ComponentOne

# HyperPanel for WPF

**ComponentOne, a division of GrapeCity**

201 South Highland Avenue, Third Floor

Pittsburgh, PA 15206 • USA

**Internet:** info@ComponentOne.com

**Web site:** http://www.componentone.com

**Sales**

E-mail: sales@componentone.com

Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

### Trademarks

The ComponentOne product name is a trademark and ComponentOne is a registered trademark of GrapeCity, Inc. All other trademarks used herein are the properties of their respective owners.

### Warranty

ComponentOne warrants that the original CD (or diskettes) are free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective CD (or disk) to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for a defective CD (or disk) by sending it and a check for $25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original CD (or disks) set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. We are not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

### Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only.

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

# Table of Contents

# ComponentOne HyperPanel for WPF Overview

**ComponentOne HyperPanel™ for WPF** is a **StackPanel** that provides an automatic zoom effect for items near the mouse. You can place any elements in the panel to achieve carousel-like effects and display a large number of elements in a relatively small container, without using scrollbars. The resulting effect is similar to the system toolbar (dock) seen in Mac OS X.

For a list of the latest features added to **ComponentOne Studio for WPF**, visit What's New in Studio for WPF.

**Getting Started**

Get started with the following topics:

- Key Features (page 1)

- Quick Start (page 3)

- Task-Based Help (page 17)

## Help with ComponentOne Studio for WPF

### Getting Started

For information on installing **ComponentOne Studio for WPF**, licensing, technical support, namespaces and creating a project with the control, please visit Getting Started with Studio for WPF.

### What's New

For a list of the latest features added to **ComponentOne Studio for WPF**, visit What's New in Studio for WPF.

# Key Features

**ComponentOne HyperPanel for WPF** allows you to create customized, rich applications. Make the most of **HyperPanel for WPF** by taking advantage of the following key features:

- **Add Dynamic Zooming**

  **ComponentOne HyperPanel™ for WPF** allows you to display a large number of items in a small space. Items far from the mouse are shrunk and don't take up much space.

- **Control the Zoom Effect**

  Determine how much zooming should be applied when the mouse moves over the panel. Use the Distribution property to control the strength of the zoom effect.

- **Limit the Zoom Effect**

  Use the **MinElementScale** property to prevent items from getting too small.

- **Control Item Opacity**

  Use the **ApplyOpacity** property to make elements near the mouse opaque. Items far from the mouse become more transparent, conveying an idea of distance.

# HyperPanel for WPF Quick Start

The following quick start guide is intended to get you up and running with **HyperPanel for WPF**. In this quick start you'll start in Visual Studio and create a new project, add a **HyperPanel for WPF** panel to your application, add controls within C1HyperPanel, and customize the appearance and behavior of the panel.

You will create a simple application using a C1HyperPanel panel that contains several items. You'll then customize the C1HyperPanel panel's appearance and behavior settings to explore the possibilities of using **HyperPanel for WPF**.

## Step 1 of 3: Setting up the Application

In this step you'll begin in Visual Studio to create a WPF application using **HyperPanel for WPF**. When you add a C1HyperPanel panel to your application and items to the panel you'll have a completely interactive way of viewing and selecting items.

To set up your project, complete the following steps:

1. Create a new WPF project in Visual Studio. For more information about creating a WPF project, see Creating a .NET Project in Visual Studio.

2. Add the follow XAML markup just under the `<Window>` tag:

   - XAML to Add

```
<Window.Resources>
    <ResourceDictionary>
        <Style x:Key="letterStyle" TargetType="Border">
            <Setter Property="Height" Value="60" />
            <Setter Property="Width" Value="60" />
            <Setter Property="Margin" Value="2" />
            <Setter Property="CornerRadius" Value="3" />
            <Setter Property="BorderThickness" Value="4" />
            <Setter Property="BorderBrush" >
                <Setter.Value>
                    <LinearGradientBrush EndPoint="0.5,1"
StartPoint="0.5,0">
                        <GradientStop Color="#FFBDBDBD"/>
                        <GradientStop Color="#FFDADADA" Offset="0.5"/>
                        <GradientStop Color="#FFBDBDBD" Offset="1"/>
                    </LinearGradientBrush>
                </Setter.Value>
            </Setter>
        </Style>
        <DataTemplate x:Key="letterTemplate" >
            <Border Background="Blue" Style="{StaticResource
letterStyle}" >
                <TextBlock Text="{Binding}"
HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32"
TextAlignment="Center" TextWrapping="Wrap" >
                   <TextBlock.Foreground>
                         <LinearGradientBrush EndPoint="0.5,1"
StartPoint="0.5,0">
                               <GradientStop Color="BlueViolet"/>
                               <GradientStop Color="Thistle"
Offset="0.4"/>
```

```
                               <GradientStop Color="AntiqueWhite"
Offset="0.6"/>
                               <GradientStop Color="White" Offset="1"/>
                         </LinearGradientBrush>
                   </TextBlock.Foreground>
                </TextBlock>
            </Border>
        </DataTemplate>
    </ResourceDictionary>
</Window.Resources>
```

This will add a style to format the content we will add to the C1HyperPanel panel.

3. Navigate to the Toolbox and double-click the **C1HyperPanel** icon to add the panel to Window1.
4. Resize window and the C1HyperPanel panel to fill the window. Later you'll add items to the C1HyperPanel panel.

You've successfully created a WPF application and added a C1HyperPanel panel to the application. In the next step you'll add controls to and customize C1HyperPanel.

## Step 2 of 3: Adding Content to the Panel

In the previous step you created a new WPF project and added a C1HyperPanel panel to the application. In this step you'll continue by adding controls to the panel.

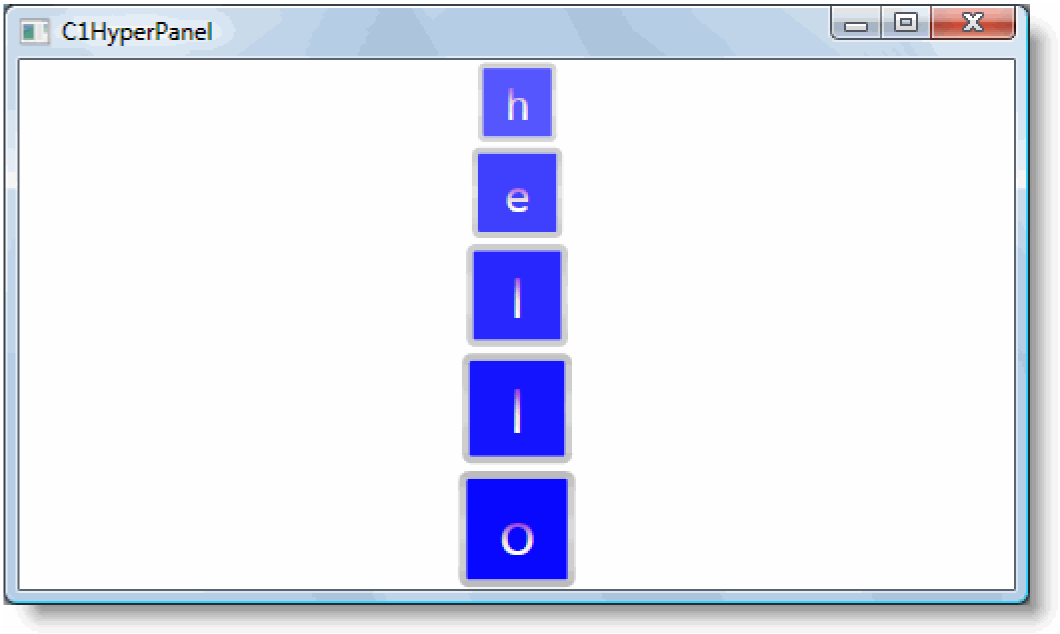
Complete the following steps:

1. Switch to XAML view and navigate to just under the `<Window>` tag.
2. In the Design pane, click once inside the C1HyperPanel panel; now when you add items they will be added inside the panel.
3. Navigate to the Toolbox and double-click the **ContentControl** icon to add the control to C1HyperPanel. That's all you have to do to add an item to a C1HyperPanel panel – you can items as you would normally to other panels include the **Grid** and **Canvas**.
4. Switch to XAML view and update the **ContentControl**'s markup so that it appears similar to the following:
   ```
   <ContentControl Content="h" ContentTemplate="{StaticResource
   letterTemplate}"/>
   ```
5. Enter the following XAML under the **ContentControl**'s markup and within the C1HyperPanel to add additional controls to the panel:

   - XAML to Add
     ```
     <ContentControl Content="e" ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content="l" ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content="l" ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content="o" ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content=" " ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content="w" ContentTemplate="{StaticResource
     letterTemplate}"/>
     <ContentControl Content="o" ContentTemplate="{StaticResource
     letterTemplate}"/>
     ```

```
<ContentControl Content="r" ContentTemplate="{StaticResource
letterTemplate}"/>
<ContentControl Content="l" ContentTemplate="{StaticResource
letterTemplate}"/>
<ContentControl Content="d" ContentTemplate="{StaticResource
letterTemplate}"/>
<ContentControl Content="!" ContentTemplate="{StaticResource
letterTemplate}"/>
```

6. From the **Project** menu, select **Test Solution** to view how your application will appear at run time. It will appear similar to the following:



Notice that you can move your mouse over the items in the panel and they will appear to move.

You've successfully created a WPF application and added C1HyperPanel and controls to the application. In the next step you'll customize C1HyperPanel.

## Step 3 of 3: Customizing the Application

In the previous steps you created a new WPF project and added a C1HyperPanel panel and several **ContentControl**s to the application. In this step you'll continue by setting properties to customize those controls.

Complete the following steps:

1. In Design view, click once on the **C1HyperPanel** control to select it.
2. Navigate to the Properties window and set the Orientation property to **Horizontal**.

   The Orientation property determines if items in the panel are displayed horizontally or vertically. By default Orientation is set to **Vertical** and the panel displays content vertically; setting Orientation property to **Horizontal** will display content horizontally.
3. In the Properties window, set the Distribution property to "0.2".

   The Distribution property consists of a number between .1 and 1.0 and controls how elements are zoomed near the center of the panel. The smaller the value, the more visible the zoom effect. By default, the property is set to "0.5". Setting Distribution to ".02" will cause elements in the center to appear even more zoomed in then elements at the edge of the panel.

4. Set the MinElementScale property to "0.5".

   The MinElementScale property consists of a number between 0 and 1.0 and determines how small elements near the edge of the panel will appear when compared to elements near the center. By default, the property is set to "0". Setting MinElementScale will ensure that elements near the edge of the panel will appear half their original size at the smallest.

5. Set the Center property to "0.1".

   The Center property consists of a number between 0 and 1.0 and sets where the center, or the most zoomed in element, of the panel is when the application is initially run. By default, the property is set to "0.5" and the center is in the middle. Setting Center will move the initially zoomed element to the left side of the control. This value is updated automatically as the mouse moves over the panel at run time.

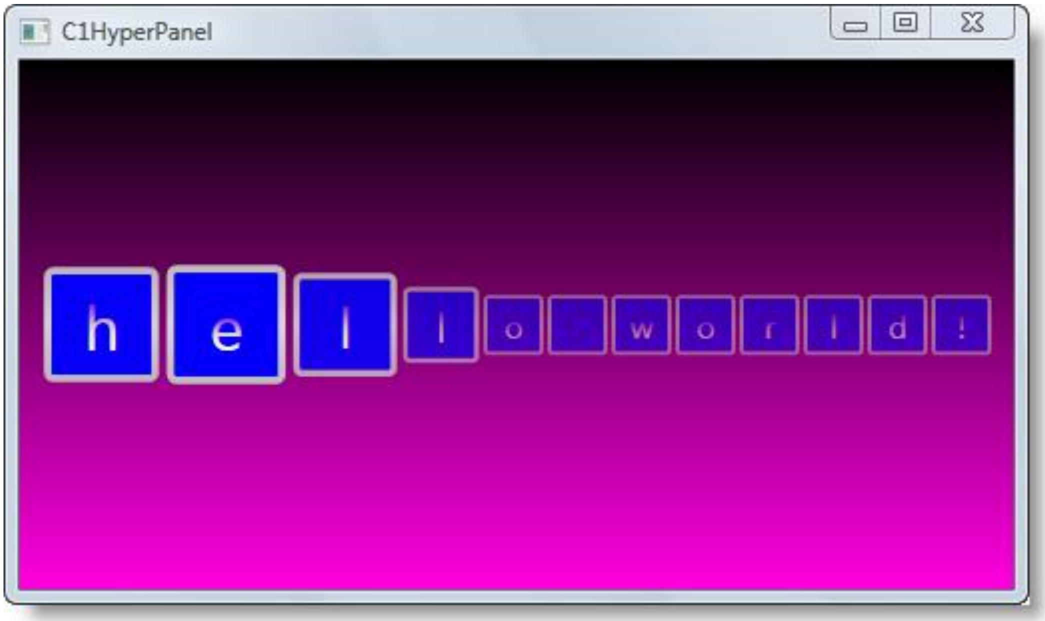
6. Switch to XAML view and add the following markup just after the `<C1HyperPanel>` tag:

```
<c1:C1HyperPanel.Background>
    <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
        <GradientStop Color="#FF000000" Offset="0"/>
        <GradientStop Color="#FFFF00DF" Offset="1"/>
    </LinearGradientBrush>
</c1:C1HyperPanel.Background>
```

