティブに参照できます。タイルを展開または折りたたんで、情報を表示する量を増減できます。この高度なビジュアル化機能と対話式機能を備えたコントロールを使用すると、Windows 8 の真にタッチ優先で流れるように高速な能力をアプリケーションで遺憾なく発揮できます。ダッシュボード、詳細ビュー、フォトギャラリーなどを作成してください。

## 主な特長

TileView for UWPには、次の主な特長があります。

- **ジェスチャベースの操作とアニメーション**

C1TileView コントロールは極めてインタラクティブなコントロールです。各タイルを3種類の状態で表示でき、ユーザーはタイルヘッダーをタップするだけで表示状態を切り替えることができます。また、タイルを任意の方向にスライドまたはフリックするだけでタイルを並べ替えることができます。タイルのある状態から別の状態への移行は、スムーズなアニメーションで行われます。

- **3つのタイル表示状態**

各タイルは、**Maximized**(最大化)、**Minimized**(最小化)、**Default**(デフォルト)の3つの状態で表示できます。項目テンプレートを使用して、各タイルに表示する情報量を簡単に増減できます。**Default** 状態では、すべてのタイルが同じ大きさで表示されます。

- **最小化位置**

1つのプロパティを設定するだけで、タイルを C1TileView コントロールの上端、左端、下端、または右端に最小化できます。デフォルト状態での行数と列数を指定することもできます。

- **柔軟なデータ連結**

**C1TileView** は、ビジネスオブジェクトの任意のコレクションに連結できる項目コントロールです。他の項目コントロールと同様に、項目テンプレート内で要素連結を指定します。

- **UI の仮想化**

C1TileView コントロールは UI の仮想化をサポートしているため、パフォーマンスに影響を与えずに大量の項目をロードして表示できます。

## クイックスタート

このクイックスタートガイドは、**TileView for UWP**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、C1TileView コントロールを使用して、簡単なプロジェクトを作成します。新しい **UWP アプリケーション**を作成し、C1TileView コントロールをアプリケーションに追加します。C1TileView コントロールに表示されるコンテンツを追加し、**TileView for UWP**に対して実行可能ないくつかの操作を確認してみます。

## 手順1:C1TileView アプリケーションの作成

この手順では、**TileView for UWP**を使用して **UWP** アプリケーションを作成します。アプリケーションに C1TileView コントロールを追加すると、コンテンツを内部に表示できるインターフェースになります。プロジェクトをセットアップし、C1TileView コントロールをアプリケーションに追加するには、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 ストア]を選択し、テンプレートリストで[新しいアプリケーション (XAML)]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、<Grid> タグと </Grid>タグの間にカーソルが置かれます。
3. ツールボックスに移動し、C1TileView アイコンをページにドラッグして、コントロールをグリッドに追加します。これで、参照と XAML 名前空間が自動的に追加されます。XAML マークアップは次のようになります。

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
  <TileView:C1TileView HorizontalAlignment="Left" Height="100"
  VerticalAlignment="Top" Width="100"/>
</Grid>
```

4. Grid 内で C1TileView コントロールを初期化し、<TileView:C1TileView> タグに x:Name="C1TileView1" を追加して、コントロールに名前を付けます。次のようになります。

```
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
  <TileView:C1TileView x:Name="C1TileView1" HorizontalAlignment="Left"
  Height="100"
  VerticalAlignment="Top" Width="100"/>
</Grid>
```

これで、"**C1TileView1**" という名前の C1TileView コントロールがアプリケーションに追加されます。

これで、アプリケーションのユーザーインターフェースが正しくセットアップされましたが、このアプリケーションを実行すると、C1TileView コントロールにコンテンツがないことがわかります。次の手順では、C1TileView コントロールにコンテンツを追加し、コントロールに対して実行可能ないくつかの操作を確認してみます。

## 手順2:C1TileView コントロールのカスタマイズ

前の手順では、**UWP アプリケーション**を作成し、プロジェクトに C1TileView コントロールを追加しました。アプリケーションをカスタマイズするには、次の手順に従います。

1. ページの **C1TileView** タグ内に `AllowDrop="True"` を追加して、ユーザーがコントロール内の項目に対してドラッグ & ドロップ操作を実行できるようにします。**XAML** マークアップは次のようになります。

