ComponentOne

Carousel for Silverlight

By GrapeCity, Inc.

Corporate Headquarters
ComponentOne, a division of GrapeCity
201 South Highland Avenue
3rd 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 Carousel for Silverlight Overview

C1CarouselPanel is a virtualizing panel control that arranges child elements along an arbitrary 3D path. You can use **C1CarouselPanel** with any items control to turn a flat list of items into a virtualizing carousel with true 3D perspectives and scrolling animation.

Help with ComponentOne Studio for Silverlight

Getting Started

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

What's New

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

Key Features

ComponentOne Carousel for Silverlight provides the C1CarouselPanel control that allows you to view content and controls – including standard or 3rd-party controls – in an interactive carousel view. Similar in concept to the **C1CoverFlow** or **C1HyperPanel** control, C1CarouselPanel is a unique way of displaying content and data. Items in the panel revolve on a path when manipulated at run time.

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

- **View Data Interactively**

  The C1CarouselPanel control allows you to display content and data in a unique, highly interactive format.

- **Define an Arbitrary Path**

  C1CarouselPanel items can navigate along any arbitrary path. Define a simple ellipse path, or create a more advanced path such as a zig-zag or a figure '8'. **C1CarouselPanel** provides a simplified 3D path definition as a composition of a 2D path and a rotation angle in a 3D space.

- **True 3D Perspective**

  When distributing elements along a 3D path, the placements of child elements in 3D space are fully honored. This means that more distant elements are overlapped by nearer ones, and more distant elements will be diminished more than the nearer ones. This creates a true 3D scene with a perspective effect.

- **Perspective Effects**

  The perspective effect is controlled via a single property that defines how much the elements will be diminished. This depends on the distance of each element from the screen plane in 3D space.

- **Scroll Animation**

  C1CarouselPanel provides scrolling animation in order to create the effect of elements movement during scroll.

- **Many Control Options**

  C1CarouselPanel provides a number of control options such as alignment of the path, placement of elements and more.

- **Use with any Items Control**

  You can use C1CarouselPanel with any items control to turn a flat list of items into a virtualizing and interactive carousel.

Carousel for Silverlight Quick Start

The following quick start guide is intended to get you up and running with **ComponentOne Carousel for Silverlight**. In this quick start you'll create a simple project using a C1CarouselPanel control. You'll create a new Silverlight application, create a **ListBox** with media, add the C1CarouselPanel control in a template, and observe some of the run-time interaction possible with **Carousel for Silverlight**.

Note that in this example, you'll use images installed with the **ComponentOne Studio for Silverlight** samples. You can substitute other images, but you will have to update the steps below.

Step 1 of 3: Creating the Silverlight Application

In this step you'll create and set up Silverlight application. Note that in this example, you'll use images installed with the **ComponentOne Studio for Silverlight** samples. If you choose, you can substitute other images but you will have to update the steps below. To set up your project, complete the following steps:

1. From the Visual Studio **File** menu select **New** and choose **Project**.

2. In the **New Project** dialog box choose a language and **Silverlight Application** in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project. You may also need to select Silverlight 4 as the Silverlight version number. 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.

3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the **C1.Silverlight** and **C1.Silverlight.Carousel** assemblies and click **OK** to add references to your project.

4. Open the XAML view of the MainPage.xaml file; in this quick start you'll use XAML markup and code.

5. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

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

   This is a unified namespace that will enable you to work with most ComponentOne WPF or Silverlight controls without adding multiple namespaces.

6. In the Solution Explorer window, right-click the **QuickStart.Web** item and select **Add** | **New Folder** to add a folder. Name the folder "Resources".

7. In the Solution Explorer window, right-click the **Resources** folder and select **Add** | **New Folder** to add a folder. Name the folder "covers".

8. Right-click the **covers** folder and select **Add Existing Item**. This will open the **Add Existing Item** dialog box.

9. In the **Add Existing Item** dialog box, locate the **covers** folder in the **C1Extended_Demo** sample folder, select all the files, and click **Add**. By default this sample should be located in the **Documents** or **MyDocuments** folder in **ComponentOne Samples\Studio for Silverlight \C1.Silverlight.Extended\C1Extended_Demo\C1Extended_Demo.Web\Resources\covers**. Several images will be added to the application.

10. Select all of the images you added in the Solutions Explorer (select the first image and press SHIFT and click on the last image) and set their **Build Action** to **Resource** in the Properties window.

You've successfully created a new Silverlight application and added image resources to the application. In the next step you'll add a **ListBox** and create a **C1CarouselPanel** template to apply to the **ListBox** control.

Step 2 of 3: Adding Content to the Application

In the previous step you created a new Silverlight application and added image resources to the application. In this step you'll add a **ListBox** and create a **C1CarouselPanel** template to apply to the **ListBox** control. Complete the following steps:

1. In the XAML view add a **Resources** tag just under the **UserControl** tag and above the **Grid** tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

2. Add an **ItemsPanelTemplate** within the **Resources** tag to define the **C1CarouselPanel**:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```

3. Add a **DataTemplate** within the **Resources** tag:

   ```
   <DataTemplate x:Key="carouselItemTemplate">

       <Image Source="{Binding}" Stretch="None" />

   </DataTemplate>
   ```

4. Add a **Style** within the **Resources** tag to define the **C1CarouselPanel**'s path:

   ```
   <Style x:Key="circlePanelStyle" TargetType="ListBox">

       <Setter Property="c1:C1CarouselPanel.PathGeometry" Value="F1 M
   466.829,27.2642C 635.339,35.6577 762.667,98.3819 762.667,173C
   762.667,254.002 613.428,319.667 429.333,319.667C 245.238,319.667
   96,254.002 96,173C 96,98.0584 224.402,35.1712 393.751,27.1714"/>

       <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
   Value="Left"/>

       <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
   Value="Top"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveAngle" Value="55"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveFactor" Value="0.4"/>

   </Style>
   ```

5. Create a **ListBox** control by adding the following **ListBox** tag within the **Grid** tags:

   ```
   <ListBox Background="Transparent" Name="carouselListBox" Grid.Row="1"
   ItemsPanel="{StaticResource carouselPanelTemplate}"
   ItemTemplate="{StaticResource carouselItemTemplate}"
   Style="{StaticResource circlePanelStyle}">

   </ListBox>
   ```

   Note that this **ListBox** uses the templates you just added. The **ItemsPanel** property assigns the C1CarouselPanel to the **ListBox** via the **ItemsPanelTemplate**.