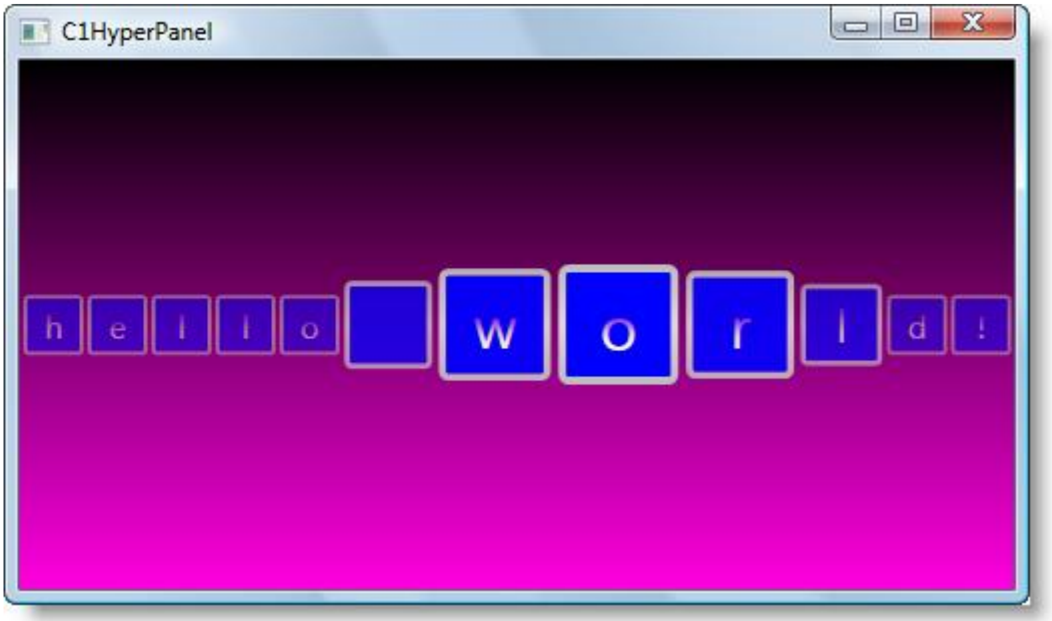
   This markup will add a gradient background to the C1HyperPanel.

7. From the **Project** menu, select **Test Solution** to view how your application will appear at run time.

   The application will initially appear similar to the following:



8. Move your mouse in the panel and notice that the Center of the content changes:



Congratulations! You've completed the **HyperPanel for WPF** quick start and created a **ComponentOne HyperPanel for WPF** application, customized the appearance and behavior of the controls, and viewed some of the run-time capabilities of your application.

# Working with HyperPanel for WPF

**ComponentOne HyperPanel for WPF** includes the C1HyperPanel panel, a simple **StackPanel** which provides an automatic zoom effect for items near the mouse. When you add the C1HyperPanel panel to a XAML window, it exists as an empty panel where you can add controls and content as you would to any other panel, the **Grid** or **Canvas**.

### Basic Properties

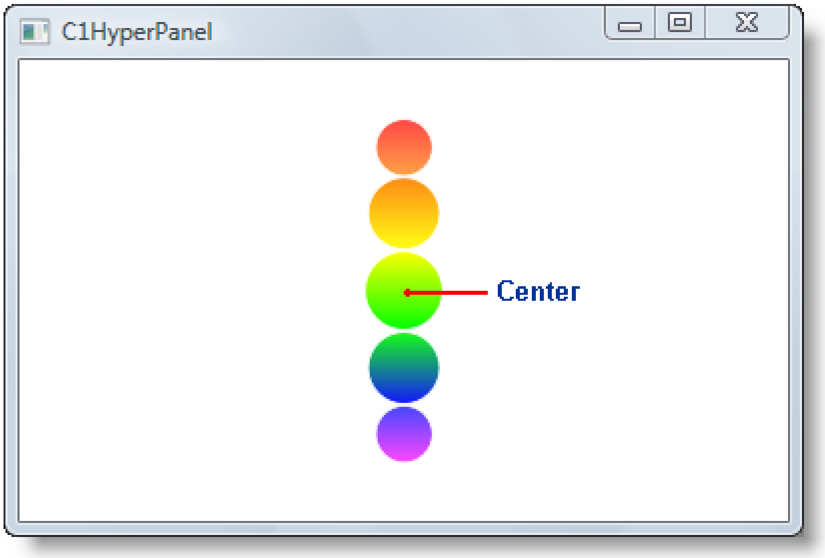
**ComponentOne HyperPanel for WPF** includes several properties that allow you to set the functionality of the panel. Some of the more important properties that vary from the standard **StackPanel** properties are listed below.

The following properties let you customize the C1HyperPanel panel:

| Property | Description |
|---|---|
| ApplyOpacity | Gets or sets a Boolean value that determines if opacity is applied to elements away from the center of the panel. |
| Center | Gets or sets the center of the **C1HyperPanel** as a percentage of the control size. |
| Distribution | Gets or sets a value between 0.1 and 1.0 that controls how much zooming should be applied to elements near the center. |
| HorizontalContentAlignment | Gets or sets the horizontal alignment of the panel's content. |
| MinElementScale | Gets or sets a value between zero and one that determines the minimum scale to be applied to elements when they are away from the center. |
| Orientation | Gets or sets a value that indicates the dimension by which child elements are stacked. |
| VerticalContentAlignment | Gets or sets the vertical alignment of the panel's content. |

## Zoom Center

The Center property determines where the initial center of the panel is located and indicates the largest element of the zoom effect of the panel. By default, the center is at the exact center of content in the panel:

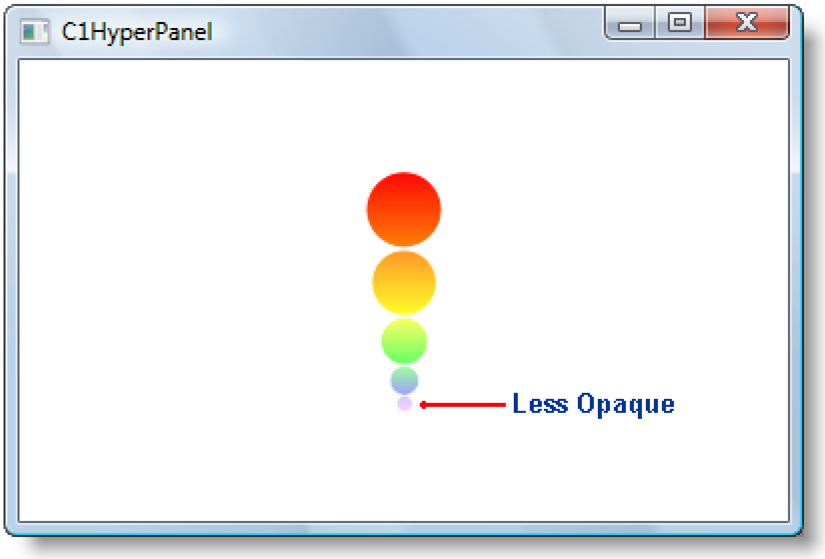


The Center can be set to a value from 0 and 1. By default, the Center is "0.5". If you set Center to "0", the Center will be the top edge of the panel (or left edge if the Orientation is set to **Horizontal**).