```
<TileView:C1TileView x:Name="C1TileView1" AllowDrop="True"/>
```

# TileView for UWP

2. **C1TileView** タグ内に3つの **C1TileViewItem** を追加します。**XAML** マークアップは次のようになります。

```
<TileView:C1TileView x:Name="C1TileView1" AllowDrop="True">
  <TileView:C1TileViewItem></TileView:C1TileViewItem>
  <TileView:C1TileViewItem></TileView:C1TileViewItem>
  <TileView:C1TileViewItem></TileView:C1TileViewItem>
</TileView:C1TileView>
```

3. 各 **C1TileViewItem** に **Background** プロパティと **Header** プロパティを追加します。マークアップは次のようになります。

```
<TileView:C1TileView x:Name="C1TileView1" AllowDrop="True">
  <TileView:C1TileViewItem Background="Red" Header="赤色">
</TileView:C1TileViewItem>
  <TileView:C1TileViewItem Background="Blue" Header="青色">
</TileView:C1TileViewItem>
  <TileView:C1TileViewItem Background="Yellow" Header="黄色">
</TileView:C1TileViewItem>
</TileView:C1TileView>
```

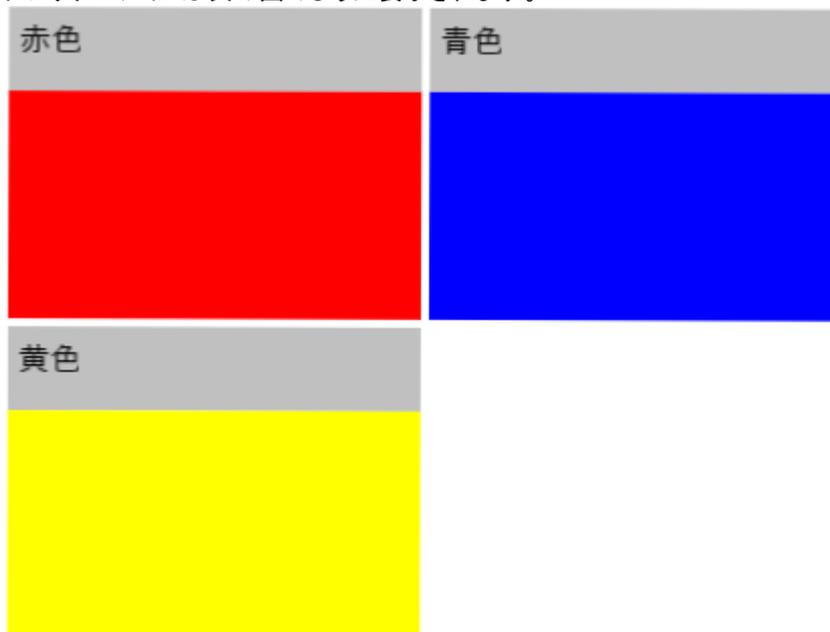
これで、各項目が異なる色で表示され、ヘッダーにテキストが表示されます。

この手順では、C1TileView コントロールにコンテンツを追加しました。次の手順では、このコントロールで可能な実行時の操作をいくつか示します。

## 手順3: C1TileView アプリケーションの実行

**UWP アプリケーション**を作成し、C1TileView コントロールをカスタマイズしました。次に、アプリケーションを実行します。アプリケーションを実行し、**TileView for UWP**の実行時の動作を確認するには、次の手順に従います。

1. **[デバッグ]**メニューから**[デバッグ開始]**を選択し、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。アプリケーションは次の図のように表示されます。



C1TileView コントロール内に3つの C1TileViewItem が表示されます。

2. 赤色の項目のヘッダーをクリックし、青色の項目に向かってドラッグします。項目が入れ替わります。

- 黄色の項目のヘッダーをクリックして項目を最大化します。他の2つの項目は最小化されます。



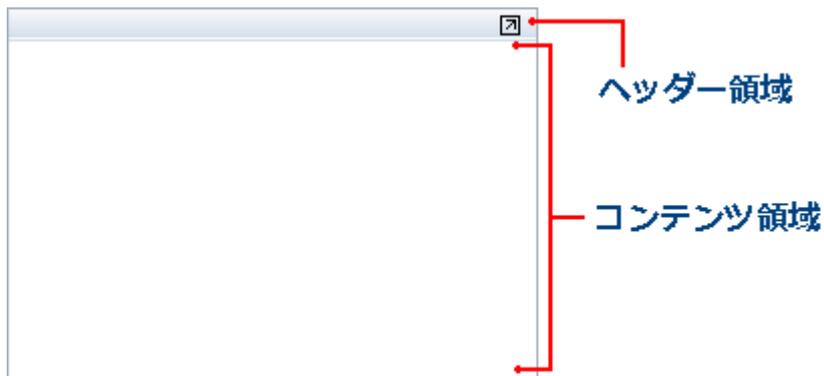
おめでとうございます。これで **TileView for UWP** クイックスタートは完了です。簡単な **UWP アプリケーション** を作成し、**TileView for UWP** コントロールを1つ追加してカスタマイズしました。その後、コントロールの実行時機能をいくつか確認しました。

## TileView の使い方

**TileView for UWP**には **C1TileView** コントロールが含まれます。これは、データを対話式に参照するためのパネルです。**XAML** ウィンドウに追加された **C1TileView** コントロールは、空のコンテナコントロールになります。このコントロールをカスタマイズしたり、コンテンツをロードすることができます。

## TileViewItem の要素

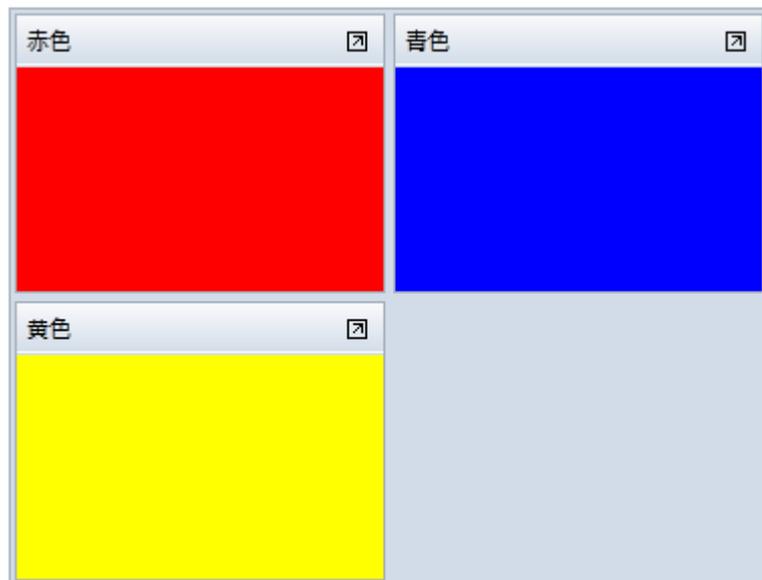
