

---

ComponentOne

# TileControl for WinForms

This manual was produced using *ComponentOne Doc-To-Help*.

# Contents

ComponentOne TileControl for WinForms Overview .....	1
Help with ComponentOne Studio for WinForms .....	1
TileControl for WinForms Key Features .....	3
TileControl for WinForms Quick Start .....	4
Step 1 of 3: Creating the C1Tile Application.....	4
Step 2 of 3: Creating the Template with Elements .....	5
Step 3 of 3: Applying the Template to the Specified Tile.....	8
Design-Time Support.....	9
C1TileControl Context Menu .....	9
C1TileControl Smart Tag .....	11
Group Tasks.....	12
Tile Tasks.....	13
C1TileControl Collection Editors.....	14
C1TileControl.Templates Collection Editor.....	14
Template.Elements Collection Editor.....	15
C1TileControl.Groups Collection Editor.....	16
Group.Tiles Collection Editor .....	17
TileControl Layout.....	18
TileControl Behavior.....	20
TileControl Scrolling.....	20
TileControl Navigation.....	22
TileControl Touchscreen Support.....	22
TileControl Templates .....	22
TileControl Groups .....	23
TileControl Tiles and Elements.....	24
Image Element .....	26
Panel Element .....	27
Text Element.....	27
TileControl for WinForms Samples.....	28
TileControl for WinForms Task-Based Help.....	29
Adding Templates to the C1TileControl.....	29
Assigning a Template to a Specified Tile .....	29
Adding Elements to a Template.....	30
Changing the BackColor of the Template.....	30
Removing Specific Templates .....	31
Alternating the Text View by a Timer .....	32
Saving and Loading TileControl as an XML File.....	35
Loading TileControl From an XML File.....	35

Saving TileControl as an XML File .....	36
Setting Text for the TileControl and Group .....	37
Adding Groups to the C1TileControl.....	38
Removing Groups from the C1TileControl .....	38
Modifying the Group's Font Properties .....	39
Setting the TileControl's BackColor.....	39
Adding Tiles to a Specific Group .....	40
Creating a CheckMark for the Tile .....	41
Increasing the Size of a Specific Tile .....	42
Adding Image Elements to a Tile .....	43
Adding a Symbol to a Tile .....	43
Drawing an Image at Runtime .....	43
Index .....	47

# ComponentOne TileControl for WinForms Overview

The Windows 8-inspired ComponentOne TileControl™ for WinForms makes it easy to replicate the Windows 8 Modern UI experience in your desktop app. Get several different tile controls that support panning and tapping gestures. Combine tiles with different containers to achieve endless layout possibilities.

For a list of the latest features added to **ComponentOne Studio for WinForms**, visit [What's New in Studio for WinForms](#).

 **Getting Started**

- [TileControl for WinForms Quick Start](#) (page 4)
- [TileControl for WinForms Key Features](#) (page 3)
- [TileControl for WinForms Task-Based Help](#) (page 29)

---

## Help with ComponentOne Studio for WinForms

### Getting Started

For information on installing ComponentOne Studio for WinForms, licensing, technical support, namespaces and creating a project with the control, please visit [Getting Started with Studio for WinForms](#).

### What's New

For a list of the latest features added to **ComponentOne Studio for WinForms**, visit [What's New in Studio for WinForms](#).



# TileControl for WinForms Key Features

The following are some of the main features of C1TileControl that you may find useful:

## Custom Tile Layout

The TileControl uses either automatic or manual layout of tiles in the groups. Tiles may be any size; they are not limited to large and small sizes only. The interior layout of the tiles (or rather tile templates) is very flexible. You can use docked and stacked panels, nested panels, text elements, and images. Additionally, you can save the layout to an XML file and load it from an XML file at any time.

## Two Display Orientations

Tile groups can be stacked vertically or horizontally.

## Images

The background image of the control can be scrolled with tiles, as on Windows 8 Start Screen. In addition to user images, there is a set of standard symbols of various sizes that can be displayed on tiles. Also, it's easy to display the "badge number" or "5-star" rating image as a part of the template. These elements can be bound to an integer value specified in a property of the tile.

## Templates

There is no need to design each tile separately. Instead, you can create one or several tile templates, then associate these templates with tiles. Tiles can provide data for templates, such as strings, colors, and images. It's possible to associate one template with several tiles, and to switch templates for a single tile; for example, to alternate text and image views by a timer. For an example of how to do this see the

## Touch Support

The TileControl supports panning, tapping, and checking tiles (with a swipe gesture) using the touch input hardware on a machine with Windows 7 or Windows 8. When users press and hold a tile with their fingertip, the tile shows its tooltip.

## Navigation

It is easy to navigate between tiles using the keyboard.

# TileControl for WinForms Quick Start

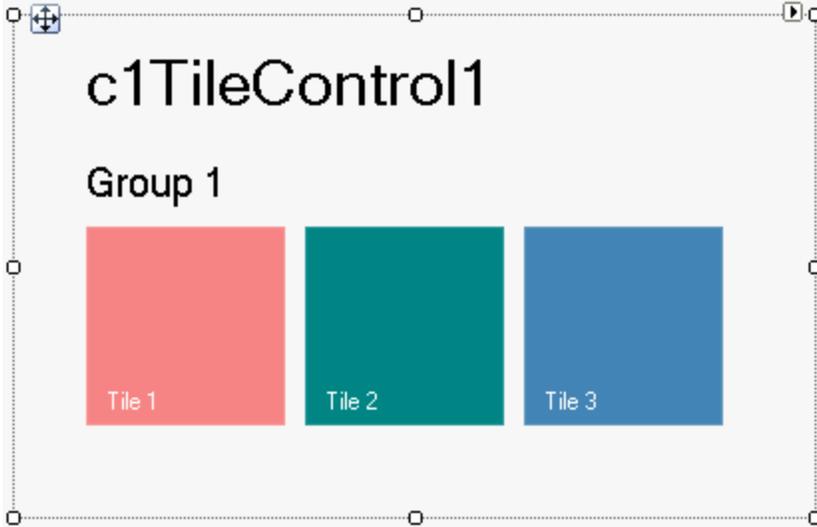
The goal of this quick start guide is to get you acquainted with **Tile Control for WinForms**. In the first step of this Quick Start guide, you will add a C1TileControl to your WinForms project. This quick start guide will also explain how to add the C1Tile control to your application, add content that will be displayed in the C1Tile control, and observe some of the run-time interactions possible with **TileControl for WinForms**.

---

## Step 1 of 3: Creating the C1Tile Application

In this step, you will create a .NET project using **Tile Control for WinForms**. When you add a C1Tile control to your application, you'll have an interface that you can display content in. To set up your project and add a C1Tile control to your application, complete the following steps:

1. Begin by creating a new Windows Forms Application. In this example the application will be named "QuickStart". If you name the project something else, in later steps you may need to change references to "QuickStart" with the name of your project.
2. In the Solution Explorer, right-click the project name and choose **Add Reference**. Select the **Browse** tab to locate **C1.Win.C1TileControl.2.dll**. In the **Add Reference** dialog box, select the C1.Win.C1TileControl.2.dll and click **OK** to add references to your project.
3. While in Design view, navigate to the Visual Studio Toolbox and double-click the **C1TileControl** to add it to your form. If it's not there right-click in the toolbox area and select **Add Tab**. Name the tab, for example, **C1TileControl**. Right-click under the C1TileControl and select **Choose Items**. The **Choose Toolbox Items** appears. Browse to the **C1.Win.C1Tile** assembly and click **OK**.
4. The **C1TileControl** control appears.



---

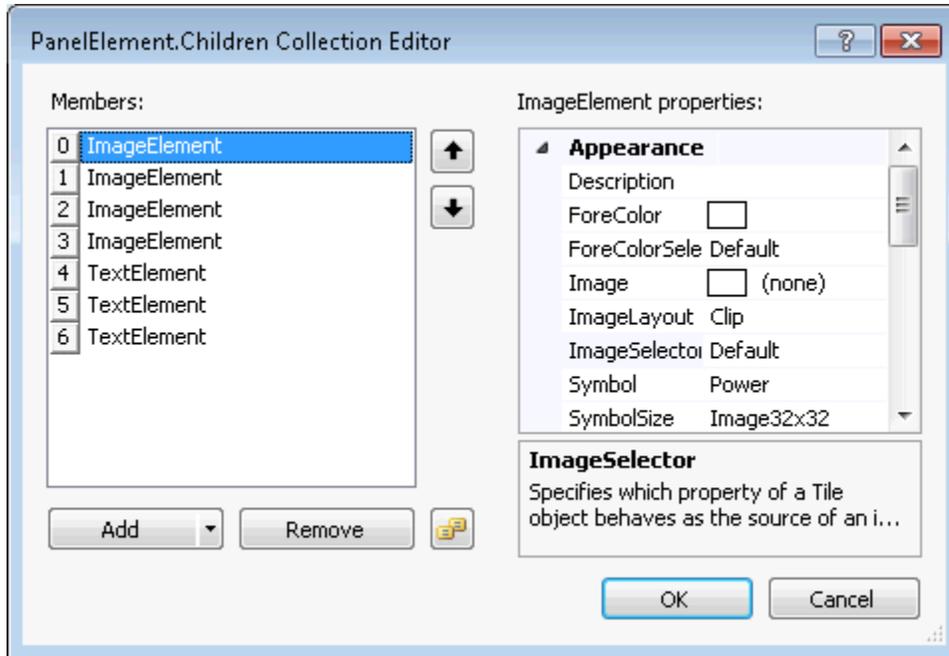
## Step 2 of 3: Creating the Template with Elements

In the previous step you created a WinForms application and added the **C1TileControl** to your project.

To add a panel to your tile with image and text elements, complete the following steps:

### To add a template with elements to the Tile at design time:

1. Right-click the default **Tile 1** and select **Edit Templates**. The **C1TileControl.Templates Collection Editor** appears.
2. Click **Add** to add a new template to Tile1.
3. Click on the ellipsis button next to **Elements** to open the **Template.Elements Collection Editor**. Select **Add** and click on the dropdown arrow to select the **PanelElement** and then the **TextElement**. This will add elements into the TemplateCollection.
4. Select **PanelElement** from the Members list and enter **9** next to the **ChildSpacing** property.
5. Click on the ellipsis button next to **Children** so the **PanelElement.Children Collection Editor** appears. Select **Add** and click on the dropdown arrow to select the **ImageElement**. Add 4 **ImageElements** and 3 **TextElements**. This will add elements into the TemplateCollection.



6. Select the first [0] **ImageElement** from the **Members** list and set its properties to the following:
  - **ImageSelector** property to **UnboundSymbol**. This will make the symbol that you select act as an image for the specified Tile.
  - **Symbol** property to **LeftToRight**. This will make the LeftToRight image appear on the specified Tile.
  - **SymbolSize** property to **Image64x64**. This will change the default symbol size from **32x32** to **64x64**.
  - **FixedHeight** to **70**. This will set the height of the contents in the panel to 70 pixels.
7. Select the second [1]**ImageElement** from the **Members** list and set its properties to the following:
  - **ForeColor** property to **255, 192, 255**.
  - **ForeColorSelector** property to **Unbound**.
  - **ImageSelector** property to **UnboundSymbol**.
  - **Symbol** property to **CircleWithPlus**.
8. Select the third [2]**ImageElement** from the **Members** list and set its properties to the following:
  - **ForeColor** property to **255, 192, 255**.
  - **ForeColorSelector** property to **Unbound**.
  - **ImageSelector** property to **UnboundSymbol**.
  - **Symbol** property to **CircleWithMinus**.
  - **Alignment** property to **TopCenter**. This will align the symbol to the TopCenter of the Panel.
9. Select the fourth [3]**ImageElement** from the **Members** list and set its properties to the following:
  - **ForeColor** property to **255, 224, 192**.
  - **ForeColorSelector** property to **Unbound**.
  - **ImageSelector** property to **UnboundSymbol**.