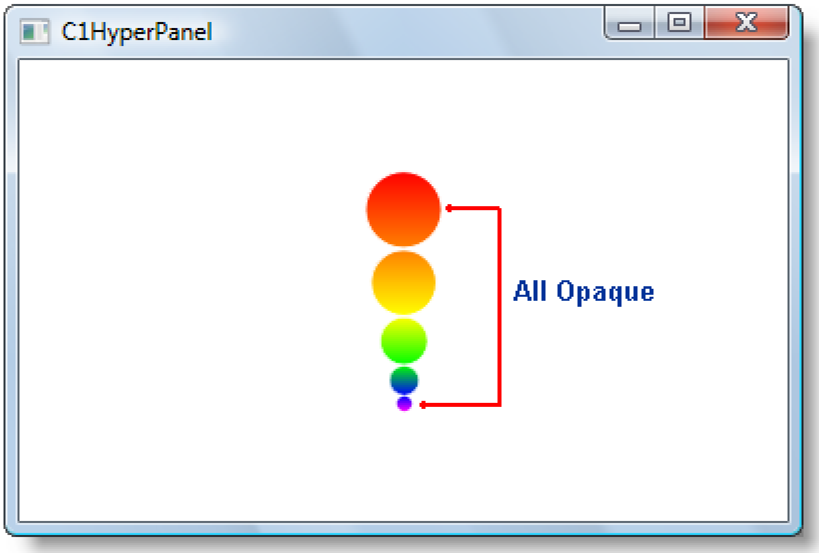
When Orientation is set to **Horizontal**, the Center property is also determined by the FlowDirection property. If Center is set to "0" and FlowDirection is set to **LeftToRight** (default) then the Center will be the left edge of the panel. If the FlowDirection is set to **RightToLeft** the Center will be the right edge of the panel.

# Edge Opacity

The ApplyOpacity property lets you control how items further from the Center of the panel appear. If ApplyOpacity is **True** (default) then items further away from the center will be less opaque:

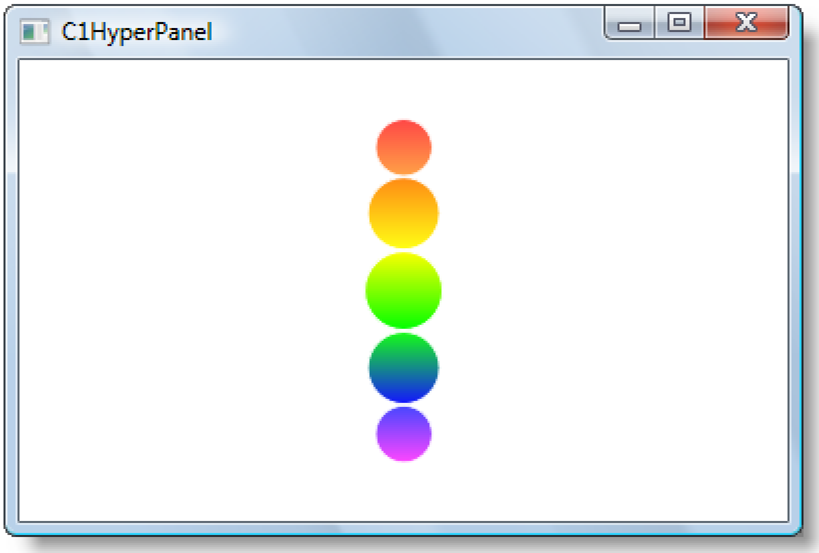


If ApplyOpacity is **False** then items further away from the center will appear at the same opacity level as the item in the center:

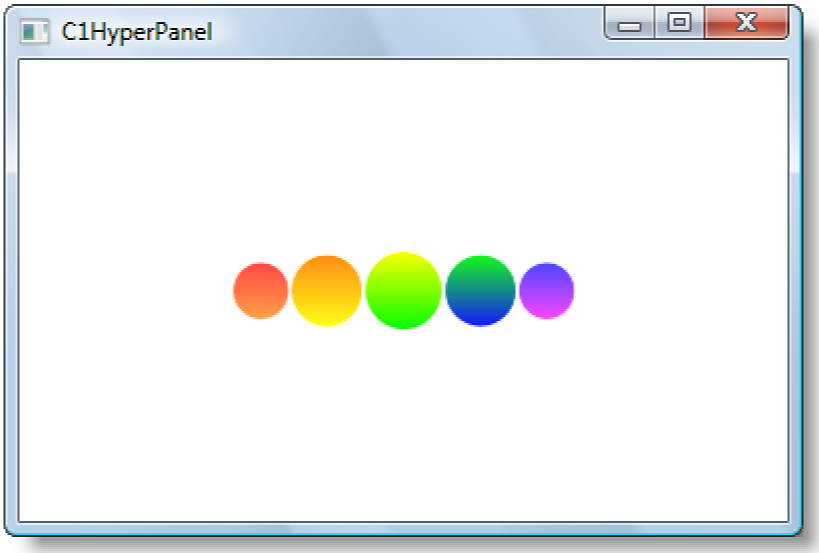


## Horizontal and Vertical Orientation

The Orientation property determines how content is laid out within the C1HyperPanel panel. By default, Orientation is set to **Vertical** and content appears stacked from top to bottom vertically in the panel:

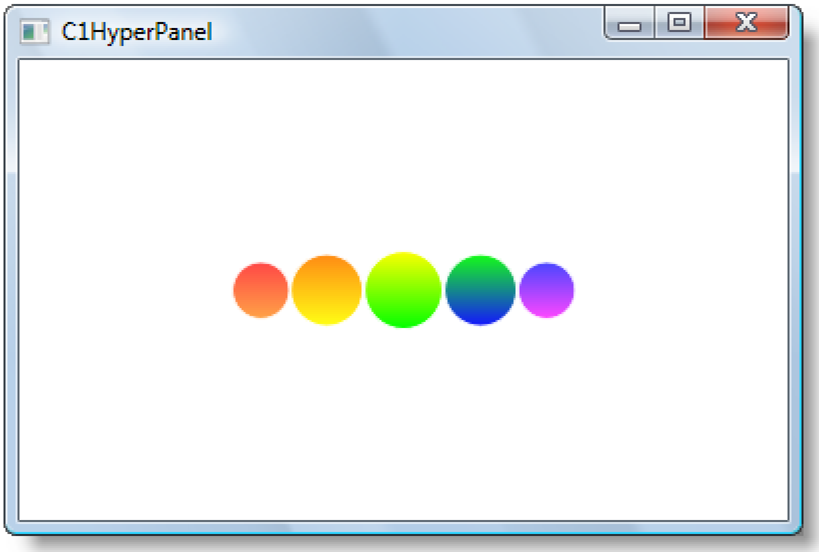


When Orientation is set to **Horizontal** content will appear stacked from left to right horizontally in the panel:

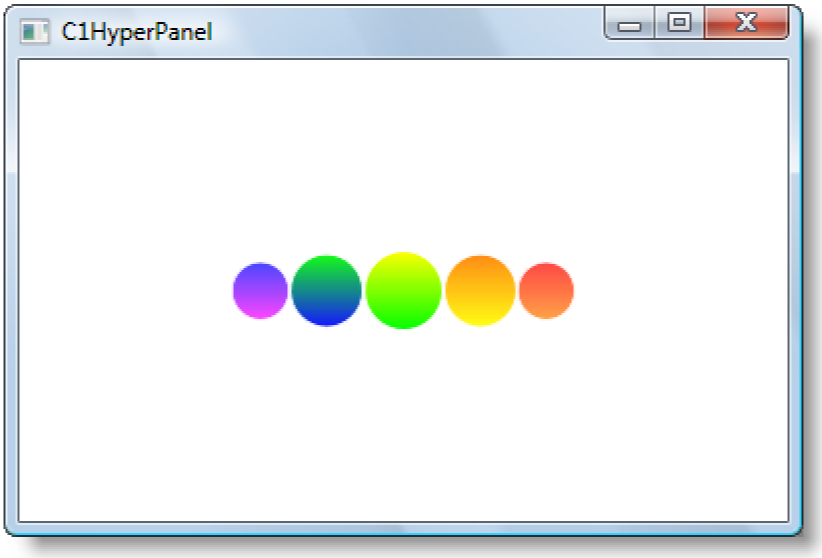


## Flow Direction

The FlowDirection property determined how content flows when the Orientation property is set to **Horizontal**. By default, FlowDirection is set to **LeftToRight** and when the Orientation property is set to **Horizontal**, content will appear stacked from left to right across the panel:



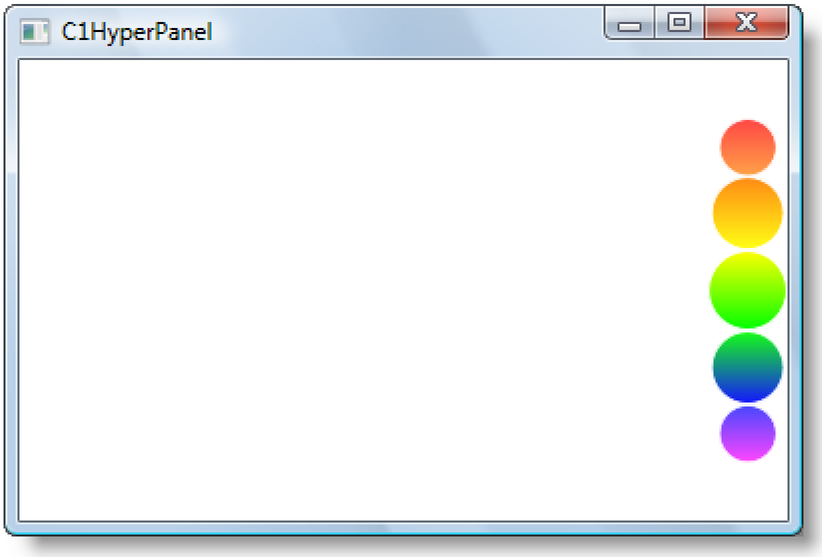
If FlowDirection is set to **LeftToRight** and the Orientation property is set to **Horizontal**, content will instead appear stacked from right to left across the panel:



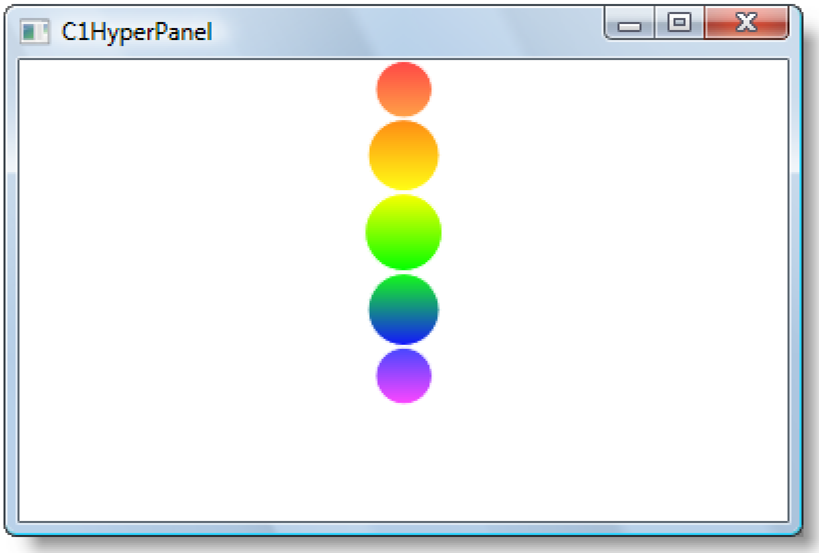
## Alignment

The **HorizontalAlignment** and **VerticalAlignment** properties control how the C1HyperPanel panel is aligned in the containing panel or window. By default, both properties are set to **Stretch** and the panel is stretched to fill the entire space available.

**HorizontalAlignment** options include **Left**, **Center**, **Right**, and **Stretch**. If, for example, **HorizontalAlignment** was set to **Right**, the panel would appear at the right of the available area like in the following image:



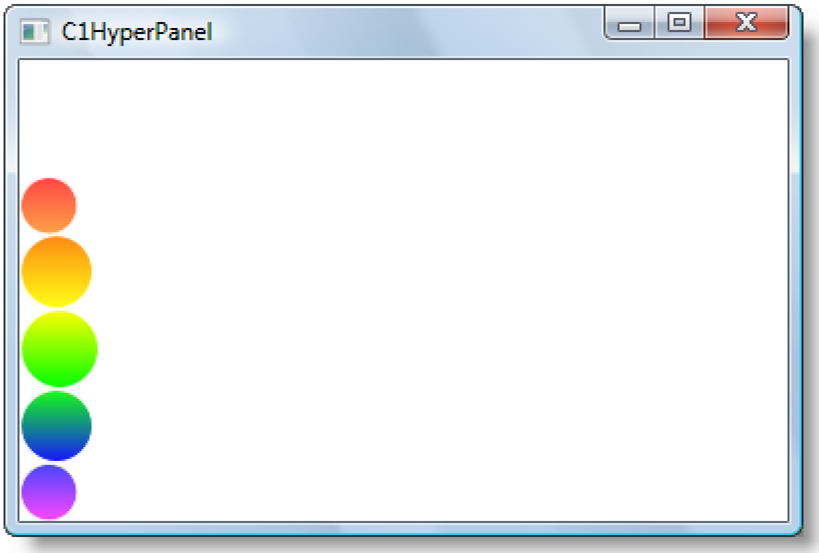
**VerticalAlignment** options include **Top**, **Center**, **Bottom**, and **Stretch**. If, for example, **VerticalAlignment** was set to **Top**, the panel would appear at the top of the available area like in the following image:



## Content Alignment

The HorizontalContentAlignment and VerticalContentAlignment properties control how content in the C1HyperPanel panel is aligned within the panel. By default, both properties are set to **Center** and content is centered within the panel.