**C1TileViewItem** コントロールは、ヘッダーとコンテンツ領域の2つの部分で構成されます。次の図に、ツールバーとコンテンツ領域を示します。



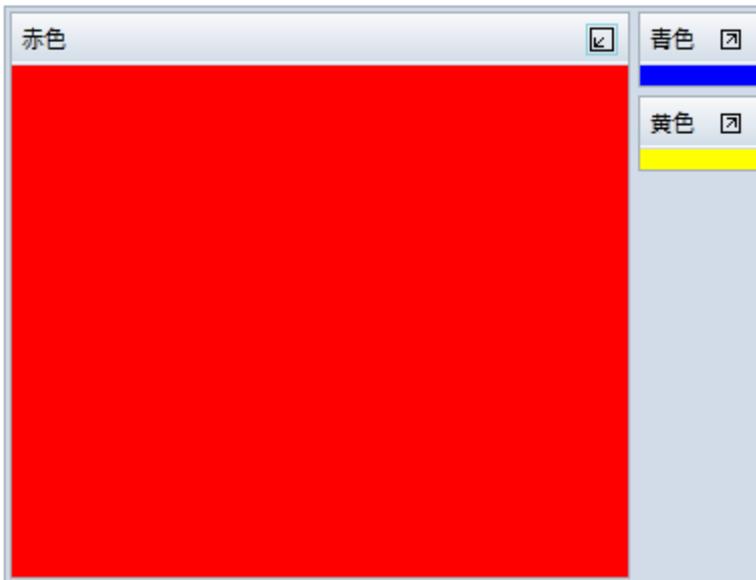
**C1TileViewItem** に追加されたコンテンツは、コンテンツ領域に表示されます。ヘッダー領域には、キャプションバータイトルを追加できます。右上隅のボタンは、**C1TileView** コントロールを最大化または最小化します。

## TileViewItem の状態

各 **C1TileViewItem** には、最小化状態、最大化状態、デフォルト状態(最小化状態でも最大化状態でもない)の3つの状態があります。たとえば、次の図では、**C1TileView** コントロール内にある3つの **C1TileViewItem** がすべてデフォルト状態で表示されています。



次の図では、赤色の **C1TileViewItem** が最大化され、他の2つの項目が最小化されています。



1つの項目が最大化されると、他の項目は最小化され、**MinimizedItem** プロパティで指定された方法で表示されます。デフォルト状態では、**Columns** プロパティと **Rows** プロパティを使用してレイアウトが決定されます。最小化/最大化状態では、**MinimizedItemPosition** プロパティを使用してレイアウトが決定されます。

## 列と行

**Columns** プロパティと **Rows** プロパティはそれぞれ、**C1TileViewItem** が配置されている列の数と行の数を取得または設定します。この値が0の場合は、スクロールの必要がない最小数が使用されます。**Columns** と **Rows** の両方が0の場合、項目は1つのマス目に配置されます。

デフォルト状態では、**Columns** プロパティと **Rows** プロパティを使用してレイアウトが決定されます。最小化/最大化状態では、**MinimizedItemPosition** プロパティを使用してレイアウトが決定されます。状態の詳細については、「**TileViewItem の状態**」を参照してください。

## 最小化項目の位置

**MinimizedItemsPosition** プロパティを使用すると、最小化された項目を **C1TileView** コントロール内のどこに表示するかを指定できます。オプションには、**[左]**、**[右]**、**[上]**、および**[下]**があります。デフォルトでは、最小化された項目はパネルの右に表示されます。

**C1TileView** のデフォルト状態では、**Columns** プロパティと **Rows** プロパティを使用してレイアウトが決定されます。最小化/最大化状態では、**MinimizedItemPosition** プロパティを使用してレイアウトが決定されます。状態の詳細については、「**TileViewItem の状態**」を参照してください。

## ドラッグ & ドロップ操作

**CanUserReorder** プロパティを設定して、**C1TileView** コントロールでドラッグ & ドロップ操作を許可するかどうかを簡単に指定できます。デフォルトでは、このプロパティが **True** に設定されており、ユーザーは実行時に項目を並べ替えることができます。反対に、このプロパティが **False** に設定されている場合は、実行時に項目を並べ替えることができません。例については、「**ドラッグ & ドロップ機能を無効にする**」を参照してください。

## 基本的なプロパティ

**TileView for UWP**には、コントロールの機能を設定するためのいくつかのプロパティがあります。主要なプロパティを次に示します。

# TileView for UWP

次のプロパティを使用して、C1TileView コントロールをカスタマイズできます。

プロパティ	説明
<b>AnimationDuration</b>	項目の並べ替えにかかる時間を取得または設定します。
<b>AnimationEasingFunction</b>	並べ替え動作を滑らかにする方法を取得または設定します。
