ComponentOne

OrgChart for WPF and Silverlight

ComponentOne, a division of GrapeCity
201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206 USA

Website: http://www.componentone.com
Sales: 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 media on which the software is delivered is 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 media 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 the defective media by sending it and a check for $25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original media on which the software is delivered is 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. ComponentOne is 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.

Table of Contents

OrgChart for WPF and Silverlight Overview 5

Help with WPF and Silverlight Edition 5

Key Features 5

Quick Start : OrgChart for WPF and Silverlight 6

Step 1 of 3: Creating the C1OrgChart Application 6

Step 2 of 3: Adding Content to the C1OrgChart Control 10

Step 3 of 3: Running the C1OrgChart Application 17

Working with C1OrgChart 17

C1OrgChart Elements 18

C1OrgChart Core Properties 18

Using Bindings in C1OrgChart Properties 20

Advanced Binding Scenarios 22

C1OrgChart Layout and Appearance 25

Layout in a Panel 26

C1OrgChart Appearance Properties 26

Orientation 27

FlowDirection 27

ChildSpacing 27

Connector 28

Alignment 29

C1OrgChart Templates 31

C1OrgChart Visual States 31

C1OrgChart Task-Based Help 31

Adding C1OrgChart to the Application 32

Changing C1OrgChart Orientation 33

Changing C1OrgChart Flow Direction 34

Customizing the C1OrgChart Item Connector 34

Expanding and Collapsing C1OrgChart Nodes 40

Using a Hierarchical Data Template 53

API Reference 59

C1.Silverlight.OrgChart.5 Assembly 59

Namespaces 59

C1.Silverlight.OrgChart Namespace 59

Overview 59

Classes..60

C1OrgChart...60

Overview 61

Members 61

C1OrgChart Constructor 69

Methods 69

OnApplyTemplate Method 71

Properties..71

ChildItemsPath Property 76

ChildNodes Property 76

ChildSpacing Property 77

ConnectorDashArray Property 77

ConnectorDashCap Property 78

ConnectorDashOffset Property 78

ConnectorEndLineCap Property..79

ConnectorStartLineCap Property 80

ConnectorStroke Property 80

ConnectorThickness Property 81

Header Property...81

IsCollapsed Property 82

ItemTemplateSelector Property 82

Orientation Property...83

Fields 84

ChildItemsPathProperty Field 85

ChildSpacingProperty Field 85

ConnectorDashArrayProperty Field 86

ConnectorDashCapProperty Field...86

ConnectorDashOffsetProperty Field 87

ConnectorEndLineCapProperty Field 87

ConnectorStartLineCapProperty Field 88

ConnectorStrokeProperty Field 89

ConnectorThicknessProperty Field 89

HeaderProperty Field...90

IsCollapsedProperty Field 90

ItemTemplateSelectorProperty Field..91

OrientationProperty Field 91

DataTemplateSelector 92

Overview 93

Members 93

DataTemplateSelector Constructor 94

Methods 94

SelectTemplate Method 95

HierarchicalDataTemplate 95

Overview 96

Members 97

HierarchicalDataTemplate Constructor 98

Properties 99

ItemsSource Property 100

ItemTemplate Property 100

C1.WPF.OrgChart.4 Assembly 101

Namespaces 101

C1.WPF.OrgChart Namespace 101

Overview 101

Classes 101

C1OrgChart 101

Overview 102

Members 103

C1OrgChart Constructor 120

Methods 121

OnApplyTemplate Method 124

Properties 125

ChildItemsPath Property 132

ChildNodes Property 132

ChildSpacing Property 133

ConnectorDashArray Property 133

ConnectorDashCap Property 134

ConnectorDashOffset Property 134

ConnectorEndLineCap Property 135

ConnectorStartLineCap Property 136

ConnectorStroke Property 136

ConnectorThickness Property 137

Header Property 137
IsCollapsed Property 138
Orientation Property 138
Fields 139
ChildItemsPathProperty Field 140
ChildSpacingProperty Field 141
ConnectorDashArrayProperty Field 141
ConnectorDashCapProperty Field 142
ConnectorDashOffsetProperty Field 142
ConnectorEndLineCapProperty Field 143
ConnectorStartLineCapProperty Field 143
ConnectorStrokeProperty Field 144
ConnectorThicknessProperty Field 144
HeaderProperty Field 145
IsCollapsedProperty Field 145
OrientationProperty Field 146
SmartAssembly.Attributes Namespace 147
Overview 147
Classes 147
PoweredByAttribute 147
Overview 148
Members 148
PoweredByAttribute Constructor 149

OrgChart for WPF and Silverlight Overview

Create hierarchical diagrams that show the structure and relationships of your data. **OrgChart for WPF and Silverlight** leverages the rich data binding mechanisms of the platform to provide a flexible, yet easy-to-use control.

See Also

Help with WPF and Silverlight Edition

Help with WPF and Silverlight Edition

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

Key Features

OrgChart for WPF and Silverlight allows you to create customized, rich applications. Make the most of **C1OrgChart** by taking advantage of the following key features:

- **Flexible Data Binding**
  The C1OrgChart control is an **ItemsControl**. Bind it to a single entity containing sub-items or an **IEnumerable** collection of items which each can contain sub-items.
- **Versatile Orientation and Flow**
  C1OrgChart can display items flowing horizontally or vertically in either direction. You can determine the flow of the chart by simply setting the Orientation and **FlowDirection** properties on the control.
- **Collapsible Nodes**
  Allow the user to hide an item's branches to create a more compact display. The C1OrgChart nodes have an IsCollapsed property that allows you to collapse or expand each node, similar to a **TreeView**.
- **Connector Line Customization**
  The C1OrgChart control exposes several properties that allow you to customize the lines used to connect nodes. These properties allow you to customize the brush, thickness and dash array used to create connector lines. You can even bind these properties to properties on the data item to customize the lines per relationship.
- **Child Spacing and Alignment Options**
  Customize the alignment and spacing of items in your **OrgChart** by simply setting a few properties. The control includes a ChildSpacing property that controls the

separation between items (in pixels), as well as horizontal and vertical alignment properties which can give dramatic changes to the visualization.

- **Complex Hierarchical Displays**
  In addition to the flexibility provided by data templates and bindings, the C1OrgChart control also supports advanced binding scenarios to create complex hierarchical displays. Use different templates for certain nodes based upon properties of specific data items like conditional formatting. For example, you could use different templates to visually distinguish among directors, managers, and clerical staff in an employee organization chart.

Quick Start : OrgChart for WPF and Silverlight

The following quick start guide is intended to get you up and running with **OrgChart for WPF and Silverlight**. In this quick start you will create a simple project using a C1OrgChart control. You'll create a new WPF or Silverlight application, add the C1OrgChart control to your application, add random data that will be displayed in the C1OrgChart control, and observe some of the run-time interactions possible with **OrgChart for WPF and Silverlight**.

See Also

Step 1 of 3: Creating the C1OrgChart Application

Step 2 of 3: Adding Content to the C1OrgChart Control

Step 3 of 3: Running the C1OrgChart Application

Step 1 of 3: Creating the C1OrgChart Application

In this step you'll create a WPF or Silverlight application using **OrgChart for WPF and Silverlight**. **C1OrgChart** allows you to create hierarchical diagrams that show the structure and relationships of your data. To set up your project and add a C1OrgChart control to your application, complete the following steps:

1. Create a new WPF or Silverlight project in Visual Studio. 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**. In the **Add Reference** dialog box, locate and select the **C1.WPF** and **C1.WPF.OrgChart** or **C1.Silverlight** and **C1.Silverlight.OrgChart** assemblies and click **OK** to add references to your project.
3. Open the XAML view of the MainWindow.xaml or MainPage.xaml file; in this quick start you'll add templates and controls using XAML markup.
4. Add the XAML namespace to the Window or UserControl tag with the following markup:

XAML	Copy Code

```
xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml" xmlns:ext="clr-
namespace:Util"
xmlns:local="clr-namespace:QuickStart"
```

5. Note that if you named your project something other than "QuickStart" you will have to update the last reference with your project name. The Window or UserControl tag will now appear similar to the following:

XAML	Copy Code

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
xmlns:ext="clr-namespace:QuickStart.Util" xmlns:local="clr-
namespace:QuickStart"
    Title="ComponentOne OrgChart for WPF" Height="275" Width="425">
```

6.

XAML	Copy Code

```
<UserControl x:Class="C1SilverlightCS111010.MainPage"
 xlns="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
 xmlns:ext="clr-namespace:Util" xmlns:local="clr-namespace:QuickStart"
 mc:Ignorable="d" d:DesignHeight="262" d:DesignWidth="399">
```

7. This is a unified namespace that will enable you to work with most of the controls in WPF and Silverlight Edition without adding multiple namespaces.
8. If creating a Silverlight application, add the following Resources just under the UserControl tag. The XAML appears similar to the following:

XAML	Copy Code

```
<UserControl.Resources>
        <!-- TemplateSelector: picks _tplDirector or _tlpOther -->
        <local:PersonTemplateSelector x:Key="_personTplSelector" />
        <!-- data template for Directors -->
        <DataTemplate x:Key="_tplDirector" >
            <Border
                Background="Gold" BorderBrush="Black"
                BorderThickness="2 2 4 4" CornerRadius="6"
                Margin="20" MaxWidth="200" >
                <StackPanel Orientation="Vertical" >
                    <Border CornerRadius="6 6 0 0" Background="Black" >
                        <StackPanel Orientation="Horizontal">
                            <Ellipse Width="12" Height="12" Fill="Gold"
Margin="4" />
                            <TextBlock Text="{Binding Name}"
FontWeight="Bold" FontSize="16" Foreground="Gold" />
                        </StackPanel>
                    </Border>
                    <TextBlock Text="{Binding Position}" Padding="6 0"
FontSize="14" FontStyle="Italic" HorizontalAlignment="Right" />
                </StackPanel>
            </Border>
        </DataTemplate>
        <!-- data template for everyone else -->
        <DataTemplate x:Key="_tplOther" >
            <Border
                Background="WhiteSmoke" BorderBrush="Black"
                BorderThickness="1 1 2 2" CornerRadius="6"
                MaxWidth="200" >
                <StackPanel Orientation="Vertical" >
                    <Border CornerRadius="6 6 0 0" Background="Black" >
                        <TextBlock Text="{Binding Name}" FontWeight="Bold"
FontSize="14" Foreground="WhiteSmoke" Padding="4 0 0 0" />
                    </Border>
                    <TextBlock Text="{Binding Notes}" Padding="6 0"
FontSize="9.5" TextWrapping="Wrap" />
                    <TextBlock Text="{Binding Position}" Padding="6 0"
FontSize="12" FontStyle="Italic" HorizontalAlignment="Right" />
                </StackPanel>
            </Border>
        </DataTemplate>
</UserControl.Resources>
```

9. Add the following row definition within the Grid tag:

XAML	Copy Code

```
<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition />
</Grid.RowDefinitions>
```

10. Add the following markup just below the row definition to add a title and a control panel that will enable you to change the display of the **C1OrgChart** control at run time:

XAML	Copy Code

```
<!-- sample title -->
        <StackPanel Orientation="Horizontal" >
            <TextBlock Text="C1OrgChart Quick Start" FontSize="16"
VerticalAlignment="Bottom" />
            <TextBlock Name="_tbTotal" VerticalAlignment="Bottom" />
        </StackPanel>
<!-- control panel -->
<StackPanel Orientation="Horizontal" VerticalAlignment="Top" Grid.Row="1" >
    <Button Content="New Data" Padding="8 0" Click="Button_Click" />
    <TextBlock Text="  Zoom: " VerticalAlignment="Center" />
    <Slider x:Name="_sliderZoom" VerticalAlignment="Center" Minimum=".01"
Maximum="1" Value="1" Width="200" />
</StackPanel>
<StackPanel Orientation="Horizontal" VerticalAlignment="Top" Grid.Row="2">
    <StackPanel Orientation="Horizontal" >
        <TextBlock Text="Orientation: " VerticalAlignment="Center" />
        <ComboBox Width="100" SelectedValue="{Binding ElementName=_orgChart,
Path=Orientation, Mode=TwoWay}" ItemsSource="{Binding
Source={ext:EnumerationExtension EnumType=Orientation}}" />
    </StackPanel>
    <StackPanel Orientation="Horizontal" >
        <TextBlock Text="  HorizontalContentAlignment: "
VerticalAlignment="Center" />
    <ComboBox Width="80" SelectedValue="{Binding ElementName=_orgChart,
Path=HorizontalContentAlignment, Mode=TwoWay}" ItemsSource="{Binding
Source={ext:EnumerationExtension EnumType=HorizontalAlignment}}" />
    </StackPanel>
    <StackPanel Orientation="Horizontal" >
        <TextBlock Text="  VerticalContentAlignment: "
VerticalAlignment="Center" />
        <ComboBox Width="80" SelectedValue="{Binding ElementName=_orgChart,
Path=VerticalContentAlignment, Mode=TwoWay}" ItemsSource="{Binding
Source={ext:EnumerationExtension EnumType=VerticalAlignment}}" />
    </StackPanel>
</StackPanel>
```

11. Add the following XAML markup just after the last **StackPanel** to add a **C1OrgChart** control within a **ScrollViewer** to the application:

XAML	Copy Code

```

<!-- org chart -->
<ScrollViewer Grid.Row="3" HorizontalScrollBarVisibility="Auto"
VerticalScrollBarVisibility="Auto" Padding="0">
 <c1:C1OrgChart x:Name="_orgChart" Grid.Row="1"
Orientation="Horizontal" ItemTemplateSelector="{StaticResource
_personTplSelector}" ConnectorStroke="Black" ConnectorThickness="2" >
<!-- scale transform bound to slider -->
    <c1:C1OrgChart.RenderTransform>
        <ScaleTransform ScaleX="{Binding Value, ElementName=_sliderZoom}"
ScaleY="{Binding Value, ElementName=_sliderZoom}" />
    </c1:C1OrgChart.RenderTransform>
</c1:C1OrgChart>
</ScrollViewer>
```

12. This will add a C1OrgChart control named "_orgChart" to the application.

You've successfully set up your application's user interface, but the C1OrgChart control currently contains no content. In the next steps you'll add content to the C1OrgChart control, and then you'll observe some of the run-time interactions possible with the control.

Step 2 of 3: Adding Content to the C1OrgChart Control

%%scrap%%

" -->

For a list of all members of this type, see C1OutlookBar members.

Public Events

	Name	Description
⚡	ContextMenuClosing	(Inherited from System.Windows.FrameworkElement)
⚡	ContextMenuOpening	(Inherited from System.Windows.FrameworkElement)
⚡	DataContextChanged	(Inherited from System.Windows.FrameworkElement)
⚡	DragEnter	(Inherited from System.Windows.UIElement)
⚡	DragLeave	(Inherited from System.Windows.UIElement)
⚡	DragOver	(Inherited from System.Windows.UIElement)
⚡	Drop	(Inherited from System.Windows.UIElement)
⚡	FocusableChanged	(Inherited from System.Windows.UIElement)
⚡	GiveFeedback	(Inherited from System.Windows.UIElement)
⚡	GotFocus	(Inherited from System.Windows.UIElement)
⚡	GotKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	GotMouseCapture	(Inherited from System.Windows.UIElement)
⚡	GotStylusCapture	(Inherited from System.Windows.UIElement)
⚡	GotTouchCapture	(Inherited from System.Windows.UIElement)
⚡	Initialized	(Inherited from System.Windows.FrameworkElement)
⚡	IsEnabledChanged	(Inherited from System.Windows.UIElement)
⚡	IsHitTestVisibleChanged	(Inherited from System.Windows.UIElement)
⚡	IsKeyboardFocusedChanged	(Inherited from System.Windows.UIElement)

⚡	IsKeyboardFocusWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseCapturedChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseCaptureWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseDirectlyOverChanged	(Inherited from System.Windows.UIElement)
⚡	IsStylusCapturedChanged	(Inherited from System.Windows.UIElement)
⚡	IsStylusCaptureWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsStylusDirectlyOverChanged	(Inherited from System.Windows.UIElement)
⚡	IsVisibleChanged	(Inherited from System.Windows.UIElement)
⚡	KeyDown	(Inherited from System.Windows.UIElement)
⚡	KeyUp	(Inherited from System.Windows.UIElement)
⚡	LayoutUpdated	(Inherited from System.Windows.UIElement)
⚡	Loaded	(Inherited from System.Windows.FrameworkElement)
⚡	LostFocus	(Inherited from System.Windows.UIElement)
⚡	LostKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	LostMouseCapture	(Inherited from System.Windows.UIElement)
⚡	LostStylusCapture	(Inherited from System.Windows.UIElement)
⚡	LostTouchCapture	(Inherited from System.Windows.UIElement)
⚡	ManipulationBoundaryFeedback	(Inherited from System.Windows.UIElement)
⚡	ManipulationCompleted	(Inherited from System.Windows.UIElement)
⚡	ManipulationDelta	(Inherited from System.Windows.UIElement)

⚡	ManipulationInertiaStarting	(Inherited from System.Windows.UIElement)
⚡	ManipulationStarted	(Inherited from System.Windows.UIElement)
⚡	ManipulationStarting	(Inherited from System.Windows.UIElement)
⚡	MouseDoubleClick	(Inherited from System.Windows.Controls.Control)
⚡	MouseDown	(Inherited from System.Windows.UIElement)
⚡	MouseEnter	(Inherited from System.Windows.UIElement)
⚡	MouseLeave	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseMove	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseUp	(Inherited from System.Windows.UIElement)
⚡	MouseWheel	(Inherited from System.Windows.UIElement)
⚡	PreviewDragEnter	(Inherited from System.Windows.UIElement)
⚡	PreviewDragLeave	(Inherited from System.Windows.UIElement)
⚡	PreviewDragOver	(Inherited from System.Windows.UIElement)
⚡	PreviewDrop	(Inherited from System.Windows.UIElement)
⚡	PreviewGiveFeedback	(Inherited from System.Windows.UIElement)
⚡	PreviewGotKeyboardFocus	(Inherited from System.Windows.UIElement)

⚡	PreviewKeyDown	(Inherited from System.Windows.UIElement)
⚡	PreviewKeyUp	(Inherited from System.Windows.UIElement)
⚡	PreviewLostKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseDoubleClick	(Inherited from System.Windows.Controls.Control)
⚡	PreviewMouseDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseLeftButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseLeftButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseMove	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseRightButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseRightButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseWheel	(Inherited from System.Windows.UIElement)
⚡	PreviewQueryContinueDrag	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusDown	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusInAirMove	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusInRange	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusMove	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusOutOfRange	(Inherited from System.Windows.UIElement)

⚡	PreviewStylusSystemGesture	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusUp	(Inherited from System.Windows.UIElement)
⚡	PreviewTextInput	(Inherited from System.Windows.UIElement)
⚡	PreviewTouchDown	(Inherited from System.Windows.UIElement)
⚡	PreviewTouchMove	(Inherited from System.Windows.UIElement)
⚡	PreviewTouchUp	(Inherited from System.Windows.UIElement)
⚡	QueryContinueDrag	(Inherited from System.Windows.UIElement)
⚡	QueryCursor	(Inherited from System.Windows.UIElement)
⚡	RequestBringIntoView	(Inherited from System.Windows.FrameworkElement)
⚡	SelectedIndexChanged	Event raised when the **SelectedIndex** property has changed.
⚡	SelectedItemChanged	Event raised when the **SelectedItem** property has changed.
⚡	SizeChanged	(Inherited from System.Windows.FrameworkElement)
⚡	SourceUpdated	(Inherited from System.Windows.FrameworkElement)
⚡	StylusButtonDown	(Inherited from System.Windows.UIElement)
⚡	StylusButtonUp	(Inherited from System.Windows.UIElement)
⚡	StylusDown	(Inherited from System.Windows.UIElement)
⚡	StylusEnter	(Inherited from System.Windows.UIElement)
⚡	StylusInAirMove	(Inherited from System.Windows.UIElement)
⚡	StylusInRange	(Inherited from System.Windows.UIElement)

⚡	StylusLeave	(Inherited from System.Windows.UIElement)
⚡	StylusMove	(Inherited from System.Windows.UIElement)
⚡	StylusOutOfRange	(Inherited from System.Windows.UIElement)
⚡	StylusSystemGesture	(Inherited from System.Windows.UIElement)
⚡	StylusUp	(Inherited from System.Windows.UIElement)
⚡	TargetUpdated	(Inherited from System.Windows.FrameworkElement)
⚡	TextInput	(Inherited from System.Windows.UIElement)
⚡	ToolTipClosing	(Inherited from System.Windows.FrameworkElement)
⚡	ToolTipOpening	(Inherited from System.Windows.FrameworkElement)
⚡	TouchDown	(Inherited from System.Windows.UIElement)
⚡	TouchEnter	(Inherited from System.Windows.UIElement)
⚡	TouchLeave	(Inherited from System.Windows.UIElement)
⚡	TouchMove	(Inherited from System.Windows.UIElement)
⚡	TouchUp	(Inherited from System.Windows.UIElement)
⚡	Unloaded	(Inherited from System.Windows.FrameworkElement)

Top

See Also

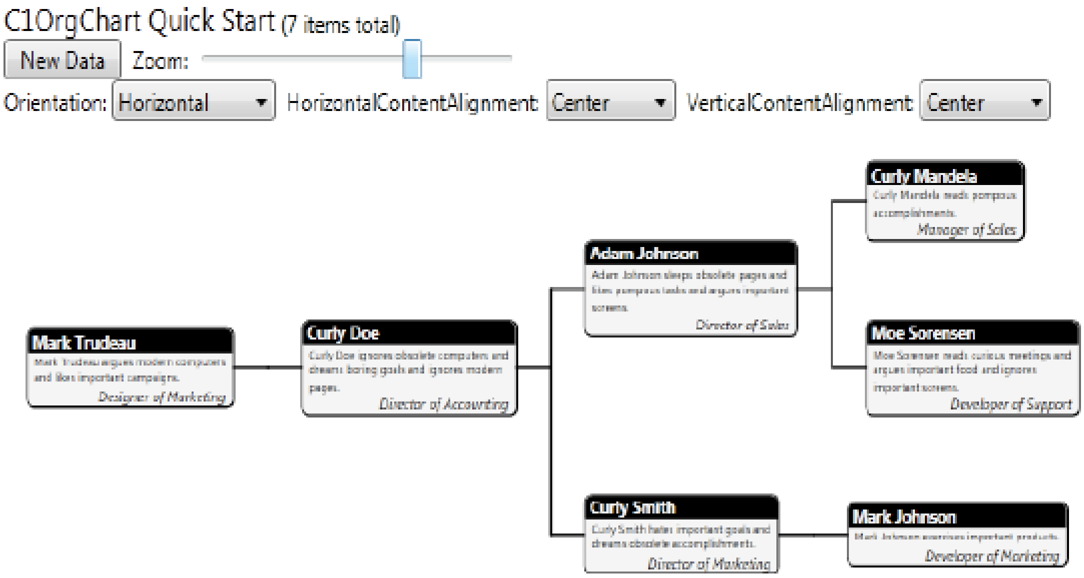
Reference

C1OutlookBar Class
C1.WPF.OutlookBar Namespace

Step 3 of 3: Running the C1OrgChart Application

Now that you've created a WPF or Silverlight application and added content to the C1OrgChart control, the only thing left to do is run your application. To run your application and observe **C1OrgChart**'s run-time behavior, 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:



C1OrgChart Quick Start (7 items total)

Orientation: Horizontal HorizontalContentAlignment Center VerticalContentAlignment Center

Mark Trudeau
Mark Trudeau argues modern computers and likes important campaigns.
Designer of Marketing

Curly Doe
Curly Doe ignores obsolete computers and dreams boring goals and ignores modern pages.
Director of Accounting

Adam Johnson
Adam Johnson sleeps obsolete pages and likes pompous tasks and argues important screens.
Director of Sales

Curly Mandela
Curly Mandela reads pompous accomplishments.
Manager of Sales

Moe Sorensen
Moe Sorensen reads curious meetings and argues important food and ignores important screens.
Developer of Support

Curly Smith
Curly Smith hates important goals and dreams obsolete accomplishments.
Director of Marketing

Mark Johnson
Mark Johnson exercises important products.
Developer of Marketing

The application contains a control panel area and a random organizational chart.

2. Drag the slider's thumb button to set the zoom level of the **C1OrgChart** control.
3. Change the orientation of the control by clicking the **Organization** drop-down box and selecting **Vertical**. The grid will appear vertically orientated.
4. Set the control's alignment by selecting options from the **HorizontalContentAlignment** and **VerticalContentAlignment** drop-down boxes.
5. Click the **New data** button to load the control with different data.

Congratulations! You've completed the **C1OrgChart** quick start and created a simple WPF or Silverlight application, added and customized a **C1OrgChart** control, and viewed some of the run-time capabilities of the control.

Working with C1OrgChart

Organizational charts (often called organization chart, org chart, or organogram) are diagrams that show the structure of an organization and the relationships and relative ranks of its parts and positions/jobs. The term is also used for similar diagrams, for example ones showing the different elements of a field of knowledge or a group of languages.

The C1OrgChart control allows you to create organizational charts that show any type of hierarchical data. The control leverages the rich data binding mechanisms in WPF and Silverlight to provide a flexible, yet easy-to-use tool.

See Also

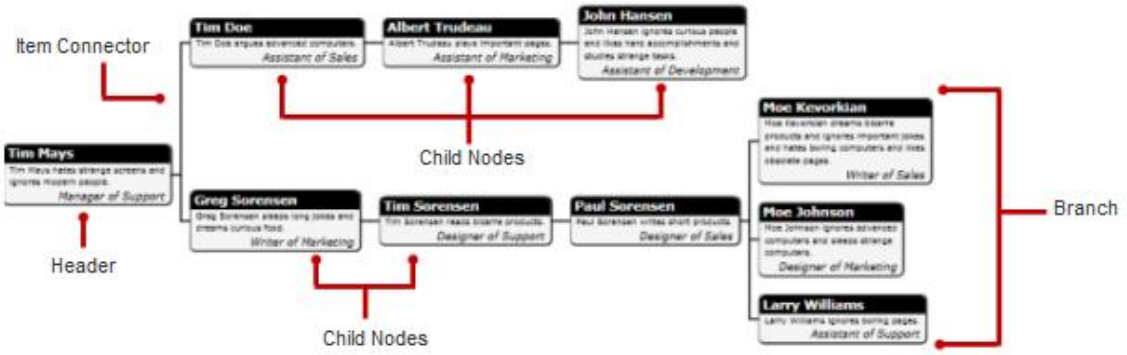
C1OrgChart Core Properties

Using Bindings in C1OrgChart Properties

Advanced Binding Scenarios

C1OrgChart Elements

The C1OrgChart control consists of several parts: the **Header**, the **ChildNodes**, and the **ItemConnector**. The image below identifies those parts. The image also identifies a Branch of the **C1OrgChart**:



C1OrgChart Core Properties

The C1OrgChart control is an **ItemsControl**. To use it, you will normally populate the control with the Header or **ItemsSource** properties and define the appearance of the items using the **ItemTemplate** property.

Use the **Header** property if you have a single data item that contains sub-items. Use the **ItemsSource** property if you have a collection of items with sub-items.

Either way, the data items must contain sub-items. In most cases, the sub-items are of the same type as the main items. For example, an **Employee** class may contain properties about the employee and a **Subordinates** property that contains a list of employees who report to the parent **Employee**:

C#	Copy Code