6. Switch to Code view by right-clicking the page and selecting **View Code**.

7. Add the following import statements at the top of the page of the Code Editor:

   - Visual Basic

     ```
     Imports System.Windows.Media.Imaging
     Imports C1.Silverlight
     Imports C1.Silverlight.Carousel
     ```

- C#

```
using System.Windows.Media.Imaging;
using C1.Silverlight;
using C1.Silverlight.Carousel;
```

8. Add code to the main class so it appears like the following:

   - Visual Basic

   ```
   Public Sub New()
       InitializeComponent()
       InitData()
   End Sub
   ```

   - C#

   ```
   public MainPage()
   {
       InitializeComponent();
       InitData();
   }
   ```

9. Add the following code below the main class:

   - Visual Basic

   ```
   Private Sub InitData()
       For i As Integer = 101 To 140
           carouselListBox.Items.Add(New
   BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
   ".jpg")))
          Next
   End Sub
   ```

   - C#

   ```
   private void InitData()
   {
       for (int i = 101; i <= 140; i++)
       {
           carouselListBox.Items.Add(new
   BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
   ".jpg")));
          }
   }
   ```

   This loads the images you added into the **ListBox** control. Note that if you added different images to your project, you may have to adapt the above code.

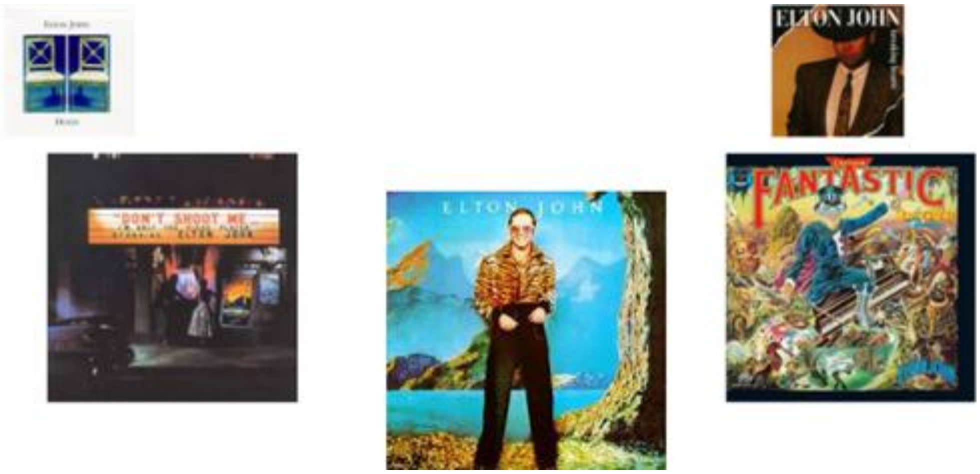
In this step you added a **ListBox** and create a **C1CarouselPanel** template to apply to the **ListBox** control. Now all that's left is to run the application.

Step 3 of 3: Running the Application

Now that you've created a Silverlight application and added content to it, the only thing left to do is run your application. To run your application and complete the following steps:

1. From the **Debug** menu, select **Start Debugging** to view how your application will appear at run time.

   The application will appear similar to the following:



2. Click on the left-most image. Notice that the carousel rotates in a circular path to the right. This path was specified in the **Style** resource.
3. Click on the right-most image. Notice that the carousel rotates in a circular path to the left.
4. Click on the horizontal scroll bar in your browser. Notice that that clicking on the scroll bar causes the carousel to rotate.

Congratulations! You've completed the **Carousel for Silverlight** quick start and created a simple Silverlight application, added and customized a **Carousel for Silverlight** control, and viewed some of the run-time capabilities of the control.

Working with Carousel for Silverlight

ComponentOne Carousel for Silverlight includes the C1CarouselPanel control, a simple viewer that allows you to add a carousel-like animation to your application. C1CarouselPanel can be used to lay out any control with child elements that can be arranged in an arbitrary panel; for example, you can use C1CarouselPanel as **ItemsPanel** of **ListBox** or as **MonthsPanel** of **Scheduler for Silverlight**'s **C1Calendar**. Use the carousel panel for a unique, interactive data presentation.

Using C1CarouselPanel

To use the **C1CarouselPanel** to create a carousel-like interactive effect, you can set it as an **ItemsControl** control's panel and assign your visual elements collection to the **ItemsControl.Items** property. So, for example, in the XAML below an **ItemsPanelTemplate** template is defined in the Window's resources and includes the **C1CarouselPanel** panel. An **ItemsControl** containing arbitrary elements later points to the **ItemsPanelTemplate** template:

```
<UserControl x:Class="C1Carousel.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <UserControl.Resources>
        <!-- An ItemsPanelTemplate template defining the C1CarouselPanel.
-->
        <ItemsPanelTemplate x:Key="carouselPanel">
            <c1:C1CarouselPanel />
        </ItemsPanelTemplate>
    </UserControl.Resources>
    <Grid>
<!-- An ItemsControl with ItemsPanel set to the ItemsPanelTemplate
defining a C1CarouselPanel. -->
<ItemsControl ItemsPanel="{StaticResource carouselPanel}">
<!-- Arbitrary controls or images within the ItemsControl. -->
            <Image Width="51" Height="51" Source="image1.png"/>
            <Image Width="51" Height="51" Source="image2.png"/>
            <Image Width="51" Height="51" Source="image3.png"/>
            <Button Height="23" Name="Button1" Width="75">Button</Button>
</ItemsControl>
    </Grid>
</UserControl>
```

Customizing the Carousel

You can customize the **C1CarouselPanel** in the example in the Using C1CarouselPanel (page 11) topic by setting properties in either the `<c1:C1CarouselPanel/>` tag in the **ItemsPanelTemplate** or in the `<ItemsControl>` tag. Setting properties in the `<ItemsControl>` tag is possible because all of the properties introduced in carousel are attached dependency properties, providing the ability to change carousel's property at run time.