<b>CanUserReorder</b>	ユーザーがこのコントロールから C1TileViewItem をドラッグ & ドロップして並べ替えできるかどうかを取得または設定します。
<b>Columns</b>	C1TileViewItem が配置されている列の数を取得または設定します。この値が0の場合は、スクロールの必要がない最小数が使用されます。 <b>Columns</b> と <b>Rows</b> の両方が0の場合、項目は1つのマス目に配置されます。
<b>ItemTemplateHeader</b>	項目のタイトルとして使用される <b>DataTemplate</b> を取得または設定します。
<b>ItemTemplateMaximized</b>	<b>Maximized()</b> 状態の項目に使用される <b>DataTemplate</b> を取得または設定します。
<b>ItemTemplateMinimized</b>	<b>Minimized()</b> 状態の項目に使用される <b>DataTemplate</b> を取得または設定します。
<b>MaximizedIndex</b>	選択されている項目の <b>Items</b> コレクション内のインデックスを取得または設定します。
<b>MaximizedItem</b>	現在強調表示されている <b>Items</b> コレクションのメンバを取得または設定します。
<b>MinimizedItemsPosition</b>	項目が最小化された状態のストリップを配置する場所を取得または設定します。スクロールバーは、ストリップの右または下にあります。
<b>Rows</b>	<b>C1TileViewItem</b> が配置されている行の数を取得または設定します。この値が0の場合は、スクロールの必要がない最小数が使用されます。 <b>Columns</b> と <b>Rows</b> の両方が0の場合、項目は1つのマス目に配置されます。
<b>ScrollBarStyle</b>	内部スクロールバーに使用されるスタイルを取得または設定します。
<b>ScrollBarVisibility</b>	スクロールバーを可視にするかどうかを取得または設定します。
<b>UpdateSourceCollection</b>	項目の順序の変更を <b>Items</b> または <b>ItemsSource</b> に書き込むかどうかを取得または設定します。

## タスク別ヘルプ

次のタスク別ヘルプトピックは、ユーザーの皆様が Visual Studio および **Expression Blend** に精通しており、C1TileView コントロールの一般的な使用方法を理解していることを前提としています。**TileView for UWP**製品を初めて使用される場合は、まず「**クイックスタート**」を参照してください。

このセクションの各トピックは、**TileView for UWP**製品を使用して特定のタスクを実行するための方法を提供します。また、ほとんどのタスク別ヘルプトピックは、新しい **UWP** プロジェクトが作成されており、そのプロジェクトに C1TileView コントロールが追加されていることを前提としています。

## アプリケーションに C1TileView を追加する

このトピックでは、アプリケーションに C1TileView コントロールを追加します。次の手順に従います。

1. Visual Studio 2012 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ストア]を選択し、テンプレートリストで[新しいアプリケーション (XAML)]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。
3. **MainPage.xaml** が開いていない場合は開きます。<Grid> タグと</Grid> タグの間にカーソルを置き、1回クリックします。
4. ツールボックスに移動し、C1TileView アイコンをダブルクリックして、コントロールをグリッドに追加します。これで、参照と XAML 名前空間が自動的に追加されます。XAML マークアップは次のようになります。