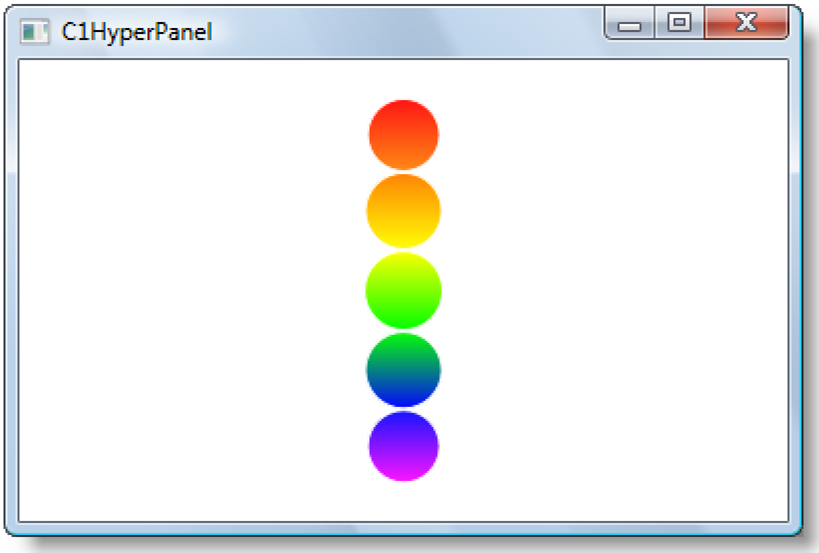
HorizontalContentAlignment options include **Left**, **Center**, **Right**, and **Stretch**. VerticalContentAlignment options include **Top**, **Center**, **Bottom**, and **Stretch**. If, for example, HorizontalContentAlignment was set to **Left** and VerticalContentAlignment was set to **Bottom**, the panel would appear in the bottom-left corner of the panel like in the following image:



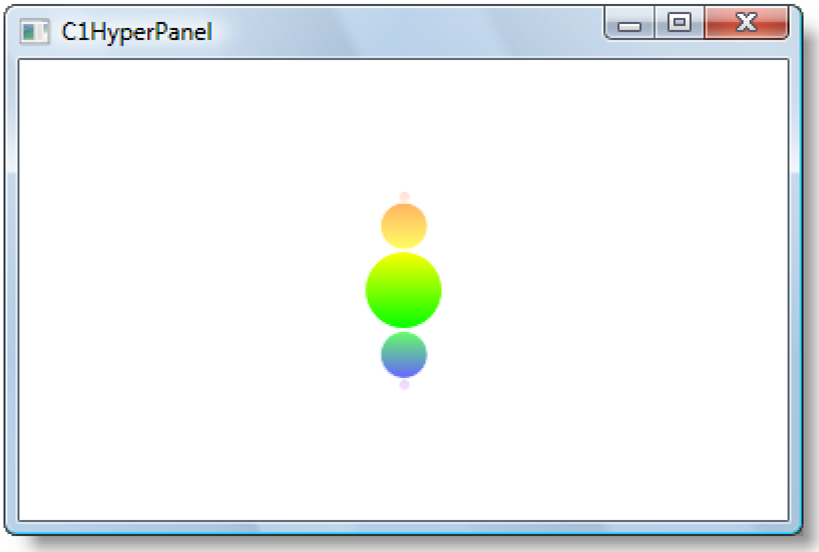
## Distribution

The Distribution property gets or sets a value between 0.1 and 1.0 that controls how much zooming should be applied to elements near the center of the panel. The smaller the value, the more distant items away from the center of the panel will appear. By default Distribution is set to "0.5".

If Distribution is set to "1", all elements will appear at the same zoom level, for example in the following image:



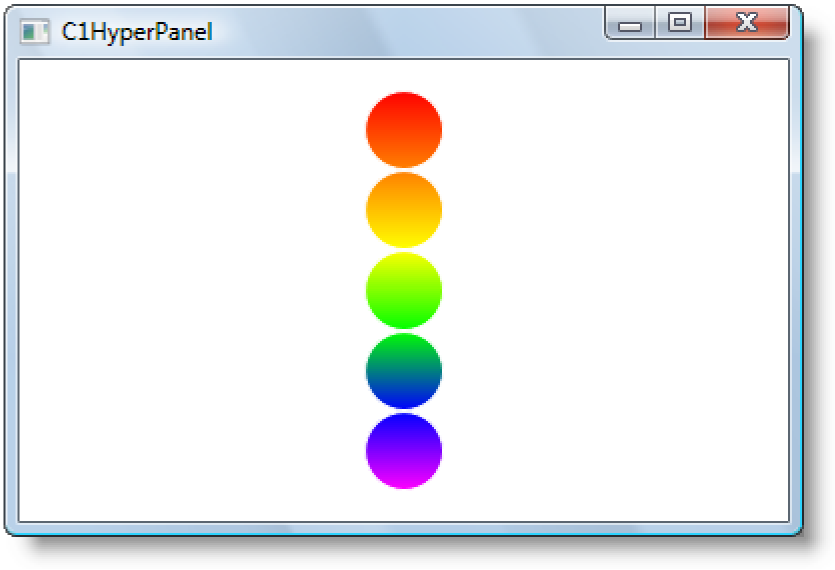
If Distribution is set to a smaller number, such as "0.2", elements away from the center will appear further zoomed out:



## Scale

The MinElementScale property gets or sets a value that determines the minimum scale to be applied to elements when they are away from the center. If MinElementScale is set to "0.1", items furthest away from the center will appear at smallest 10% of their original size.

By default, MinElementScale is set to "0" and items furthest away from the center of the panel appear completely zoomed out. The larger the number, the more zoomed in elements further from the center appear. For example, when MinElementScale is set to "1" as in the image below, all items appear to be the same distance away and elements appear at 100% of their original size. When the mouse is moved over elements in the panel they are not zoomed in or out at all and appear static:


C1HyperPanel

# HyperPanel for WPF Samples

Please be advised that this ComponentOne software tool is accompanied by various sample projects and/or demos, which may make use of other ComponentOne development tools included with the ComponentOne Studios. Samples can be accessed from the **ComponentOne Studio for WPF ControlExplorer**. To view samples, on your desktop, click the **Start** button and then click **All Programs | ComponentOne | Studio for WPF | Samples | WPF ControlExplorer**.

### C# Samples

The following C# sample is included:

| Sample | Description |
|---|---|
| ControlExplorer | The **HyperPanel** page in the **ControlExplorer** sample demonstrates how to add content to and customize the C1HyperPanel container. |

# HyperPanel for WPF Task-Based Help

The task-based help assumes that you are familiar with programming in Visual Studio .NET and know how to use the C1HyperPanel panel in general. If you are unfamiliar with the **ComponentOne HyperPanel for WPF** product, please see the HyperPanel for WPF Quick Start (page 3) first.

Each topic in this section provides a solution for specific tasks using the **ComponentOne HyperPanel for WPF** product.

Each task-based help topic also assumes that you have created a new WPF project and added a C1HyperPanel panel to your project.

## Adding a Control to the Panel

By default the C1HyperPanel panel appears empty and contains no content. Adding controls and other content to the panel is as easy as adding content to any other panel, **Canvas**, or **Grid**. In the following steps you'll add a button to your C1HyperPanel.

### At Design Time

To add a button to C1HyperPanel at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Visual Studio Toolbox and double-click the **Button** control. The control will be added to the panel.