In the C1CarouselPanel Tag

So, for example, to limit the number of visible elements in the C1CarouselPanel, you can set the PageSize property in the `<c1:C1CarouselPanel>` tag:

```
<UserControl.Resources>
     <!-- An ItemsPanelTemplate template defining the C1CarouselPanel. -->
    <ItemsPanelTemplate x:Key="carouselPanel">
        <!-- Limit the number of visible elements to 3 with the PageSize
property. -->
        <c1:C1CarouselPanel PageSize="3" />
    </ItemsPanelTemplate>
</UserControl.Resources>
```

In the ItemsControl Tag

To limit the number of visible elements in the C1CarouselPanel, you can also set the PageSize property in the in the `<ItemsControl>` tag:

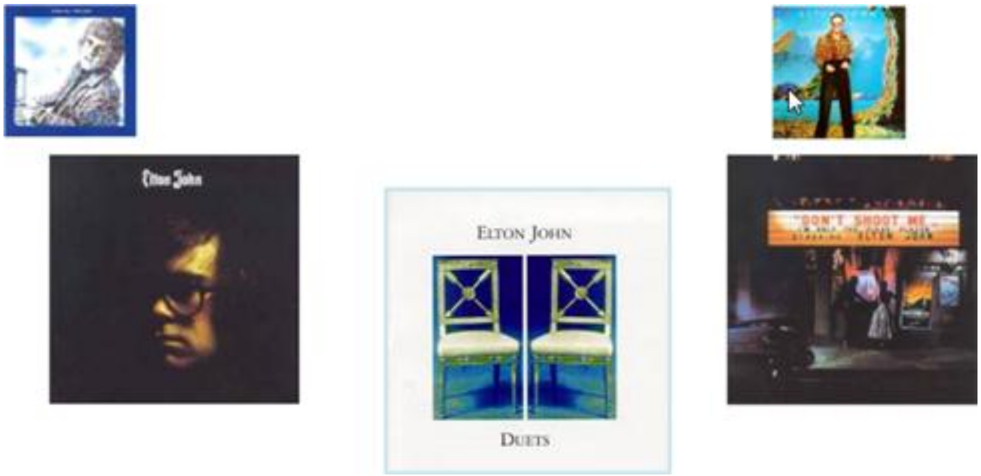
```
<!-- An ItemsControl with ItemsPanel set to the ItemsPanelTemplate
defining a C1CarouselPanel. The PageSize property limits the number of
visible elements to 3. -->

<ItemsControl ItemsPanel="{StaticResource carouselPanel}"
c1:C1CarouselPanel.PageSize="3">

<!-- Arbitrary controls or images within the ItemsControl. -->

            <Image Width="51" Height="51" Source="image1.png"/>

            <Image Width="51" Height="51" Source="image2.png"/>

            <Image Width="51" Height="51" Source="image3.png"/>

            <Button Height="23" Name="Button1" Width="75">Button</Button>

</ItemsControl>
```

Run-Time Interaction

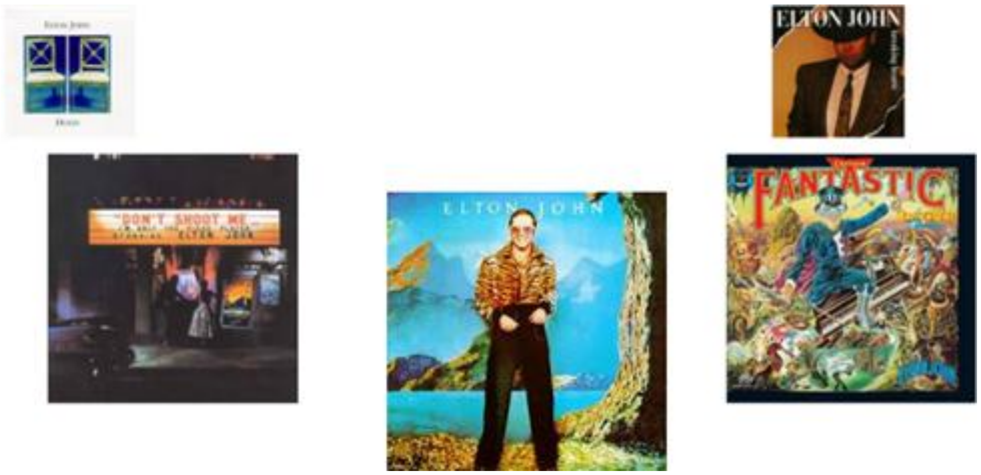
Users can interact with items in the toolbar and content area of the C1CarouselPanel control at run time. Users can move and drag content in the content area or use the scrollbar to manipulate the items displayed in the content area.

Moving Content

To move content, click on any item in the **C1CarouselPanel**. The content in the **C1CarouselPanel** will shift to the left or right according to the path you have specified and the clicked item will become the front-most item. For example, clicking the right-most item in the following image:

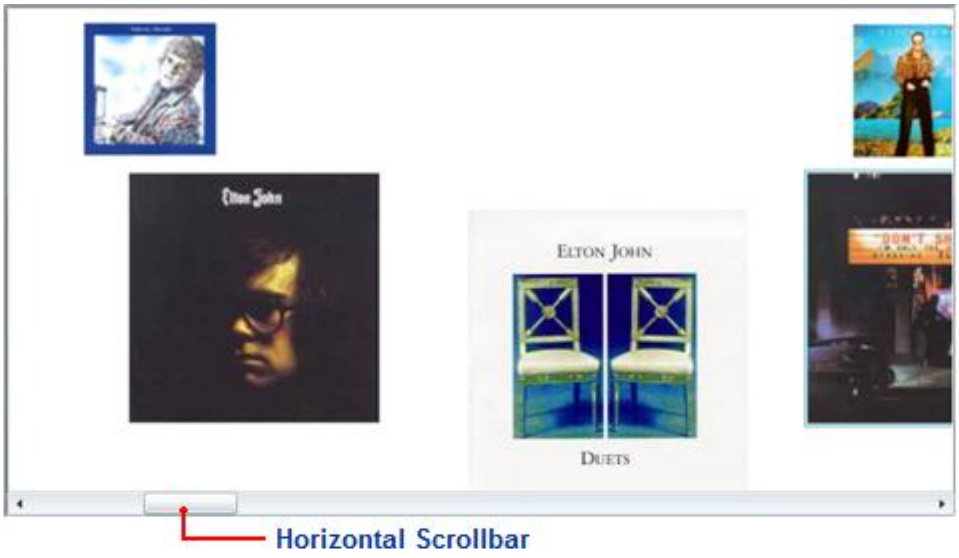


Will move that item to the center of the **C1CarouselPanel**:



Scrolling Content

You can also move the items in the C1CarouselPanel control by clicking in the horizontal scrollbar in your browser:



Click and drag on the thumb button to move content, or click on the scrollbar next to the thumb button to move the content in the direction you choose.

Carousel for Silverlight Task Based Help

The following task-based help topics assume that you are familiar with Visual Studio and Expression Blend and know how to use the C1Carousel control in general. If you are unfamiliar with the **ComponentOne Carousel for Silverlight** product, please see the Carousel for Silverlight Quick Start (page 7) first.

Each topic in this section provides a solution for specific tasks using the **ComponentOne Carousel for Silverlight** product. Most task-based help topics also assume that you have created a new Silverlight project and added a C1Carousel control to the project – for information about creating the control, see Adding C1Carousel to the Application (page 15).