- **Symbol** property to **CircleWithMultiply**.
  - **Alignment** property to **BottomCenter**.
10. Select the fifth [4]**TextElement** from the **Members** list and set its properties to the following:
    - **ForeColor** property to **192, 192, 255**.
    - **ForeColorSelector** property to **Unbound**.
    - **Text** property to **Top**.
    - **TextSelector** to **Unbound**.
    - **Alignment** to **TopCenter**.
  11. Select the sixth [5]**TextElement** from the **Members** list and set its properties to the following:
    - **BackColorSelector** property to **Unbound**.
    - **ForeColor** property to **255, 224, 192**.
    - **ForeColorSelector** property to **Unbound**.
    - **Text** property to **Bottom**.
    - **TextSelector** property to **Unbound**. This will make the new text Bottom appear rather than the default text.
    - **Alignment** property to **BottomCenter**.
    - **DirectionVertical** property to **True**.
  12. Select the seventh [6]**TextElement** from the **Members** list and set its properties to the following:
    - **BackColorSelector** property to **Unbound**.
    - **ForeColor** property to **255, 192, 255**.
    - **ForeColorSelector** property to **Unbound**.
    - **Text** property to **Middle**.
    - **TextSelector** property to **Unbound**.
  13. Click **OK** to save and close the **PanelElement.Children Collection Editor**.
  14. In the **Template.Elements Collection Editor** select the [1] **TextElement** from the **Members** list.
  15. Select the second [1]**TextElement** from the **Members** list and set its properties to the following:
    - **Alignment** property to **BottomCenter**.
    - **Margin** property to **0, 0, 0, 5**.
  16. Click **OK** to save and close the **Template.Elements Collection Editor**.
  17. Click **OK** to save and close the **C1TileControl.Templates Collection Editor**.

**Run and observe the following:**

Tile1 will appear the same since the template that we created, Template1, has not been applied to the Tile's **Template** property.

In the next step you will learn how to assign the new template to the specified tile as well as modify a few of the Tile's properties using the **C1TileControl Tasks** menu.

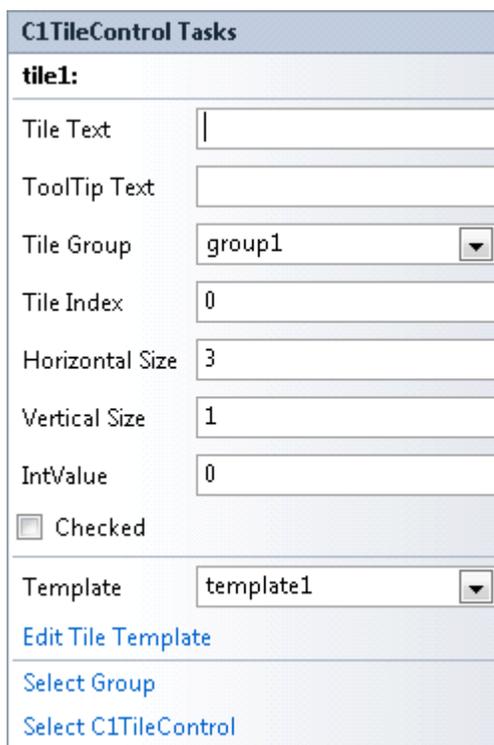
---

## Step 3 of 3: Applying the Template to the Specified Tile

In the previous step you created a template and added template elements such as panels, images, and text. You also set the alignment and layout properties for the elements in the template. In this step we will apply the template to the specified tile and set a few tile properties such as the `BackColor`, `Template`, `HorizontalSize`, and `Text` property.

To apply the template to the first tile as well as modify a few properties for the first tile of the `C1TileControl`, complete the following steps:

1. Select **Tile 1** to open its **C1TileControl Tasks** menu.
2. In the **C1TileControl Tasks** menu remove the default text, `Tile1`, set the **Template** property to **Template** and **HorizontalSize** property to **3**.



The screenshot shows the **C1TileControl Tasks** dialog box for **tile1**. The dialog has a title bar with the text "C1TileControl Tasks" and a sub-header "tile1:". Below the sub-header are several input fields and a checkbox:

- Tile Text**: An empty text box.
- ToolTip Text**: An empty text box.
- Tile Group**: A dropdown menu with "group1" selected.
- Tile Index**: A text box containing "0".
- Horizontal Size**: A text box containing "3".
- Vertical Size**: A text box containing "1".
- IntValue**: A text box containing "0".
- Checked**: A checkbox that is currently unchecked.
- Template**: A dropdown menu with "template1" selected.

At the bottom of the dialog, there are three blue links: [Edit Tile Template](#), [Select Group](#), and [Select C1TileControl](#).

3. Right-click on the first Tile you have modified and select **Edit Groups**. The **C1TileControl.Groups Collection Editor** appears.
4. Click on the ellipsis button next to the **Tiles** to open the **Group.Tiles Collection Editor** and select `tile1[]`.
5. Set the **BackColor** property to **DimGrey** for `tile1[]`.
6. Click **OK** to save and close the **Group.Tiles Collection Editor** and click **OK** to save and close the **C1TileControl.Groups Collection Editor**.

### Run and observe the following:

Template1 is applied to the first Tile as well as the Tile settings.



### ✔ What You've Accomplished

Congratulations! You have successfully completed the `C1TileControl` quick start. In this topic, you added a `C1TileControl` to your windows form, created a template for a specific tile, and set a few of the tile's properties.

# Design-Time Support

`C1TileControl` provides customized context menus, smart tags, and a designer that offers rich design-time support and simplifies working with the object model.

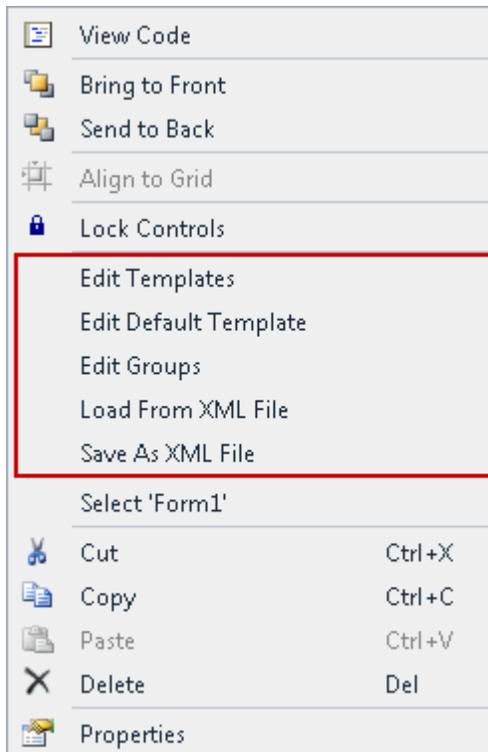
The following topics describe how to use `C1TileControl` design-time environment to configure `C1TileControl`.

---

## C1TileControl Context Menu

The `C1TileControl` control provides a context menu for additional functionality to use at design time.

To access `C1TileControl`'s context menu, right-click on the `C1TileControl` control and the context menu for it appears like the following:



The **C1TileControl** context menu operates as follows:

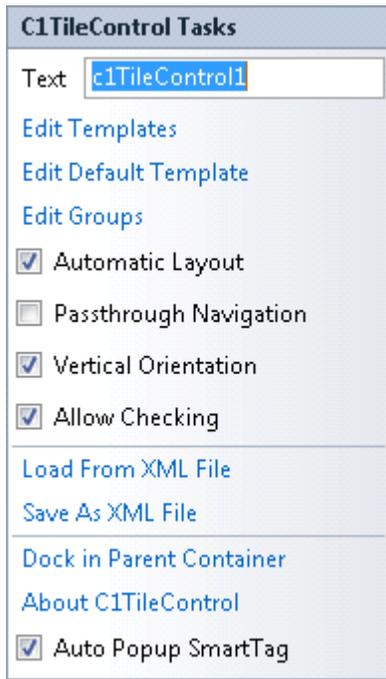
- **Edit Templates**  
Selecting the **Edit Templates** opens the **C1TileControl.Templates Collection Editor** where you can add, remove, or modify templates.
- **Edit Default Template**  
Selecting the **Edit Default Template** item opens the **Template.Elements Collection Editor** where you can add text, image, and panels to the templates in the C1TileControl.
- **Edit Groups**  
Selecting the **Edit Groups** item opens the **C1TileControl.Groups Collection Editor** where you can add, remove, or modify groups for the C1Tile control.
- **Load From Xml File**  
Selecting the **Load from Xml File** opens the **Load From Xml File** dialog box where you browse to the .xml file you wish to load.
- **Save As Xml File**  
Selecting the **Save As Xml File** opens the **Save As Xml File** dialog box where you browse to the .xml file you wish to save.
- **About C1TileControl**  
Clicking **About C1TileControl** shows the **About ComponentOne** dialog box. This dialog box displays the version number and licensing information for the ComponentOne C1TileControl product.

---

# C1TileControl Smart Tag

In Visual Studio, each component in **TileControl for WinForms** includes a smart tag. A smart tag represents a short-cut tasks menu that provides the most commonly used properties in each control.

To access the **C1TileControl Tasks** menu, click the smart tag (▾) in the upper right corner of the **C1TileControl** control. This will open the **C1TileControl Tasks** menu.



The **C1TileControl Tasks** menu operates as follows:

## Text

Clicking in the textbox next to the Text item will create text that appears on the top of the TileControl.

## Edit Templates

Clicking the **Edit Templates** opens the **C1TileControl.Templates Collection Editor** where you can add, remove, or modify templates..

## Edit Default Template

Clicking the **Edit Default Template** item opens the **Template.Elements Collection Editor** where you can add text, image, and panels to the templates in the **C1TileControl**.

## Edit Groups

Clicking the **Edit Groups** item opens the **C1TileControl.Groups Collection Editor** where you can add, remove, or modify groups for the C1TileControl.

## Vertical Orientation

Selecting the **Vertical Orientation** checkbox will align the C1TileControl vertically

## Automatic Layout

When selected, enables automatic layout.

### **Passthrough Navigation**

When selected, enables passthrough navigation.

### **Allow Checking**

When selected, enables checking.

### **Load From Xml File**

Clicking the **Load from Xml File** opens the **Load From Xml File** dialog box where you browse to the .xml file you wish to load.

### **Save As Xml File**

Clicking the **Save As Xml File** opens the **Save As Xml File** dialog box where you browse to the .xml file you wish to save.

### **Dock in Parent Container**

Clicking Dock in Parent Container will dock the C1TileControl in its parent container.

### **About C1TileControl**

Clicking **About C1TileControl** shows the **About ComponentOne** dialog box. This dialog box displays the version number and licensing information for the ComponentOne GanttView product.

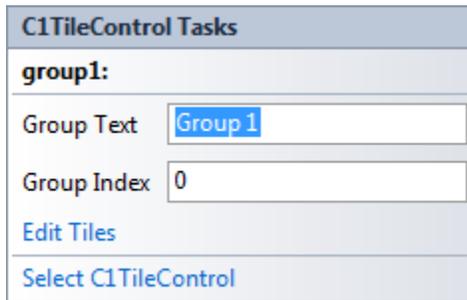
### **Auto Popup SmartTag**

Unselecting the **Auto Popup SmartTag** checkbox will disable the popup smart tag when you click on the group or each tile/panel.

---

## **Group Tasks**

To access the **C1TileControl Tasks** menu, click any of the groups in the **C1TileControl** control. This will open the **C1TileControl Tasks** menu.



The **C1TileControl Tasks** menu operates as follows:

#### **Group Text**

Clicking in the textbox next to the **Group Text** item will create text that appears on the top of the **Group** in the C1TileControl.

#### **Group Index**

Specifies the position of the Tile within the group.

#### **Edit Tiles**

Clicking the **Edit Tiles** item opens the **Group.Tiles Collection Editor** where you can add, remove, or modify the tiles within the group.

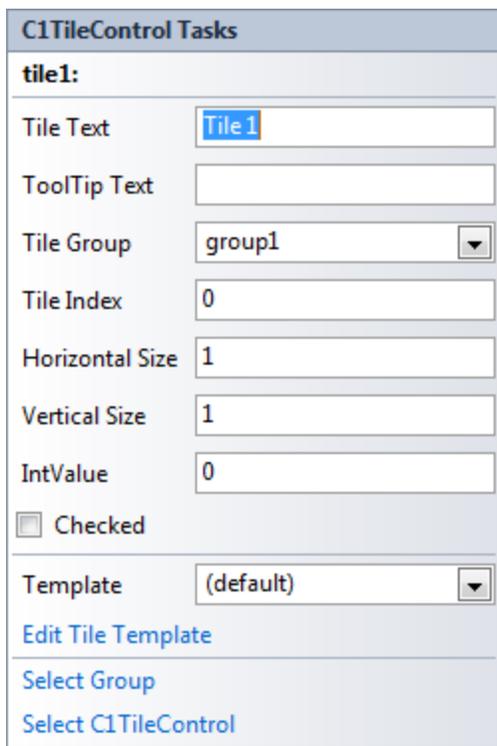
### Select C1TileControl

Clicking the **Select C1TileControl** item selects the C1TileControl.