```
<Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">
    <TileView:C1TileView/>
</Grid>
```

### 🟢 ここまでの成果

これで、アプリケーションのユーザーインターフェイスが正しくセットアップされましたが、このアプリケーションを実行すると、**C1TileView** コントロールにコンテンツがないことがわかります。詳細については、「**C1TileView に項目を追加する**」を参照してください。

 **メモ:** C1TileView コントロールが Visual Studio のツールボックスにインストールされている場合は、ページにコントロールをドラッグするだけで、上のすべての手順が自動的に実行されます。

## C1TileView に項目を追加する

このトピックでは、C1TileView コントロールに C1TileViewItem を追加します。このトピックは、空の C1TileView コントロールがアプリケーションに追加されていることを前提としています。

<TileView:C1TileView x:Name="C1TileView1" /> タグを編集して、いくつかの **C1TileViewItems** を追加します。XAML は次のようになります。

```
<TileView:C1TileView Name="C1TileView1">
    <TileView:C1TileViewItem Background="Red" Header="赤色"></TileView:C1TileViewItem>
    <TileView:C1TileViewItem Background="Orange" Header="オレンジ色">
</TileView:C1TileViewItem>
    <TileView:C1TileViewItem Background="Yellow" Header="黄色">
</TileView:C1TileViewItem>
    <TileView:C1TileViewItem Background="Green" Header="緑色">
```

# TileView for UWP

```
</TileView:C1TileViewItem>
    <TileView:C1TileViewItem Background="Blue" Header="青色">
</TileView:C1TileViewItem>
    <TileView:C1TileViewItem Background="Purple" Header="紫色">
</TileView:C1TileViewItem>
</TileView:C1TileView>
```

これで、C1TileView コントロールに6つの C1TileViewItem が追加されました。

## ドラッグ & ドロップ機能を無効にする

デフォルトでは、ドラッグ & ドロップ機能が有効化されており、ユーザーは実行時に C1TileViewItem 要素を並べ替えることができます。ただし、必要に応じて、**CanUserReorder** プロパティを **False** に設定することで、ドラッグ & ドロップ機能を無効にすることができます。

### 設計時

設計時に[プロパティ]ウィンドウで C1TileView コントロールでドラッグ & ドロップ機能を無効にするには、次の手順に従います。

1. C1TileView コントロールをクリックして選択します。
2. [プロパティ]ウィンドウに移動し、**CanUserReorder** プロパティを見つけます。
3. **CanUserReorder** プロパティの横にあるドロップダウン矢印をクリックし、**False** を選択します。  
これで、ドラッグ & ドロップ機能が無効になります。

### XAML の場合

XAML で C1TileView コントロールのドラッグ & ドロップ機能を無効にするには **CanUserReorder="False"** を **<TileView:TileView>** タグに追加します。次のようになります。