Adding C1Carousel to the Application

Follow these steps to add a C1Carousel control to your project:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
    - C1.Silverlight.dll
    - C1.Silverlight.Carousel.dll
4. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

    The namespaces will now appear similar to the following:

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

5. In the XAML view add a **Resources** tag just under the **UserControl** tag and above the **Grid** tag with the following markup:

    ```
    <UserControl.Resources>

    </UserControl.Resources>
    ```

    You will add templates within this tag.

6. Add an **ItemsPanelTemplate** within the **Resources** tag to define the **C1CarouselPanel**:

    ```
    <ItemsPanelTemplate x:Key="carouselPanelTemplate">

        <c1:C1CarouselPanel Padding="0, 10, 50, 50"
    VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
    VerticalItemAnchorOnPath="Center"/>

    </ItemsPanelTemplate>
    ```

You have now added a C1CarouselPanel to the page. If you run your application now, there will be an empty Silverlight application displayed.

Adding and Moving Content in C1Carousel

The following Help topics will walk you through adding content to the Carousel control through adding items to the Resources folder and then adding code to call items from the Resources folder. They will also walk you through creating a path along which your content can move. There are several different paths along which you can move content in the C1Carousel control.

Moving Content Along an Elliptical Path

Follow these steps to move content along an Elliptical path:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
   - C1.Silverlight.dll
   - C1.Silverlight.Carousel.dll
4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

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

8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">
   ```

```
    <c1:C1CarouselPanel Padding="0, 10, 50, 50"
VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
VerticalItemAnchorOnPath="Center"/>

</ItemsPanelTemplate>
```

10. Insert a `<DataTemplate>` to define the image source:

```
<DataTemplate x:Key="carouselItemTemplate">

            <Image Source="{Binding}" Stretch="None" />

        </DataTemplate>
```

11. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

```
<Style x:Key="circlePanelStyle" TargetType="ListBox">

       <Setter Property="c1:C1CarouselPanel.PathGeometry"

        Value="F1 M 466.829,27.2642C 635.339,35.6577 762.667,98.3819
762.667,173C 762.667,254.002 613.428,319.667 429.333,319.667C
245.238,319.667 96,254.002 96,173C 96,98.0584 224.402,35.1712
393.751,27.1714"/>

       <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
Value="Left"/>

       <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
Value="Top"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveAngle"
Value="55"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveFactor"
Value="0.4"/>

        </Style>
```

12. Add the following **ListBox** control within the `<Grid>`:

```
<ListBox Background="Transparent" Name="carouselListBox" Grid.Row="1"

                 ItemsPanel="{StaticResource carouselPanelTemplate}"

                 ItemTemplate="{StaticResource carouselItemTemplate}"

                 Style="{StaticResource circlePanelStyle}"/>
```

13. Right-click the **MainPage.xaml** page and select **View Code** from the list.

14. Add the following namespaces to the top of the page:

    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`
    - `C1.Silverlight.Carousel.dll`

15. Insert the following method directly below the **InitializeComponent()** method:

    - Visual Basic

      ```
      InitData()
      ```

    - C#

      ```
      InitData();
      ```

16. Call the following method to populate the C1CarouselPanel with data:

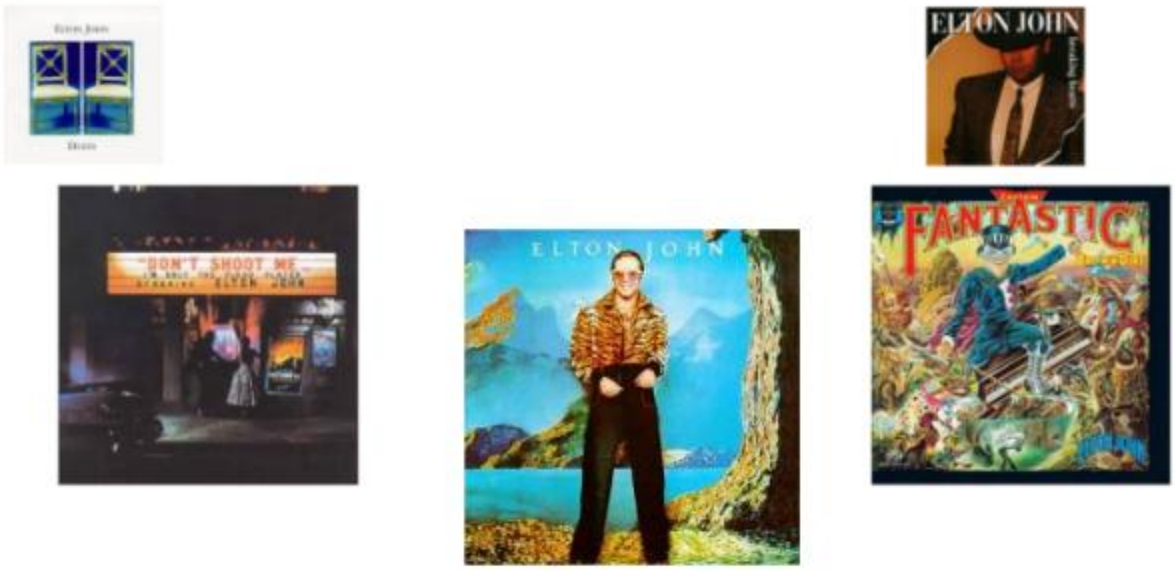
- Visual Basic

```
Private Sub InitData()
   For i As Integer = 101 To 140
         carouselListBox.Items.Add(New
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
".jpg")))
   Next
End Sub
```

- C#

```
private void InitData()
      {
          for (int i = 101; i <= 140; ++i)
          {
              carouselListBox.Items.Add(new
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
".jpg")));
          }
```

17. Press F5 to run your application. The carousel should resemble the following image:



Moving Content Along a ZigZag Path

Follow these steps:

1. From the Visual Studio **File** menu select **New** and choose **Project**.

2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
   - C1.Silverlight.dll
   - C1.Silverlight.Carousel.dll
4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

   ```
   <UserControl x:Class="QuickStart.MainPage"
   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
   xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
   mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
   ```
8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.
9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```
10. Insert a `<DataTemplate>` to define the image source:

    ```
       <DataTemplate x:Key="carouselItemTemplate">

                   <Image Source="{Binding}" Stretch="None" />

               </DataTemplate>
    ```
11. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

    ```
       <Style x:Key="zigzagPanelStyle" TargetType="ListBox">

              <Setter Property="c1:C1CarouselPanel.PathGeometry"
    ```