---

## Tile Tasks

To access the **C1TileControl Tasks** menu, click any of the tiles in the **C1TileControl** control. This will open the **C1TileControl Tasks** menu.



The screenshot shows a dialog box titled "C1TileControl Tasks". It contains several input fields and a checkbox:

- tile1:** (header)
- Tile Text:** Textbox containing "Tile 1" (highlighted in blue).
- ToolTip Text:** Empty textbox.
- Tile Group:** Dropdown menu showing "group1".
- Tile Index:** Textbox containing "0".
- Horizontal Size:** Textbox containing "1".
- Vertical Size:** Textbox containing "1".
- IntValue:** Textbox containing "0".
- Checked**
- Template:** Dropdown menu showing "(default)".
- [Edit Tile Template](#)
- [Select Group](#)
- [Select C1TileControl](#)

The **C1TileControl Tasks** menu operates as follows:

### Tile Text

Clicking in the textbox next to the **Tile Text** item will create text that appears on the top of the **Tile** in the C1TileControl.

### ToolTip Text

Clicking in the textbox next to the **ToolTip Text** item will create text that appears when you hover over the text in the **Tile** of the C1TileControl.

### Tile Group

Clicking the dropdown arrow will show a menu that lists the existing groups. Select the group that you wish the tile to be in.

### Tile Index

Specifies the position of the Tile within the group.

### Horizontal Size

Specifies the horizontal size of the **Tile**.

### Vertical Size

Specifies the vertical size of the **Tile**.

### IntValue

Specifies the integer value of the **Tile**.

### Checked

Clicking on the checkbox will enable the checkmark so the **Tile** will have a checkmark on it that appears like the following:



### Edit Tile Template

Clicking the **Edit Tile Template** item opens the **Template.Elements Collection Editor** where you can add text, image, and panels to the templates in the **Tiles**.

### Select Group

Clicking the **Select Group** item selects the Group where the **Tile** is located within.

### Select C1TileControl

Clicking the **Select C1TileControl** item selects the **C1TileControl**.

---

## C1TileControl Collection Editors

**C1TileControl** provides the following collection editors that allow you to apply properties to the **C1TileControl** elements at design time:

- **C1TileControl.Templates Collection Editor**
- **Template.Elements Collection Editor**
- **C1TileControl.Groups Collection Editor**
- **Group.Tiles Collection Editor**

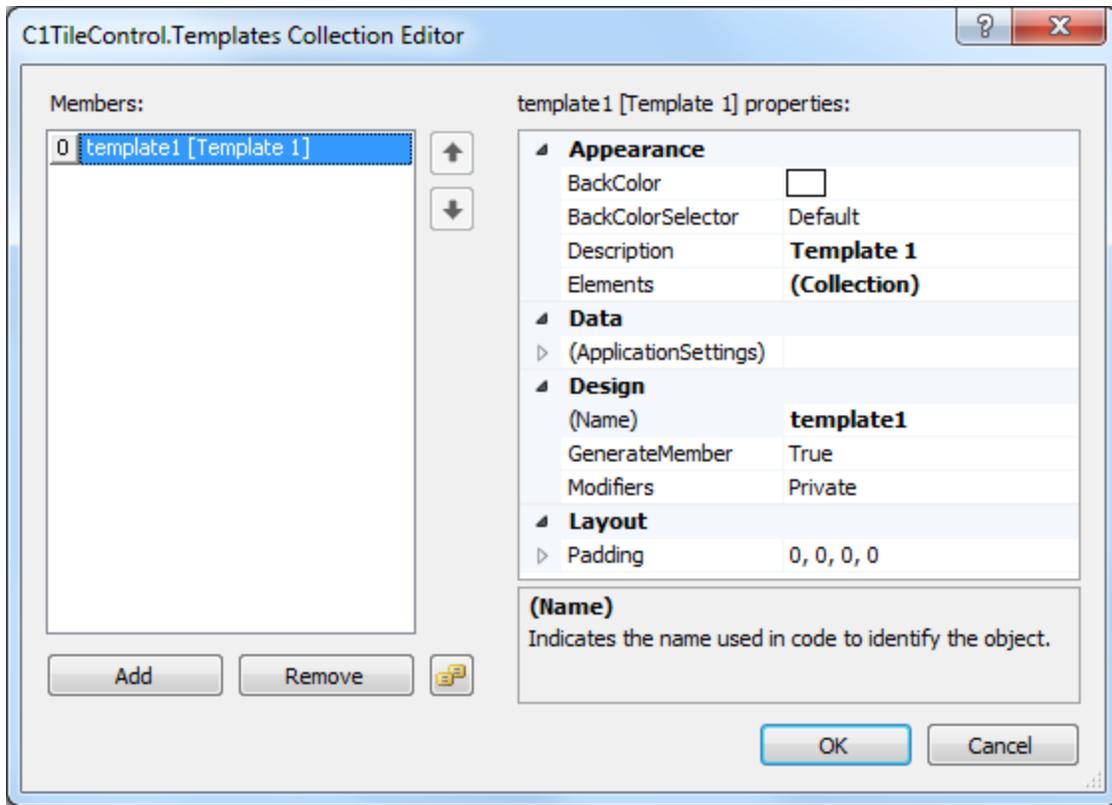
The following topics provide an overview of each **C1TileControl** collection editor and show how to access each of them:

### C1TileControl.Templates Collection Editor

The **C1TileControl.Templates Collection Editor** is used for adding templates to the **C1TileControl**. A template can hold elements such as text, images, and panels. These elements can be added to each template at design time through the **Template.Elements Collection Editor** or programatically. Each text, image, or panel element can hold children elements (text, image, and/or panel). For example, a panel element can include multiple text and image elements.

### To Access the C1TileControl.Templates Collection Editor

Right-click on the **C1TileControl** and select **Edit Templates** from its context menu. The **C1TileControl.Templates Collection Editor** appears like the following when a member is added to the collection:

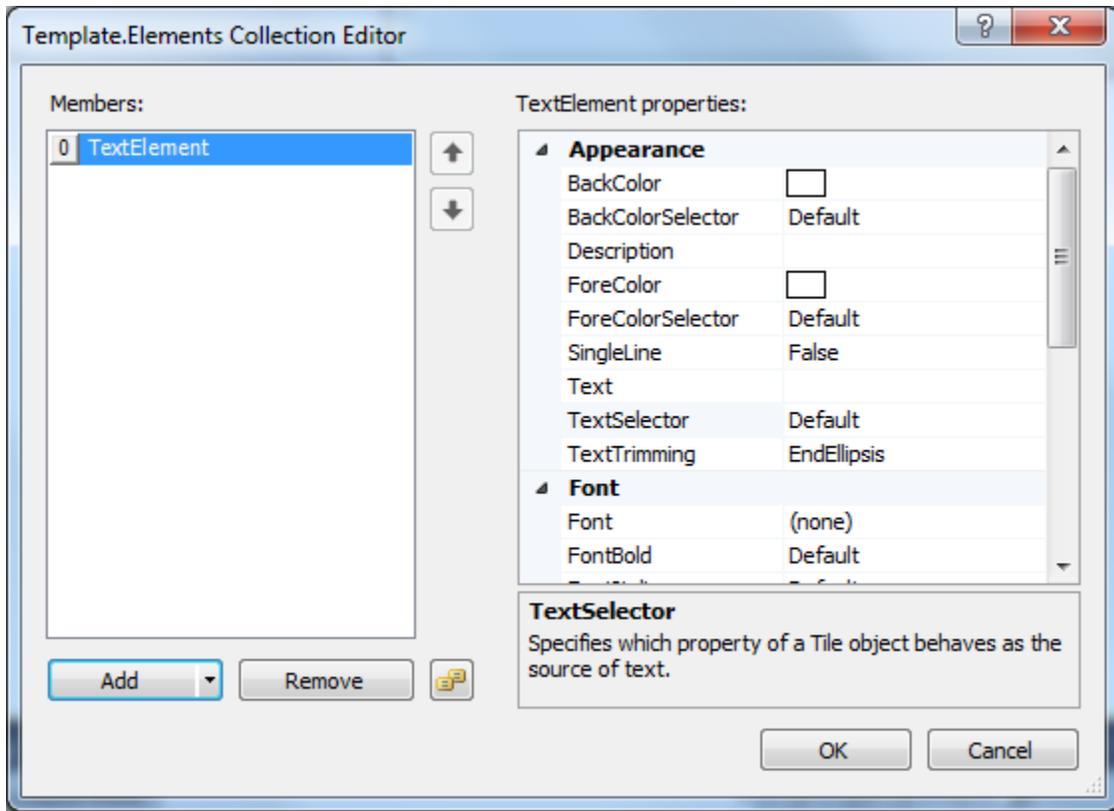


### Template.Elements Collection Editor

The **Template.Elements Collection Editor** is used for adding, removing, or modifying panel elements such as at design time.

#### To Access the Template.Elements Collection Editor

Right-click on the **C1TileControl** control and select **Edit Templates** from its context menu. The **C1TileControl.Templates Collection Editor** appears. Click **Add** to add a template item to the collection. Click on the ellipsis button in the **Elements** property. The **Template.Elements Collection Editor** appears. Click on the dropdown listbox and select a member such as **TextElement** to modify the **TextElement**'s properties.

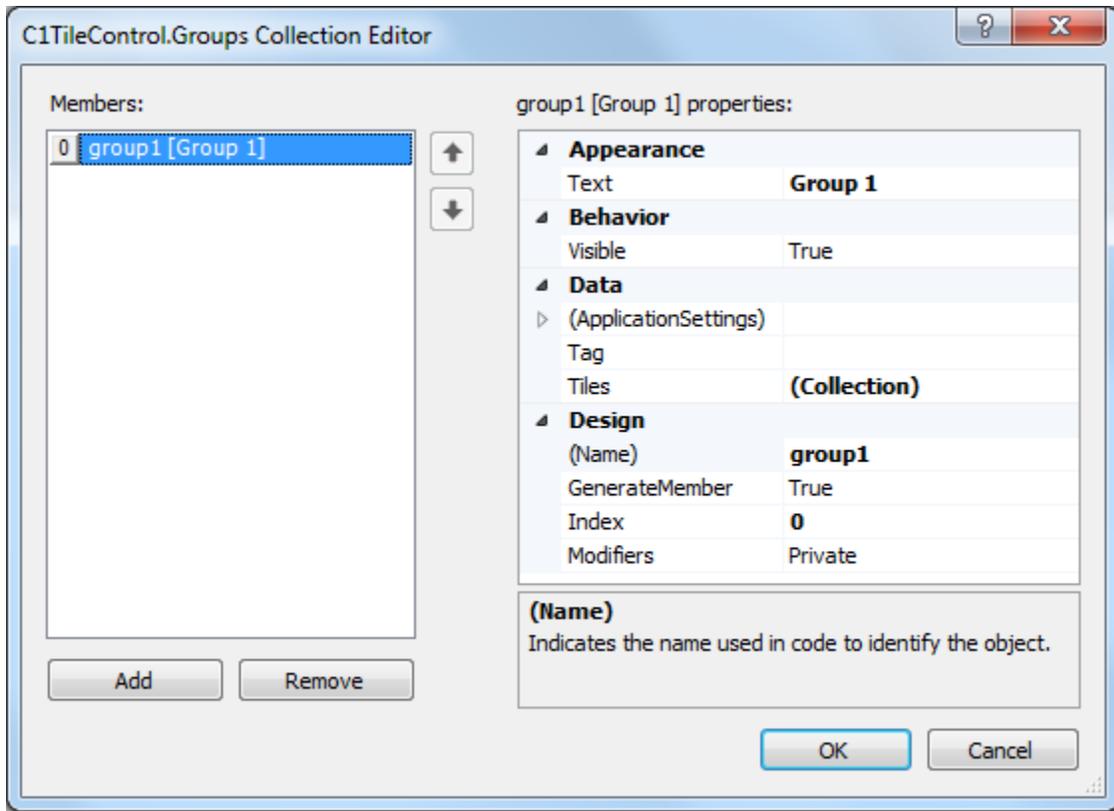


## C1TileControl.Groups Collection Editor

The **C1TileControl.Groups Collection Editor** is used for adding, removing, or modifying groups within the **C1TileControl** at design time.