### In XAML

To add a button to C1HyperPanel add the `<Button>` tag after the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1">
    <Button Height="50" Name="button1" Width="50"></Button>
</c1:C1HyperPanel>
```

### In Code

To add a button to C1HyperPanel, double-click the window to switch to Code view and add the **Window_Loaded** event handler; then add code so it appears like the following:

- Visual Basic

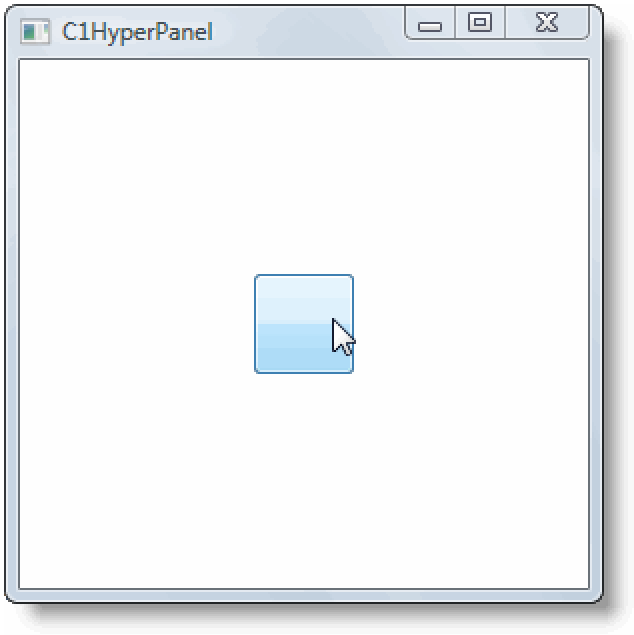
```
Private Sub Window1_Loaded(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles MyBase.Loaded
    Dim button1 = New Button
    button1.Height = 50
    button1.Width = 50
    Me.C1HyperPanel1.Children.Add(button1)
End Sub
```

- C#

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    Button button1 = new Button();
    button1.Height = 50;
    button1.Width = 50;
    this.c1HyperPanel1.Children.Add(button1);
}
```

**Run your project and observe:**

The C1HyperPanel panel will appear with a button:



## Changing the Background Color

By default the C1HyperPanel panel appears transparent and all elements below the panel will show through. If you choose, you can set the color of the panel using the **Background** property. For example, the following steps will detail how to change the color of the background to black.

### At Design Time

To change **C1HyperPanel**'s background color to black at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Properties window and locate the **Background** property.
3. Select the drop-down arrow next to the property name and select **Black** from the list.

### In XAML

To change **C1HyperPanel**'s background color to black in XAML add `Background="Black"` to the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1" Background="Black">
```

### In Code

For example, to change the background color add the following code to your project:

- Visual Basic

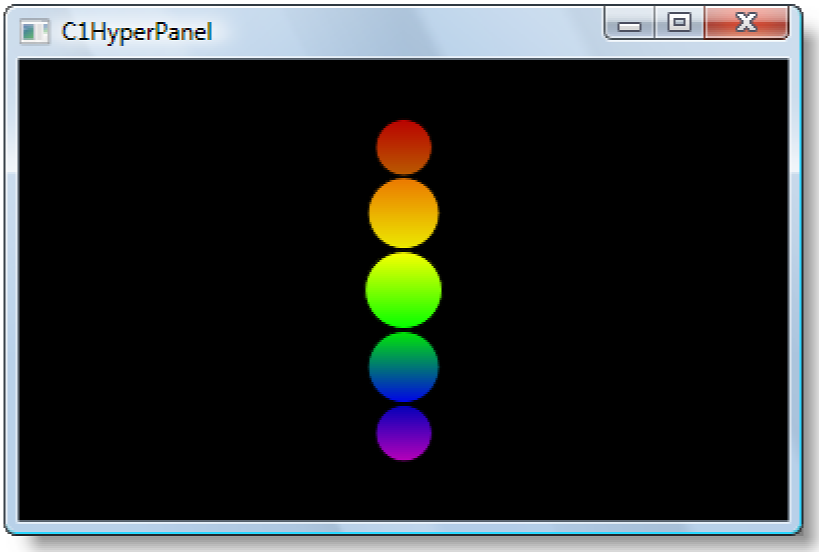
```
Me.C1HyperPanel1.Background = System.Windows.Media.Brushes.Black
```

- C#

```
this.c1HyperPanel1.Background = System.Windows.Media.Brushes.Black;
```

### Run your project and observe:

The C1HyperPanel panel will appear with a black background:



## Setting the Start Position

The Center property determines where the initial center of the panel is located. The "center" here does not actually mean the actual center of the panel, but rather indicates the largest element of the zoom effect of the panel. For example, the following steps will detail how to change the Center of the panel to the first item.

### At Design Time

To change **C1HyperPanel**'s Center at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Properties window and locate the Center property.
3. Click in the text box next to the Center property and enter "0".

### In XAML

To change **C1HyperPanel**'s Center in XAML add `Center="0"` to the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1" Center="0">
```

### In Code

To change the Center in code, add the following code to your project:

- Visual Basic

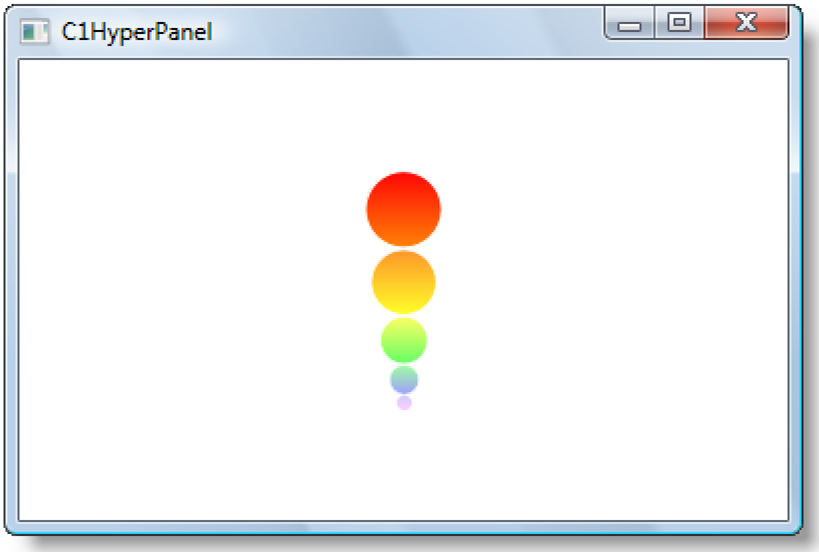
```
Me.C1HyperPanel1.Center = "0"
```

- C#

```
this.c1HyperPanel1.Center = "0";
```

### Run your project and observe:

The C1HyperPanel panel will appear centered on the first item in the panel:



## Changing the Scale

The MinElementScale property determines how small elements farthest away from the Center of the C1HyperPanel will appear. For example, using MinElementScale you can choose to set all elements to the same size which would remove the zoom effect.

### At Design Time

To set all C1HyperPanel elements to the same size at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Properties window and locate the MinElementScale property.
3. Click in the text box next to the MinElementScale property and enter "1".