```
        Value="M 259.228,50.3333C 361.691,64.8311 464.153,79.3289
525.213,93.8267C 586.274,108.324 605.932,122.822 607.421,137.32C
608.911,151.818 592.231,166.316 566.913,176.05C 541.595,185.784
507.64,190.755 440.027,198.004C 372.413,205.252 271.142,214.78
216.039,226.171C 160.936,237.562 152,250.817 152,263.244C 152,275.67
160.936,287.268 191.615,299.281C 222.294,311.293 274.717,323.72
328.331,331.59C 381.945,339.46 436.75,342.774 492.449,346.916C
548.148,351.059 604.741,356.029 661.333,361"/>

       <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
Value="Left"/>

       <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
Value="Top"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveAngle"
Value="31"/>

       <Setter Property="c1:C1CarouselPanel.PerspectiveFactor"
Value="1.08"/>

        </Style>
```

12. Add the following **ListBox** control within the `<Grid>`:

```
<ListBox Background="Transparent" Name="carouselListBox"
Grid.Row="1"

     ItemsPanel="{StaticResource carouselPanelTemplate}"

     ItemTemplate="{StaticResource carouselItemTemplate}"

     Style="{StaticResource zigzagPanelStyle}"/>
```

13. Right-click the **MainPage.xaml** page and select **View Code** from the list.
14. Add the following namespaces to the top of the page:
    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`
    - `C1.Silverlight.Carousel.dll`
15. Insert the following method directly below the **InitializeComponent()** method:
    - Visual Basic

    ```
    InitData()
    ```

    - C#

    ```
    InitData();
    ```

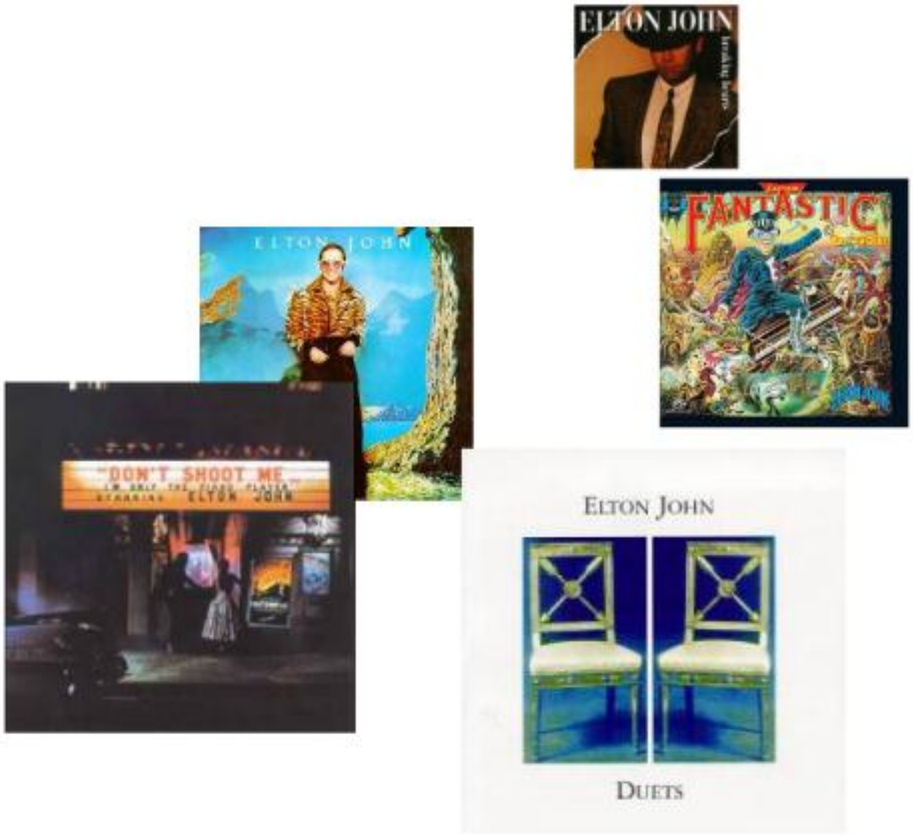
16. Call the following method to populate the C1CarouselPanel with data:
    - Visual Basic

```
Private Sub InitData()

   For i As Integer = 101 To 140

         carouselListBox.Items.Add(New
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
".jpg")))

   Next

End Sub
```

- C#

```
private void InitData()
      {
          for (int i = 101; i <= 140; ++i)
          {
              carouselListBox.Items.Add(new
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
".jpg")));
          }
```

17. Press F5 to run your application. The carousel should resemble the following image:



Moving Content Along a Diagonal Path

Follow these steps to move items along a diagonal path:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:

- C1.Silverlight.dll
- C1.Silverlight.Carousel.dll

4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

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

8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```

10. Insert a `<DataTemplate>` to define the image source:

    ```
    <DataTemplate x:Key="carouselItemTemplate">

                <Image Source="{Binding}" Stretch="None" />

            </DataTemplate>
    ```

11. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

    ```
    <Style x:Key="diagonalPanelStyle" TargetType="ListBox">

      <Setter Property="c1:C1CarouselPanel.PathGeometry"         Value="M
    216, 60 L 600, 390"/>

      <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
    Value="Left"/>

      <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
    Value="Top"/>
    ```

```
  <Setter Property="c1:C1CarouselPanel.PerspectiveAngle" Value="65"/>

  <Setter Property="c1:C1CarouselPanel.PerspectiveFactor"
Value="0.12"/>

        </Style>
```

12. Add the following **ListBox** control within the `<Grid>`:

```
<ListBox Background="Transparent" Name="carouselListBox"
Grid.Row="1"

     ItemsPanel="{StaticResource carouselPanelTemplate}"

     ItemTemplate="{StaticResource carouselItemTemplate}"