```
public class Employee
 {
   List<Employee> _list = new List<Employee>();
   public string Name { get; set; }
   public string Position { get; set; }
   public string Notes { get; set; }
   public IEnumerable<Employee> Subordinates{ get; set; }
 }
```

If you assign an **Employee** object to the **Header** property, the **C1OrgChart** will automatically detect that the **Subordinates** property contains a collection of **Employee** objects, and that is enough to establish the hierarchy of the data.

If your data class contains multiple collection properties, or if the collection is of a generic type (for example **IEnumerable**), then you should use the ChildItemsPath property to specify the name of the property that contains the child (subordinate) items.

If the items contain sub-items of different types, then you should use a **HierarchicalDataTemplate** to specify the items at each level. This is discussed later in this document.

The **ItemTemplate** property specifies how the **C1OrgChart** control should display the data items. This is a standard Silverlight/WPF **DataTemplate**, which you can define in XAML as follows:

XAML	Copy Code

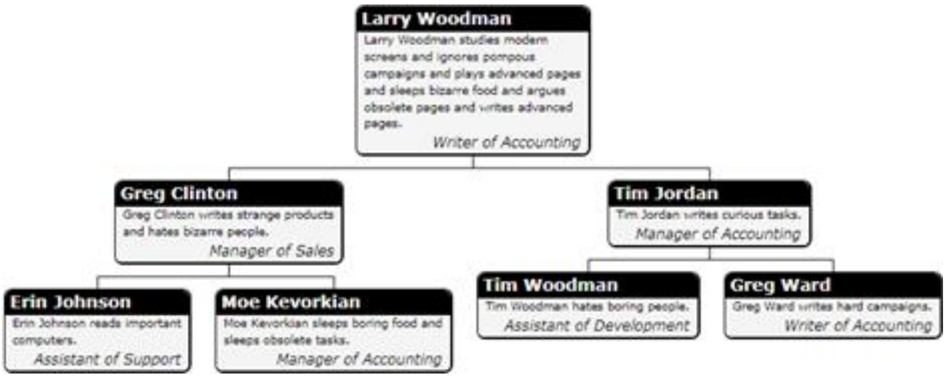
```
<Window.Resources>
   <!-- C1OrgChart node content -->
   <DataTemplate x:Key="EmployeeTemplate" >
     <Border
       Background="WhiteSmoke" BorderBrush="Black"
       BorderThickness="1 1 2 2" CornerRadius="6"
       MaxWidth="200" >
       <StackPanel Orientation="Vertical" >
         <TextBlock Text="{Binding Name}" FontSize="14" />
         <TextBlock Text="{Binding Notes}" FontSize="9.5" />
         <TextBlock Text="{Binding Position}" FontSize="12" />
       </StackPanel>
     </Border>
  </DataTemplate>
 </Window.Resources>
```

Once you have a **DataTemplate** defined as a resource, you can use it in a **C1OrgChart** control as follows:

XAML	Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```

To illustrate, the chart below was created with a slightly enhanced version of this template and some randomly-generated employees:


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

Using Bindings in C1OrgChart Properties

The **ItemTemplate** used in the examples above use bindings to show properties of the **Employee** class as visual elements. But you can also bind elements to properties of the C1OrgChart.

The most useful of these scenarios is binding a **CheckBox.IsChecked** property to the **C1OrgChart's IsCollapsed** property. This allows you to create collapsible **C1OrgCharts** that behave similarly to a **TreeView** control.

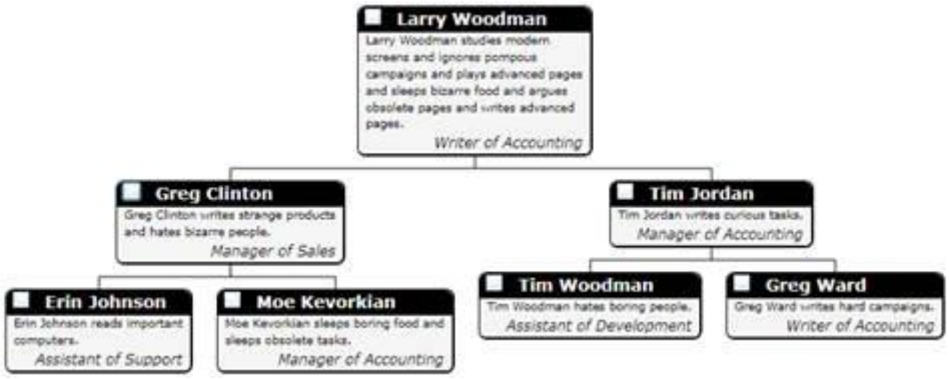
For example, here is a slightly modified version of the data template we assigned to the **C1OrgChart's ItemTemplate** property:

XAML	Copy Code

```
<Window.Resources>
   <!-- C1OrgChart node content -->
   <DataTemplate x:Key="EmployeeTemplate" >
     <Border
       Background="WhiteSmoke" BorderBrush="Black"
       BorderThickness="1 1 2 2" CornerRadius="6"
       MaxWidth="200" >
```

```
      <StackPanel Orientation="Vertical" >
        <!-- CheckBox bound to C1OrgChart's IsCollapsed property -->
        <CheckBox Margin="4 0"
          IsChecked="{Binding IsCollapsed, Mode=TwoWay,
          RelativeSource={RelativeSource AncestorType=c1:C1OrgChart}}"/>
       <TextBlock Text="{Binding Name}" FontSize="14" />
        <TextBlock Text="{Binding Notes}" FontSize="9.5" />
        <TextBlock Text="{Binding Position}" FontSize="12" />
      </StackPanel>
    </Border>
 </DataTemplate>
</Window.Resources
```

The effect of this change is shown below:



□ Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting

□ Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales

□ Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting

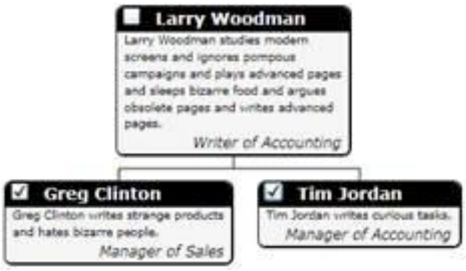
□ Erin Johnson
Erin Johnson reads important computers.
Assistant of Support

□ Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting

□ Tim Woodman
Tim Woodman hates boring people.
Assistant of Development

□ Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

Clicking the checkboxes collapses the branches, resulting in this more compact display:



□ Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting

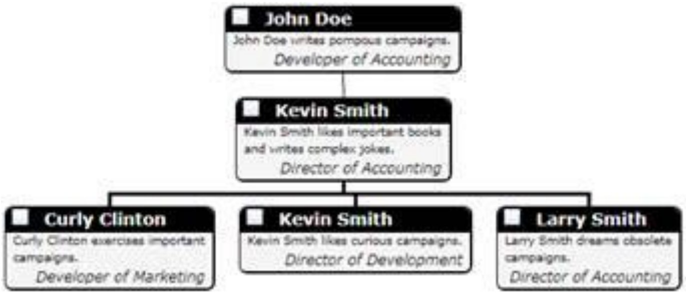
☑ Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales

☑ Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting

You can also use bindings to customize the connector lines. For example, the XAML below generates a chart where the thickness of the connector line corresponds to the number of child nodes:

XAML | Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   ConnectorThickness="{Binding Path=Subordinates.Count}"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```


John Doe
John Doe writes pompous campaigns.
Developer of Accounting
Kevin Smith
Kevin Smith likes important books and writes complex jokes.
Director of Accounting
Curly Clinton
Curly Clinton exercises important campaigns.
Developer of Marketing
Kevin Smith
Kevin Smith likes curious campaigns.
Director of Development
Larry Smith
Larry Smith dreams obsolete campaigns.
Director of Accounting

Since John Doe has only one direct subordinate, his connector line is one pixel thick. Kevin Smith has three direct subordinates, so his connector lines are tree pixels thick.

Advanced Binding Scenarios

In addition to the flexibility provided by data templates and bindings, the C1OrgChart control also supports advanced binding scenarios using two standard classes: **DataTemplateSelector** and **HierarchicalDataTemplate**.

DataTemplateSelector class: This class allows you to select different templates based on features of specific data items. For example, you could use different templates to display directors, managers, and clerical staff.

To do this, you would create a custom class that inherits from **DataTemplateSelector** and override the **SelectTemplate** object. You would then create an instance of this class and assign it to the **ItemTemplateSelector** property of the **C1OrgChart** control.

The example below shows a simple **DataTemplateSelector** implementation:

C#	Copy Code

```
/// <summary>
 /// Class used to dynamically select templates for items being created.
 /// </summary>
public class EmployeeTemplateSelector : DataTemplateSelector
 {
   public override DataTemplate SelectTemplate(object item, DependencyObject
ctnr)
   {
     var p = item as Employee;
     var e = Application.Current.RootVisual as FrameworkElement;
     return p.Position.IndexOf("Director") > -1
       ? e.Resources["DirectorTemplate"] as DataTemplate
       : e.Resources["EmployeeTemplate"] as DataTemplate;
   }
 }
```

Notice the following important points:

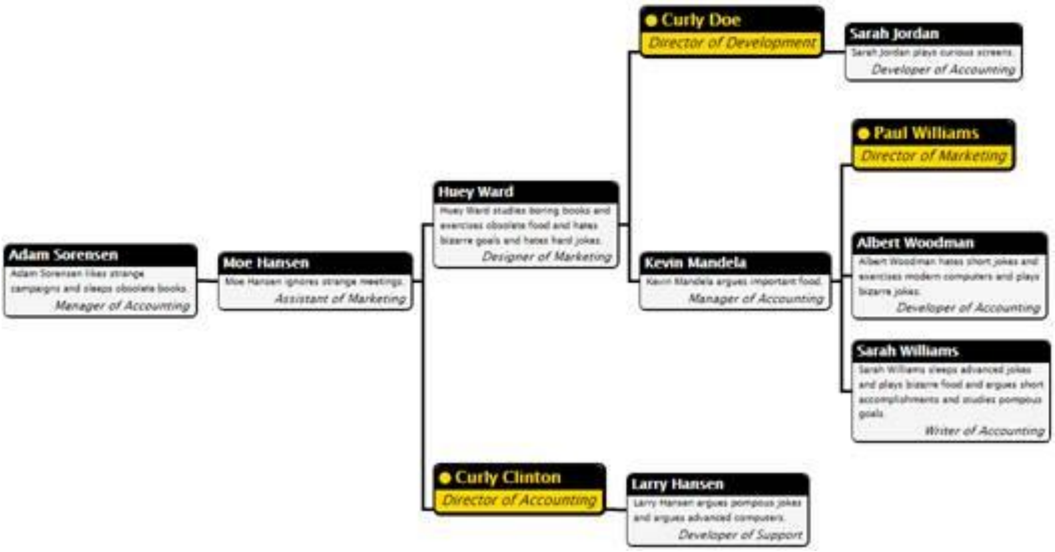
- In WPF, the **DataTemplateSelector** class is defined in the *System.Windows.Controls* namespace. In Silverlight, it is defined in the *C1.Silverlight.C1OrgChart* namespace.
- The **DataTemplateSelector** class is not specific to the **C1OrgChart** control. It is a standard WPF class, used with any **ItemsControl** object (for example, **ListBox.ItemTemplateSelector**).
- The code assumes that the “DirectorTemplate”and “EmployeeTemplate” data templates are defined as resources under the application’s root visual.

Once the custom **DataTemplateSelector** is available, you can use it in XAML as usual:

XAML	Copy Code

```
<Window.Resources>
   <!-- to pick templates based on employee position -->
   <local:PersonTemplateSelector x:Key="TemplateSelector" />
   <!-- data template for Directors -->
   <DataTemplate x:Key="DirectorTemplate" >
   …
  </DataTemplate>
   <!-- data template for other Employees -->
   <DataTemplate x:Key="EmployeeTemplate" >
   …
  </DataTemplate>
 </Window.Resources>
   …
  <c1:C1OrgChart
     ItemTemplateSelector="{StaticResource TemplateSelector}"
     ConnectorStroke="Black" ConnectorThickness="3" />
```

The image below shows the result:



Adam Sorensen
Adam Sorensen likes strange campaigns and sleeps obsolete books.
Manager of Accounting

Moe Hansen
Moe Hansen ignores strange meetings.
Assistant of Marketing

Huey Ward
Huey Ward studies boring books and exercises obsolete food and hates bizarre goals and hates hard jokes.
Designer of Marketing

● Curly Clinton
Director of Accounting

Larry Hansen
Larry Hansen argues pompous jokes and argues advanced computers.
Developer of Support

● Curly Doe
Director of Development

Sarah Jordan
Sarah Jordan plays curious streets.
Developer of Accounting

Kevin Mandela
Kevin Mandela argues important food.
Manager of Accounting

● Paul Williams
Director of Marketing

Albert Woodman
Albert Woodman hates short jokes and exercises modern computers and plays bizarre jokes.
Developer of Accounting

Sarah Williams
Sarah Williams sleeps advanced jokes and plays bizarre food and argues short accomplishments and studies pompous goals.
Writer of Accounting

The **ItemTemplateSelector** is useful when the data items are of the same type, but you want to display certain items differently based on the properties of specific data items.

HierarchicalDataTemplate class: This class allows you to bind the **C1OrgChart** control to items that contain items of different types. For example, you could create a chart that displays leagues, divisions within each league, and teams within each division.

To do this, you would create a **HierarchicalDataTemplate** for each of the classes with sub-items, and a regular data template for the last class in the hierarchy. In our example, you would create a **HierarchicalDataTemplate** for the leagues, another for the divisions, and finally a regular data template for the teams:

XAML	Copy Code

```
<Window.Resources>
   <!-- regular template for Team objects -->
   <DataTemplate x:Key="TeamTemplate" >
     <Border Background="LightBlue" Padding="4" >
       <TextBlock FontStyle="Italic" Text="{Binding Path=Name}" />
     </Border>
   </DataTemplate>
   <!-- hierarchical template for Division objects -->
   <c1:HierarchicalDataTemplate x:Key="DivisionTemplate"
     ItemsSource="{Binding Path=Teams}"
     ItemTemplate="{StaticResource TeamTemplate}">
    <Border Background="Gold" >
       <TextBlock Text="{Binding Path=Name}" FontWeight="Bold" Padding="20"
/>
     </Border>
   </c1:HierarchicalDataTemplate>
   <!-- hierarchical template for League objects -->
   <c1:HierarchicalDataTemplate x:Key="LeagueTemplate"
     ItemsSource="{Binding Path=Divisions}"
     ItemTemplate="{StaticResource DivisionTemplate}">
     <Border Background="LightCoral" >
       <TextBlock Text="{Binding Path=Name}" FontWeight="Bold" Padding="40"
/>
     </Border>
   </c1:HierarchicalDataTemplate>
 </Window.Resources>
```

The top-level template is the **LeagueTemplate**. In addition to defining how **League** object should be displayed (as regular templates do), its **ItemsSource** property specifies that subordinate objects should be retrieved from the **League.Divisions** property. Finally, the

ItemTemplate property specifies the template that should be used to display the subordinate objects.

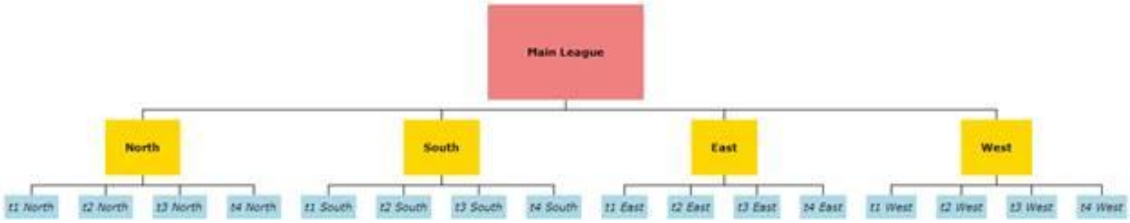
In this case, the **ItemTemplate** is **DivisionTemplate**, another **HierarchicalDataTemplate** that specifies how **Division** objects should be displayed, that the **Division.Teams** property exposes subordinate objects, and that the subordinate objects should be displayed using the **TeamTemplate**, which is a regular (non-hierarchical) data template.

Notice the following important points:

- In WPF, the **HierarchicalDataTemplate** class is defined in the *System.Windows* namespace. In Silverlight, it is defined in the *C1.Silverlight.C1OrgChart* namespace. (In Silverlight, you can also use the **HierarchicalDataTemplate** classes defined in the Microsoft SDK or in the C1.Silverlight.dll assembly).
- The **HierarchicalDataTemplate** class derives from the standard **DataTemplate** class and adds two properties: **ItemsSource** specifies the property that contains sub-items, and **ItemTemplate** specifies the template that should be used for the subordinate items.

Once the templates have been defined, using them is just a matter of setting the **ItemTemplate** property as usual:

```
<c1:C1OrgChart ItemTemplate="{StaticResourceLeagueTemplate}" />
```



Notice how the **C1OrgChart** uses the hierarchical data templates to navigate the hierarchy picking the right child collections and data templates. This is the same mechanism used with WPF **HeaderedItemControl** classes such as **Menu** and **TreeView**.

C1OrgChart Layout and Appearance

The following topics detail how to customize the C1OrgChart control's layout and appearance. You can use built-in layout options to lay your controls out in panels such as Grids or Canvases. Themes allow you to customize the appearance of the grid and take advantage of WPF's XAML-based styling. You can also use templates to format and lay out the control and to customize the control's actions.

See Also

Layout in a Panel

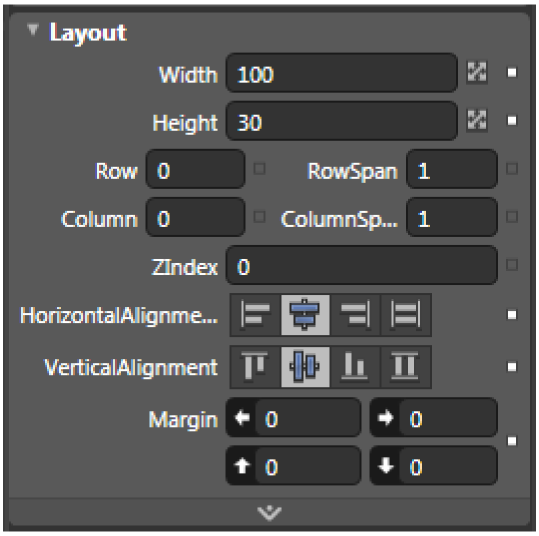
C1OrgChart Appearance Properties

C1OrgChart Templates

C1OrgChart Visual States

Layout in a Panel

You can easily lay out the C1OrgChart and other controls in your WPF application, using the attached layout properties. For example, you can lay out your control in a **Grid** panel with its **Row**, **ColumnSpan**, and **RowSpan** properties and in a **Canvas** panel with its **Left** and **Top** properties. For example, the C1OrgChart control includes the following **Layout** properties when located within a **Grid** panel:



Layout

Width 100

Height 30

Row 0

RowSpan 1

Column 0

ColumnSp... 1

ZIndex 0

HorizontalAlignme...

VerticalAlignment

Margin 0 0 0 0

You can change the sizing, alignment, and location of the C1OrgChart control within the **Grid** panel.

C1OrgChart Appearance Properties

The **ItemTemplate** property gives you complete flexibility to specify the appearance of the nodes in the **C1OrgChart**. The **C1OrgChart** exposes several properties that allow you to customize the appearance of the chart itself.

See Also

Orientation

FlowDirection

ChildSpacing

Connector

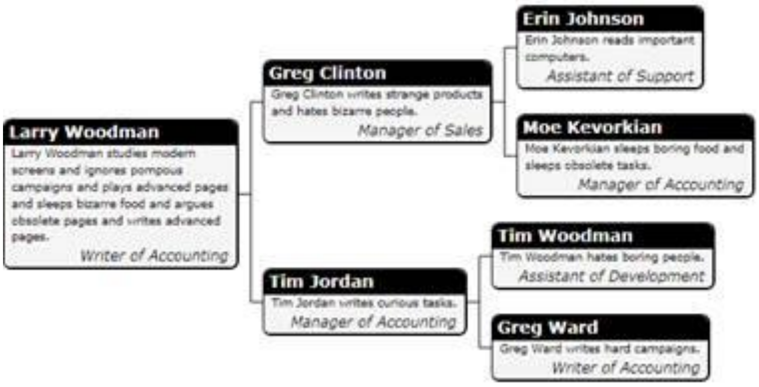
Alignment

Orientation

The Orientation property allows you to specify whether the chart should flow in the vertical or horizontal direction. By default a vertical **C1OrgChart** is displayed. Setting the **Orientation** property to **Horizontal** would have the following effect:

XAML	Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart" Orientation="Horizontal"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

FlowDirection

The **FlowDirection** property allows you to specify whether the chart should flow right to left or left to right. By default, the **C1OrgChart** flows left to right. You can change the **FlowDirection** property to **RightToLeft** with the following markup:

XAML	Copy Code

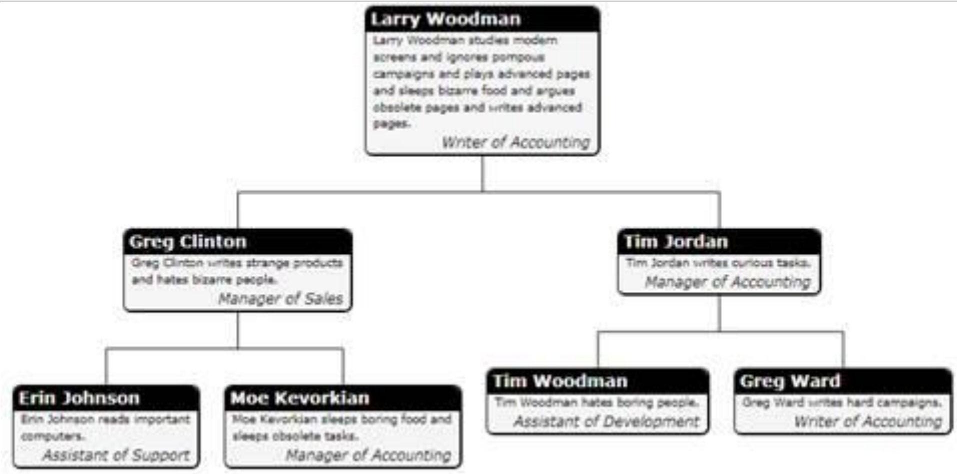
```
<c1:C1OrgChart
   x:Name="_orgChart" FlowDirection="RightToLeft"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```

ChildSpacing

ChildSpacing property: This property allows you to control the separation between items, in pixels. The default value is (20, 20), which spaces items by 20 pixels in the horizontal and vertical directions. For example, setting the **ChildSpacing** property to (20, 60) would have the following effect:

XAML	Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   ChildSpacing="20 60"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

Notice how the vertical spacing between items increased compared to the default setting.

You can also control the spacing between items by specifying a **Margin** value in your **ItemTemplate**. This approach is slightly more flexible because it allows you to specify the amount of space to be left on each side of the node elements.

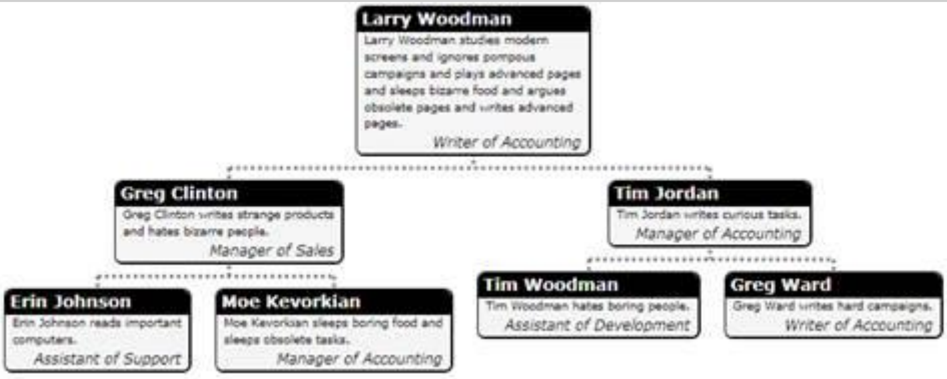
Connector

The **C1OrgChart** exposes several properties that allow you to customize the lines used to connect the nodes. These properties include: ConnectorStroke (specifies the **Brush** used to create the connectors), ConnectorThickness (the thickness of the lines), ConnectorDashArray (used to create dashed lines), and a few others. These properties are analogous to the ones in the **Line** element.

For example, if you wanted to connect the items using gray dotted lines you could use this XAML markup:

XAML	Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   ConnectorStroke="Gray"
   ConnectorThickness="3"
   ConnectorDashArray="1 1"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

The **ConnectorDashArray** property uses a collection of **double** values that specify the widths of dashes and blank spaces, expressed in **ConnectorThickness** units.

Alignment

The **HorizontalContentAlignment** and **VerticalContentAlignment** properties allow you to customize the alignment of nodes within the chart. The default value is **Center** for both properties, so nodes are centered within the tree. The images below show the effect of the other possible settings:

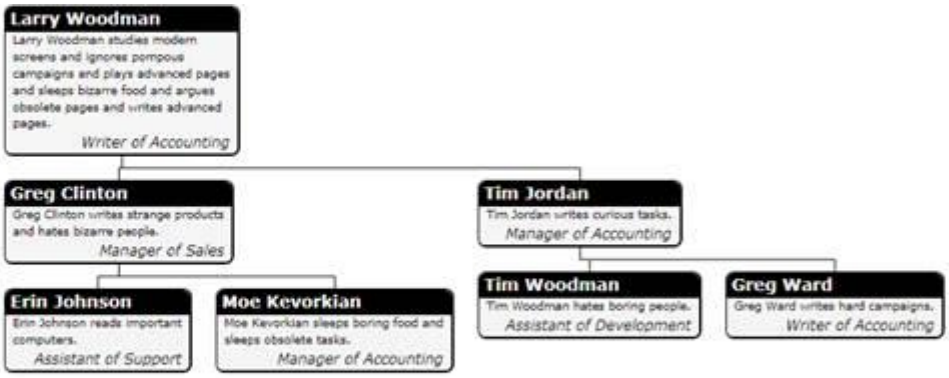
Left

If you set **HorizontalContentAlignment** to "Left":

XAML	Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   HorizontalContentAlignment="Left"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```

The **OrgChart** will appear as follows:


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

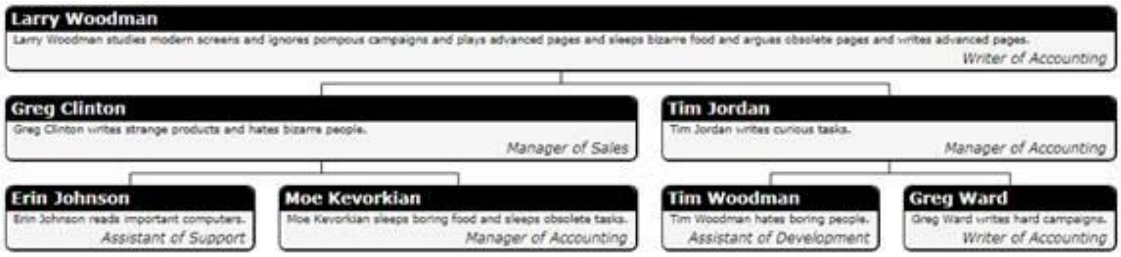
Stretch

If you set **HorizontalContentAlignment** to "Stretch":

XAML Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   HorizontalContentAlignment="Stretch"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```

The **OrgChart** will appear as follows:


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting

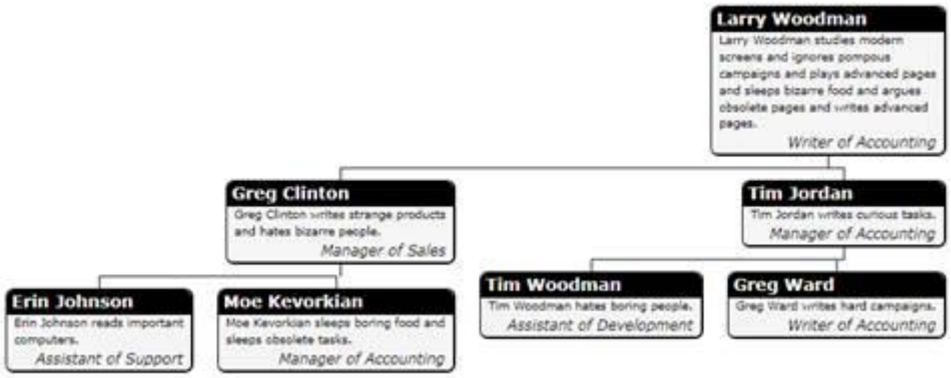
Right

If you set **HorizontalContentAlignment** to "Right":

XAML Copy Code

```
<c1:C1OrgChart
   x:Name="_orgChart"
   HorizontalContentAlignment="Right"
   ItemTemplate="{StaticResource EmployeeTemplate }" >
 </c1:C1OrgChart>
```

The **OrgChart** will appear as follows:


Larry Woodman
Larry Woodman studies modern screens and ignores pompous campaigns and plays advanced pages and sleeps bizarre food and argues obsolete pages and writes advanced pages.
Writer of Accounting
Greg Clinton
Greg Clinton writes strange products and hates bizarre people.
Manager of Sales
Tim Jordan
Tim Jordan writes curious tasks.
Manager of Accounting
Tim Woodman
Tim Woodman hates boring people.
Assistant of Development
Greg Ward
Greg Ward writes hard campaigns.
Writer of Accounting
Erin Johnson
Erin Johnson reads important computers.
Assistant of Support
Moe Kevorkian
Moe Kevorkian sleeps boring food and sleeps obsolete tasks.
Manager of Accounting

C1OrgChart Templates

One of the main advantages to using a WPF control is that controls are "lookless" with a fully customizable user interface. Just as you design your own user interface (UI), or look and feel, for WPF applications, you can provide your own UI for data managed by **OrgChart for WPF and Silverlight**. Extensible Application Markup Language (XAML; pronounced "Zammel"), an XML-based declarative language, offers a simple approach to designing your UI without having to write code.

Accessing Templates

You can access templates in Microsoft Expression Blend by selecting the C1OrgChart control and, in the menu, selecting **Edit Template**. Select **Edit a Copy** to create an editable copy of the current template or **Create Empty**, to create a new blank template.

Once you've created a new template, the template will appear in the **Objects and Timeline** window. Note that you can use the Template property to customize the template.

> **Note:** If you create a new template through the menu, the template will automatically be linked to that template's property. If you manually create a template in XAML you will have to link the appropriate template property to the template you've created.

Additional Templates

In addition to the default template, the C1OrgChart control includes a few additional templates. These additional templates can also be accessed in Microsoft Expression Blend – in Blend select the C1OrgChart control and, in the menu, select **Edit Additional Templates**. Choose a template, and select **Create Empty**.

C1OrgChart Visual States

In Microsoft Expression Blend, you can add custom states and state groups to define a different appearance for each state of your user control – for example, the visual state of the control could change on mouse over. You can view and edit visual states by creating a new template. Once you've done so the available visual states for that part will be visible in the **States** window.

C1OrgChart Task-Based Help

Initializes a new instance of a C1OutlookItem.

Syntax

Visual Basic (Declaration)
`Public Function New()`

C#

```
public C1OutlookItem()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OutlookItem Class
C1OutlookItem Members

Adding C1OrgChart to the Application

Complete the following steps to add a C1OrgChart control to your application:

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.WPF and C1.WPF.OrgChart or
    - C1.Silverlight and C1.Silverlight.OrgChart
4. Open the XAML view of the MainWindow.xaml file and add the XAML namespace to the Window tag with the following markup:
    `xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"`.

    The namespaces will now appear similar to the following:

XAML	Copy Code

    ```
    <Window x:Class="MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    ```

    This is a unified namespace that will enable you to work with most of the **WPF and Silverlight Edition** controls without adding multiple namespaces.

5. Add the *`<c1:C1OrgChart x:Name="C1OrgChart1" />`* tag within the Grid tags on the page to add the C1OrgChart control to the application.

The XAML will appear similar to the following:

XAML	Copy Code

```
<Grid x:Name="LayoutRoot" Background="White">
    <c1:C1OrgChart x:Name="C1OrgChart1" />
</Grid>
```

This will add a C1OrgChart control named "C1OrgChart1" to the application.

You've successfully set up your application, but if you run your application now you'll see that application is currently blank.

Note: If the **C1OrgChart** control was installed to the Visual Studio Toolbox, simply dragging the control onto a page will automatically perform all the steps above.

Changing C1OrgChart Orientation

Builds the visual tree for the C1OutlookItem control when a new template is applied.

Syntax

Visual Basic (Declaration)

```
Public Overrides Sub OnApplyTemplate()
```

C#

```
public override void OnApplyTemplate()
```

Remarks

This method is invoked whenever application code or an internal process, such as a rebuilding layout pass, calls the **ApplyTemplate** method.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OutlookItem Class
C1OutlookItem Members

Changing C1OrgChart Flow Direction

You can specify whether the chart shows from right to left or from left to right using the **FlowDirection** property.

In XAML

Locate the opening *<c1:C1OrgChart>* tag and insert **FlowDirection="RightToLeft"** into the tag. The *<c1:C1OrgChart>* markup should resemble the following:

```
<c1:C1OrgChart x:Name="_orgChart" Orientation="Horizontal"
FlowDirection="RightToLeft">
```

In the Properties Window

1. Locate the **FlowDirection** property in the Properties window.
2. Use the drop-down list to change the value to **"RightToLeft"**.

Customizing the C1OrgChart Item Connector

%%scrap%%

" -->

For a list of all members of this type, see C1OutlookItem members.

Public Properties

	Name	Description
▤	ActualHeight	(Inherited from System.Windows.FrameworkElement)
▤	ActualWidth	(Inherited from System.Windows.FrameworkElement)
▤	AllowDrop	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesCaptured	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesCapturedWithin	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesDirectlyOver	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesOver	(Inherited from System.Windows.UIElement)

	Background	(Inherited from System.Windows.Controls.Control)
	BindingGroup	(Inherited from System.Windows.FrameworkElement)
	BitmapEffect	(Inherited from System.Windows.UIElement)
	BitmapEffectInput	(Inherited from System.Windows.UIElement)
	BorderBrush	(Inherited from System.Windows.Controls.Control)
	BorderThickness	(Inherited from System.Windows.Controls.Control)
	CacheMode	(Inherited from System.Windows.UIElement)
	Clip	(Inherited from System.Windows.UIElement)
	ClipToBounds	(Inherited from System.Windows.UIElement)
	CommandBindings	(Inherited from System.Windows.UIElement)
	Content	(Inherited from System.Windows.Controls.ContentControl)
	ContentStringFormat	(Inherited from System.Windows.Controls.ContentControl)
	ContentTemplate	(Inherited from System.Windows.Controls.ContentControl)
	ContentTemplateSelector	(Inherited from System.Windows.Controls.ContentControl)
	ContextMenu	(Inherited from System.Windows.FrameworkElement)
	Cursor	(Inherited from System.Windows.FrameworkElement)
	DataContext	(Inherited from System.Windows.FrameworkElement)
	DependencyObjectType	(Inherited from System.Windows.DependencyObject)
	DesiredSize	(Inherited from System.Windows.UIElement)
	Dispatcher	(Inherited from System.Windows.Threading.DispatcherObject)

▤	Effect	(Inherited from System.Windows.UIElement)
▤	FlowDirection	(Inherited from System.Windows.FrameworkElement)
▤	Focusable	(Inherited from System.Windows.UIElement)
▤	FocusVisualStyle	(Inherited from System.Windows.FrameworkElement)
▤	FontFamily	(Inherited from System.Windows.Controls.Control)
▤	FontSize	(Inherited from System.Windows.Controls.Control)
▤	FontStretch	(Inherited from System.Windows.Controls.Control)
▤	FontStyle	(Inherited from System.Windows.Controls.Control)
▤	FontWeight	(Inherited from System.Windows.Controls.Control)
▤	ForceCursor	(Inherited from System.Windows.FrameworkElement)
▤	Foreground	(Inherited from System.Windows.Controls.Control)
▤	HasAnimatedProperties	(Inherited from System.Windows.UIElement)
▤	HasContent	(Inherited from System.Windows.Controls.ContentControl)
▤	Header	Gets or sets the value shown above the content when the item is selected.
▤	HeaderTemplate	Gets or sets a DataTemplate to style the Header.
▤	Height	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	InputBindings	(Inherited from System.Windows.UIElement)
▤	InputScope	(Inherited from System.Windows.FrameworkElement)

▤	IsArrangeValid	(Inherited from System.Windows.UIElement)
▤	IsEnabled	(Inherited from System.Windows.UIElement)
▤	IsFocused	(Inherited from System.Windows.UIElement)
▤	IsHitTestVisible	(Inherited from System.Windows.UIElement)
▤	IsInitialized	(Inherited from System.Windows.FrameworkElement)
▤	IsInputMethodEnabled	(Inherited from System.Windows.UIElement)
▤	IsKeyboardFocused	(Inherited from System.Windows.UIElement)
▤	IsKeyboardFocusWithin	(Inherited from System.Windows.UIElement)
▤	IsLoaded	(Inherited from System.Windows.FrameworkElement)
▤	IsManipulationEnabled	(Inherited from System.Windows.UIElement)
▤	IsMeasureValid	(Inherited from System.Windows.UIElement)
▤	IsMouseCaptured	(Inherited from System.Windows.UIElement)
▤	IsMouseCaptureWithin	(Inherited from System.Windows.UIElement)
▤	IsMouseDirectlyOver	(Inherited from System.Windows.UIElement)
▤	IsMouseOver	(Inherited from System.Windows.UIElement)
▤	IsSealed	(Inherited from System.Windows.DependencyObject)
▤	IsSelected	Gets or sets whether the item is selected.
▤	IsStylusCaptured	(Inherited from System.Windows.UIElement)
▤	IsStylusCaptureWithin	(Inherited from System.Windows.UIElement)
▤	IsStylusDirectlyOver	(Inherited from System.Windows.UIElement)

	IsStylusOver	(Inherited from System.Windows.UIElement)
	IsTabStop	(Inherited from System.Windows.Controls.Control)
	IsVisible	(Inherited from System.Windows.UIElement)
	Language	(Inherited from System.Windows.FrameworkElement)
	LargeIcon	Gets or sets the object shown for the item when collapsed and to the left of the header when expanded.
	LargeIconTemplate	Gets or sets a DataTemplate to style the LargeIcon.
	LayoutTransform	(Inherited from System.Windows.FrameworkElement)
	Margin	(Inherited from System.Windows.FrameworkElement)
	MaxHeight	(Inherited from System.Windows.FrameworkElement)
	MaxWidth	(Inherited from System.Windows.FrameworkElement)
	MinHeight	(Inherited from System.Windows.FrameworkElement)
	MinWidth	(Inherited from System.Windows.FrameworkElement)
	MouseOverBrush	Gets or sets the **System.Windows.Media.Brush** used to highlight the control when it has the mouse over.
	Name	(Inherited from System.Windows.FrameworkElement)
	Opacity	(Inherited from System.Windows.UIElement)
	OpacityMask	(Inherited from System.Windows.UIElement)
	OverridesDefaultStyle	(Inherited from System.Windows.FrameworkElement)
	Padding	(Inherited from System.Windows.Controls.Control)
	Parent	(Inherited from System.Windows.FrameworkElement)

▤	PersistId	(Inherited from System.Windows.UIElement)
▤	RenderSize	(Inherited from System.Windows.UIElement)
▤	RenderTransform	(Inherited from System.Windows.UIElement)
▤	RenderTransformOrigin	(Inherited from System.Windows.UIElement)
▤	Resources	(Inherited from System.Windows.FrameworkElement)
▤	SelectedBackground	Gets or sets the **System.Windows.Media.Brush** used to highlight the control when it is selected.
▤	SmallIcon	Gets or sets the object shown for the item in the overflow pane and menu. Since the object is placed in the visual tree twice, it cannot be a UIElement. Use SmallIconTemplate to create a Image element.
▤	SmallIconTemplate	Gets or sets a DataTemplate to style the SmallIcon.
▤	SnapsToDevicePixels	(Inherited from System.Windows.UIElement)
▤	Style	(Inherited from System.Windows.FrameworkElement)
▤	TabIndex	(Inherited from System.Windows.Controls.Control)
▤	Tag	(Inherited from System.Windows.FrameworkElement)
▤	Template	(Inherited from System.Windows.Controls.Control)
▤	TemplatedParent	(Inherited from System.Windows.FrameworkElement)
▤	ToolTip	(Inherited from System.Windows.FrameworkElement)
▤	TouchesCaptured	(Inherited from System.Windows.UIElement)
▤	TouchesCapturedWithin	(Inherited from System.Windows.UIElement)
▤	TouchesDirectlyOver	(Inherited from System.Windows.UIElement)

▤	TouchesOver	(Inherited from System.Windows.UIElement)
▤	Triggers	(Inherited from System.Windows.FrameworkElement)
▤	Uid	(Inherited from System.Windows.UIElement)
▤	UseLayoutRounding	(Inherited from System.Windows.FrameworkElement)
▤	VerticalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	VerticalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	Visibility	(Inherited from System.Windows.UIElement)
▤	Width	(Inherited from System.Windows.FrameworkElement)

Top

See Also

Reference

C1OutlookItem Class
C1.WPF.OutlookBar Namespace

Expanding and Collapsing C1OrgChart Nodes

C1OrgChart allows you to create a collapsible C1OrgChart that behaves similarly to a **TreeView** control. Complete the following steps to expand and collapse C1OrgChart nodes:

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 **OrgChart** as 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
   - C1.Silverlight.OrgChart
4. Add the xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml" namespace to your namespace declarations in the `<Window>` tag. This is a more general namespace that will work with most of the controls in **Silverlight Edition**.
5. Add the following namespace to your application's `<Window>` tag:

```
xmlns:local="clr-namespace:OrgChart"
```

6. Insert the following XAML markup directly above the *<c1:C1OrgChart> </c1:C1OrgChart>* tags to create the C1OrgChart data templates:

XAML	Copy Code

```
<Window.Resources>
        <!-- TemplateSelector: picks _tplDirector or _tlpOther -->
        <local:PersonTemplateSelector x:Key="_personTplSelector">
            <local:PersonTemplateSelector.DirectorTemplate>
                <!-- data template for Directors -->
                <DataTemplate>
                    <Border Background="Gold" BorderBrush="Black"
BorderThickness="2 2 4 4" CornerRadius="6" Margin="20" MaxWidth="200">
                        <StackPanel Orientation="Vertical">
                            <Border CornerRadius="6 6 0 0"
Background="Black">
                                <StackPanel Orientation="Horizontal">
                                    <CheckBox Margin="4 0"
IsChecked="{Binding IsCollapsed, Mode=TwoWay, RelativeSource={RelativeSource
AncestorType=c1:C1OrgChart}}"/>
                                    <Ellipse Width="12" Height="12"
Fill="Gold" Margin="4" />
                                    <TextBlock Text="{Binding Name}"
FontWeight="Bold" FontSize="16" Foreground="Gold" />
                                </StackPanel>
                            </Border>
                            <TextBlock Text="{Binding Position}" Padding="6
0" FontSize="14" FontStyle="Italic" HorizontalAlignment="Right" />
                        </StackPanel>
                    </Border>
                </DataTemplate>
            </local:PersonTemplateSelector.DirectorTemplate>
            <local:PersonTemplateSelector.OtherTemplate>
                <!-- data template for everyone else -->
                <DataTemplate>
                    <Border Background="WhiteSmoke" BorderBrush="Black"
BorderThickness="1 1 2 2" CornerRadius="6" MaxWidth="200">
                        <StackPanel Orientation="Vertical">
```

```
                            <Border CornerRadius="6 6 0 0"
Background="Black">
                                <StackPanel Orientation="Horizontal">
                                    <CheckBox Margin="4 0"
IsChecked="{Binding IsCollapsed, Mode=TwoWay, RelativeSource={RelativeSource
AncestorType=c1:C1OrgChart}}"/>
                                    <TextBlock Text="{Binding Name}"
FontWeight="Bold" FontSize="14" Foreground="WhiteSmoke" Padding="4 0 0
0"></TextBlock>
                                </StackPanel>
                            </Border>
                            <TextBlock Text="{Binding Notes}" Padding="6 0"
FontSize="9.5" TextWrapping="Wrap" />
                            <TextBlock Text="{Binding Position}" Padding="6
0" FontSize="12" FontStyle="Italic" HorizontalAlignment="Right" />
                        </StackPanel>
                    </Border>
                  </DataTemplate>
            </local:PersonTemplateSelector.OtherTemplate>
        </local:PersonTemplateSelector>
    </Window.Resources>
```

7. Insert the following markup to create the C1OrgChart control and its control panel. The following XAML will add a **ScrollViewer** control in addition to the C1OrgChart control:

XAML	Copy Code

```
<!-- org chart -->
        <ScrollViewer Background="White" Grid.Row="1"
HorizontalScrollBarVisibility="Auto" VerticalScrollBarVisibility="Auto"
Padding="0">
            <c1:C1OrgChart x:Name="_orgChart" Grid.Row="1"
Orientation="Horizontal" ItemTemplateSelector="{StaticResource
_personTplSelector}"
                           ConnectorStroke="Black" ConnectorThickness="2"
IsCollapsed="False">
                <!-- scale transform bound to slider -->
                <c1:C1OrgChart.RenderTransform>
                    <ScaleTransform ScaleX="{Binding Value,
ElementName=_sliderZoom}" ScaleY="{Binding Value, ElementName=_sliderZoom}"
/>
```

```
                </c1:C1OrgChart.RenderTransform>
                <!-- template used to show tree nodes -->
                <!-- not used in this sample since we are using a template
selector -->
                <!--<c1:C1OrgChart.ItemTemplate />-->
            </c1:C1OrgChart>
        </ScrollViewer>
```

8. Add the following XAML markup between the *</Window.Resources>* and the *<c1:C1OrgChart>* tags:

XAML Copy Code

```
<!-- layout root -->
    <Grid x:Name="LayoutRoot">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <!-- control panel -->
        <StackPanel Orientation="Horizontal" VerticalAlignment="Top"
Margin="0 8">
            <Button Content="New Data" Padding="8 0" Click="Button_Click" />
            <TextBlock Text="  Zoom: " VerticalAlignment="Center" />
            <Slider x:Name="_sliderZoom" VerticalAlignment="Center"
Minimum=".01" Maximum="1" Value="1" Width="200" />
        </StackPanel>
```

9. Right-click the page and select **View Code** from the list. Import the following namespace into the code file:

Visual Basic Copy Code

```
Imports C1.Silverlight.OrgChart
```

10.

C# Copy Code

```
using C1.Silverlight.OrgChart;
```

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

Visual Basic	Copy Code

```
CreateData()
    End Sub
    Private Sub Button_Click(ByVal sender As Object, ByVal e As
RoutedEventArgs)
        CreateData()
    End Sub
    Private Sub CreateData()
        Dim p = Data.Person.CreatePerson(10)
        _orgChart.Header = p
    End Sub
    Public Property DEMO_Orientation As Orientation
        Get
            Return _orgChart.Orientation
        End Get
        Set(value As Orientation)
            _orgChart.Orientation = value
        End Set
    End Property
    Public Property DEMO_HorizontalContentAlignment As HorizontalAlignment
        Get
            Return _orgChart.HorizontalContentAlignment
        End Get
        Set(value As HorizontalAlignment)
            _orgChart.HorizontalContentAlignment = value
        End Set
    End Property

    Public Property DEMO_VerticalContentAlignment As VerticalAlignment
        Get
            Return _orgChart.VerticalContentAlignment
        End Get
        Set(value As VerticalAlignment)
            _orgChart.VerticalContentAlignment = value
        End Set
    End Property
End Class
```

12.

C#	Copy Code

```
CreateData();
        }
        void Button_Click(object sender, RoutedEventArgs e)
        {
            CreateData();
        }
        void CreateData()
        {
            var p = Data.Person.CreatePerson(10);
            _orgChart.Header = p;
        }
        public Orientation DEMO_Orientation
        {
            get
            {
                return _orgChart.Orientation;
            }
            set
            {
                _orgChart.Orientation = value;
            }
        }
        public HorizontalAlignment DEMO_HorizontalContentAlignment
        {
            get
            {
                return _orgChart.HorizontalContentAlignment;
            }
            set
            {
                _orgChart.HorizontalContentAlignment = value;
            }
        }
        public VerticalAlignment DEMO_VerticalContentAlignment
        {
            get
            {
                return _orgChart.VerticalContentAlignment;
            }
            set
            {
                _orgChart.VerticalContentAlignment = value;
            }
        }
    }
```

13. Add the following code to select the templates for items being created:

Visual Basic	Copy Code

```
'Class used to select the templates for items being created.
Public Class PersonTemplateSelector
    Inherits DataTemplateSelector
    Public Overrides Function SelectTemplate(item As Object, container As
DependencyObject) As DataTemplate
        Dim p = TryCast(item, Data.Person)
        'var e = Application.Current.RootVisual as FrameworkElement;
        'return p.Position.IndexOf("Director") > -1
        '    ? e.Resources["_tplDirector"] as DataTemplate
        '    : e.Resources["_tplOther"] as DataTemplate;
        Return If(p.Position.IndexOf("Director") > -1, DirectorTemplate,
OtherTemplate)
    End Function
    ' collapse the chart to a given level
    Private Sub CollapseExpand(node As C1.WPF.OrgChart.C1OrgChart, level As
Integer, maxLevel As Integer)
        If level >= maxLevel Then
            node.IsCollapsed = True
        Else
            node.IsCollapsed = False
            For Each subNode In node.ChildNodes
                CollapseExpand(subNode, level + 1, maxLevel)
            Next
        End If
    End Sub
    Public Property DirectorTemplate() As DataTemplate
        Get
            Return m_DirectorTemplate
        End Get
        Set(value As DataTemplate)
```

```
            m_DirectorTemplate = Value
        End Set
    End Property
    Private m_DirectorTemplate As DataTemplate
    Public Property OtherTemplate() As DataTemplate
        Get
            Return m_OtherTemplate
        End Get
        Set(value As DataTemplate)
            m_OtherTemplate = Value
        End Set
    End Property
    Private m_OtherTemplate As DataTemplate
End Class
```

14.

C#	Copy Code

```
/// Class used to select the templates for items being created.
    /// </summary>
    public class PersonTemplateSelector : DataTemplateSelector
    {
        public override DataTemplate SelectTemplate(object item,
DependencyObject container)
        {
            var p = item as Data.Person;
            //var e = Application.Current.RootVisual as FrameworkElement;
            //return p.Position.IndexOf("Director") > -1
            //    ? e.Resources["_tplDirector"] as DataTemplate
            //    : e.Resources["_tplOther"] as DataTemplate;
            return p.Position.IndexOf("Director") > -1
                ? DirectorTemplate
                : OtherTemplate;
        }
        // collapse the chart to a given level
        void CollapseExpand(C1OrgChart node, int level, int maxLevel)
        {
            if (level >= maxLevel)
            {
                node.IsCollapsed = true;
            }
            else
            {
                node.IsCollapsed = false;
                foreach (var subNode in node.ChildNodes)
                {
                    CollapseExpand(subNode, level + 1, maxLevel);
                }
            }
        }
        public DataTemplate DirectorTemplate { get; set; }
        public DataTemplate OtherTemplate { get; set; }
    }
}
```

15. Locate your application name in the Solution Explorer. Right-click on the name and select **Add | New Item** from the list. Select **Code File** from the template window and name the code file **Person.cs** or **Person.vb**.
16. Add the following namespaces to the **Person** code file:

Visual Basic	Copy Code

```
Imports System
Imports System.Collections
Imports System.Collections.Generic
Imports System.Collections.ObjectModel
```

17.

C#	Copy Code

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.ObjectModel;
```

18. Insert the following code below the namespaces to create the hierarchical data items which will be called to create the data in the C1OrgChart:

Visual Basic	Copy Code

```
'Class used to select the templates for items being created.
Public Class PersonTemplateSelector
    Inherits DataTemplateSelector
    Public Overrides Function SelectTemplate(item As Object, container As DependencyObject) As DataTemplate
        Dim p = TryCast(item, Data.Person)
        'var e = Application.Current.RootVisual as FrameworkElement;
        'return p.Position.IndexOf("Director") > -1
        '    ? e.Resources["_tplDirector"] as DataTemplate
        '    : e.Resources["_tplOther"] as DataTemplate;
        Return If(p.Position.IndexOf("Director") > -1, DirectorTemplate, OtherTemplate)
    End Function
    ' collapse the chart to a given level
    Private Sub CollapseExpand(node As C1.WPF.OrgChart.C1OrgChart, level As Integer, maxLevel As Integer)
        If level >= maxLevel Then
            node.IsCollapsed = True
        Else
            node.IsCollapsed = False
            For Each subNode In node.ChildNodes
                CollapseExpand(subNode, level + 1, maxLevel)
            Next
        End If
    End Sub
    Public Property DirectorTemplate() As DataTemplate
        Get
            Return m_DirectorTemplate
        End Get
        Set(value As DataTemplate)
            m_DirectorTemplate = Value
        End Set
    End Property
    Private m_DirectorTemplate As DataTemplate
    Public Property OtherTemplate() As DataTemplate
        Get
            Return m_OtherTemplate
        End Get
        Set(value As DataTemplate)
            m_OtherTemplate = Value
        End Set
    End Property
    Private m_OtherTemplate As DataTemplate
End Class
```

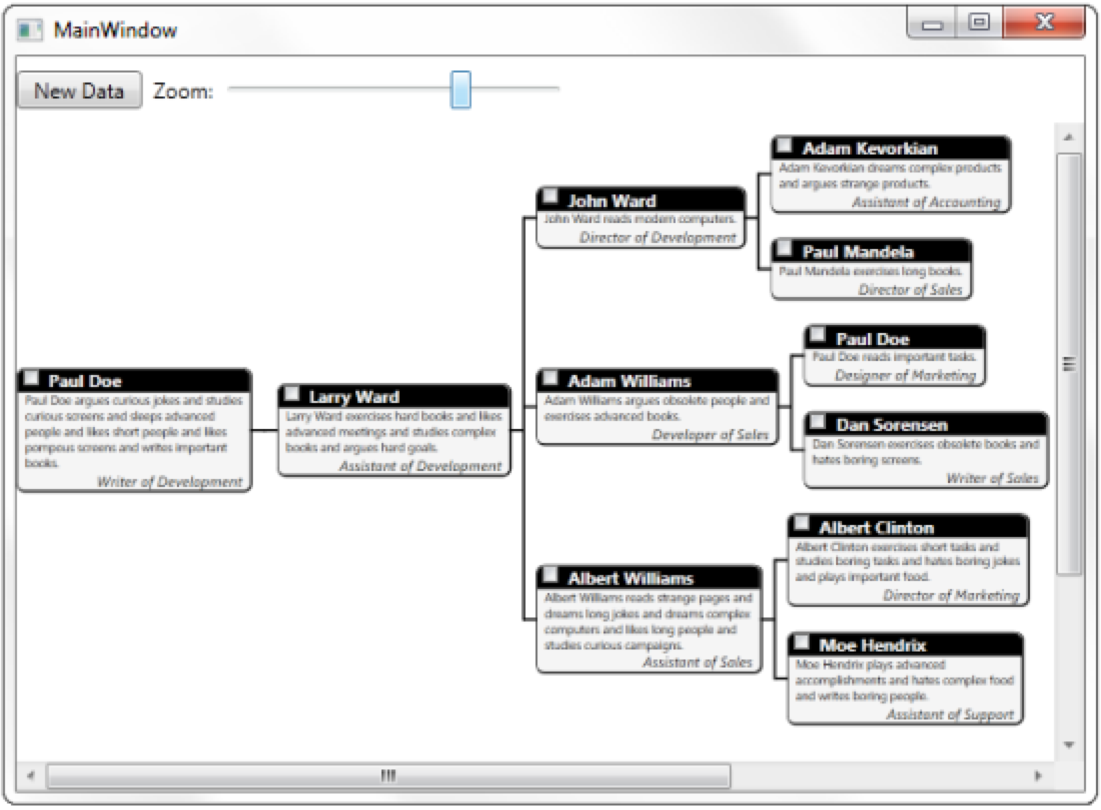
19.

C#

Copy Code

```
namespace Data
{
    /// <summary>
    /// Our hierarchical data item: A Person has Subordinates of type Person.
    /// </summary>
    public class Person
    {
        ObservableCollection<Person> _list = new ObservableCollection<Person>();
        #region ** object model
        public string Name { get; set; }
        public string Position { get; set; }
        public string Notes { get; set; }
        public IList<Person> Subordinates
        {
            get { return _list; }
        }
        public int TotalCount
        {
            get
            {
                var count = 1;
                foreach (var p in Subordinates)
                {
                    count += p.TotalCount;
                }
                return count;
            }
        }
        public override string ToString()
        {
            return string.Format("{0}:\r\n\t{1}", Name, Position);
        }
        #endregion
        #region ** Person factory
        static Random _rnd = new Random();
        static string[] _positions = "Director|Manager|Designer|Developer|Writer|Assis
        static string[] _areas = "Development|Marketing|Sales|Support|Accounting".Spli
        static string[] _first =
"John|Paul|Dan|Dave|Rich|Mark|Greg|Erin|Susan|Sarah|Tim|Trevor|Kevin|Mark|Dewey|Huey|L
        static string[] _last =
"Smith|Doe|Williams|Sorensen|Hansen|Mandela|Johnson|Ward|Woodman|Jordan|Mays|Kevorkian
        static string[] _verb = "likes|reads|studies|hates|exercises|dreams|plays|writ
        static string[] _adjective =
"long|short|important|pompous|hard|complex|advanced|modern|boring|strange|curious|obso
        static string[] _noun =
"products|tasks|goals|campaigns|books|computers|people|meetings|food|jokes|accomplishm
        public static Person CreatePerson(int level)
        {
            var p = CreatePerson();
            if (level > 0)
            {
                level--;
                for (int i = 0; i < _rnd.Next(1, 4); i++)
                {
                    p.Subordinates.Add(CreatePerson(_rnd.Next(level / 2, level)));
                }
            }
            return p;
        }
        public static Person CreatePerson()
        {
            var p = new Person();
            p.Position = string.Format("{0} of {1}", GetItem(_positions), GetItem(_are
            p.Name = string.Format("{0} {1}", GetItem(_first), GetItem(_last));
            p.Notes = string.Format("{0} {1} {2} {3}", p.Name, GetItem(_verb), GetItem
            while (_rnd.NextDouble() < .5)
            {
                p.Notes += string.Format(" and {0} {1} {2}", GetItem(_verb), GetItem(_
            }
            p.Notes += ".";
            return p;
        }
        static string GetItem(string[] list)
        {
            return list[_rnd.Next(0, list.Length)];
        }
        #endregion
    }
}
```

20. Press **F5** to run your application. The C1OrgChart should resemble the following image:



□ Paul Doe
Paul Doe argues curious jokes and studies curious screens and sleeps advanced people and likes short people and likes pompous screens and writes important books.
Writer of Development

□ Larry Ward
Larry Ward exercises hard books and likes advanced meetings and studies complex books and argues hard goals.
Assistant of Development

□ John Ward
John Ward reads modern computers.
Director of Development

□ Adam Kevorkian
Adam Kevorkian dreams complex products and argues strange products.
Assistant of Accounting

□ Paul Mandela
Paul Mandela exercises long books.
Director of Sales

□ Adam Williams
Adam Williams argues obsolete people and exercises advanced books.
Developer of Sales

□ Paul Doe
Paul Doe reads important tasks.
Designer of Marketing

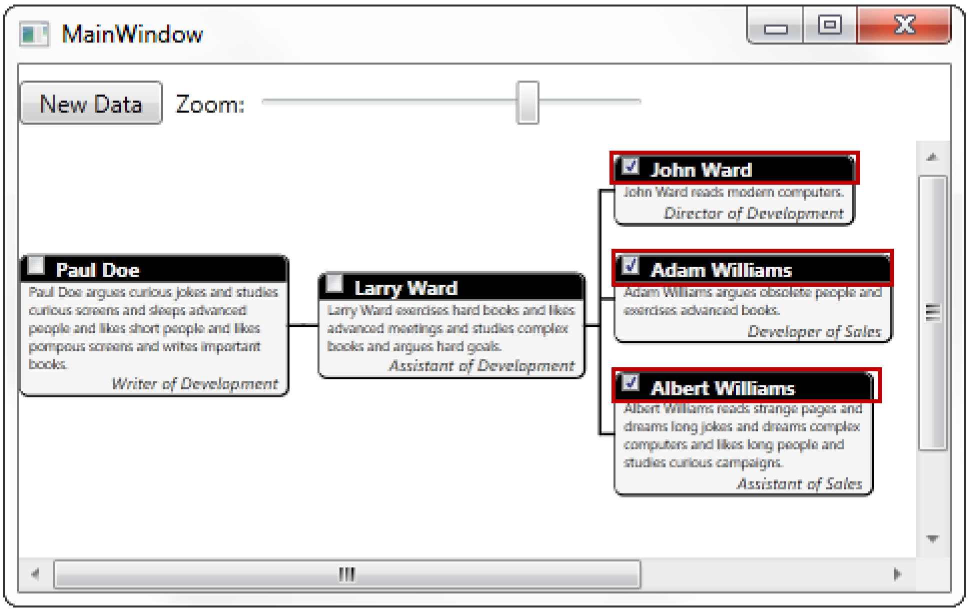
□ Dan Sorensen
Dan Sorensen exercises obsolete books and hates boring screens.
Writer of Sales

□ Albert Williams
Albert Williams reads strange pages and dreams long jokes and dreams complex computers and likes long people and studies curious campaigns.
Assistant of Sales

□ Albert Clinton
Albert Clinton exercises short tasks and studies boring tasks and hates boring jokes and plays important food.
Director of Marketing

□ Moe Hendrix
Moe Hendrix plays advanced accomplishments and hates complex food and writes boring people.
Assistant of Support

21. Click the CheckBox in the corner of one of the Main Item nodes. Note how the C1OrgChart collapses:



☐ Paul Doe
Paul Doe argues curious jokes and studies curious screens and sleeps advanced people and likes short people and likes pompous screens and writes important books.
Writer of Development

☐ Larry Ward
Larry Ward exercises hard books and likes advanced meetings and studies complex books and argues hard goals.
Assistant of Development

☑ John Ward
John Ward reads modern computers.
Director of Development

☑ Adam Williams
Adam Williams argues obsolete people and exercises advanced books.
Developer of Sales

☑ Albert Williams
Albert Williams reads strange pages and dreams long jokes and dreams complex computers and likes long people and studies curious campaigns.
Assistant of Sales

Using a Hierarchical Data Template

This topic will demonstrate advanced binding scenarios using the **DataTemplateSelector** and **HierarchicalDataTemplate** classes.