### To Access the C1TileControl.Groups Collection Editor

Right-click on the **C1TileControl** and select **Edit Groups** from its context menu. The **C1TileControl.Groups Collection Editor** appears like the following when a member is added to the collection:

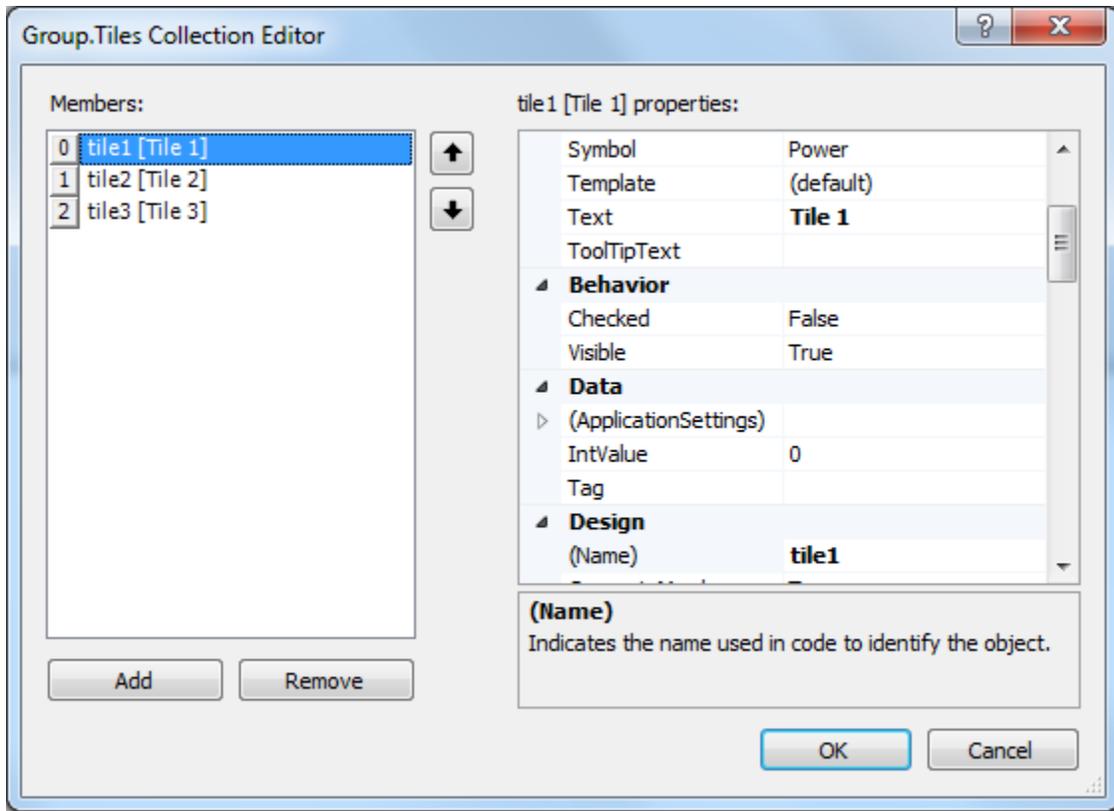


## Group.Tiles Collection Editor

The **Group.Tiles Collection Editor** is used for adding, removing, or modifying tiles within the group of the C1TileControl.

### To Access the Group.Tiles Collection Editor

Click on any group in the C1TileControl. Select **Edit Tiles** from the C1TileControl-Group tasks menu. The **Group.Tiles Collection Editor** appears:



# TileControl Layout

The TileControl uses either automatic or manual layout of tiles in the groups. The AutomaticLayout property determines whether the tiles are arranged automatically or manually. When manual layout is used, you can drag and drop the tiles anywhere on the form. Tiles may be any size; they are not limited to large and small sizes only. The interior layout of the tiles (or rather tile templates) is very flexible. You can use docked and stacked panels, nested panels, text elements, and images. Additionally, you can save the layout to an XML file and load it from an XML file at any time.

The layout of the Tiles in the TileControl is fully customizable through the layout properties. The tiles are arranged in each group horizontally by default, but can be changed to vertically through the Orientation property. Use the horizontal layout mode to make the Tile control appear like the Windows 8 UI and use the vertical layout to make the Tile control appear like an advanced listbox.

The following table lists the common surface/layout properties for all Tiles in the C1TileControl:

Property	Description
----------	-------------

<b>AutomaticLayout</b>	Indicates whether the tiles should be arranged automatically or manually.
<b>CellHeight</b>	Specifies the height of a single tile cell.
<b>CellSpacing</b>	Specifies the gap between tile cells in a group.
<b>CellWidth</b>	Specifies the width of a single tile cell.
<b>MaximumRowsOrColumns</b>	Specifies the maximum number of cell rows or columns in automatic layout mode.
<b>Orientation</b>	Specifies the method of arranging of the tile groups.
<b>ScrollBarStyle</b>	Specifies whether the default or system scroll bar should appear.
<b>ScrollOffset</b>	Specifies the negative or zero offset of the scrollable area.
<b>SurfaceContentAlignment</b>	Specifies the alignment of groups on the scrollable surface.
<b>SurfacePadding</b>	The interior spacing of the scrollable surface.

Once you apply the settings to the preceding Surface properties any new Tiles added will have those same settings. If you want different layouts for the tiles then you will need to apply unique templates to the tiles.

When a new Group is added the Tiles appear horizontal by its default Orientation setting. The following image illustrates the Horizontal orientation of the Tiles.

## c1TileControl1

### Group 1



### Group 2



# TileControl Behavior

The following section details the behavior properties used to control the behavior of the tiles in the TileControl.

---

## TileControl Scrolling

The Tiles in the TileControl are scrollable by default.

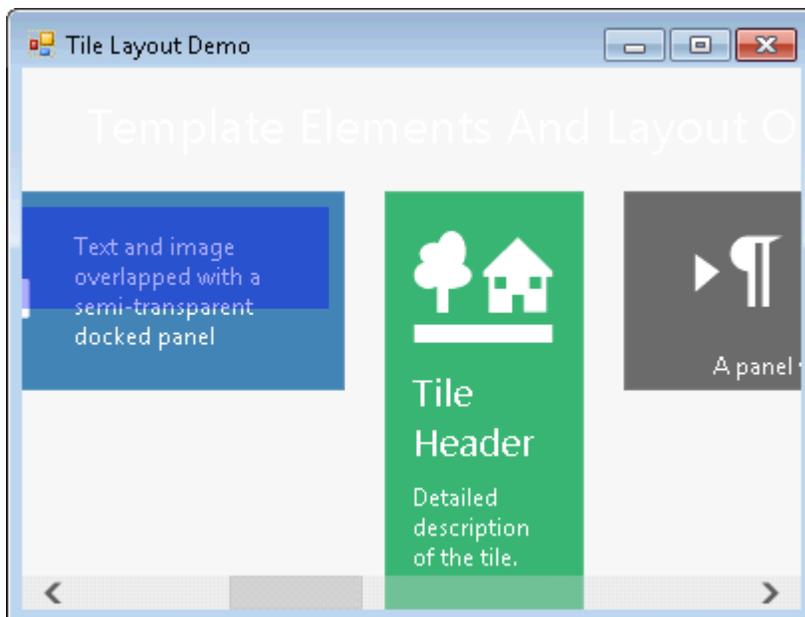
C1TileControl's scrollbar appearance can be determined by the ScrollBarStyle property.

The following table represents the two difference scroll bar styles to choose from when you set its ScrollBarStyle property:

Value	Description
Default	Specifies the default scrollbar for the C1TileControl.
System	Specifies the System scrollbar for the C1TileControl.

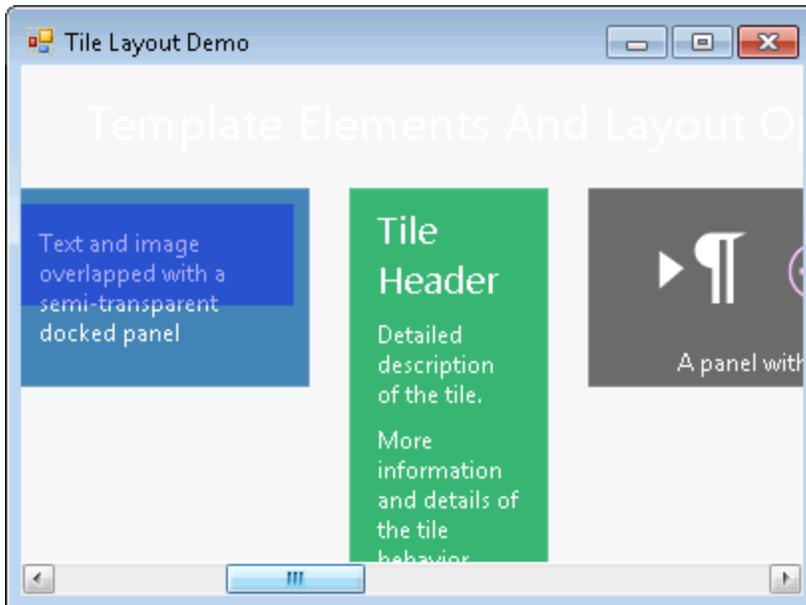
### Default scrollbar

When you hover over the default scrollbar, the bar changes color. The default scrollbar appears like the following:



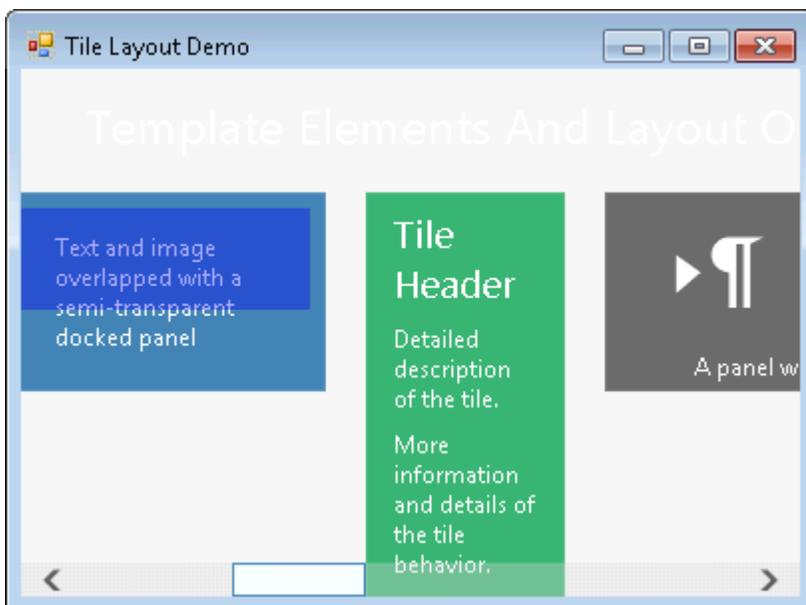
## System scrollbar

When you hover over the system scrollbar, the bar changes color. The system scrollbar appears like the following:



The color of the scrollbar thumb border and the scrollbar thumb interior can be specified using the `SBThumbBorderColor` and `SBThumbInnerColor` properties respectively. The negative or zero offset of the scrollable area can be determined using the `ScrollOffset` property.

The following image illustrates the effects of the `SBThumbBorderColor` and `SBThumbInnerColor` properties. The `SBThumbBorderColor` is set to **SteelBlue** and the `SBThumbInnerColor` is set to **AliceBlue**.



---

## TileControl Navigation

The `PassthroughNavigation` property gets or sets whether the keyboard should navigate to the next row/column after focusing the last tile in the current row/column for vertical and/or horizontal layout.

---

## TileControl Touchscreen Support

The `TileControl` supports panning, tapping, and checking tiles with a swipe gesture using the touch input hardware on a machine with Windows 7 or Windows 8. A visual cue is shown when you reach the end of the pannable area of the `C1TileControl`. To disable the visual cue, set the `AllowPanningFeedback` to **False**.