     Style="{StaticResource diagonalPanelStyle}"/>
```

13. Right-click the **MainPage.xaml** page and select **View Code** from the list.
14. Add the following namespaces to the top of the page:
    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`
    - `C1.Silverlight.Carousel.dll`
15. Insert the following method directly below the **InitializeComponent()** method:
    - Visual Basic

    ```
    InitData()
    ```

    - C#

    ```
    InitData();
    ```

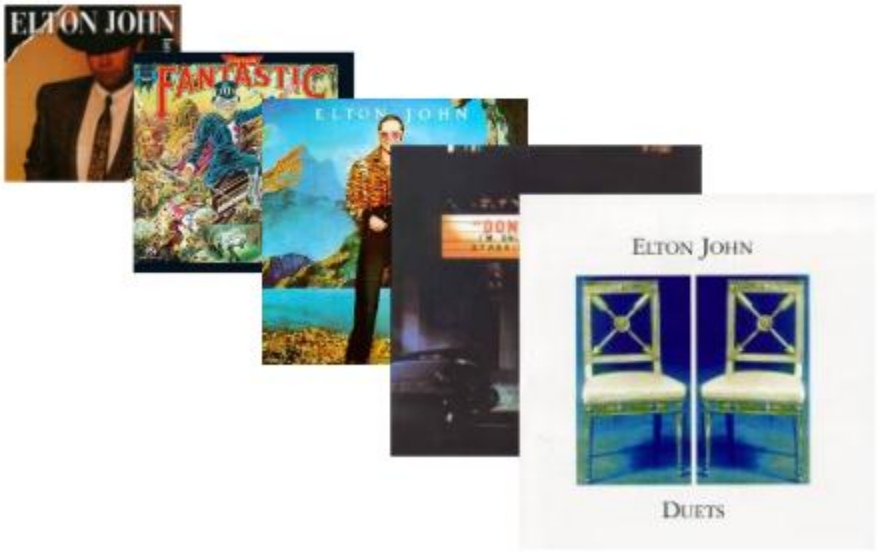
16. Call the following method to populate the C1CarouselPanel with data:
    - Visual Basic

    ```
    Private Sub InitData()

       For i As Integer = 101 To 140

             carouselListBox.Items.Add(New
    BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
    ".jpg")))

       Next

    End Sub
    ```

    - C#

    ```
    private void InitData()

          {

              for (int i = 101; i <= 140; ++i)

              {

                  carouselListBox.Items.Add(new
    BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
    ".jpg")));

              }
    ```

Press F5 to run your application. The carousel should resemble the following image:



Moving Content Along a Parabolic Path

Follow these steps:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
    - C1.Silverlight.dll
    - C1.Silverlight.Carousel.dll
4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

    The namespaces will now appear similar to the following:

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

8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```

10. Insert a `<DataTemplate>` to define the image source:

    ```
       <DataTemplate x:Key="carouselItemTemplate">

                   <Image Source="{Binding}" Stretch="None" />

               </DataTemplate>
    ```

11. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

    ```
       <Style x:Key="parabolaPanelStyle" TargetType="ListBox">

           <Setter Property="c1:C1CarouselPanel.PathGeometry"        Value="F1
       M 164.564,73.0518C 201.647,164.183 238.73,255.315 284.817,323.23C
       330.903,391.146 385.993,435.845 445.921,431.137C
       505.849,426.43 570.614,372.315 614.53,304.361C 658.446,236.407
       681.512,154.614 704.578,72.8207"/>

           <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
       Value="Left"/>

           <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
       Value="Top"/>

           <Setter Property="c1:C1CarouselPanel.PerspectiveAngle" Value="90"/>

           <Setter Property="c1:C1CarouselPanel.PerspectiveFactor" Value="-
       0.317"/>

               </Style>
    ```

12. Add the following **ListBox** control within the `<Grid>`:

    ```
       <ListBox Background="Transparent" Name="carouselListBox"
       Grid.Row="1"

            ItemsPanel="{StaticResource carouselPanelTemplate}"

            ItemTemplate="{StaticResource carouselItemTemplate}"

            Style="{StaticResource parabolaPanelStyle}"/>
    ```

13. Right-click the **MainPage.xaml** page and select **View Code** from the list.

14. Add the following namespaces to the top of the page:

    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`

- `C1.Silverlight.Carousel.dll`

15. Insert the following method directly below the **InitializeComponent()** method:

- Visual Basic

```
InitData()
```

- C#

```
InitData();
```

16. Call the following method to populate the C1CarouselPanel with data:

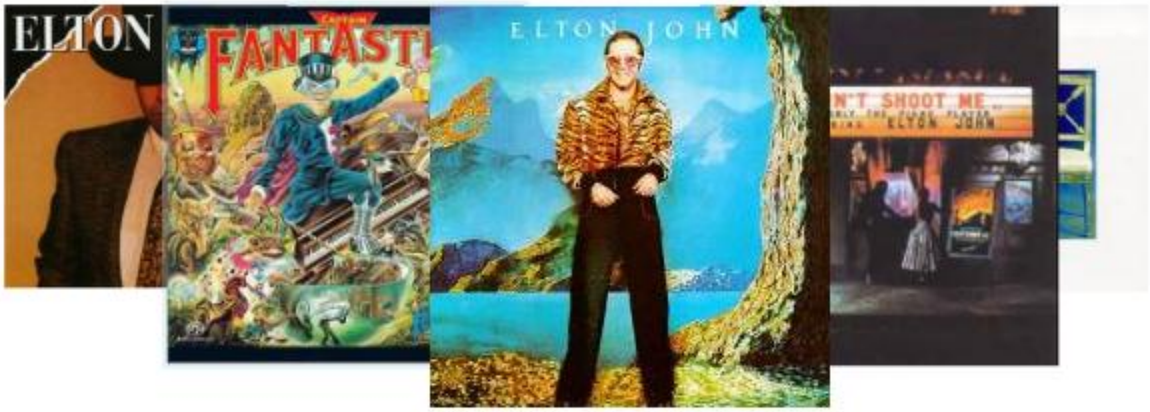
- Visual Basic

```
Private Sub InitData()
   For i As Integer = 101 To 140
         carouselListBox.Items.Add(New
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
".jpg")))
   Next
End Sub
```

- C#

```
private void InitData()
      {
          for (int i = 101; i <= 140; ++i)
          {
              carouselListBox.Items.Add(new
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
".jpg")));
          }
```

Press F5 to run your application. The carousel should resemble the following image:



Moving Content Along a Spiral Path

Follow these steps:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
   - C1.Silverlight.dll
   - C1.Silverlight.Carousel.dll
4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

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

8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```

10. Insert a `<DataTemplate>` to define the image source:

    ```
    <DataTemplate x:Key="carouselItemTemplate">

                <Image Source="{Binding}" Stretch="None" />

            </DataTemplate>
    ```

11. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