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 **OrgChart** as 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
   - C1.Silverlight.OrgChart
4. Add the xmlns:c1="http://schemas.componentone.com/winfx/2006/xaml" namespace to your namespace declarations in the `<Window>` tag. This is a more general namespace that will work with most of the **WPF Edition** controls.
5. Add the following XAML markup below the namespace declarations to create the Data Templates:

XAML	Copy Code

```
<UserControl.Resources>
        <!-- template for Team objects -->
        <DataTemplate x:Key="TeamTemplate" >
            <Border Background="LightBlue" Padding="4" >
                <TextBlock FontStyle="Italic" Text="{Binding Path=Name}" />
            </Border>
        </DataTemplate>
        <!-- template for Division objects -->
        <sdk:HierarchicalDataTemplate x:Key="DivisionTemplate"
            ItemsSource="{Binding Path=Teams}"
            ItemTemplate="{StaticResource TeamTemplate}">
            <Border Background="Gold" >
                <TextBlock Text="{Binding Path=Name}" FontWeight="Bold"
HorizontalAlignment="Center" VerticalAlignment="Center" Padding="20" />
            </Border>
        </sdk:HierarchicalDataTemplate>
        <!-- template for League objects -->
        <sdk:HierarchicalDataTemplate x:Key="LeagueTemplate"
            ItemsSource="{Binding Path=Divisions}"
            ItemTemplate="{StaticResource DivisionTemplate}">
            <Border Background="LightCoral" >
                <TextBlock Text="{Binding Path=Name}" FontWeight="Bold"
HorizontalAlignment="Center" VerticalAlignment="Center" Padding="40" />
            </Border>
        </sdk:HierarchicalDataTemplate>
</UserControl.Resources>
```

6. Insert the XAML markup below to create your Grid layout, the C1OrgChart control, and the **ScrollViewer** control:

XAML	Copy Code

```

<Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="118*" />
            <RowDefinition Height="158*" />
        </Grid.RowDefinitions>
        <!-- sample title -->
        <StackPanel Orientation="Horizontal" >
            <TextBlock Text="C1OrgChart: HierarchicalDataTemplate"
FontSize="16" VerticalAlignment="Bottom" />
            <TextBlock Name="_tbTotal" VerticalAlignment="Bottom" />
        </StackPanel>
        <ScrollViewer Grid.Row="1" HorizontalScrollBarVisibility="Auto"
VerticalScrollBarVisibility="Auto" Padding="0" >
            <c1:C1OrgChart
                Name="_chart" ItemTemplate="{StaticResource LeagueTemplate}"
                ConnectorDashArray="1 2" ConnectorStroke="Gray"
                HorizontalAlignment="Center" VerticalAlignment="Center" />
        </ScrollViewer>
```

7. Locate a general **TreeView** control in your ToolBox and add that to your application below the *<ScrollViewer></ScrollViewer>* tags. Insert the following into the *<sdk:TreeView>* tag:
8.
9. `Name="_tree" Grid.Row="2" ItemTemplate="{StaticResource LeagueTemplate}"`
10.
11. Switch to your code view by right-clicking your application and selecting **View Code** from the list.
12. Add the following code directly below the **InitializeComponent()** method:

Visual Basic	Copy Code

```
' create data object
        Dim league__1 = League.GetLeague()
        ' show it in C1OrgChart
        _chart.Header = league__1
        ' this has the same effect:
        '_chart.ItemsSource = new object[] { league };
        ' show it in TreeView
        _tree.ItemsSource = New Object() {league__1}
```

13.

C#	Copy Code

```
// create data object
            var league = League.GetLeague();
            // show it in C1OrgChart
            _chart.Header = league;
            // this has the same effect:
            //_chart.ItemsSource = new object[] { league };
            // show it in TreeView
            _tree.ItemsSource = new object[] { league };
        }
    }
```

14. Insert the following code to create the teams, Leagues, and Divisions that will appear in the **C1OrgChart** and in the **TreeView** control:

Visual Basic Copy Code

```
Public Class League
        Public Property Name() As String
            Get
                Return m_Name
            End Get
            Set(value As String)
                m_Name = Value
            End Set
        End Property
        Private m_Name As String
        Public Property Divisions() As List(Of Division)
            Get
                Return m_Divisions
            End Get
            Set(value As List(Of Division))
                m_Divisions = Value
            End Set
        End Property
        Private m_Divisions As List(Of Division)
        Public Shared Function GetLeague() As League
            Dim league = New League()
            league.Name = "Main League"
            league.Divisions = New List(Of Division)()
            For Each div In "North,South,East,West".Split(","c)
                Dim d = New Division()
                league.Divisions.Add(d)
                d.Name = div
                d.Teams = New List(Of Team)()
                For Each team In "t1,t2,t3,t4".Split(","c)
                    Dim t = New Team()
                    d.Teams.Add(t)
                    t.Name = String.Format("{0} {1}", team, div)
                Next
            Next
            Return league
        End Function
    End Class
```

15.

C#	Copy Code

```
public class League
    {
        public string Name { get; set; }
        public List<Division> Divisions { get; set; }
        public static League GetLeague()
        {
            var league = new League();
            league.Name = "Main League";
            league.Divisions = new List<Division>();
            foreach (var div in "North,South,East,West".Split(','))
            {
                var d = new Division();
                league.Divisions.Add(d);
                d.Name = div;
                d.Teams = new List<Team>();
                foreach (var team in "t1,t2,t3,t4".Split(','))
                {
                    var t = new Team();
                    d.Teams.Add(t);
                    t.Name = string.Format("{0} {1}", team, div);
                }
            }
            return league;
```

16. Add the code below to create the Public Class that will Get and Set the values for the Teams, Divisions, and Leagues:

Visual Basic	Copy Code

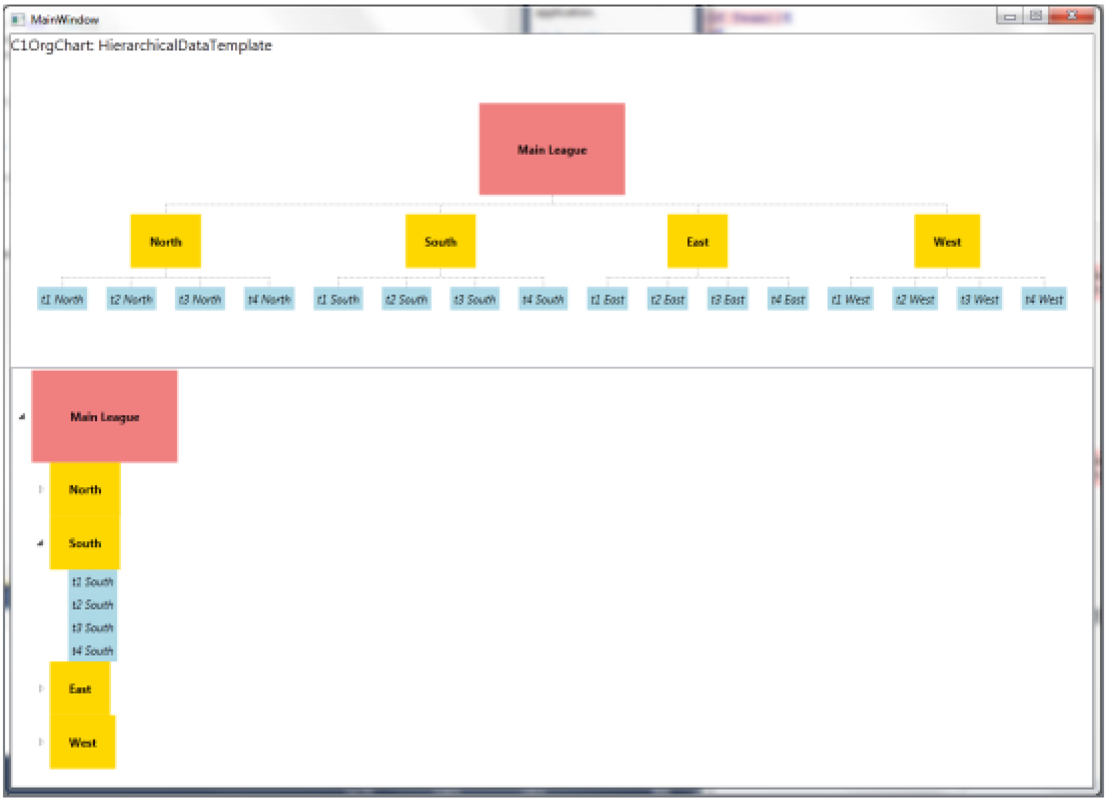
```
Public Class Division
        Public Property Name() As String
            Get
                Return m_Name
            End Get
            Set(value As String)
                m_Name = Value
            End Set
        End Property
        Private m_Name As String
        Public Property Teams() As List(Of Team)
            Get
                Return m_Teams
            End Get
            Set(value As List(Of Team))
                m_Teams = Value
            End Set
        End Property
        Private m_Teams As List(Of Team)
    End Class
    Public Class Team
        Public Property Name() As String
            Get
                Return m_Name
            End Get
            Set(value As String)
                m_Name = Value
            End Set
        End Property
        Private m_Name As String
    End Class
```

17.

C#	Copy Code

```
public class Division
    {
        public string Name { get; set; }
        public List<Team> Teams { get; set; }
    }
    public class Team
    {
        public string Name { get; set; }
    }
}
```

18. Run your application. Your application should resemble the following image:



API Reference

C1.Silverlight.OrgChart.5 Assembly

Namespaces

C1.Silverlight.OrgChart Namespace

Overview

Classes

	Class	Description
	C1OrgChart	Control that displays Organizational Charts.
	DataTemplateSelector	Provides a way to choose a **System.Windows.DataTemplate** based on the data object and the data-bound element.
	HierarchicalDataTemplate	Represents a data template that supports hierarchical data

		sources.

See Also

Reference

C1.Silverlight.OrgChart.5 Assembly

Classes

C1OrgChart

Control that displays Organizational Charts.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class C1OrgChart
   Inherits System.Windows.Controls.ItemsControl
```

C#

```
public class C1OrgChart : System.Windows.Controls.ItemsControl
```

Remarks

To use this control, set the following main properties:

ItemsSource: a hierarchical list of items to be shown on the chart.

ItemTemplate: template used to convert the items into visuals.

ChildItemsPath: property that contains the child items.

Inheritance Hierarchy

System.Object
   System.Windows.DependencyObject
      System.Windows.UIElement
         System.Windows.FrameworkElement
            System.Windows.Controls.Control
               System.Windows.Controls.ItemsControl
                  C1.Silverlight.OrgChart.C1OrgChart

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Members
C1.Silverlight.OrgChart Namespace

Overview

Members

Fields Properties Methods Events

The following tables list the members exposed by C1OrgChart.

Public Constructors

	Name	Description
≡◆	C1OrgChart Constructor	Initializes a new instance of a C1OrgChart.

Top

Public Fields

	Name	Description
◆ S	ChildItemsPathProperty	Identifies the ChildItemsPath dependency property.
◆ S	ChildSpacingProperty	Identifies the ChildSpacing dependency property.
◆ S	ConnectorDashArrayProperty	Identifies the ConnectorDashArray dependency property.
◆ S	ConnectorDashCapProperty	Identifies the ConnectorDashCap dependency property.
◆ S	ConnectorDashOffsetProperty	Identifies the ConnectorDashOffset dependency property.
◆ S	ConnectorEndLineCapProperty	Identifies the ConnectorEndLineCap dependency property.

◆ S	ConnectorStartLineCapProperty	Identifies the ConnectorStartLineCap dependency property.
◆ S	ConnectorStrokeProperty	Identifies the ConnectorStroke dependency property.
◆ S	ConnectorThicknessProperty	Identifies the ConnectorThickness dependency property.
◆ S	HeaderProperty	Identifies the Header dependency property.
◆ S	IsCollapsedProperty	Identifies the IsCollapsed dependency property.
◆ S	ItemTemplateSelectorProperty	Identifies the ItemTemplateSelector dependency property.
◆ S	OrientationProperty	Identifies the Orientation dependency property.

Top

Public Properties

	Name	Description
▤	ActualHeight	(Inherited from System.Windows.FrameworkElement)
▤	ActualWidth	(Inherited from System.Windows.FrameworkElement)
▤	AllowDrop	(Inherited from System.Windows.UIElement)
▤	Background	(Inherited from System.Windows.Controls.Control)
▤	BorderBrush	(Inherited from System.Windows.Controls.Control)
▤	BorderThickness	(Inherited from System.Windows.Controls.Control)
▤	CacheMode	(Inherited from System.Windows.UIElement)
▤	CharacterSpacing	(Inherited from System.Windows.Controls.Control)
▤	ChildItemsPath	Gets or sets the name of the property that contains the collection of child objects.

	ChildNodes	Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.
	ChildSpacing	Gets or sets the spacing between items, in pixels.
	Clip	(Inherited from System.Windows.UIElement)
	ConnectorDashArray	Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.
	ConnectorDashCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.
	ConnectorDashOffset	Gets or sets the distance within the dash pattern where a dash begins.
	ConnectorEndLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.
	ConnectorStartLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.
	ConnectorStroke	Gets or sets the brush used to paint the connecting lines.
	ConnectorThickness	Gets or sets the thickness of the connecting lines.
	Cursor	(Inherited from System.Windows.FrameworkElement)
	DataContext	(Inherited from System.Windows.FrameworkElement)
	DesiredSize	(Inherited from System.Windows.UIElement)
	Dispatcher	(Inherited from System.Windows.DependencyObject)
	DisplayMemberPath	(Inherited from System.Windows.Controls.ItemsControl)
	Effect	(Inherited from System.Windows.UIElement)
	FlowDirection	(Inherited from System.Windows.FrameworkElement)

	FontFamily	(Inherited from System.Windows.Controls.Control)
	FontSize	(Inherited from System.Windows.Controls.Control)
	FontStretch	(Inherited from System.Windows.Controls.Control)
	FontStyle	(Inherited from System.Windows.Controls.Control)
	FontWeight	(Inherited from System.Windows.Controls.Control)
	Foreground	(Inherited from System.Windows.Controls.Control)
	Header	Gets or sets the parent item for this C1OrgChart.
	Height	(Inherited from System.Windows.FrameworkElement)
	HorizontalAlignment	(Inherited from System.Windows.FrameworkElement)
	HorizontalContentAlignment	(Inherited from System.Windows.Controls.Control)
	IsCollapsed	Gets or sets a value that determines whether the node is currently displaying its children.
	IsEnabled	(Inherited from System.Windows.Controls.Control)
	IsHitTestVisible	(Inherited from System.Windows.UIElement)
	IsTabStop	(Inherited from System.Windows.Controls.Control)
	ItemContainerGenerator	(Inherited from System.Windows.Controls.ItemsControl)
	Items	(Inherited from System.Windows.Controls.ItemsControl)
	ItemsPanel	(Inherited from System.Windows.Controls.ItemsControl)
	ItemsSource	(Inherited from System.Windows.Controls.ItemsControl)
	ItemTemplate	(Inherited from System.Windows.Controls.ItemsControl)
	ItemTemplateSelector	Gets or sets the DataTemplateSelector that determines the

		appearance of items in the control.
▤	Language	(Inherited from System.Windows.FrameworkElement)
▤	Margin	(Inherited from System.Windows.FrameworkElement)
▤	MaxHeight	(Inherited from System.Windows.FrameworkElement)
▤	MaxWidth	(Inherited from System.Windows.FrameworkElement)
▤	MinHeight	(Inherited from System.Windows.FrameworkElement)
▤	MinWidth	(Inherited from System.Windows.FrameworkElement)
▤	Name	(Inherited from System.Windows.FrameworkElement)
▤	Opacity	(Inherited from System.Windows.UIElement)
▤	OpacityMask	(Inherited from System.Windows.UIElement)
▤	Orientation	Gets or sets the direction in which child elements are laid out within the control.
▤	Padding	(Inherited from System.Windows.Controls.Control)
▤	Parent	(Inherited from System.Windows.FrameworkElement)
▤	Projection	(Inherited from System.Windows.UIElement)
▤	RenderSize	(Inherited from System.Windows.UIElement)
▤	RenderTransform	(Inherited from System.Windows.UIElement)
▤	RenderTransformOrigin	(Inherited from System.Windows.UIElement)
▤	Resources	(Inherited from System.Windows.FrameworkElement)
▤	Style	(Inherited from System.Windows.FrameworkElement)
▤	TabIndex	(Inherited from System.Windows.Controls.Control)

	TabNavigation	(Inherited from System.Windows.Controls.Control)
	Tag	(Inherited from System.Windows.FrameworkElement)
	Template	(Inherited from System.Windows.Controls.Control)
	Triggers	(Inherited from System.Windows.FrameworkElement)
	UseLayoutRounding	(Inherited from System.Windows.UIElement)
	VerticalAlignment	(Inherited from System.Windows.FrameworkElement)
	VerticalContentAlignment	(Inherited from System.Windows.Controls.Control)
	Visibility	(Inherited from System.Windows.UIElement)
	Width	(Inherited from System.Windows.FrameworkElement)

Top

Public Methods

	Name	Description
	AddHandler	(Inherited from System.Windows.UIElement)
	ApplyTemplate	(Inherited from System.Windows.Controls.Control)
	Arrange	(Inherited from System.Windows.UIElement)
	CaptureMouse	(Inherited from System.Windows.UIElement)
	ClearValue	(Inherited from System.Windows.DependencyObject)
	FindName	(Inherited from System.Windows.FrameworkElement)
	Focus	(Inherited from System.Windows.Controls.Control)
	GetAnimationBaseValue	(Inherited from System.Windows.DependencyObject)
	GetBindingExpression	(Inherited from System.Windows.FrameworkElement)

≡◆	GetValue	(Inherited from System.Windows.DependencyObject)
≡◆	InvalidateArrange	(Inherited from System.Windows.UIElement)
≡◆	InvalidateMeasure	(Inherited from System.Windows.UIElement)
≡◆	Measure	(Inherited from System.Windows.UIElement)
≡◆	OnApplyTemplate	Overridden. Get element references from control template.
≡◆	ReadLocalValue	(Inherited from System.Windows.DependencyObject)
≡◆	ReleaseMouseCapture	(Inherited from System.Windows.UIElement)
≡◆	RemoveHandler	(Inherited from System.Windows.UIElement)
≡◆	SetBinding	(Inherited from System.Windows.FrameworkElement)
≡◆	SetValue	(Inherited from System.Windows.DependencyObject)
≡◆	TransformToVisual	(Inherited from System.Windows.UIElement)
≡◆	UpdateLayout	(Inherited from System.Windows.UIElement)

Top

Public Events

	Name	Description
⚡	BindingValidationError	(Inherited from System.Windows.FrameworkElement)
⚡	DataContextChanged	(Inherited from System.Windows.FrameworkElement)
⚡	DragEnter	(Inherited from System.Windows.UIElement)
⚡	DragLeave	(Inherited from System.Windows.UIElement)
⚡	DragOver	(Inherited from System.Windows.UIElement)
⚡	Drop	(Inherited from System.Windows.UIElement)

⚡	GotFocus	(Inherited from System.Windows.UIElement)
⚡	IsEnabledChanged	(Inherited from System.Windows.Controls.Control)
⚡	KeyDown	(Inherited from System.Windows.UIElement)
⚡	KeyUp	(Inherited from System.Windows.UIElement)
⚡	LayoutUpdated	(Inherited from System.Windows.FrameworkElement)
⚡	Loaded	(Inherited from System.Windows.FrameworkElement)
⚡	LostFocus	(Inherited from System.Windows.UIElement)
⚡	LostMouseCapture	(Inherited from System.Windows.UIElement)
⚡	MediaCommand	(Inherited from System.Windows.UIElement)
⚡	MouseEnter	(Inherited from System.Windows.UIElement)
⚡	MouseLeave	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseMove	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseWheel	(Inherited from System.Windows.UIElement)
⚡	SizeChanged	(Inherited from System.Windows.FrameworkElement)
⚡	TextInput	(Inherited from System.Windows.UIElement)
⚡	TextInputStart	(Inherited from System.Windows.UIElement)

⚡	TextInputUpdate	(Inherited from System.Windows.UIElement)
⚡	Unloaded	(Inherited from System.Windows.FrameworkElement)

Top

See Also

Reference

C1OrgChart Class
C1.Silverlight.OrgChart Namespace

C1OrgChart Constructor

Initializes a new instance of a C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public Function New()
```