```
<TileView:C1TileView Name="C1TileView1" CanUserReorder="False">
```

### コードの場合

ウィンドウを右クリックし、[コードの表示]を選択してコードエディタを開きます。コードをメインクラスに追加します。次のようになります。

#### ▶ Visual Basic コードの書き方

```
Visual Basic
Public Sub New()
    InitializeComponent()
    Me.C1TileView1.CanUserReorder = False
End Sub
```

#### ▶ C# コードの書き方

```
C#
public MainPage()
{
    InitializeComponent();
    this.C1TileView1.CanUserReorder = false;
}
```

```
}

```

## プロジェクトの実行と確認

実行時にドラッグ & ドロップ操作を実行できなくなります。

## ヘッダーの外観をカスタマイズする

**C1TileView** には、**C1TileViewItem** の **Header** の外観を変更できるプロパティがいくつかあります。これらのプロパティは、**Header**、**HeaderBackground**、**HeaderFontFamily**、**HeaderFontSize**、**HeaderFontStretch**、**HeaderFontStyle**、**HeaderFontWeight**、**HeaderForeground**、**HeaderPadding**、および **HeaderTemplate** です。

たとえば、次のマークアップは、これらのプロパティのいくつかを設定します。

```
<TileView:C1TileViewItem Header="ニュース" HeaderPadding="10 5 5 5" HeaderForeground="#FF507494"
HeaderFontFamily="Trebuchet MS" HeaderFontSize="16">
  <TileView:C1TileViewItem.HeaderBackground>
    <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
      <GradientStop Color="#FFE9ECF0" Offset="0" />
      <GradientStop Color="#FFDDE1E7" Offset="0.2" />

      <GradientStop Color="#FFCCD3DC" Offset="0.2" />

      <GradientStop Color="#FFFAFAFB" Offset="0.647" />
    </LinearGradientBrush>
  </TileView:C1TileViewItem.HeaderBackground>
</TileView:C1TileViewItem>
```

## 最大化/最小化項目テンプレートを作成する

**C1TileView** コントロールの項目を最小化および最大化した場合の表示方法をカスタマイズできます。たとえば、最小化された項目のコンテンツ領域にアイコンを表示して、その項目に含まれるコンテンツの種類を示すことができます。**ContentMinimized** プロパティおよび **ContentMaximized** プロパティを使用して、表示テンプレートを設定できます。これらのプロパティが設定されていない場合は、**Content** が使用されます。

たとえば、次のマークアップは、**ContentMinimized** テンプレートと **ContentMaximized** テンプレートを追加します。

```
<TileView:C1TileView Name="C1TileView1">
  <TileView:C1TileViewItem Background="Red" Header="赤色">
    <TileView:C1TileViewItem.ContentMinimized>
      <TextBlock Text="最小化された赤色" Height="28" Name="TextBlock1"
Foreground="White"/>
    </TileView:C1TileViewItem.ContentMinimized>
    <TileView:C1TileViewItem.ContentMaximized>
      <TextBlock Text="最大化された赤色" Height="28" Name="TextBlock2"
Foreground="White"/>
    </TileView:C1TileViewItem.ContentMaximized>
  </TileView:C1TileViewItem>
  <TileView:C1TileViewItem Background="Orange" Header="オレンジ">
    <TileView:C1TileViewItem.ContentMinimized >
      <TextBlock Text="最小化されたオレンジ" Height="28" Name="TextBlock3"
Foreground="White"/>
    </TileView:C1TileViewItem.ContentMinimized>
    <TileView:C1TileViewItem.ContentMaximized >
      <TextBlock Text="最大化されたオレンジ" Height="28" Name="TextBlock4"
Foreground="White"/>
    </TileView:C1TileViewItem.ContentMaximized>
  </TileView:C1TileViewItem>
</TileView:C1TileView>
```

# TileView for UWP

```
</TileView:C1TileViewItem>  
</TileView:C1TileView>
```

## ✔ ここまでの成果

C1TileView の最小化および最大化された状態に使用されるテンプレートを追加しました。アプリケーションを実行し、項目の1つを最大化して、最小化された項目も最大化された項目もコンテンツが変更されることを確認してください。最小化された項目を最大化すると、各項目のコンテンツが再度変更されます。