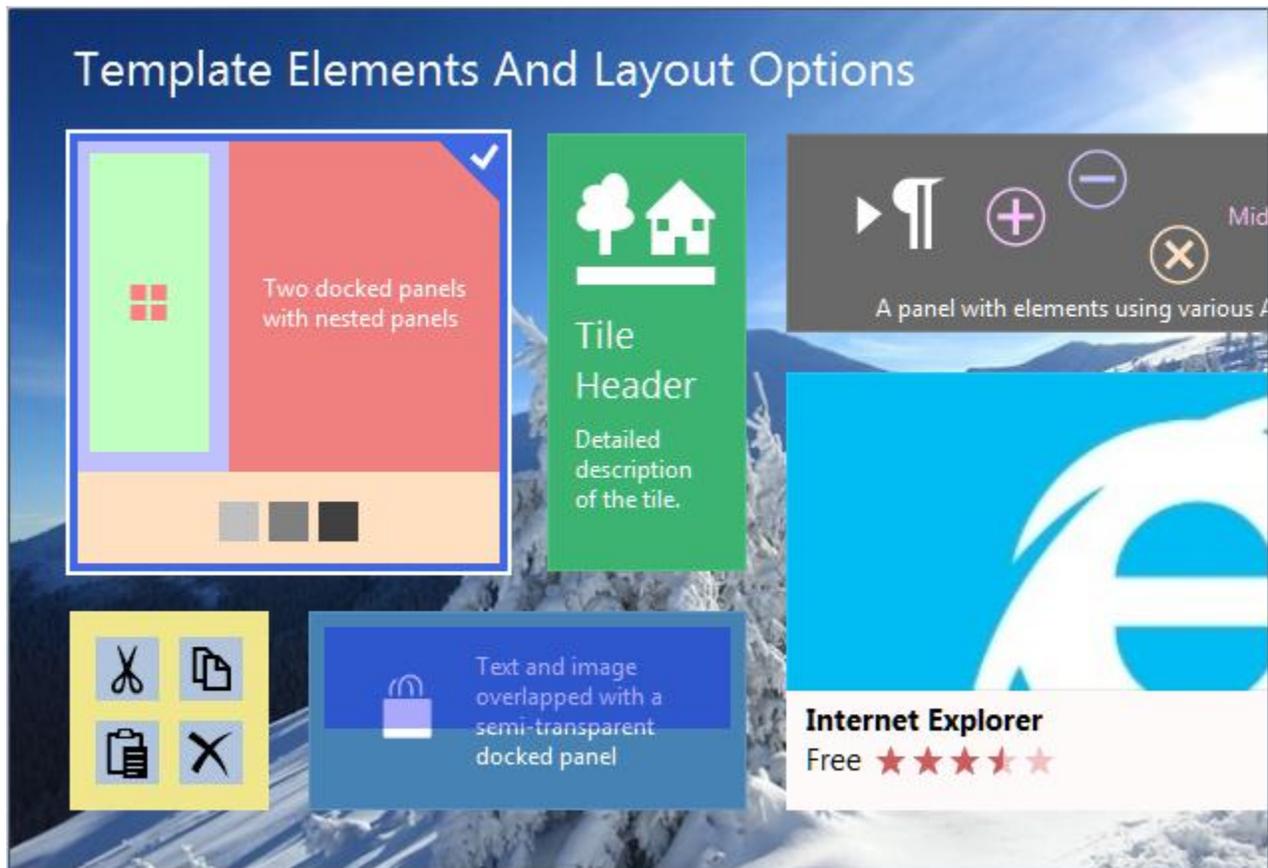
When the `AllowChecking` property is enabled you can check/uncheck tiles using a swipe gesture or right-clicking the mouse.

The `ShowTooltips` property is enabled by default so the tooltips appear when you press and hold a tile with your fingertip on a touchscreen. You can specify the time, in milliseconds, that passes before the tooltips appear using the `ToolTipInitialDelay` property.

# TileControl Templates

Tiles and templates are the most important components of the `C1TileControl`. `Tile` provides the data and the `template` provides the visualization pattern. You can switch the templates for the same tile. For example, the first template may show the tile image and the second may show the detailed tile text. Also, you can apply the same template to multiple tiles.

Templates may consist of three possible elements: panel, text, and image. Panel elements of the type `PanelElement` class may contain child elements including nested panels. The child elements can be added at design time through the designer or programatically through the `Children` property. Text elements of the type `TextElement` can be added at design time through the designer or programatically through `TextElement` class. Image elements of the type `ImageElement` can be added at design time through the designer or programatically through the `ImageElement` class. There are a number of layout settings that give you full freedom in laying out the template elements.



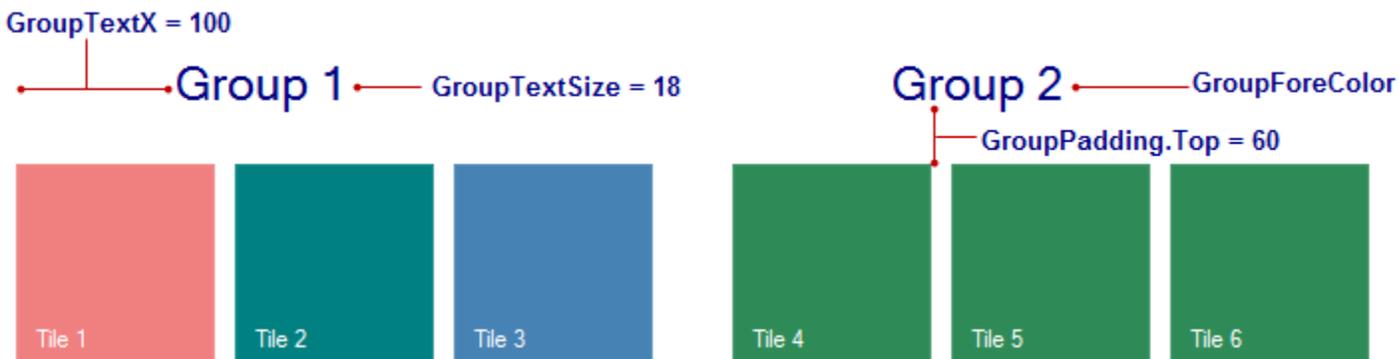
# TileControl Groups

TileControl Groups are of the type Group class. A TileControl can have one or more groups. Each group can have one or more tiles of the type Tile class. Each group may include a caption that represents the name for the group. The group's caption is specified by the Text property. The group caption's font, forecolor, padding, text size, and position can be modified. The group's caption can also be bolded.

A group arranges its items according to the Orientation property. The spaces between the groups can be specified through the GroupSpacing property so a different group can appear detached from the other group.

A Group can be added to the **C1TileControl** at design time through the **C1TileControl.Groups Collection Editor** or programmatically through the Groups property. When a Group is added to the designer it will appear empty on the C1TileControl. You will need to add tiles to the group. The tiles can be added by clicking on the ellipsis button next to the Tiles property and clicking the Add button. Once the tiles are added the default group caption will appear. To see how to add groups to a C1TileControl see [Adding Groups to the C1TileControl](#) (page 38).

The following image illustrates the effects of the Group's appearance properties:



The C1TileControl Groups appearance and layout properties are listed in the table below:

Property	Description
<b>GroupFont</b>	Specifies the font for group captions.
<b>GroupForeColor</b>	Specifies the foreground color of a group caption.
<b>GroupPadding</b>	The interior spacing of a tile group.
<b>GroupTextBold</b>	Indicates whether the group caption font is bold. This property takes precedence over the GroupFont properties.
<b>GroupTextSize</b>	Specifies the font size for the group captions. This property takes precedence over the GroupFont properties. The default text size is 15 pixels.
<b>GroupTextX</b>	Specifies the horizontal offset of a group caption. The default horizontal offset size is 20 pixels.
<b>GroupTextY</b>	Specifies the vertical offset of a group caption. The default vertical offset size is 5 pixels.

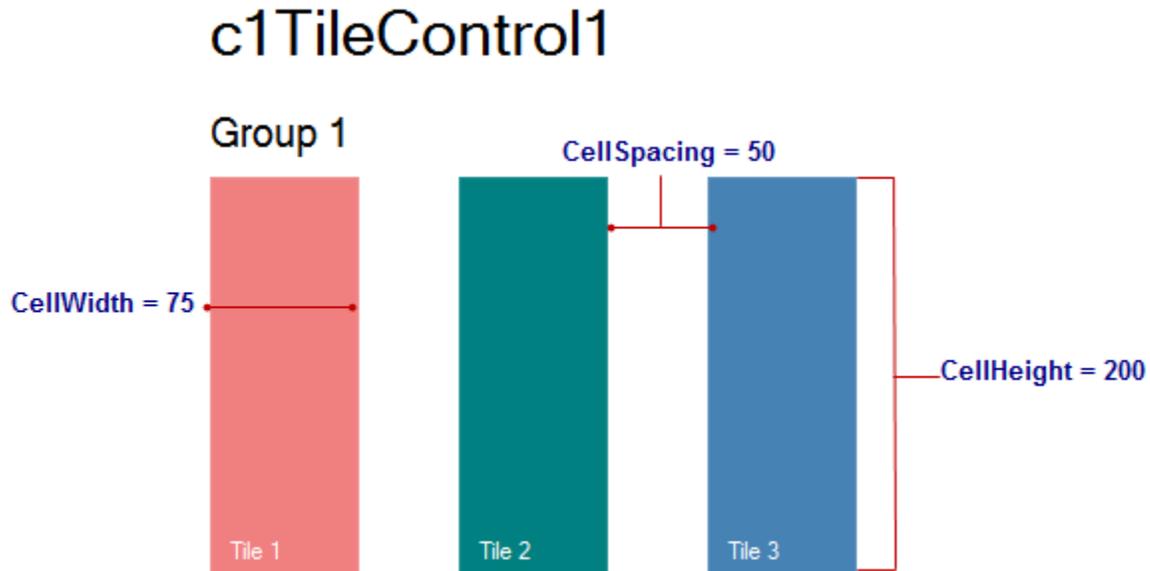
## TileControl Tiles and Elements

The Tile class represents a single Tile which can hold several types of elements such as images (ImageElement), text (TextElement), and panels (PanelElement). The Tiles are one of the most important component of the TileControl; they are responsible for the data. The Tiles in the TileControl in their simplest form appear like button controls. The images, text, and panels can be easily be formatted through the properties.

A Tile can be added to the **C1TileControl** at design time through the **Group.Tiles Collection Editor** or programmatically through the Tiles property. When a Tile is added to the designer it will be added to the specified

group of the C1TileControl. A single group appears by default so you can easily start adding the tiles into the group. The tiles can be added at design time by clicking on the ellipsis button next to the **Tiles** property and clicking the **Add** button. To see how to add tiles to a specific group in the C1TileControl see [Adding Tiles to a Specific Group](#) (page 40).

The following image illustrates the effects of a few of the Tile's appearance properties:



The C1TileControl Groups appearance and layout properties are listed in the table below:

Property	Description
BackColor	Gets or sets the background color for the tile.
BackColor1	Gets or sets the first additional background color for the tile.
BackColor2	Gets or sets the second additional background color for the tile.
BackColor3	Gets or sets the third additional background color for the tile.
BackColor4	Gets or sets the fourth additional background color for the tile.
BackColor5	Gets or sets the fifth additional background color for the tile.
Checked	Gets or sets whether the tile is checked.
ForeColor	Gets or sets the foreground color for the tile.
ForeColor1	Gets or sets the first additional foreground color for the tile.
ForeColor2	Gets or sets the second additional foreground color for the tile.
ForeColor3	Gets or sets the third additional foreground color for the tile.
ForeColor4	Gets or sets the fourth additional foreground color for the tile.
ForeColor5	Gets or sets the fifth additional foreground color for the tile.
Height	Gets or sets the height of the tile, in pixels.

HorizontalSize	Gets or sets the width of the tile, in pixels.
Image	Gets or sets an image displayed on the tile.
Image1	Gets or sets the first additional image that can be displayed on the tile.
Image2	Gets or sets the second additional image that can be displayed on the tile.
Image3	Gets or sets the third additional image that can be displayed on the tile.
Image4	Gets or sets the fourth additional image that can be displayed on the tile.
Image5	Gets or sets the fifth additional image that can be displayed on the tile.
Symbol	Gets or sets a symbol associated with the tile.
Template	Gets or sets the tile template.
Text	Gets or sets the text on the tile.
Text1	Gets or sets the first additional text string for the tile.
Text2	Gets or sets the second additional text string for the tile.
Text3	Gets or sets the third additional text string for the tile.
Text4	Gets or sets the fourth additional text string for the tile.
Text5	Gets or sets the fifth additional text string for the tile.
Text6	Gets or sets the sixth additional text string for the tile.
Text7	Gets or sets the seventh additional text string for the tile.
Text8	Gets or sets the eighth additional text string for the tile.
Text9	Gets or sets the ninth additional text string for the tile.
ToolTipText	Gets or sets the tooltip text for the tile.
VerticalSize	Gets or sets the height of the tile, in cells.
Width	Gets the width of the tile, in pixels.