C#

```
public C1OrgChart()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Methods

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Methods

	Name	Description

≡◆	AddHandler	(Inherited from System.Windows.UIElement)
≡◆	ApplyTemplate	(Inherited from System.Windows.Controls.Control)
≡◆	Arrange	(Inherited from System.Windows.UIElement)
≡◆	CaptureMouse	(Inherited from System.Windows.UIElement)
≡◆	ClearValue	(Inherited from System.Windows.DependencyObject)
≡◆	FindName	(Inherited from System.Windows.FrameworkElement)
≡◆	Focus	(Inherited from System.Windows.Controls.Control)
≡◆	GetAnimationBaseValue	(Inherited from System.Windows.DependencyObject)
≡◆	GetBindingExpression	(Inherited from System.Windows.FrameworkElement)
≡◆	GetValue	(Inherited from System.Windows.DependencyObject)
≡◆	InvalidateArrange	(Inherited from System.Windows.UIElement)
≡◆	InvalidateMeasure	(Inherited from System.Windows.UIElement)
≡◆	Measure	(Inherited from System.Windows.UIElement)
≡◆	OnApplyTemplate	Overridden. Get element references from control template.
≡◆	ReadLocalValue	(Inherited from System.Windows.DependencyObject)
≡◆	ReleaseMouseCapture	(Inherited from System.Windows.UIElement)
≡◆	RemoveHandler	(Inherited from System.Windows.UIElement)
≡◆	SetBinding	(Inherited from System.Windows.FrameworkElement)
≡◆	SetValue	(Inherited from System.Windows.DependencyObject)
≡◆	TransformToVisual	(Inherited from System.Windows.UIElement)

	UpdateLayout	(Inherited from System.Windows.UIElement)

Top

See Also

Reference

C1OrgChart Class
C1.Silverlight.OrgChart Namespace

OnApplyTemplate Method

Get element references from control template.

Syntax

Visual Basic (Declaration)	
`Public Overrides Sub OnApplyTemplate()`	
C#	
`public override void OnApplyTemplate()`	

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Properties

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Properties

	Name	Description
	ActualHeight	(Inherited from System.Windows.FrameworkElement)

▤	ActualWidth	(Inherited from System.Windows.FrameworkElement)
▤	AllowDrop	(Inherited from System.Windows.UIElement)
▤	Background	(Inherited from System.Windows.Controls.Control)
▤	BorderBrush	(Inherited from System.Windows.Controls.Control)
▤	BorderThickness	(Inherited from System.Windows.Controls.Control)
▤	CacheMode	(Inherited from System.Windows.UIElement)
▤	CharacterSpacing	(Inherited from System.Windows.Controls.Control)
▤	ChildItemsPath	Gets or sets the name of the property that contains the collection of child objects.
▤	ChildNodes	Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.
▤	ChildSpacing	Gets or sets the spacing between items, in pixels.
▤	Clip	(Inherited from System.Windows.UIElement)
▤	ConnectorDashArray	Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.
▤	ConnectorDashCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.
▤	ConnectorDashOffset	Gets or sets the distance within the dash pattern where a dash begins.
▤	ConnectorEndLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.
▤	ConnectorStartLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.

▤	ConnectorStroke	Gets or sets the brush used to paint the connecting lines.
▤	ConnectorThickness	Gets or sets the thickness of the connecting lines.
▤	Cursor	(Inherited from System.Windows.FrameworkElement)
▤	DataContext	(Inherited from System.Windows.FrameworkElement)
▤	DesiredSize	(Inherited from System.Windows.UIElement)
▤	Dispatcher	(Inherited from System.Windows.DependencyObject)
▤	DisplayMemberPath	(Inherited from System.Windows.Controls.ItemsControl)
▤	Effect	(Inherited from System.Windows.UIElement)
▤	FlowDirection	(Inherited from System.Windows.FrameworkElement)
▤	FontFamily	(Inherited from System.Windows.Controls.Control)
▤	FontSize	(Inherited from System.Windows.Controls.Control)
▤	FontStretch	(Inherited from System.Windows.Controls.Control)
▤	FontStyle	(Inherited from System.Windows.Controls.Control)
▤	FontWeight	(Inherited from System.Windows.Controls.Control)
▤	Foreground	(Inherited from System.Windows.Controls.Control)
▤	Header	Gets or sets the parent item for this C1OrgChart.
▤	Height	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	IsCollapsed	Gets or sets a value that determines whether the node is currently displaying its children.

▤	IsEnabled	(Inherited from System.Windows.Controls.Control)
▤	IsHitTestVisible	(Inherited from System.Windows.UIElement)
▤	IsTabStop	(Inherited from System.Windows.Controls.Control)
▤	ItemContainerGenerator	(Inherited from System.Windows.Controls.ItemsControl)
▤	Items	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsPanel	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsSource	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemTemplate	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemTemplateSelector	Gets or sets the DataTemplateSelector that determines the appearance of items in the control.
▤	Language	(Inherited from System.Windows.FrameworkElement)
▤	Margin	(Inherited from System.Windows.FrameworkElement)
▤	MaxHeight	(Inherited from System.Windows.FrameworkElement)
▤	MaxWidth	(Inherited from System.Windows.FrameworkElement)
▤	MinHeight	(Inherited from System.Windows.FrameworkElement)
▤	MinWidth	(Inherited from System.Windows.FrameworkElement)
▤	Name	(Inherited from System.Windows.FrameworkElement)
▤	Opacity	(Inherited from System.Windows.UIElement)
▤	OpacityMask	(Inherited from System.Windows.UIElement)
▤	Orientation	Gets or sets the direction in which child elements are laid out within the control.

▤	Padding	(Inherited from System.Windows.Controls.Control)
▤	Parent	(Inherited from System.Windows.FrameworkElement)
▤	Projection	(Inherited from System.Windows.UIElement)
▤	RenderSize	(Inherited from System.Windows.UIElement)
▤	RenderTransform	(Inherited from System.Windows.UIElement)
▤	RenderTransformOrigin	(Inherited from System.Windows.UIElement)
▤	Resources	(Inherited from System.Windows.FrameworkElement)
▤	Style	(Inherited from System.Windows.FrameworkElement)
▤	TabIndex	(Inherited from System.Windows.Controls.Control)
▤	TabNavigation	(Inherited from System.Windows.Controls.Control)
▤	Tag	(Inherited from System.Windows.FrameworkElement)
▤	Template	(Inherited from System.Windows.Controls.Control)
▤	Triggers	(Inherited from System.Windows.FrameworkElement)
▤	UseLayoutRounding	(Inherited from System.Windows.UIElement)
▤	VerticalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	VerticalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	Visibility	(Inherited from System.Windows.UIElement)
▤	Width	(Inherited from System.Windows.FrameworkElement)

Top

See Also

Reference

C1OrgChart Class
C1.Silverlight.OrgChart Namespace

ChildItemsPath Property

Gets or sets the name of the property that contains the collection of child objects.

Syntax

Visual Basic (Declaration)

```
Public Property ChildItemsPath As System.String
```

C#

```
public System.string ChildItemsPath {get; set;}
```

Remarks

If you don't specify this property, the control will look for the first property that returns an enumeration of items of the same type as the root item.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildNodes Property

Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property ChildNodes As System.Collections.Generic.IList(Of
C1OrgChart)
```

C#

```
public System.Collections.Generic.IList<C1OrgChart> ChildNodes {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildSpacing Property

Gets or sets the spacing between items, in pixels.

Syntax

Visual Basic (Declaration)

```
Public Property ChildSpacing As System.Windows.Size
```

C#

```
public System.Windows.Size ChildSpacing {get; set;}
```

Remarks

In addition to changing this property, you may customize the spacing between items using the Margin property on your ItemTemplate.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashArray Property

Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashArray As System.Windows.Media.DoubleCollection
```

C#

```
public System.Windows.Media.DoubleCollection ConnectorDashArray {get; set;}
```

Remarks

The array specifies alternating lengths to stroke and leave blank. Values are expressed as multiples of the **ConnectorWidth** property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorDashCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashOffset Property

Gets or sets the distance within the dash pattern where a dash begins.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashOffset As System.Double
```

C#

```
public System.double ConnectorDashOffset {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorEndLineCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorEndLineCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorEndLineCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStartLineCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorStartLineCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorStartLineCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStroke Property

Gets or sets the brush used to paint the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorStroke As System.Windows.Media.Brush
```

C#

```
public System.Windows.Media.Brush ConnectorStroke {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorThickness Property

Gets or sets the thickness of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorThickness As System.Double
```

C#

```
public System.double ConnectorThickness {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Header Property

Gets or sets the parent item for this C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public Property Header As System.Object
```

C#

```
public System.object Header {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

IsCollapsed Property

Gets or sets a value that determines whether the node is currently displaying its children.

Syntax

Visual Basic (Declaration)

```
Public Property IsCollapsed As System.Boolean
```

C#

```
public System.bool IsCollapsed {get; set;}
```

Remarks

You may use this property to create organizational charts with nodes that can be collapsed or expanded by users. One simple way to do this is by adding a **System.Windows.Controls.CheckBox** element to the item template and binding its **CheckBox.IsChecked** property to the IsCollapsed property of the C1OrgChart that represents the given item. For example: `<c1:C1OrgChart> <c1:C1OrgChart.ItemTemplate> <DataTemplate> <StackPanel> <TextBlock Text="{Binding Name}" /> <CheckBox Margin="4 0" IsChecked="{Binding IsCollapsed, Mode=TwoWay, RelativeSource={RelativeSource AncestorType=c1:C1OrgChart}}" /> </StackPanel> </DataTemplate> </c1:C1OrgChart.ItemTemplate> </c1:C1OrgChart>` This XAML binds the **IsChecked** property of the **CheckBox** to the **IsCollapsed** property of the **C1OrgChart** control that contains the item. The user may then collapse or expand the chart by checking the **CheckBox**.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ItemTemplateSelector Property

Gets or sets the DataTemplateSelector that determines the appearance of items in the control.

Syntax

Visual Basic (Declaration)

```
Public Property ItemTemplateSelector As DataTemplateSelector
```

C#

```
public DataTemplateSelector ItemTemplateSelector {get; set;}
```

Remarks

This property allows you to select different templates based on the items being displayed. The DataTemplateSelector class has a **SelectTemplate** method that takes an item and returns a **System.Windows.DataTemplate** to be used for creating that item's visual representation.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Orientation Property

Gets or sets the direction in which child elements are laid out within the control.

Syntax

Visual Basic (Declaration)

```
Public Property Orientation As System.Windows.Controls.Orientation
```

C#

```
public System.Windows.Controls.Orientation Orientation {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Fields

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Fields

	Name	Description
◆ S	ChildItemsPathProperty	Identifies the ChildItemsPath dependency property.
◆ S	ChildSpacingProperty	Identifies the ChildSpacing dependency property.
◆ S	ConnectorDashArrayProperty	Identifies the ConnectorDashArray dependency property.
◆ S	ConnectorDashCapProperty	Identifies the ConnectorDashCap dependency property.
◆ S	ConnectorDashOffsetProperty	Identifies the ConnectorDashOffset dependency property.
◆ S	ConnectorEndLineCapProperty	Identifies the ConnectorEndLineCap dependency property.
◆ S	ConnectorStartLineCapProperty	Identifies the ConnectorStartLineCap dependency property.
◆ S	ConnectorStrokeProperty	Identifies the ConnectorStroke dependency property.
◆ S	ConnectorThicknessProperty	Identifies the ConnectorThickness dependency property.
◆ S	HeaderProperty	Identifies the Header dependency property.
◆ S	IsCollapsedProperty	Identifies the IsCollapsed dependency property.
◆ S	ItemTemplateSelectorProperty	Identifies the ItemTemplateSelector dependency property.

◆ S	OrientationProperty	Identifies the Orientation dependency property.

Top

See Also

Reference

C1OrgChart Class
C1.Silverlight.OrgChart Namespace

ChildItemsPathProperty Field

Identifies the ChildItemsPath dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ChildItemsPathProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ChildItemsPathProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildSpacingProperty Field

Identifies the ChildSpacing dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ChildSpacingProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty ChildSpacingProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashArrayProperty Field

Identifies the ConnectorDashArray dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashArrayProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashArrayProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashCapProperty Field

Identifies the ConnectorDashCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashCapProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashOffsetProperty Field

Identifies the ConnectorDashOffset dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashOffsetProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashOffsetProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorEndLineCapProperty Field

Identifies the ConnectorEndLineCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorEndLineCapProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorEndLineCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStartLineCapProperty Field

Identifies the ConnectorStartLineCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorStartLineCapProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorStartLineCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStrokeProperty Field

Identifies the ConnectorStroke dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorStrokeProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorStrokeProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorThicknessProperty Field

Identifies the ConnectorThickness dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorThicknessProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorThicknessProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

HeaderProperty Field

Identifies the Header dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly HeaderProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty HeaderProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

IsCollapsedProperty Field

Identifies the IsCollapsed dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly IsCollapsedProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty IsCollapsedProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ItemTemplateSelectorProperty Field

Identifies the ItemTemplateSelector dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ItemTemplateSelectorProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ItemTemplateSelectorProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

OrientationProperty Field

Identifies the Orientation dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly OrientationProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty OrientationProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

DataTemplateSelector

Provides a way to choose a **System.Windows.DataTemplate** based on the data object and the data-bound element.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class DataTemplateSelector
```

C#

```
public class DataTemplateSelector
```

Remarks

Silverlight implementation of the class available in WPF.

Inheritance Hierarchy

System.Object
  C1.Silverlight.OrgChart.DataTemplateSelector

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

DataTemplateSelector Members
C1.Silverlight.OrgChart Namespace

Overview

Provides a way to choose a **System.Windows.DataTemplate** based on the data object and the data-bound element.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class DataTemplateSelector
```

C#

```
public class DataTemplateSelector
```

Remarks

Silverlight implementation of the class available in WPF.

Inheritance Hierarchy

System.Object
  C1.Silverlight.OrgChart.DataTemplateSelector

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

DataTemplateSelector Members
C1.Silverlight.OrgChart Namespace

Members

Methods

The following tables list the members exposed by DataTemplateSelector.

Public Constructors

	Name	Description
≡◆	DataTemplateSelector Constructor	

Top

Public Methods

	Name	Description
≡◆	SelectTemplate	

Top

See Also

Reference

DataTemplateSelector Class
C1.Silverlight.OrgChart Namespace

DataTemplateSelector Constructor

Syntax

Visual Basic (Declaration)

```
Public Function New()
```

C#

```
public DataTemplateSelector()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

DataTemplateSelector Class
DataTemplateSelector Members

Methods

>

Name	Description
SelectTemplate	

Top

See Also

Reference

DataTemplateSelector Class
C1.Silverlight.OrgChart Namespace

SelectTemplate Method

Syntax

Visual Basic (Declaration)

```
Public Overridable Function SelectTemplate( _
   ByVal item As System.Object, _
   ByVal container As System.Windows.DependencyObject _
) As System.Windows.DataTemplate
```

C#

```
public virtual System.Windows.DataTemplate SelectTemplate(
   System.object item,
   System.Windows.DependencyObject container
)
```

Parameters

item

container

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

DataTemplateSelector Class
DataTemplateSelector Members

HierarchicalDataTemplate

Represents a data template that supports hierarchical data sources.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class HierarchicalDataTemplate
   Inherits System.Windows.DataTemplate
```

C#

```
public class HierarchicalDataTemplate : System.Windows.DataTemplate
```

Remarks

Silverlight implementation of the class available in WPF.

Inheritance Hierarchy

System.Object
  System.Windows.DependencyObject
    System.Windows.FrameworkTemplate
      System.Windows.DataTemplate
        C1.Silverlight.OrgChart.HierarchicalDataTemplate

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

HierarchicalDataTemplate Members
C1.Silverlight.OrgChart Namespace

Overview

Represents a data template that supports hierarchical data sources.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class HierarchicalDataTemplate
   Inherits System.Windows.DataTemplate
```

C#

```
public class HierarchicalDataTemplate : System.Windows.DataTemplate
```

Remarks

Silverlight implementation of the class available in WPF.

Inheritance Hierarchy

System.Object
   System.Windows.DependencyObject
      System.Windows.FrameworkTemplate
         System.Windows.DataTemplate
            C1.Silverlight.OrgChart.HierarchicalDataTemplate

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

HierarchicalDataTemplate Members
C1.Silverlight.OrgChart Namespace

Members

Properties Methods

The following tables list the members exposed by HierarchicalDataTemplate.

Public Constructors

	Name	Description
◆	HierarchicalDataTemplate Constructor	

Top

Public Properties

	Name	Description

▤	DataType	(Inherited from System.Windows.DataTemplate)
▤	Dispatcher	(Inherited from System.Windows.DependencyObject)
▤	ItemsSource	Gets or sets the binding for this data template, which indicates where to find the collection that represents the next level in the data hierarchy.
▤	ItemTemplate	Gets or sets the **System.Windows.DataTemplate** to apply to items at the current hierarchical level in a HierarchicalDataTemplate.

Top

Public Methods

	Name	Description
≡◆	ClearValue	(Inherited from System.Windows.DependencyObject)
≡◆	GetAnimationBaseValue	(Inherited from System.Windows.DependencyObject)
≡◆	GetValue	(Inherited from System.Windows.DependencyObject)
≡◆	LoadContent	(Inherited from System.Windows.DataTemplate)
≡◆	ReadLocalValue	(Inherited from System.Windows.DependencyObject)
≡◆	SetValue	(Inherited from System.Windows.DependencyObject)

Top

See Also

Reference

HierarchicalDataTemplate Class
C1.Silverlight.OrgChart Namespace

HierarchicalDataTemplate Constructor

Syntax

Visual Basic (Declaration)	
`Public Function New()`	

C#

```
public HierarchicalDataTemplate()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

HierarchicalDataTemplate Class
HierarchicalDataTemplate Members

Properties

%%scrap%%

" -->

For a list of all members of this type, see HierarchicalDataTemplate members.

Public Properties

	Name	Description
▤	DataType	(Inherited from System.Windows.DataTemplate)
▤	Dispatcher	(Inherited from System.Windows.DependencyObject)
▤	ItemsSource	Gets or sets the binding for this data template, which indicates where to find the collection that represents the next level in the data hierarchy.
▤	ItemTemplate	Gets or sets the **System.Windows.DataTemplate** to apply to items at the current hierarchical level in a HierarchicalDataTemplate.

Top

See Also

Reference

HierarchicalDataTemplate Class
C1.Silverlight.OrgChart Namespace

ItemsSource Property

Gets or sets the binding for this data template, which indicates where to find the collection that represents the next level in the data hierarchy.

Syntax

Visual Basic (Declaration)

```
Public Property ItemsSource As System.Windows.Data.Binding
```

C#

```
public System.Windows.Data.Binding ItemsSource {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

HierarchicalDataTemplate Class
HierarchicalDataTemplate Members

ItemTemplate Property

Gets or sets the **System.Windows.DataTemplate** to apply to items at the current hierarchical level in a HierarchicalDataTemplate.

Syntax

Visual Basic (Declaration)

```
Public Property ItemTemplate As System.Windows.DataTemplate
```

C#

```
public System.Windows.DataTemplate ItemTemplate {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

HierarchicalDataTemplate Class
HierarchicalDataTemplate Members

C1.WPF.OrgChart.4 Assembly

Namespaces

C1.WPF.OrgChart Namespace

Overview

Classes

	Class	Description
	C1OrgChart	Control that displays Organizational Charts.

See Also

Reference

C1.WPF.OrgChart.4 Assembly

Classes

C1OrgChart

Control that displays Organizational Charts.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class C1OrgChart
   Inherits System.Windows.Controls.ItemsControl
```

C#

```
public class C1OrgChart : System.Windows.Controls.ItemsControl
```

Remarks

To use this control, set the following main properties:

ItemsSource : a hierarchical list of items to be shown on the chart.

ItemTemplate : template used to convert the items into visuals.

ChildItemsPath : property that contains the child items.