```
<Style x:Key="spiralPanelStyle" TargetType="ListBox">
    <Setter Property="c1:C1CarouselPanel.PathGeometry"        Value="
F1 M 426.39877,214.56664
C 477.32999,252.5027 425.16229,295.34012 402.57449,303.93842
C 385.32977,310.50281 302.99234,311.50055 289.99231,310.50055
C 277.30948,309.52493 190.99216,292.50021 178.99214,286.50021
C 171.53923,282.77377 143.14252,246.34085 140.99213,237.50035
C 131.99211,200.50027 144.99214,178.50027 191.99223,149.5002
C 200.50449,144.24796 259.86719,123.92526 276.99231,120.50028
C 291.99237,117.50031 363.99265,118.50018 382.01099,122.78677
C 402.75034,127.7207 436.07599,133.93291 457.07605,144.93292
C 474.27582,153.94232 489.89432,159.93607 502.89435,169.93608
C 513.04492,177.74423 542.43744,201.83525 552.03833,213.18535
C 564.9931,228.50035 568.22015,248.1823 573.71283,266.7619
C 576.95355,277.72397 559.99274,316.50018 546.99274,333.50021
C 533.24768,351.47452 473.77701,392.49255 450.77695,402.49255
C 424.47693,413.92734 387.50757,418.44592 358.99243,421.50061
C 330.99237,424.50009 300.22705,418.91339 272.99228,416.50018
C 247.2047,414.21521 219.99205,409.50064 193.94289,403.37714
C 177.01965,399.39893 137.99185,389.50015 123.99122,379.50012
C 103.3692,364.77072 64.991714,336.50052 56.991692,318.50049
C 46.535854,294.97488 40.107754,267.91367 35.991642,242.50037
C 32.747829,222.47273 49.97628,161.97603 61.236015,142.52779
C 72.236038,123.52813 102.94307,100.53529 122.99121,85.49958
C 146.99126,67.500015 189.99133,51.499714 217.99138,42.499702
C 243.25951,34.377808 297.45969,22.180216 323.99234,21.49992
C 401.99304,19.499987 423.3172,21.685627 497.32446,56.185596
C 532.84088,72.742287 591.66083,116.8151 614.9928,144.50041"/>
    <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
Value="Left"/>
    <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
Value="Top"/>
    <Setter Property="c1:C1CarouselPanel.PerspectiveAngle" Value="90"/>
    <Setter Property="c1:C1CarouselPanel.PerspectiveFactor" Value="-
0.317"/>
        </Style>
```

12. Add the following **ListBox** control within the `<Grid>`:

```
<ListBox Background="Transparent" Name="carouselListBox"
Grid.Row="1"
```

```
        ItemsPanel="{StaticResource carouselPanelTemplate}"
        ItemTemplate="{StaticResource carouselItemTemplate}"
        Style="{StaticResource sprialPanelStyle}"/>
```

13. Right-click the **MainPage.xaml** page and select **View Code** from the list.
14. Add the following namespaces to the top of the page:
    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`
    - `C1.Silverlight.Carousel.dll`
15. Insert the following method directly below the **InitializeComponent()** method:
    - Visual Basic

      ```
      InitData()
      ```

    - C#

      ```
      InitData();
      ```

16. Call the following method to populate the C1CarouselPanel with data:
    - Visual Basic

      ```
      Private Sub InitData()
         For i As Integer = 101 To 140
               carouselListBox.Items.Add(New
      BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
      ".jpg")))
         Next
      End Sub
      ```

    - C#

      ```
      private void InitData()
            {
                for (int i = 101; i <= 140; ++i)
                {
                    carouselListBox.Items.Add(new
      BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
      ".jpg")));
                }
      ```

Press F5 to run your application. The items in the C1Carousel control should move through a spiral.

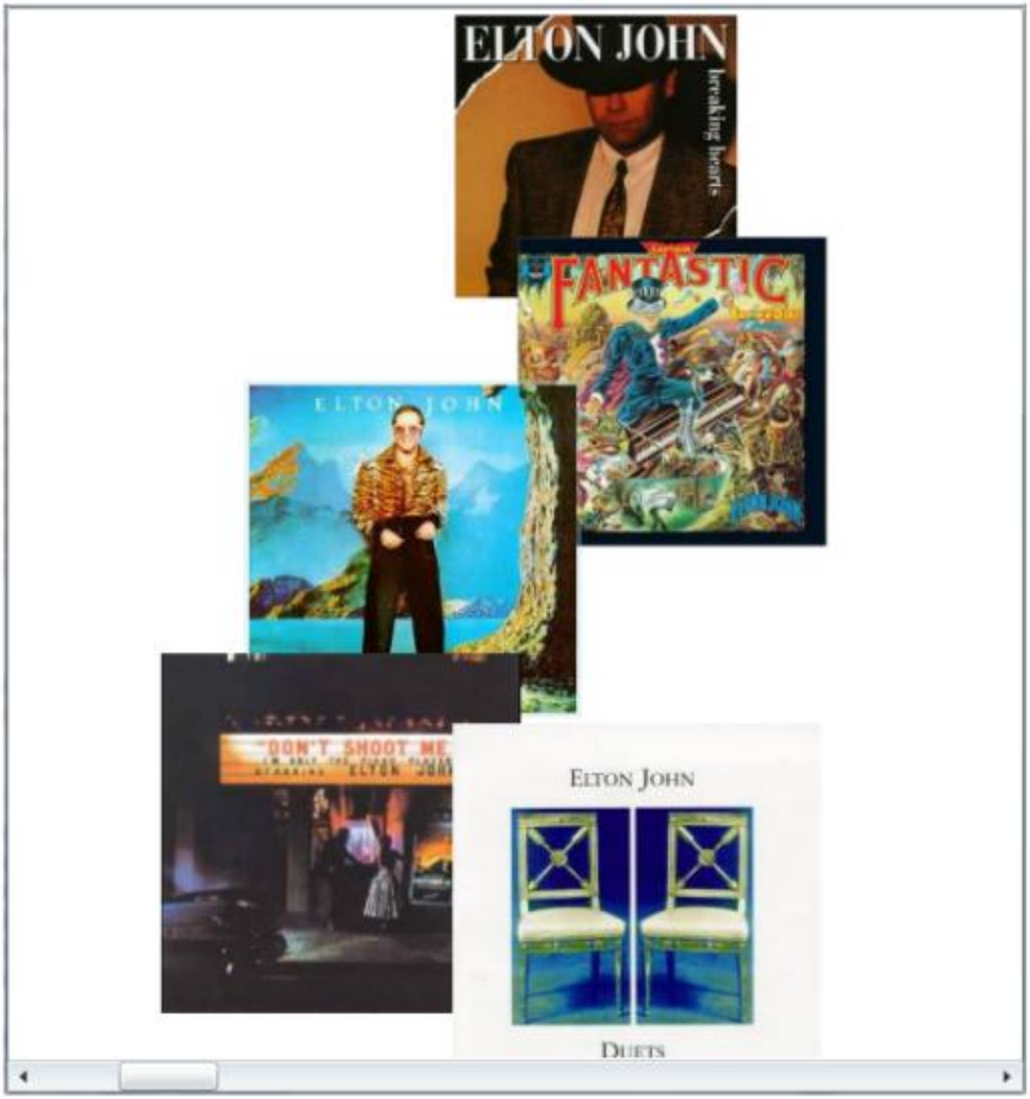
Using the ScrollViewer

You can allow users to scroll through C1Carousel content either by clicking or by using the ScrollViewer. This topic will walk you through setting the ScrollViewer to visible and setting some of its properties.

This Help topic assumes that you have completed one of the previous Adding and Moving Content topics and are using that project for this Help.

Follow these steps to set the ScrollViewer:

1. Select the C1CarouselPanel control in your application to open the Properties window.
2. Locate the CanHorizontallyScroll property in the list and make sure it is checked.
3. Locate the ScrollOwner property in the list and use the drop-down list to select ScrollViewer. This will open the ScrollOwner sub-properties.
4. Locate the HorizontalScrollBar property and use the drop-down list to set it to **Visible**. Scroll down to the VerticalScrollBar property and set that to **Disabled**.
5. Press F5 to run your application. It should appear as in the following image:



You will be able to scroll through the content using the scrollbar at the bottom of the page.

Limiting the Number of Visible Elements

You can customize the number of visible elements in the C1CarouselPanel by setting properties in the `<c1:C1CarouselPanel>` tag in the **ItemsPanelTemplate** .

In the C1CarouselPanel Tag

To limit the number of visible elements in the C1CarouselPanel follow these steps:

1. From the Visual Studio **File** menu select **New** and choose **Project**.
2. In the **New Project** dialog box choose a language in the left-side menu, choose **.NET Framework 4** in the **Framework** drop-down list, and enter a name for the project.
3. In the Solution Explorer, right-click the project name and choose **Add Reference**. In the **Add Reference** dialog box, locate and select the following assemblies and click **OK** to add references to your project:
   - C1.Silverlight.dll
   - C1.Silverlight.Carousel.dll
4. Right-click on the **yourprojectname .Web** project and select **Add | Folder** from the list. Name the folder **Resources**.
5. Right-click on the **Resources** folder and select **Add | Folder** from the list. Name the new sub-folder **covers**.
6. Right-click on the **covers** folder and select **Add | Existing Item** from the list. For this Help, locate the covers that are included with the **C1Carousel_Demo** sample application. Select all the files by pressing Shift and then clicking the first and last covers. Click **Add** to add the files to your folder.
7. Add the XAML namespace to the UserControl tag with the following markup: `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

   The namespaces will now appear similar to the following:

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

8. In the XAML view add a `<Resources>` tag just under the `<UserControl>` tag and above the `<Grid>` tag with the following markup:

   ```
   <UserControl.Resources>