The TileControl Tiles can include the following elements:

- Images – The images are represented by the class, ImageElement.
- Panels – The panels are represent by the class, PanelElement.
- Text – The text is represent the by class, TextElement.

---

## Image Element

Tiles can display one or several images. An image can be specified in the Tile using its Image, ImageKey, or Symbol properties. Also, it can be specified as a part of the template's ImageElement or stored in one of the CommonImage objects that belong to C1TileControl.

There are a few tricks when working with images. For example, you can create a big image that consists of N images in width and M images in height. If so, the ImageColumns property must be set to N, and ImageRows property must be set to M. Also, the ColumnIndex and RowIndex properties can be used to pick a small image from large matrix.

The ImageList collection should be specified first in order to use the ImageIndex property.

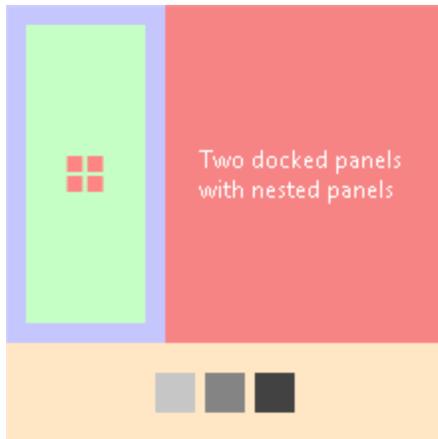
For more information see [Adding Image Elements to a Tile](#) (page 43).

---

## Panel Element

Tiles can display one or more panels. A panel can have nested panels. Panel elements of the type PanelElement class may contain child elements including nested panels. The child elements can be added at design time through the **PanelElement.Children Collection** editor or programatically through the Children property.

The following image displays a Tile with two docked panels with nested panels.



---

## Text Element

Tiles can display one or more text elements. The text elements are represented by the TextElement class. The child elements can be added at design time through the designer or programatically through the Text, Text1 through Text6 properties.

The following image displays a Tile with a few text element and a badge if IntValue is greater than zero.



# TileControl for WinForms Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos which may make use of other development tools included with the ComponentOne Studios.

You can access samples from the **ComponentOne TileControl**. To view samples, click the **Start** button and then click **ComponentOne | Studio for WinForms | TileControl**. The following table provides a short overview of each sample.

Sample	Description
TileImages	The sample shows how to display images on tiles using various techniques.
TileLayout	The sample shows various methods of laying out the template elements.

# TileControl for WinForms Task-Based Help

The task-based help section assumes that you are familiar with programming in the Visual Studio NET environment and have a general understanding of the **ComponentOne TileControl**.

Each topic provides a solution for specific tasks using the **C1TileControl**. By following the steps outlined in each topic, you will be able to create projects using a variety of **C1TileControl** features.

---

## Adding Templates to the C1TileControl

Templates can be created at design time or programmatically. At design time they can be created using the **C1TileControl.Templates Collection Editor**. Templates can also be created programmatically through the **TemplateCollection** class. The template can be inserted into the **TemplateCollection** at the specified index using the **InsertItem** method.

### Design-Time

To add a template at design time, complete the following:

1. Select the **C1TileControl** and click on its smart tag to open the **C1TileControl Tasks** menu.
2. Select **Edit Tiles**.  
The **C1TileControl.Templates Collection Editor** editor appears.
3. Click **Add** to add a template to the Members list.

---

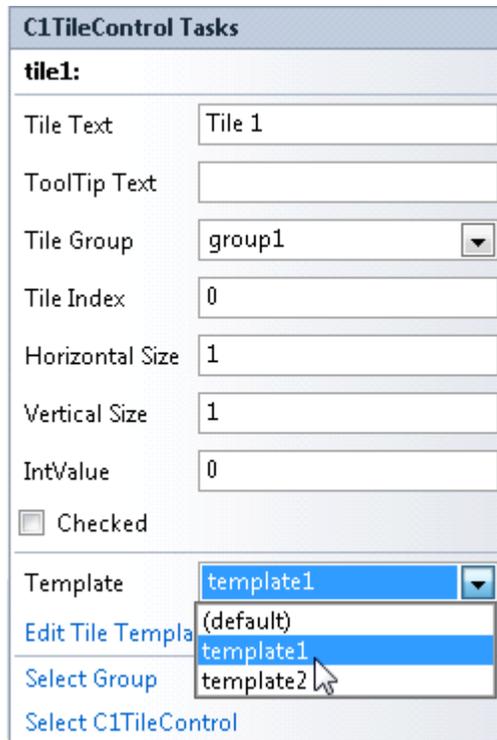
## Assigning a Template to a Specified Tile

Once a template is created at design time through the **C1TileControl.Templates Collection Editor** or programmatically through the **TemplateCollection** class it then can be assigned to a specified tile.

### Design-Time

To assign a template to a specified tile, complete the following:

1. Select the tile you wish to add the template to. The **C1TileControl Tasks** menu appears for the selected tile.
2. Select the template from the **Template** dropdown listbox you wish to assign to the specified tile. For example, **template1** is being assigned to **tile1**.



---

## Adding Elements to a Template

Template elements can be created at design time or programmatically. At design time they can be created using the **C1TileControl.Templates Collection Editor**. Template elements can also be created programmatically through the `TemplateCollection` class. The template can be inserted into the `TemplateCollection` at the specified index using the `InsertItem` method.

### Design Time

To add elements to the template at design time, complete the following:

1. Select the **C1TileControl** and click on its smart tag to open the **C1TileControl Tasks** menu.
2. Select **Edit Tiles**.  
The **C1TileControl.Templates Collection Editor** editor appears.
3. Click **Add** to add a template to the **Members** list.
4. Select the ellipsis button next to the **Elements** property. The **Template.Elements Collection Editor** appears.
5. Click on the dropdown arrow and select the **PanelElement** twice. Two panels are added to the template.

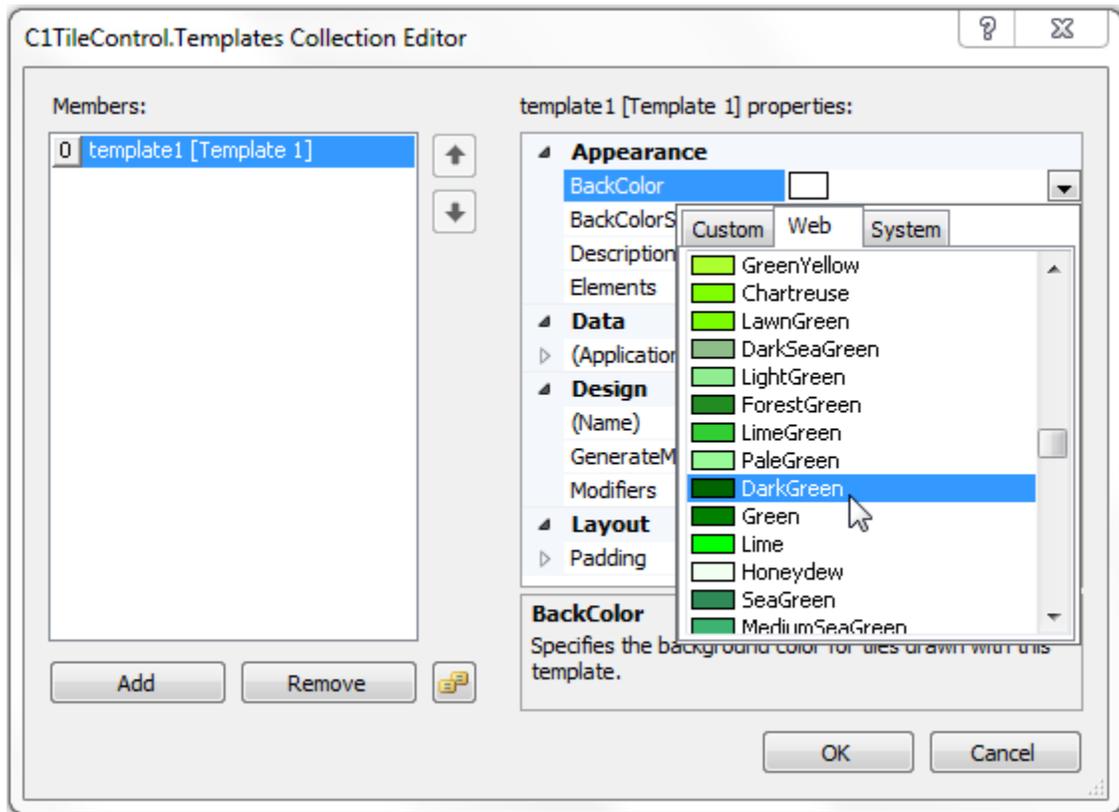
---

## Changing the BackColor of the Template

To change the `BackColor` of the Template at design time, complete the following:

1. Select the **C1TileControl** and click on its smart tag to open the **C1TileControl Tasks** menu.
2. Select **Edit Tiles**.  
The **C1TileControl.Templates Collection Editor** editor appears.
3. Click **Add** to add a template to the Members list.

4. Set the **BackColor** to **DarkGreen**.



5. Set the **BackColorSelector** to **Unbound**.

This will assign the DarkGreen backcolor to this template rather than the default backcolor. Once the Template is assigned to the specified Tile the new BackColor will appear.

6. Select the first Tile and set its Template property to Template1. The template with the new backcolor will be updated for the first Tile.

✔ This topic illustrates the following:

The new backcolor for the template appears in the first Tile.



---

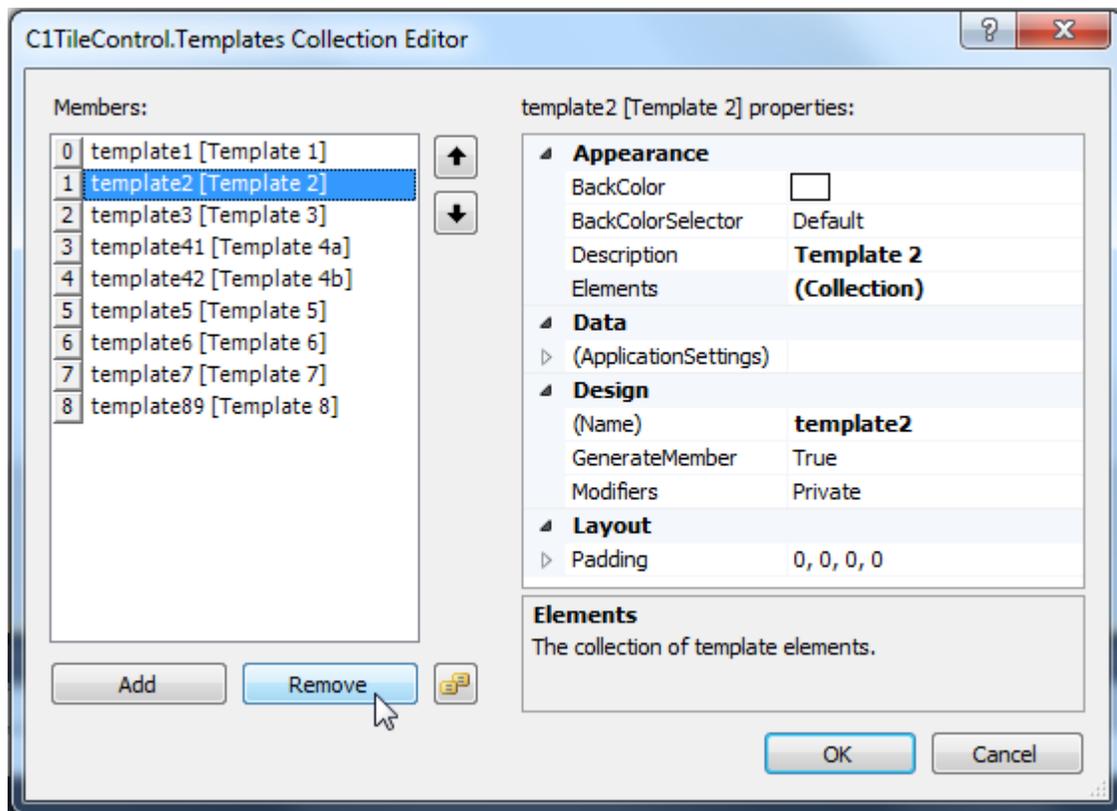
## Removing Specific Templates

The TileControl's template can be removed programmatically or at design time.