Inheritance Hierarchy

```
System.Object
   System.Windows.Threading.DispatcherObject
      System.Windows.DependencyObject
         System.Windows.Media.Visual
            System.Windows.UIElement
               System.Windows.FrameworkElement
                  System.Windows.Controls.Control
                     System.Windows.Controls.ItemsControl
                        C1.WPF.OrgChart.C1OrgChart
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Members
C1.WPF.OrgChart Namespace

Overview

Control that displays Organizational Charts.

Object Model

Syntax

Visual Basic (Declaration)

```
Public Class C1OrgChart
   Inherits System.Windows.Controls.ItemsControl
```

C#

```
public class C1OrgChart : System.Windows.Controls.ItemsControl
```

Remarks

To use this control, set the following main properties:

ItemsSource : a hierarchical list of items to be shown on the chart.

ItemTemplate : template used to convert the items into visuals.

ChildItemsPath : property that contains the child items.

Inheritance Hierarchy

System.Object
  System.Windows.Threading.DispatcherObject
    System.Windows.DependencyObject
      System.Windows.Media.Visual
        System.Windows.UIElement
          System.Windows.FrameworkElement
            System.Windows.Controls.Control
              System.Windows.Controls.ItemsControl
                C1.WPF.OrgChart.C1OrgChart

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Members
C1.WPF.OrgChart Namespace

Members

Fields Properties Methods Events

The following tables list the members exposed by C1OrgChart.

Public Constructors

	Name	Description
=◆	C1OrgChart Constructor	Initializes a new instance of a C1OrgChart.

Top

Public Fields

	Name	Description

◆ S	ChildItemsPathProperty	Identifies the ChildItemsPath dependency property.
◆ S	ChildSpacingProperty	Identifies the ChildSpacing dependency property.
◆ S	ConnectorDashArrayProperty	Identifies the ConnectorDashArray dependency property.
◆ S	ConnectorDashCapProperty	Identifies the ConnectorDashCap dependency property.
◆ S	ConnectorDashOffsetProperty	Identifies the ConnectorDashOffset dependency property.
◆ S	ConnectorEndLineCapProperty	Identifies the ConnectorEndLineCap dependency property.
◆ S	ConnectorStartLineCapProperty	Identifies the ConnectorStartLineCap dependency property.
◆ S	ConnectorStrokeProperty	Identifies the ConnectorStroke dependency property.
◆ S	ConnectorThicknessProperty	Identifies the ConnectorThickness dependency property.
◆ S	HeaderProperty	Identifies the Header dependency property.
◆ S	IsCollapsedProperty	Identifies the IsCollapsed dependency property.
◆ S	OrientationProperty	Identifies the Orientation dependency property.

Top

Public Properties

	Name	Description
▤	ActualHeight	(Inherited from System.Windows.FrameworkElement)
▤	ActualWidth	(Inherited from System.Windows.FrameworkElement)
▤	AllowDrop	(Inherited from System.Windows.UIElement)

▤	AlternationCount	(Inherited from System.Windows.Controls.ItemsControl)
▤	AreAnyTouchesCaptured	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesCapturedWithin	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesDirectlyOver	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesOver	(Inherited from System.Windows.UIElement)
▤	Background	(Inherited from System.Windows.Controls.Control)
▤	BindingGroup	(Inherited from System.Windows.FrameworkElement)
▤	BitmapEffect	(Inherited from System.Windows.UIElement)
▤	BitmapEffectInput	(Inherited from System.Windows.UIElement)
▤	BorderBrush	(Inherited from System.Windows.Controls.Control)
▤	BorderThickness	(Inherited from System.Windows.Controls.Control)
▤	CacheMode	(Inherited from System.Windows.UIElement)
▤	ChildItemsPath	Gets or sets the name of the property that contains the collection of child objects.
▤	ChildNodes	Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.
▤	ChildSpacing	Gets or sets the spacing between items, in pixels.
▤	Clip	(Inherited from System.Windows.UIElement)
▤	ClipToBounds	(Inherited from System.Windows.UIElement)
▤	CommandBindings	(Inherited from System.Windows.UIElement)
▤	ConnectorDashArray	Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.

▤	ConnectorDashCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.
▤	ConnectorDashOffset	Gets or sets the distance within the dash pattern where a dash begins.
▤	ConnectorEndLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.
▤	ConnectorStartLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.
▤	ConnectorStroke	Gets or sets the brush used to paint the connecting lines.
▤	ConnectorThickness	Gets or sets the thickness of the connecting lines.
▤	ContextMenu	(Inherited from System.Windows.FrameworkElement)
▤	Cursor	(Inherited from System.Windows.FrameworkElement)
▤	DataContext	(Inherited from System.Windows.FrameworkElement)
▤	DependencyObjectType	(Inherited from System.Windows.DependencyObject)
▤	DesiredSize	(Inherited from System.Windows.UIElement)
▤	Dispatcher	(Inherited from System.Windows.Threading.DispatcherObject)
▤	DisplayMemberPath	(Inherited from System.Windows.Controls.ItemsControl)
▤	Effect	(Inherited from System.Windows.UIElement)
▤	FlowDirection	(Inherited from System.Windows.FrameworkElement)
▤	Focusable	(Inherited from System.Windows.UIElement)
▤	FocusVisualStyle	(Inherited from System.Windows.FrameworkElement)

▤	FontFamily	(Inherited from System.Windows.Controls.Control)
▤	FontSize	(Inherited from System.Windows.Controls.Control)
▤	FontStretch	(Inherited from System.Windows.Controls.Control)
▤	FontStyle	(Inherited from System.Windows.Controls.Control)
▤	FontWeight	(Inherited from System.Windows.Controls.Control)
▤	ForceCursor	(Inherited from System.Windows.FrameworkElement)
▤	Foreground	(Inherited from System.Windows.Controls.Control)
▤	GroupStyle	(Inherited from System.Windows.Controls.ItemsControl)
▤	GroupStyleSelector	(Inherited from System.Windows.Controls.ItemsControl)
▤	HasAnimatedProperties	(Inherited from System.Windows.UIElement)
▤	HasItems	(Inherited from System.Windows.Controls.ItemsControl)
▤	Header	Gets or sets the parent item for this C1OrgChart.
▤	Height	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	InputBindings	(Inherited from System.Windows.UIElement)
▤	InputScope	(Inherited from System.Windows.FrameworkElement)
▤	IsArrangeValid	(Inherited from System.Windows.UIElement)
▤	IsCollapsed	Gets or sets a value that determines whether the node is currently displaying its children.
▤	IsEnabled	(Inherited from System.Windows.UIElement)

	IsFocused	(Inherited from System.Windows.UIElement)
	IsGrouping	(Inherited from System.Windows.Controls.ItemsControl)
	IsHitTestVisible	(Inherited from System.Windows.UIElement)
	IsInitialized	(Inherited from System.Windows.FrameworkElement)
	IsInputMethodEnabled	(Inherited from System.Windows.UIElement)
	IsKeyboardFocused	(Inherited from System.Windows.UIElement)
	IsKeyboardFocusWithin	(Inherited from System.Windows.UIElement)
	IsLoaded	(Inherited from System.Windows.FrameworkElement)
	IsManipulationEnabled	(Inherited from System.Windows.UIElement)
	IsMeasureValid	(Inherited from System.Windows.UIElement)
	IsMouseCaptured	(Inherited from System.Windows.UIElement)
	IsMouseCaptureWithin	(Inherited from System.Windows.UIElement)
	IsMouseDirectlyOver	(Inherited from System.Windows.UIElement)
	IsMouseOver	(Inherited from System.Windows.UIElement)
	IsSealed	(Inherited from System.Windows.DependencyObject)
	IsStylusCaptured	(Inherited from System.Windows.UIElement)
	IsStylusCaptureWithin	(Inherited from System.Windows.UIElement)
	IsStylusDirectlyOver	(Inherited from System.Windows.UIElement)
	IsStylusOver	(Inherited from System.Windows.UIElement)
	IsTabStop	(Inherited from System.Windows.Controls.Control)

▤	IsTextSearchCaseSensitive	(Inherited from System.Windows.Controls.ItemsControl)
▤	IsTextSearchEnabled	(Inherited from System.Windows.Controls.ItemsControl)
▤	IsVisible	(Inherited from System.Windows.UIElement)
▤	ItemBindingGroup	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerGenerator	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerStyle	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerStyleSelector	(Inherited from System.Windows.Controls.ItemsControl)
▤	Items	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsPanel	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsSource	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemStringFormat	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemTemplate	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemTemplateSelector	(Inherited from System.Windows.Controls.ItemsControl)
▤	Language	(Inherited from System.Windows.FrameworkElement)
▤	LayoutTransform	(Inherited from System.Windows.FrameworkElement)
▤	Margin	(Inherited from System.Windows.FrameworkElement)
▤	MaxHeight	(Inherited from System.Windows.FrameworkElement)
▤	MaxWidth	(Inherited from System.Windows.FrameworkElement)
▤	MinHeight	(Inherited from System.Windows.FrameworkElement)
▤	MinWidth	(Inherited from System.Windows.FrameworkElement)

▤	Name	(Inherited from System.Windows.FrameworkElement)
▤	Opacity	(Inherited from System.Windows.UIElement)
▤	OpacityMask	(Inherited from System.Windows.UIElement)
▤	Orientation	Gets or sets the direction in which child elements are laid out within the control.
▤	OverridesDefaultStyle	(Inherited from System.Windows.FrameworkElement)
▤	Padding	(Inherited from System.Windows.Controls.Control)
▤	Parent	(Inherited from System.Windows.FrameworkElement)
▤	PersistId	(Inherited from System.Windows.UIElement)
▤	RenderSize	(Inherited from System.Windows.UIElement)
▤	RenderTransform	(Inherited from System.Windows.UIElement)
▤	RenderTransformOrigin	(Inherited from System.Windows.UIElement)
▤	Resources	(Inherited from System.Windows.FrameworkElement)
▤	SnapsToDevicePixels	(Inherited from System.Windows.UIElement)
▤	Style	(Inherited from System.Windows.FrameworkElement)
▤	TabIndex	(Inherited from System.Windows.Controls.Control)
▤	Tag	(Inherited from System.Windows.FrameworkElement)
▤	Template	(Inherited from System.Windows.Controls.Control)
▤	TemplatedParent	(Inherited from System.Windows.FrameworkElement)
▤	ToolTip	(Inherited from System.Windows.FrameworkElement)
▤	TouchesCaptured	(Inherited from System.Windows.UIElement)

	TouchesCapturedWithin	(Inherited from System.Windows.UIElement)
	TouchesDirectlyOver	(Inherited from System.Windows.UIElement)
	TouchesOver	(Inherited from System.Windows.UIElement)
	Triggers	(Inherited from System.Windows.FrameworkElement)
	Uid	(Inherited from System.Windows.UIElement)
	UseLayoutRounding	(Inherited from System.Windows.FrameworkElement)
	VerticalAlignment	(Inherited from System.Windows.FrameworkElement)
	VerticalContentAlignment	(Inherited from System.Windows.Controls.Control)
	Visibility	(Inherited from System.Windows.UIElement)
	Width	(Inherited from System.Windows.FrameworkElement)

Top

Public Methods

	Name	Description
	AddHandler	Overloaded. (Inherited from System.Windows.UIElement)
	AddToEventRoute	(Inherited from System.Windows.UIElement)
	ApplyAnimationClock	Overloaded. (Inherited from System.Windows.UIElement)
	ApplyTemplate	(Inherited from System.Windows.FrameworkElement)
	Arrange	(Inherited from System.Windows.UIElement)
	BeginAnimation	Overloaded. (Inherited from System.Windows.UIElement)
	BeginInit	(Inherited from System.Windows.Controls.ItemsControl)
	BeginStoryboard	Overloaded. (Inherited from

		System.Windows.FrameworkElement)
≡◆	BringIntoView	Overloaded. (Inherited from System.Windows.FrameworkElement)
≡◆	CaptureMouse	(Inherited from System.Windows.UIElement)
≡◆	CaptureStylus	(Inherited from System.Windows.UIElement)
≡◆	CaptureTouch	(Inherited from System.Windows.UIElement)
≡◆	ClearValue	Overloaded. (Inherited from System.Windows.DependencyObject)
≡◆	CoerceValue	(Inherited from System.Windows.DependencyObject)
≡◆	ContainerFromElement	(Inherited from System.Windows.Controls.ItemsControl)
≡◆	EndInit	(Inherited from System.Windows.Controls.ItemsControl)
≡◆	Equals	(Inherited from System.Windows.DependencyObject)
≡◆	FindCommonVisualAncestor	(Inherited from System.Windows.Media.Visual)
≡◆	FindName	(Inherited from System.Windows.FrameworkElement)
≡◆	FindResource	(Inherited from System.Windows.FrameworkElement)
≡◆	Focus	(Inherited from System.Windows.UIElement)
≡◆	GetAnimationBaseValue	(Inherited from System.Windows.UIElement)
≡◆	GetBindingExpression	(Inherited from System.Windows.FrameworkElement)
≡◆	GetHashCode	(Inherited from System.Windows.DependencyObject)
≡◆	GetLocalValueEnumerator	(Inherited from System.Windows.DependencyObject)
≡◆	GetValue	(Inherited from System.Windows.DependencyObject)

≡◆	InputHitTest	(Inherited from System.Windows.UIElement)
≡◆	InvalidateArrange	(Inherited from System.Windows.UIElement)
≡◆	InvalidateMeasure	(Inherited from System.Windows.UIElement)
≡◆	InvalidateProperty	(Inherited from System.Windows.DependencyObject)
≡◆	InvalidateVisual	(Inherited from System.Windows.UIElement)
≡◆	IsAncestorOf	(Inherited from System.Windows.Media.Visual)
≡◆	IsDescendantOf	(Inherited from System.Windows.Media.Visual)
≡◆	IsItemItsOwnContainer	(Inherited from System.Windows.Controls.ItemsControl)
≡◆	Measure	(Inherited from System.Windows.UIElement)
≡◆	MoveFocus	(Inherited from System.Windows.FrameworkElement)
≡◆	OnApplyTemplate	Overridden. Get element references from control template.
≡◆	PointFromScreen	(Inherited from System.Windows.Media.Visual)
≡◆	PointToScreen	(Inherited from System.Windows.Media.Visual)
≡◆	PredictFocus	(Inherited from System.Windows.FrameworkElement)
≡◆	RaiseEvent	(Inherited from System.Windows.UIElement)
≡◆	ReadLocalValue	(Inherited from System.Windows.DependencyObject)
≡◆	RegisterName	(Inherited from System.Windows.FrameworkElement)
≡◆	ReleaseAllTouchCaptures	(Inherited from System.Windows.UIElement)
≡◆	ReleaseMouseCapture	(Inherited from System.Windows.UIElement)
≡◆	ReleaseStylusCapture	(Inherited from System.Windows.UIElement)

	ReleaseTouchCapture	(Inherited from System.Windows.UIElement)
	RemoveHandler	(Inherited from System.Windows.UIElement)
	SetBinding	Overloaded. (Inherited from System.Windows.FrameworkElement)
	SetCurrentValue	(Inherited from System.Windows.DependencyObject)
	SetResourceReference	(Inherited from System.Windows.FrameworkElement)
	SetValue	Overloaded. (Inherited from System.Windows.DependencyObject)
	ToString	(Inherited from System.Windows.Controls.ItemsControl)
	TransformToAncestor	Overloaded. (Inherited from System.Windows.Media.Visual)
	TransformToDescendant	(Inherited from System.Windows.Media.Visual)
	TransformToVisual	(Inherited from System.Windows.Media.Visual)
	TranslatePoint	(Inherited from System.Windows.UIElement)
	TryFindResource	(Inherited from System.Windows.FrameworkElement)
	UnregisterName	(Inherited from System.Windows.FrameworkElement)
	UpdateDefaultStyle	(Inherited from System.Windows.FrameworkElement)
	UpdateLayout	(Inherited from System.Windows.UIElement)

Top

Public Events

	Name	Description
	ContextMenuClosing	(Inherited from System.Windows.FrameworkElement)
	ContextMenuOpening	(Inherited from System.Windows.FrameworkElement)

⚡	DataContextChanged	(Inherited from System.Windows.FrameworkElement)
⚡	DragEnter	(Inherited from System.Windows.UIElement)
⚡	DragLeave	(Inherited from System.Windows.UIElement)
⚡	DragOver	(Inherited from System.Windows.UIElement)
⚡	Drop	(Inherited from System.Windows.UIElement)
⚡	FocusableChanged	(Inherited from System.Windows.UIElement)
⚡	GiveFeedback	(Inherited from System.Windows.UIElement)
⚡	GotFocus	(Inherited from System.Windows.UIElement)
⚡	GotKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	GotMouseCapture	(Inherited from System.Windows.UIElement)
⚡	GotStylusCapture	(Inherited from System.Windows.UIElement)
⚡	GotTouchCapture	(Inherited from System.Windows.UIElement)
⚡	Initialized	(Inherited from System.Windows.FrameworkElement)
⚡	IsEnabledChanged	(Inherited from System.Windows.UIElement)
⚡	IsHitTestVisibleChanged	(Inherited from System.Windows.UIElement)
⚡	IsKeyboardFocusedChanged	(Inherited from System.Windows.UIElement)
⚡	IsKeyboardFocusWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseCapturedChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseCaptureWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsMouseDirectlyOverChanged	(Inherited from System.Windows.UIElement)

⚡	IsStylusCapturedChanged	(Inherited from System.Windows.UIElement)
⚡	IsStylusCaptureWithinChanged	(Inherited from System.Windows.UIElement)
⚡	IsStylusDirectlyOverChanged	(Inherited from System.Windows.UIElement)
⚡	IsVisibleChanged	(Inherited from System.Windows.UIElement)
⚡	KeyDown	(Inherited from System.Windows.UIElement)
⚡	KeyUp	(Inherited from System.Windows.UIElement)
⚡	LayoutUpdated	(Inherited from System.Windows.UIElement)
⚡	Loaded	(Inherited from System.Windows.FrameworkElement)
⚡	LostFocus	(Inherited from System.Windows.UIElement)
⚡	LostKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	LostMouseCapture	(Inherited from System.Windows.UIElement)
⚡	LostStylusCapture	(Inherited from System.Windows.UIElement)
⚡	LostTouchCapture	(Inherited from System.Windows.UIElement)
⚡	ManipulationBoundaryFeedback	(Inherited from System.Windows.UIElement)
⚡	ManipulationCompleted	(Inherited from System.Windows.UIElement)
⚡	ManipulationDelta	(Inherited from System.Windows.UIElement)
⚡	ManipulationInertiaStarting	(Inherited from System.Windows.UIElement)
⚡	ManipulationStarted	(Inherited from System.Windows.UIElement)
⚡	ManipulationStarting	(Inherited from System.Windows.UIElement)
⚡	MouseDoubleClick	(Inherited from System.Windows.Controls.Control)

⚡	MouseDown	(Inherited from System.Windows.UIElement)
⚡	MouseEnter	(Inherited from System.Windows.UIElement)
⚡	MouseLeave	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseLeftButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseMove	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonDown	(Inherited from System.Windows.UIElement)
⚡	MouseRightButtonUp	(Inherited from System.Windows.UIElement)
⚡	MouseUp	(Inherited from System.Windows.UIElement)
⚡	MouseWheel	(Inherited from System.Windows.UIElement)
⚡	PreviewDragEnter	(Inherited from System.Windows.UIElement)
⚡	PreviewDragLeave	(Inherited from System.Windows.UIElement)
⚡	PreviewDragOver	(Inherited from System.Windows.UIElement)
⚡	PreviewDrop	(Inherited from System.Windows.UIElement)
⚡	PreviewGiveFeedback	(Inherited from System.Windows.UIElement)
⚡	PreviewGotKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	PreviewKeyDown	(Inherited from System.Windows.UIElement)
⚡	PreviewKeyUp	(Inherited from System.Windows.UIElement)
⚡	PreviewLostKeyboardFocus	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseDoubleClick	(Inherited from System.Windows.Controls.Control)

⚡	PreviewMouseDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseLeftButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseLeftButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseMove	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseRightButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseRightButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseUp	(Inherited from System.Windows.UIElement)
⚡	PreviewMouseWheel	(Inherited from System.Windows.UIElement)
⚡	PreviewQueryContinueDrag	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusButtonDown	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusButtonUp	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusDown	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusInAirMove	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusInRange	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusMove	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusOutOfRange	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusSystemGesture	(Inherited from System.Windows.UIElement)
⚡	PreviewStylusUp	(Inherited from System.Windows.UIElement)
⚡	PreviewTextInput	(Inherited from System.Windows.UIElement)
⚡	PreviewTouchDown	(Inherited from System.Windows.UIElement)

⚡	PreviewTouchMove	(Inherited from System.Windows.UIElement)
⚡	PreviewTouchUp	(Inherited from System.Windows.UIElement)
⚡	QueryContinueDrag	(Inherited from System.Windows.UIElement)
⚡	QueryCursor	(Inherited from System.Windows.UIElement)
⚡	RequestBringIntoView	(Inherited from System.Windows.FrameworkElement)
⚡	SizeChanged	(Inherited from System.Windows.FrameworkElement)
⚡	SourceUpdated	(Inherited from System.Windows.FrameworkElement)
⚡	StylusButtonDown	(Inherited from System.Windows.UIElement)
⚡	StylusButtonUp	(Inherited from System.Windows.UIElement)
⚡	StylusDown	(Inherited from System.Windows.UIElement)
⚡	StylusEnter	(Inherited from System.Windows.UIElement)
⚡	StylusInAirMove	(Inherited from System.Windows.UIElement)
⚡	StylusInRange	(Inherited from System.Windows.UIElement)
⚡	StylusLeave	(Inherited from System.Windows.UIElement)
⚡	StylusMove	(Inherited from System.Windows.UIElement)
⚡	StylusOutOfRange	(Inherited from System.Windows.UIElement)
⚡	StylusSystemGesture	(Inherited from System.Windows.UIElement)
⚡	StylusUp	(Inherited from System.Windows.UIElement)
⚡	TargetUpdated	(Inherited from System.Windows.FrameworkElement)
⚡	TextInput	(Inherited from System.Windows.UIElement)

⚡	ToolTipClosing	(Inherited from System.Windows.FrameworkElement)
⚡	ToolTipOpening	(Inherited from System.Windows.FrameworkElement)
⚡	TouchDown	(Inherited from System.Windows.UIElement)
⚡	TouchEnter	(Inherited from System.Windows.UIElement)
⚡	TouchLeave	(Inherited from System.Windows.UIElement)
⚡	TouchMove	(Inherited from System.Windows.UIElement)
⚡	TouchUp	(Inherited from System.Windows.UIElement)
⚡	Unloaded	(Inherited from System.Windows.FrameworkElement)

Top

See Also

Reference

C1OrgChart Class
C1.WPF.OrgChart Namespace

C1OrgChart Constructor

Initializes a new instance of a C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public Function New()
```