### In XAML

To set all C1HyperPanel elements to the same size in XAML add `MinElementScale="1"` to the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1" MinElementScale="1">
```

### In Code

To set all C1HyperPanel elements to the same size in code, add the following code to your project:

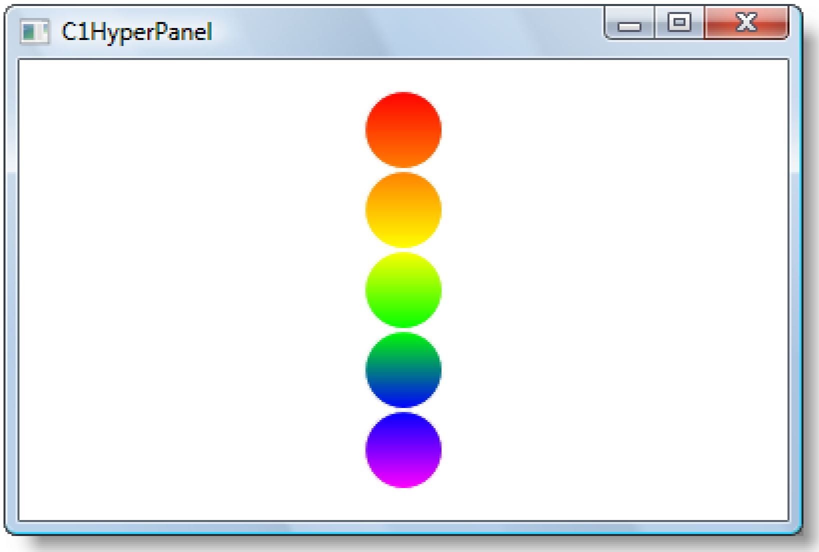
- Visual Basic

```
Me.C1HyperPanel1.MinElementScale = "1"
```

- C#
```
this.c1HyperPanel1.MinElementScale = "1";
```

**Run your project and observe:**

All items in the C1HyperPanel panel will appear the same size and when you mouse over items there will be no zoom effect:



## Setting the Orientation

By default the Orientation property is set to **Vertical** and content appears stacked from top to bottom vertically in the panel. If you choose, you can change the Orientation so that content appears stacked horizontally instead.

**At Design Time**

To set the Orientation so that content appears stacked horizontally at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Properties window and locate the Orientation property.
3. Select the drop-down arrow next to the Orientation property and choose **Horizontal**.

**In XAML**

To set the Orientation so that content appears stacked horizontally in XAML add `Orientation="Horizontal"` to the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1" Orientation="Horizontal">
```

**In Code**

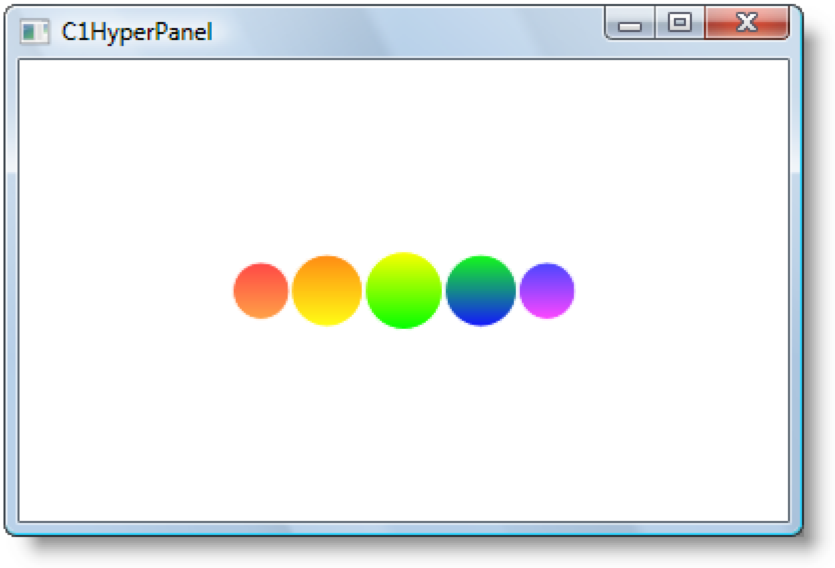
To set the Orientation so that content appears stacked horizontally in code, add the following code to your project:

- Visual Basic
```
Me.C1HyperPanel1.Orientation = Orientation.Horizontal
```

- C#
```
this.c1HyperPanel1.Orientation = Orientation.Horizontal;
```

**Run your project and observe:**

When Orientation is set to **Horizontal** content will appear stacked from left to right horizontally in the panel:



## Setting Element Distribution

The Distribution property controls how much zooming should be applied to elements near the center of the panel. The smaller the value, the smaller items away from the center of the panel will appear. In the following steps you'll set the Distribution so that items away from the center appear more zoomed out.

### At Design Time

To set the Distribution so that items away from the center appear more zoomed out at design time, complete the following:

1. Click once on the **C1HyperPanel** to select it.
2. Navigate to the Properties window and locate the Distribution property.
3. Click in the text box next to the Distribution property and enter "0.2".

### In XAML

To set the Distribution so that items away from the center appear more zoomed out in XAML add `Distribution="0.2"` to the `<c1:C1HyperPanel>` tag so that it appears similar to the following:

```
<c1:C1HyperPanel Name="C1HyperPanel1" Distribution="0.2">
```

### In Code

To set the Distribution so that items away from the center appear more zoomed out in code, add the following code to your project:

- Visual Basic

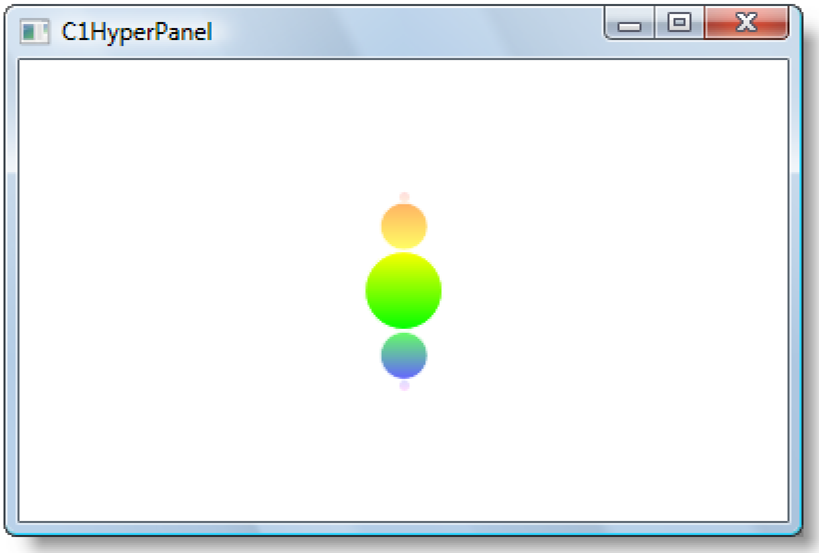
```
Me.C1HyperPanel1.Distribution = "0.2"
```

- C#

```
this.c1HyperPanel1.Distribution = "0.2";
```

### Run your project and observe:

With the Distribution property set to a smaller number, elements away from the center appear further zoomed out:


C1HyperPanel