### Design-Time

To remove a specific template from the TileControl at design time, complete the following:

1. Right-click on the C1TileControl and select **Edit Templates**. The **C1TileControl.Templates Collection Editor** appears.
2. Select the Template from the Members: list and click **Remove**.



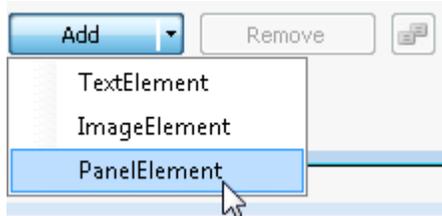
---

## Alternating the Text View by a Timer

To alternate the text view by a timer, complete the following:

### Add the First Template

1. Right-click the Tile control and select **Edit Templates**. The **C1TileControl.Templates Collection Editor** appears.
2. Click **Add** twice to add two templates to the C1TileControl.
3. Select **template1** and click on the ellipsis button next to the **Elements** collection. The **Template.Elements** collection editor appears.
4. Select the **PanelElement** from the **Add** dropdown listbox.



5. Set the **PanelElement**'s properties to the following:
  - Alignment property to **TopLeft**.
  - **ChildSpacing** property to **0**. This will decrease the default spacing between the child elements from 5 pixels to 0 pixels.
  - Orientation property to **Vertical**.
6. Click on the ellipsis button next to the **Children** property.
7. Add two **TextElements** to the **PanelElement**.
8. Select the second text element, **[1] TextElement** and set its **TextSelector** property to **Text1**. This will assign the value of the **Text1** property to this template.
9. Click **OK** to save and close the **PanelElement.Children Collection Editor** and click **OK** to save and close the **Template.Elements Collection Editor**.

#### Add the Second Template

10. Select **template2** in the **C1TileControl.Templates Collection Editor**.
11. Click on the Ellipsis button next to the **Elements Collection**. The **Template.Elements Collection Editor** appears.
12. Click the dropdown arrow next to the **Add** button to add a **PanelElement**.
13. Set the **[0]Panel Element** properties to the following:
  - Alignment property to **TopLeft**.
  - **ChildSpacing** property to **0**.
  - Orientation property to **Vertical**.
14. Click the ellipsis button next to the **Children (Collection)** property and add two **TextElements**.
15. Select the first text element, **[0] TextElement** and set its **TextSelector** property to **Text1**.
16. Select the second text element, **[1] TextElement** and set its **TextSelector** property to **Text2**.
17. Click **OK** to save and close the **PanelElement.Children Collection Editor** and click **OK** to save and close the **Template.Elements Collection Editor**.
18. Right-click on **Tile1** and select **Edit Groups**. The **C1TileControl.Groups Collection Editor** appears.
19. Click on the ellipsis button next to the **Tiles Collection**.
20. Select **tile1** and set its properties to the following:
  - **Template** property to **template1**. The settings for **template1** are applied to **Tile1**.
  - **Text1** property to **Detailed description of the Tile**.
  - **Text2** property to **More information and details of the Tile behavior**.
21. Click **OK** to save and close the **Group.Tiles Collection Editor**.

Tile1 should appear like the following:



### Add a Timer to alternate the template views for Tile1.

22. Double-click on the WindowsForm **Timer** control to add it to your component tray.
23. Set the timer1 **Interval** property to **3000** and the **Enabled** property **True**.
24. Right-click on the TileControl and select **View Code**.
25. Add the following code to your project to create an animation that alternates the text views of each template:

- Visual Basic

```
Public Partial Class Form1
    Inherits Form
    Private _tile1Flipped As Boolean

    Public Sub New()
        InitializeComponent()
    End Sub

    Private Sub timer1_Tick(sender As Object, e As EventArgs)
        Dim a As Boolean = _tile1Flipped
        tile1.Template = If(a, template1, template2)
        _tile1Flipped = Not a
    End Sub
End Class
```

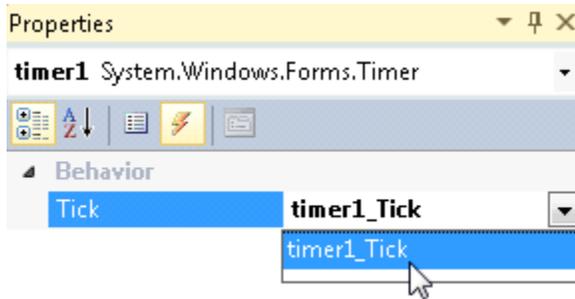
- C#

```
public partial class Form1 : Form
{
    bool _tile1Flipped;

    public Form1()
    {
        InitializeComponent();
    }

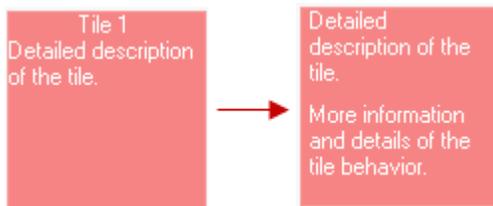
    private void timer1_Tick(object sender, EventArgs e)
    {
        bool a = _tile1Flipped;
        tile1.Template = a ? template1 : template2;
        _tile1Flipped = !a;
    }
}
```

26. In **Design view**, assign the **timer\_Tick** event handler to **timer1**.



**✔ This topic illustrates the following:**

The Tile alternates templates based upon a timer. The first template is displayed for a few seconds and then the second template for the tile appears in place of the first.




---

## Saving and Loading TileControl as an XML File

This topic shows how to save the **C1TileControl** as an XML file and how to load an existing **C1TileControl** from an xml file.

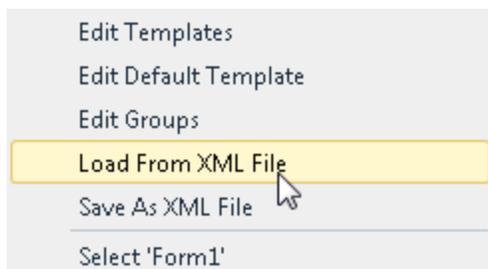
### Loading TileControl From an XML File

This task shows how to load the C1TileControl as an XML File at run time and in code.

#### Load C1TileControl as an XML file at run time

To load the C1TileControl as an XML file at run time, complete the following:

1. Right-click the C1TileControl and select **Load From XML File** item from the context menu.



The **Load From Xml File** dialog box appears.

2. Browse to the location you wish to load the xml file.
3. Click Open in the **Load From Xml File** dialog box.

## Load C1TileControl from XML file in code

To load template1 as an XML file in code, complete the following:

- **C#**

```
private void btnLoadXml_Click(object sender, EventArgs e)
{
    using (OpenFileDialog dlg = new OpenFileDialog())
    {
        dlg.DefaultExt = ".xml";
        dlg.Filter = "XML files|*.xml|All files|*.*";
        dlg.Title = "Load From Xml File";
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            try
            {
                Tilecontrol.LoadXml(dlg.FileName);
            }
            catch
            {
                MessageBox.Show("Bad tilecontrol XML.", dlg.Title);
            }
        }
    }
}
```
- **Visual Basic**

```
Private Sub btnLoadXml_Click(sender As Object, e As EventArgs)
    Using dlg As New OpenFileDialog()
        dlg.DefaultExt = ".xml"
        dlg.Filter = "XML files|*.xml|All files|*.*"
        dlg.Title = "Load From Xml File"
        If dlg.ShowDialog() = DialogResult.OK Then
            Try
                template1.LoadXml(dlg.FileName)
            Catch
                MessageBox.Show("Bad tilecontrol XML.",
                    dlg.Title)
            End Try
        End If
    End Using
End Sub
```

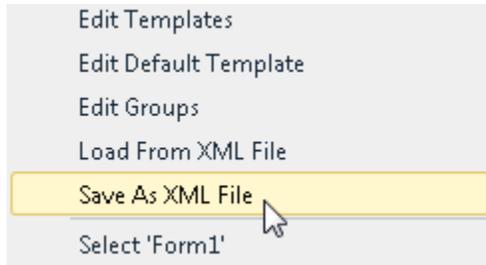
## Saving TileControl as an XML File

This task shows how to save the C1TileControl as an XML File at design time and in code.

### Save C1TileControl as an XML file at design time

To save the C1TileControl as an XML file at design time, complete the following:

1. Right-click the C1TileControl and select **Save as XML File** item from the context menu.



The **Save As Xml File** dialog box appears.

2. Browse to the location you wish to save the .xml file.
3. Click **Save** in the **Save As Xml File** dialog box.

### Save C1TileControl from XML file in code

To save the C1TileControl as an XML file in code, complete the following:

- **C#**

```
private void menuItemSaveXml_Click(object sender, EventArgs e)
{
    using (SaveFileDialog dlg = new SaveFileDialog())
    {
        dlg.DefaultExt = ".xml";
        dlg.FileName = "tilecontrol";
        dlg.Filter = "XML files|*.xml|All files|*.*";
        dlg.Title = "Save As Xml File";
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            TileControl.SaveXml(dlg.FileName);
        }
    }
}
```
- **Visual Basic**

```
Private Sub menuItemSaveXml_Click(sender As Object, e As EventArgs)
    Using dlg As New SaveFileDialog()
        dlg.DefaultExt = ".xml"
        dlg.FileName = "tilecontrol"
        dlg.Filter = "XML files|*.xml|All files|*.*"
        dlg.Title = "Save As Xml File"
        If dlg.ShowDialog() = DialogResult.OK Then
            TileControl.SaveXml(dlg.FileName)
        End If
    End Using
End Sub
```

---

## Setting Text for the TileControl and Group

The TileControl's text and its font size and color as well as the group's text, font size, and color can be applied to the TileControl and group programmatically or at design time.

To modify the TileControl and Group text, complete the following:

1. Right-click the **C1TileControl** and select **Properties**. The C1TileControl properties pane appears.

2. Enter the text inside the **Text** textbox that you wish to appear on the C1TileControl, for example **Layout Options**.
3. Set the **ForeColor** property to **Navy**. Note that this sets the group's forecolor as well. To change the group's forecolor it can be specified in the GroupForeColor property.
4. Set the **TextSize** property to **16**. This overrides the **Font.Size** property.
5. Expand the **Groups** node in the C1TileControl's properties pane.
6. Set the **GroupFont** to **True** so the group's font appears bold.
7. Set the **GroupForeColor** to **DarkGreen**.

✔ This topic illustrates the following:

The C1TileControl and Group's Text is modified.



---

## Adding Groups to the C1TileControl

Groups can be created at design time or programmatically. At design time they can be created using the **C1TileControl.Groups Collection Editor**. Groups can also be created programmatically through the Group class. The group can be inserted into the GroupCollection at the specified index using the InsertItem method.

### Design-Time

To add a template at design time, complete the following:

1. Right-click the **C1TileControl** and select **Edit Groups** to open the **C1TileControl.Groups Collection Editor**.
2. Click the **Add** button to add a new group to the C1TileControl. The Group's text will appear empty and there will be no tiles until you add the tiles to the group.

---

## Removing Groups from the C1TileControl

Groups can be removed at design time or programmatically. At design time they can be removed using the **C1TileControl.Groups Collection Editor**. Groups can also be removed programmatically through the GroupCollection class. The group can be removed from the GroupCollection at the specified index using the RemoveItem method.

### Design-Time

To remove a group from the C1TileControl complete the following:

1. Add the TileControl to the windows Form.

2. Right-click the **C1TileControl** and select **Edit Groups** to open the **C1TileControl.Groups Collection Editor**.
3. Select the Group from the Members list that you want to remove and click the **Remove** button to remove a group from the C1TileControl.

---

## Modifying the Group's Font Properties

To modify the Group's font properties complete the following:

1. Right-click on the **C1TileControl** and select **Properties**.
2. Locate the **GroupFont** property under the **Groups** node and click on the ellipsis button.
3. Set the Font to Calisto MT, FontStyle to Bold, and Size to 20. Note that the GroupTextBold and GroupTextSize will override these settings so if you set the FontStyle to Bold, but the GroupTextBold is set to False then the Group will inherit the setting from the Group.TextBold property.
4. Click **OK** to close and save the Font dialog box.
5. Set the GroupTextSize to 16. Note that this property takes precedence over the Font Size property applied in the Font dialog box.