   </UserControl.Resources>
   ```

   You will add templates within this tag.

9. Add an **ItemsPanelTemplate** within the **Resources** tag to define the C1CarouselPanel:

   ```
   <ItemsPanelTemplate x:Key="carouselPanelTemplate">

       <c1:C1CarouselPanel Padding="0, 10, 50, 50"
   VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
   VerticalItemAnchorOnPath="Center"/>

   </ItemsPanelTemplate>
   ```

10. Locate the `<c1:C1CarouselPanel>` tag and add the following XAML markup to limit the number of visible elements:

```
PageSize="3"
```

The total `<c1:C1CarouselPanel>` markup should resemble the following:

```
<c1:C1CarouselPanel Padding="0, 10, 50, 50"
VerticalPathAlignment="Center" HorizontalItemAnchorOnPath="Center"
VerticalItemAnchorOnPath="Center" PageSize="3"/>
```

11. Insert a `<DataTemplate>` to define the image source:

```
<DataTemplate x:Key="carouselItemTemplate">

            <Image Source="{Binding}" Stretch="None" />

        </DataTemplate>
```

12. Add the following `<Style>` under the `<DataTemplate>` to set the PathGeometry for the C1Carousel control:

```
<Style x:Key="diagonalPanelStyle" TargetType="ListBox">

  <Setter Property="c1:C1CarouselPanel.PathGeometry"         Value="M
216, 60 L 600, 390"/>

  <Setter Property="c1:C1CarouselPanel.HorizontalPathAlignment"
Value="Left"/>

  <Setter Property="c1:C1CarouselPanel.VerticalPathAlignment"
Value="Top"/>

  <Setter Property="c1:C1CarouselPanel.PerspectiveAngle" Value="65"/>

  <Setter Property="c1:C1CarouselPanel.PerspectiveFactor"
Value="0.12"/>

</Style>
```

13. Add the following **ListBox** control within the `<Grid>`:

```
<ListBox Background="Transparent" Name="carouselListBox"
Grid.Row="1"

     ItemsPanel="{StaticResource carouselPanelTemplate}"

     ItemTemplate="{StaticResource carouselItemTemplate}"

     Style="{StaticResource diagonalPanelStyle}"/>
```

14. Right-click the **MainPage.xaml** page and select **View Code** from the list.

15. Add the following namespaces to the top of the page:

    - `System.Windows.Media.Imaging.dll`
    - `C1.Silverlight.dll`
    - `C1.Silverlight.Carousel.dll`

16. Insert the following method directly below the **InitializeComponent()** method:

    - Visual Basic

    ```
    InitData()
    ```

    - C#

    ```
    InitData();
    ```

17. Call the following method to populate the C1CarouselPanel with data:

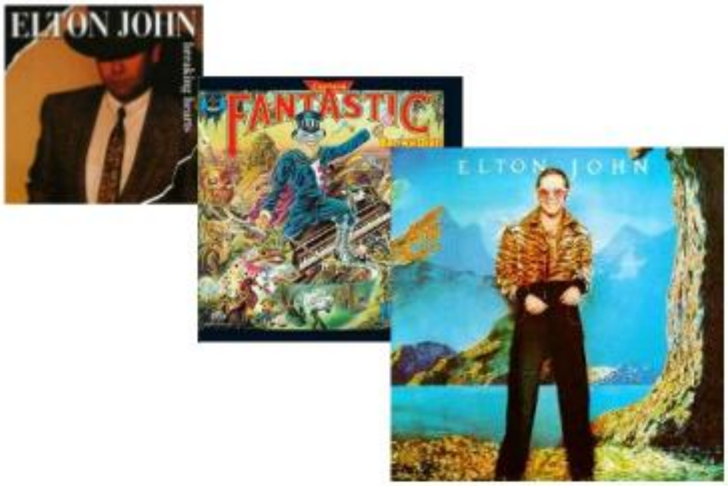
    - Visual Basic

```
Private Sub InitData()
   For i As Integer = 101 To 140
         carouselListBox.Items.Add(New
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" & i &
".jpg")))
   Next
End Sub
```

- C#

```
private void InitData()
      {
          for (int i = 101; i <= 140; ++i)
          {
              carouselListBox.Items.Add(new
BitmapImage(Extensions.GetAbsoluteUri("Resources/covers/cover" + i +
".jpg")));
          }
```

Press F5 to run your application. The carousel should resemble the following image:


ELTON JOHN
FANTASTIC
ELTON JOHN