✔ This topic illustrates the following:



---

## Setting the TileControl's BackColor

The TileControl's BackColor can be applied to the TileControl programatically or at design time.

To set the TileControl's BackColor using the Properties window, complete the following:

1. Right-click on the **C1TileControl** and select **Properties**.
2. Set the **BackColor** property to **LightYellow**.

✔ This topic illustrates the following:

# c1TileControl1

## Group 1



---

## Adding Tiles to a Specific Group

Tiles can be created at design time or programmatically. At design time they can be created using the **C1TileControl.Groups Collection Editor** and **Group.Tiles Collection Editor**. Tiles can also be created programmatically through the **Tile** class. The tiles can be inserted into the **TilesCollection** at the specified index using the **InsertItem** method. You can then add the tiles to the appropriate group using the **Group.Tiles** property.

### Design-Time

To add a template at design time, complete the following:

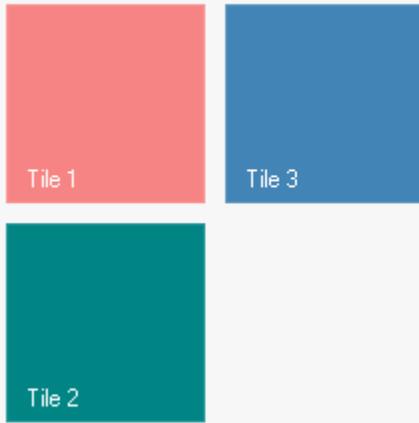
1. Right-click the **C1TileControl** and select **Edit Groups** to open the **C1TileControl.Groups Collection Editor**.
2. Select the Group from the members list where you want to add the tiles, for example **Group2**.
3. Click on the ellipsis button next to **Tiles** collection property.
4. Click add three times to add three **Tiles** to the second group, **group2**.
5. Click **OK** to save and close the **Group.Tiles Collection Editor** and click **OK** to save and close the **C1TileControl.Groups Collection Editor**.

✔ This topic illustrates the following:

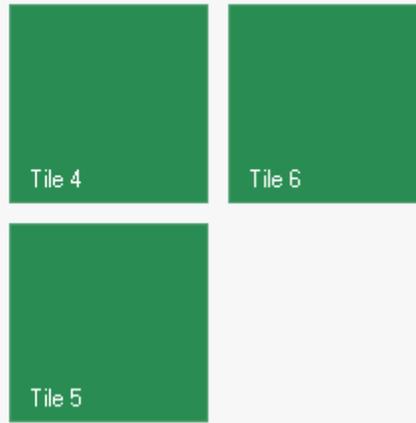
The three Tiles appear horizontally by default and they all have the same default dark green color:

# c1TileControl1

## Group 1



## Group 2



---

## Creating a CheckMark for the Tile

A checkmark can be displayed in the upper right corner of the tile by setting the `Checked` property to `True`.

### Design-Time

1. To apply a checkmark to a specific tile, select **tile2** and check the checkbox next to the `Checked` property.

The screenshot shows the Visual Studio Properties window for a `C1TileControl` control. The title bar reads "C1TileControl Tasks". The selected instance is "tile2". The properties are as follows:

Tile Text	Tile 2
ToolTip Text	
Tile Group	group1
Tile Index	1
Horizontal Size	1
Vertical Size	1
IntValue	0
<input checked="" type="checkbox"/> Checked	
Template	(default)

Below the properties, there are three links: [Edit Tile Template](#), [Select Group](#), and [Select C1TileControl](#).

2. Right-click the **C1TileControl** and select **Properties**.
3. Under the **Appearance** node set the CheckMarkColor property to **Silver**.

✔ **This topic illustrates the following:**

A checkmark appears in the upper right corner of tile2.



## Increasing the Size of a Specific Tile

A Tile's size can be increased using the `HorizontalSize` and `VerticalSize` properties.

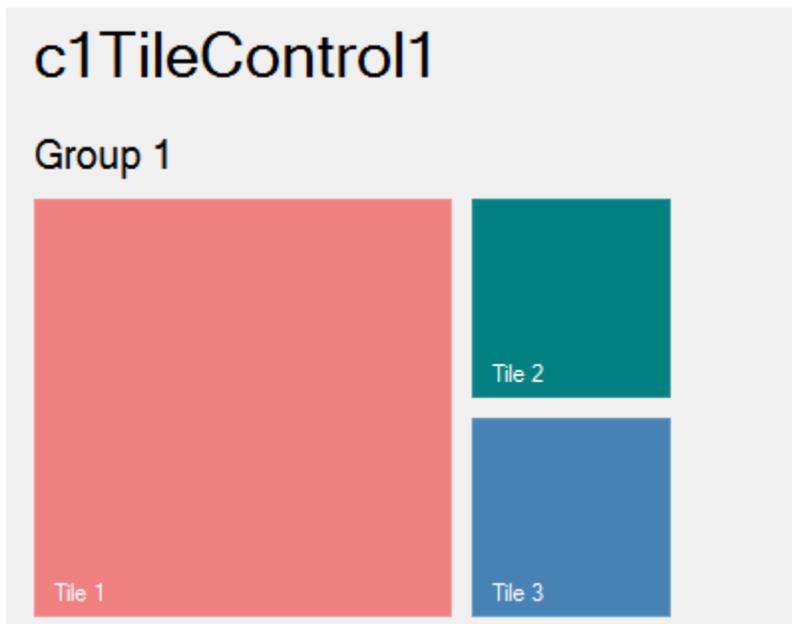
### Design-Time

To increase the Tile's size at design-time, complete the following:

1. Select the Group and click on **Edit Tiles** from the `C1TileControl` – Group Tasks menu.
2. In the **Group.Tiles Collection Editor** select `tile1[Tile 1]` and set the **HorizontalSize** to **2** and the **VerticalSize** to **2**.
3. Click **OK** to save and close the **Group.Tiles Collection Editor**.

✔ **This topic illustrates the following:**

Tile 1's size is increased so it appears twice as large as the other two tiles.



---

## Adding Image Elements to a Tile

The following tasks show different methods on how to add images to the tiles.

---

## Adding a Symbol to a Tile

To add a symbol to a tile, complete the following:

1. Right-click on the **C1TileControl** and select **Edit Templates**. The **C1TileControl.Templates Collection Editor** appears.
2. Click **Add** to add a template.
3. Click on the ellipsis button next to the **Elements Collection** property. The **Template.Elements Collection Editor** appears.
4. Select **ImageElement** from the **Add** dropdown listbox.
5. Select **Library** from the **Symbol** dropdown listbox.
6. Select **Symbol** from the dropdown listbox of the **ImageSelector** property. This binds the value of the Symbol property to the specified Tile.
7. Click **OK** to save and close the **Template.Elements Collection Editor**.
8. Right-click the **C1TileControl** and select **Edit Groups**.
9. Click on the ellipsis button next to Tiles. The **Group.Tiles Collection Editor** appears.
10. Select the first Tile from the members list.
11. Set the **Symbol** property to **Home** and **Template** property to **template1**.

✔ **This topic illustrates the following:**

The Home symbol is applied to the first Tile.



## Drawing an Image at Runtime

Images can be drawn at runtime using the Paint event, like the following:

1. Right-click on the **C1TileControl** and select **Edit Templates**.
2. Click **Add** to add a template to the **C1TileControl**.
3. Add the following code to your project to draw an image at runtime:
  - Visual Basic

```
Private Sub template1_Paint(sender As Object, e As C1.Win.C1Tile.TemplatePaintEventArgs)
    Dim g As Graphics = e.Graphics
    g.SmoothingMode =
        System.Drawing.Drawing2D.SmoothingMode.HighQuality
    Dim rect As Rectangle = e.ClipRectangle
```

```

rect.X += (rect.Width - 28) \ 2
rect.Y += (rect.Height - 28) \ 2
rect.Width = 28
rect.Height = 28
Dim brush As Brush = New SolidBrush(e.Tile.GetBackColor())
Dim pen As New Pen(e.Tile.GetForeColor())
Select Case e.Tile.IntValue
    Case 1
        g.FillPie(brush, rect, 50F, 270F)
        g.DrawPie(pen, rect, 50F, 270F)
        Exit Select
    Case 2
        g.FillRectangle(brush, rect)
        g.DrawRectangle(pen, rect)
        Exit Select
    Case Else
        g.FillEllipse(brush, rect)
        g.DrawEllipse(pen, rect)
        Exit Select
End Select
brush.Dispose()
pen.Dispose()
End Sub

```

- **C#**

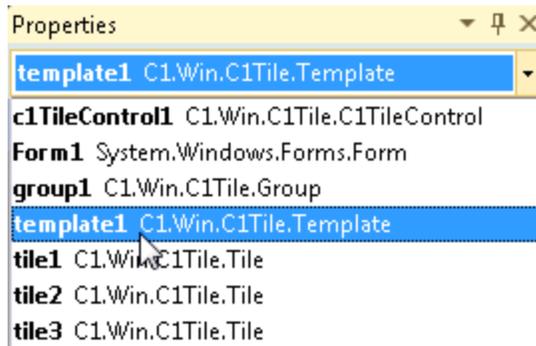
```

private void template8_Paint(object sender,
C1.Win.C1Tile.TemplatePaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.SmoothingMode =
System.Drawing.Drawing2D.SmoothingMode.HighQuality;
    Rectangle rect = e.ClipRectangle;
    rect.X += (rect.Width - 28) / 2;
    rect.Y += (rect.Height - 28) / 2;
    rect.Width = 28;
    rect.Height = 28;
    Brush brush = new SolidBrush(e.Tile.GetBackColor());
    Pen pen = new Pen(e.Tile.GetForeColor());
    switch (e.Tile.IntValue)
    {
        case 1:
            g.FillPie(brush, rect, 50f, 270f);
            g.DrawPie(pen, rect, 50f, 270f);
            break;
        case 2:
            g.FillRectangle(brush, rect);
            g.DrawRectangle(pen, rect);
            break;
        default:
            g.FillEllipse(brush, rect);
            g.DrawEllipse(pen, rect);
            break;
    }
    brush.Dispose();
    pen.Dispose();
}

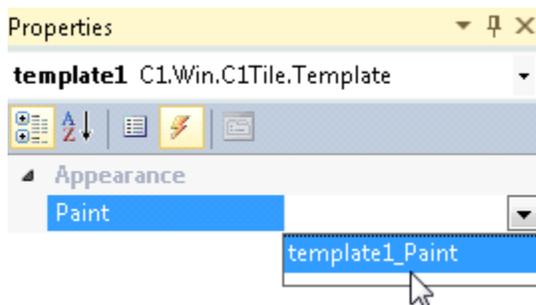
```

4. In design view, right-click on the **C1TileControl** and select **Properties**.

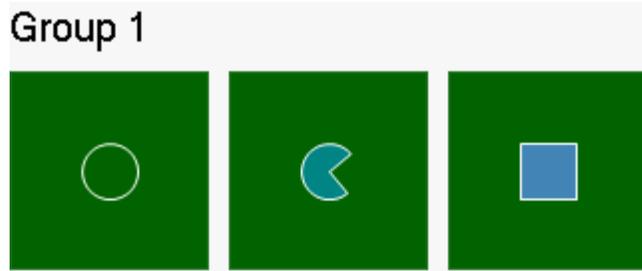
5. Select **template1** from the Windows dropdown listbox.



6. Click on the events button and set the Paint event to template1\_Paint.



7. Select tile1 so its **C1TileControl Tasks** menu appears and set its properties to the following:
  - **Horizontal Size** to 1.
  - **Vertical Size** to 1.
  - **IntValue** to 0. This will apply the first drawing to the first tile.
  - **Template** to **template1**.
8. Select tile2 so its **C1TileControl Tasks** menu appears and set its properties to the following:
  - **Horizontal Size** to 1.
  - **Vertical Size** to 1.
  - **IntValue** to 1. This will apply the second drawing to the second tile.
  - **Template** to **template1**.
9. Select tile2 so its **C1TileControl Tasks** menu appears and set its properties to the following:
  - **Horizontal Size** to 1.
  - **Vertical Size** to 1.
  - **IntValue** to 2. This will apply the third drawing to the third tile.
  - **Template** to **template1**.
10. Run your project and observe the drawings made on the tiles.



# Index

No index entries found.