C#

```
public C1OrgChart()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Methods

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Methods

	Name	Description
◆	AddHandler	Overloaded. (Inherited from System.Windows.UIElement)
◆	AddToEventRoute	(Inherited from System.Windows.UIElement)
◆	ApplyAnimationClock	Overloaded. (Inherited from System.Windows.UIElement)
◆	ApplyTemplate	(Inherited from System.Windows.FrameworkElement)
◆	Arrange	(Inherited from System.Windows.UIElement)
◆	BeginAnimation	Overloaded. (Inherited from System.Windows.UIElement)
◆	BeginInit	(Inherited from System.Windows.Controls.ItemsControl)
◆	BeginStoryboard	Overloaded. (Inherited from System.Windows.FrameworkElement)
◆	BringIntoView	Overloaded. (Inherited from System.Windows.FrameworkElement)
◆	CaptureMouse	(Inherited from System.Windows.UIElement)
◆	CaptureStylus	(Inherited from System.Windows.UIElement)
◆	CaptureTouch	(Inherited from System.Windows.UIElement)
◆	ClearValue	Overloaded. (Inherited from System.Windows.DependencyObject)

	CoerceValue	(Inherited from System.Windows.DependencyObject)
	ContainerFromElement	(Inherited from System.Windows.Controls.ItemsControl)
	EndInit	(Inherited from System.Windows.Controls.ItemsControl)
	Equals	(Inherited from System.Windows.DependencyObject)
	FindCommonVisualAncestor	(Inherited from System.Windows.Media.Visual)
	FindName	(Inherited from System.Windows.FrameworkElement)
	FindResource	(Inherited from System.Windows.FrameworkElement)
	Focus	(Inherited from System.Windows.UIElement)
	GetAnimationBaseValue	(Inherited from System.Windows.UIElement)
	GetBindingExpression	(Inherited from System.Windows.FrameworkElement)
	GetHashCode	(Inherited from System.Windows.DependencyObject)
	GetLocalValueEnumerator	(Inherited from System.Windows.DependencyObject)
	GetValue	(Inherited from System.Windows.DependencyObject)
	InputHitTest	(Inherited from System.Windows.UIElement)
	InvalidateArrange	(Inherited from System.Windows.UIElement)
	InvalidateMeasure	(Inherited from System.Windows.UIElement)
	InvalidateProperty	(Inherited from System.Windows.DependencyObject)
	InvalidateVisual	(Inherited from System.Windows.UIElement)
	IsAncestorOf	(Inherited from System.Windows.Media.Visual)
	IsDescendantOf	(Inherited from System.Windows.Media.Visual)

≡◆	IsItemItsOwnContainer	(Inherited from System.Windows.Controls.ItemsControl)
≡◆	Measure	(Inherited from System.Windows.UIElement)
≡◆	MoveFocus	(Inherited from System.Windows.FrameworkElement)
≡◆	OnApplyTemplate	Overridden. Get element references from control template.
≡◆	PointFromScreen	(Inherited from System.Windows.Media.Visual)
≡◆	PointToScreen	(Inherited from System.Windows.Media.Visual)
≡◆	PredictFocus	(Inherited from System.Windows.FrameworkElement)
≡◆	RaiseEvent	(Inherited from System.Windows.UIElement)
≡◆	ReadLocalValue	(Inherited from System.Windows.DependencyObject)
≡◆	RegisterName	(Inherited from System.Windows.FrameworkElement)
≡◆	ReleaseAllTouchCaptures	(Inherited from System.Windows.UIElement)
≡◆	ReleaseMouseCapture	(Inherited from System.Windows.UIElement)
≡◆	ReleaseStylusCapture	(Inherited from System.Windows.UIElement)
≡◆	ReleaseTouchCapture	(Inherited from System.Windows.UIElement)
≡◆	RemoveHandler	(Inherited from System.Windows.UIElement)
≡◆	SetBinding	Overloaded. (Inherited from System.Windows.FrameworkElement)
≡◆	SetCurrentValue	(Inherited from System.Windows.DependencyObject)
≡◆	SetResourceReference	(Inherited from System.Windows.FrameworkElement)
≡◆	SetValue	Overloaded. (Inherited from System.Windows.DependencyObject)

◆	ToString	(Inherited from System.Windows.Controls.ItemsControl)
◆	TransformToAncestor	Overloaded. (Inherited from System.Windows.Media.Visual)
◆	TransformToDescendant	(Inherited from System.Windows.Media.Visual)
◆	TransformToVisual	(Inherited from System.Windows.Media.Visual)
◆	TranslatePoint	(Inherited from System.Windows.UIElement)
◆	TryFindResource	(Inherited from System.Windows.FrameworkElement)
◆	UnregisterName	(Inherited from System.Windows.FrameworkElement)
◆	UpdateDefaultStyle	(Inherited from System.Windows.FrameworkElement)
◆	UpdateLayout	(Inherited from System.Windows.UIElement)

Top

See Also

Reference

C1OrgChart Class
C1.WPF.OrgChart Namespace

OnApplyTemplate Method

Get element references from control template.

Syntax

Visual Basic (Declaration)

```
Public Overrides Sub OnApplyTemplate()
```

C#

```
public override void OnApplyTemplate()
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Properties

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Properties

	Name	Description
▤	ActualHeight	(Inherited from System.Windows.FrameworkElement)
▤	ActualWidth	(Inherited from System.Windows.FrameworkElement)
▤	AllowDrop	(Inherited from System.Windows.UIElement)
▤	AlternationCount	(Inherited from System.Windows.Controls.ItemsControl)
▤	AreAnyTouchesCaptured	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesCapturedWithin	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesDirectlyOver	(Inherited from System.Windows.UIElement)
▤	AreAnyTouchesOver	(Inherited from System.Windows.UIElement)
▤	Background	(Inherited from System.Windows.Controls.Control)
▤	BindingGroup	(Inherited from System.Windows.FrameworkElement)
▤	BitmapEffect	(Inherited from System.Windows.UIElement)
▤	BitmapEffectInput	(Inherited from System.Windows.UIElement)
▤	BorderBrush	(Inherited from System.Windows.Controls.Control)
▤	BorderThickness	(Inherited from System.Windows.Controls.Control)

▤	CacheMode	(Inherited from System.Windows.UIElement)
▤	ChildItemsPath	Gets or sets the name of the property that contains the collection of child objects.
▤	ChildNodes	Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.
▤	ChildSpacing	Gets or sets the spacing between items, in pixels.
▤	Clip	(Inherited from System.Windows.UIElement)
▤	ClipToBounds	(Inherited from System.Windows.UIElement)
▤	CommandBindings	(Inherited from System.Windows.UIElement)
▤	ConnectorDashArray	Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.
▤	ConnectorDashCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.
▤	ConnectorDashOffset	Gets or sets the distance within the dash pattern where a dash begins.
▤	ConnectorEndLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.
▤	ConnectorStartLineCap	Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.
▤	ConnectorStroke	Gets or sets the brush used to paint the connecting lines.
▤	ConnectorThickness	Gets or sets the thickness of the connecting lines.
▤	ContextMenu	(Inherited from System.Windows.FrameworkElement)
▤	Cursor	(Inherited from System.Windows.FrameworkElement)

▤	DataContext	(Inherited from System.Windows.FrameworkElement)
▤	DependencyObjectType	(Inherited from System.Windows.DependencyObject)
▤	DesiredSize	(Inherited from System.Windows.UIElement)
▤	Dispatcher	(Inherited from System.Windows.Threading.DispatcherObject)
▤	DisplayMemberPath	(Inherited from System.Windows.Controls.ItemsControl)
▤	Effect	(Inherited from System.Windows.UIElement)
▤	FlowDirection	(Inherited from System.Windows.FrameworkElement)
▤	Focusable	(Inherited from System.Windows.UIElement)
▤	FocusVisualStyle	(Inherited from System.Windows.FrameworkElement)
▤	FontFamily	(Inherited from System.Windows.Controls.Control)
▤	FontSize	(Inherited from System.Windows.Controls.Control)
▤	FontStretch	(Inherited from System.Windows.Controls.Control)
▤	FontStyle	(Inherited from System.Windows.Controls.Control)
▤	FontWeight	(Inherited from System.Windows.Controls.Control)
▤	ForceCursor	(Inherited from System.Windows.FrameworkElement)
▤	Foreground	(Inherited from System.Windows.Controls.Control)
▤	GroupStyle	(Inherited from System.Windows.Controls.ItemsControl)
▤	GroupStyleSelector	(Inherited from System.Windows.Controls.ItemsControl)
▤	HasAnimatedProperties	(Inherited from System.Windows.UIElement)
▤	HasItems	(Inherited from System.Windows.Controls.ItemsControl)

▤	Header	Gets or sets the parent item for this C1OrgChart.
▤	Height	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	HorizontalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	InputBindings	(Inherited from System.Windows.UIElement)
▤	InputScope	(Inherited from System.Windows.FrameworkElement)
▤	IsArrangeValid	(Inherited from System.Windows.UIElement)
▤	IsCollapsed	Gets or sets a value that determines whether the node is currently displaying its children.
▤	IsEnabled	(Inherited from System.Windows.UIElement)
▤	IsFocused	(Inherited from System.Windows.UIElement)
▤	IsGrouping	(Inherited from System.Windows.Controls.ItemsControl)
▤	IsHitTestVisible	(Inherited from System.Windows.UIElement)
▤	IsInitialized	(Inherited from System.Windows.FrameworkElement)
▤	IsInputMethodEnabled	(Inherited from System.Windows.UIElement)
▤	IsKeyboardFocused	(Inherited from System.Windows.UIElement)
▤	IsKeyboardFocusWithin	(Inherited from System.Windows.UIElement)
▤	IsLoaded	(Inherited from System.Windows.FrameworkElement)
▤	IsManipulationEnabled	(Inherited from System.Windows.UIElement)
▤	IsMeasureValid	(Inherited from System.Windows.UIElement)
▤	IsMouseCaptured	(Inherited from System.Windows.UIElement)

▤	IsMouseCaptureWithin	(Inherited from System.Windows.UIElement)
▤	IsMouseDirectlyOver	(Inherited from System.Windows.UIElement)
▤	IsMouseOver	(Inherited from System.Windows.UIElement)
▤	IsSealed	(Inherited from System.Windows.DependencyObject)
▤	IsStylusCaptured	(Inherited from System.Windows.UIElement)
▤	IsStylusCaptureWithin	(Inherited from System.Windows.UIElement)
▤	IsStylusDirectlyOver	(Inherited from System.Windows.UIElement)
▤	IsStylusOver	(Inherited from System.Windows.UIElement)
▤	IsTabStop	(Inherited from System.Windows.Controls.Control)
▤	IsTextSearchCaseSensitive	(Inherited from System.Windows.Controls.ItemsControl)
▤	IsTextSearchEnabled	(Inherited from System.Windows.Controls.ItemsControl)
▤	IsVisible	(Inherited from System.Windows.UIElement)
▤	ItemBindingGroup	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerGenerator	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerStyle	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemContainerStyleSelector	(Inherited from System.Windows.Controls.ItemsControl)
▤	Items	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsPanel	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemsSource	(Inherited from System.Windows.Controls.ItemsControl)
▤	ItemStringFormat	(Inherited from System.Windows.Controls.ItemsControl)

	ItemTemplate	(Inherited from System.Windows.Controls.ItemsControl)
	ItemTemplateSelector	(Inherited from System.Windows.Controls.ItemsControl)
	Language	(Inherited from System.Windows.FrameworkElement)
	LayoutTransform	(Inherited from System.Windows.FrameworkElement)
	Margin	(Inherited from System.Windows.FrameworkElement)
	MaxHeight	(Inherited from System.Windows.FrameworkElement)
	MaxWidth	(Inherited from System.Windows.FrameworkElement)
	MinHeight	(Inherited from System.Windows.FrameworkElement)
	MinWidth	(Inherited from System.Windows.FrameworkElement)
	Name	(Inherited from System.Windows.FrameworkElement)
	Opacity	(Inherited from System.Windows.UIElement)
	OpacityMask	(Inherited from System.Windows.UIElement)
	Orientation	Gets or sets the direction in which child elements are laid out within the control.
	OverridesDefaultStyle	(Inherited from System.Windows.FrameworkElement)
	Padding	(Inherited from System.Windows.Controls.Control)
	Parent	(Inherited from System.Windows.FrameworkElement)
	PersistId	(Inherited from System.Windows.UIElement)
	RenderSize	(Inherited from System.Windows.UIElement)
	RenderTransform	(Inherited from System.Windows.UIElement)
	RenderTransformOrigin	(Inherited from System.Windows.UIElement)

▤	Resources	(Inherited from System.Windows.FrameworkElement)
▤	SnapsToDevicePixels	(Inherited from System.Windows.UIElement)
▤	Style	(Inherited from System.Windows.FrameworkElement)
▤	TabIndex	(Inherited from System.Windows.Controls.Control)
▤	Tag	(Inherited from System.Windows.FrameworkElement)
▤	Template	(Inherited from System.Windows.Controls.Control)
▤	TemplatedParent	(Inherited from System.Windows.FrameworkElement)
▤	ToolTip	(Inherited from System.Windows.FrameworkElement)
▤	TouchesCaptured	(Inherited from System.Windows.UIElement)
▤	TouchesCapturedWithin	(Inherited from System.Windows.UIElement)
▤	TouchesDirectlyOver	(Inherited from System.Windows.UIElement)
▤	TouchesOver	(Inherited from System.Windows.UIElement)
▤	Triggers	(Inherited from System.Windows.FrameworkElement)
▤	Uid	(Inherited from System.Windows.UIElement)
▤	UseLayoutRounding	(Inherited from System.Windows.FrameworkElement)
▤	VerticalAlignment	(Inherited from System.Windows.FrameworkElement)
▤	VerticalContentAlignment	(Inherited from System.Windows.Controls.Control)
▤	Visibility	(Inherited from System.Windows.UIElement)
▤	Width	(Inherited from System.Windows.FrameworkElement)

Top

See Also

Reference

C1OrgChart Class
C1.WPF.OrgChart Namespace

ChildItemsPath Property

Gets or sets the name of the property that contains the collection of child objects.

Syntax

Visual Basic (Declaration)

```
Public Property ChildItemsPath As System.String
```

C#

```
public System.string ChildItemsPath {get; set;}
```

Remarks

If you don't specify this property, the control will look for the first property that returns an enumeration of items of the same type as the root item.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildNodes Property

Gets a list of C1OrgChart objects that represent the child nodes under this C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public ReadOnly Property ChildNodes As System.Collections.Generic.IList(Of C1OrgChart)
```

C#

```
public System.Collections.Generic.IList<C1OrgChart> ChildNodes {get;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildSpacing Property

Gets or sets the spacing between items, in pixels.

Syntax

Visual Basic (Declaration)

```
Public Property ChildSpacing As System.Windows.Size
```

C#

```
public System.Windows.Size ChildSpacing {get; set;}
```

Remarks

In addition to changing this property, you may customize the spacing between items using the Margin property on your ItemTemplate.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashArray Property

Gets or sets the collection of values that define the pattern of dashes and gaps used to paint the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashArray As System.Windows.Media.DoubleCollection
```

C#

```
public System.Windows.Media.DoubleCollection ConnectorDashArray {get; set;}
```

Remarks

The array specifies alternating lengths to stroke and leave blank. Values are expressed as multiples of the **ConnectorWidth** property.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the ends of dashes in the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorDashCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashOffset Property

Gets or sets the distance within the dash pattern where a dash begins.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorDashOffset As System.Double
```

C#

```
public System.double ConnectorDashOffset {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorEndLineCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the end of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorEndLineCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorEndLineCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStartLineCap Property

Gets or sets a **System.Windows.Media.PenLineCap** that specifies how to paint the start of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorStartLineCap As System.Windows.Media.PenLineCap
```

C#

```
public System.Windows.Media.PenLineCap ConnectorStartLineCap {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStroke Property

Gets or sets the brush used to paint the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorStroke As System.Windows.Media.Brush
```

C#

```
public System.Windows.Media.Brush ConnectorStroke {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorThickness Property

Gets or sets the thickness of the connecting lines.

Syntax

Visual Basic (Declaration)

```
Public Property ConnectorThickness As System.Double
```

C#

```
public System.double ConnectorThickness {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Header Property

Gets or sets the parent item for this C1OrgChart.

Syntax

Visual Basic (Declaration)

```
Public Property Header As System.Object
```

C#

```
public System.object Header {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

IsCollapsed Property

Gets or sets a value that determines whether the node is currently displaying its children.

Syntax

Visual Basic (Declaration)

```
Public Property IsCollapsed As System.Boolean
```

C#

```
public System.bool IsCollapsed {get; set;}
```

Remarks

You may use this property to create organizational charts with nodes that can be collapsed or expanded by users. One simple way to do this is by adding a **System.Windows.Controls.CheckBox** element to the item template and binding its **CheckBox.IsChecked** property to the IsCollapsed property of the C1OrgChart that represents the given item. For example: `<c1:C1OrgChart> <c1:C1OrgChart.ItemTemplate> <DataTemplate> <StackPanel> <TextBlock Text="{Binding Name}" /> <CheckBox Margin="4 0" IsChecked="{Binding IsCollapsed, Mode=TwoWay, RelativeSource={RelativeSource AncestorType=c1:C1OrgChart}}" /> </StackPanel> </DataTemplate> </c1:C1OrgChart.ItemTemplate> </c1:C1OrgChart>` This XAML binds the **IsChecked** property of the **CheckBox** to the **IsCollapsed** property of the **C1OrgChart** control that contains the item. The user may then collapse or expand the chart by checking the **CheckBox**.

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Orientation Property

Gets or sets the direction in which child elements are laid out within the control.

Syntax

Visual Basic (Declaration)

```
Public Property Orientation As System.Windows.Controls.Orientation
```

C#

```
public System.Windows.Controls.Orientation Orientation {get; set;}
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

Fields

%%scrap%%

" -->

For a list of all members of this type, see C1OrgChart members.

Public Fields

	Name	Description
◆ S	ChildItemsPathProperty	Identifies the ChildItemsPath dependency property.
◆ S	ChildSpacingProperty	Identifies the ChildSpacing dependency property.
◆ S	ConnectorDashArrayProperty	Identifies the ConnectorDashArray dependency property.
◆ S	ConnectorDashCapProperty	Identifies the ConnectorDashCap dependency property.
◆ S	ConnectorDashOffsetProperty	Identifies the ConnectorDashOffset dependency property.
◆ S	ConnectorEndLineCapProperty	Identifies the ConnectorEndLineCap dependency property.

◆ S	ConnectorStartLineCapProperty	Identifies the ConnectorStartLineCap dependency property.
◆ S	ConnectorStrokeProperty	Identifies the ConnectorStroke dependency property.
◆ S	ConnectorThicknessProperty	Identifies the ConnectorThickness dependency property.
◆ S	HeaderProperty	Identifies the Header dependency property.
◆ S	IsCollapsedProperty	Identifies the IsCollapsed dependency property.
◆ S	OrientationProperty	Identifies the Orientation dependency property.

Top

See Also

Reference

C1OrgChart Class
C1.WPF.OrgChart Namespace

ChildItemsPathProperty Field

Identifies the ChildItemsPath dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ChildItemsPathProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ChildItemsPathProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ChildSpacingProperty Field

Identifies the ChildSpacing dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ChildSpacingProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty ChildSpacingProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashArrayProperty Field

Identifies the ConnectorDashArray dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashArrayProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashArrayProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashCapProperty Field

Identifies the ConnectorDashCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashCapProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorDashOffsetProperty Field

Identifies the ConnectorDashOffset dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorDashOffsetProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorDashOffsetProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorEndLineCapProperty Field

Identifies the ConnectorEndLineCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorEndLineCapProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty ConnectorEndLineCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStartLineCapProperty Field

Identifies the ConnectorStartLineCap dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorStartLineCapProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorStartLineCapProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorStrokeProperty Field

Identifies the ConnectorStroke dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorStrokeProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorStrokeProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

ConnectorThicknessProperty Field

Identifies the ConnectorThickness dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly ConnectorThicknessProperty As
System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty
ConnectorThicknessProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

HeaderProperty Field

Identifies the Header dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly HeaderProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty HeaderProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

IsCollapsedProperty Field

Identifies the IsCollapsed dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly IsCollapsedProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty IsCollapsedProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

OrientationProperty Field

Identifies the Orientation dependency property.

Syntax

Visual Basic (Declaration)

```
Public Shared ReadOnly OrientationProperty As System.Windows.DependencyProperty
```

C#

```
public static readonly System.Windows.DependencyProperty OrientationProperty
```

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

C1OrgChart Class
C1OrgChart Members

SmartAssembly.Attributes Namespace

Overview

Inheritance Hierarchy

Classes

	Class	Description
	PoweredByAttribute	

See Also

Reference

C1.WPF.OrgChart.4 Assembly

Classes

PoweredByAttribute

Object Model

Syntax

Visual Basic (Declaration)

```
Public NotInheritable Class PoweredByAttribute
   Inherits System.Attribute
```

C#

```
public sealed class PoweredByAttribute : System.Attribute
```

Inheritance Hierarchy

System.Object
  System.Attribute
    SmartAssembly.Attributes.PoweredByAttribute

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

PoweredByAttribute Members
SmartAssembly.Attributes Namespace

Overview

Object Model

Syntax

Visual Basic (Declaration)

```
Public NotInheritable Class PoweredByAttribute
   Inherits System.Attribute
```

C#

```
public sealed class PoweredByAttribute : System.Attribute
```

Inheritance Hierarchy

System.Object
  System.Attribute
    SmartAssembly.Attributes.PoweredByAttribute

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

PoweredByAttribute Members
SmartAssembly.Attributes Namespace

Members

Properties Methods

The following tables list the members exposed by PoweredByAttribute.

Public Constructors

	Name	Description
	PoweredByAttribute Constructor	

Top

Public Properties

	Name	Description
	TypeId	(Inherited from System.Attribute)

Top

Public Methods

	Name	Description
	Equals	(Inherited from System.Attribute)
	GetHashCode	(Inherited from System.Attribute)
	IsDefaultAttribute	(Inherited from System.Attribute)
	Match	(Inherited from System.Attribute)

Top

See Also

Reference

PoweredByAttribute Class
SmartAssembly.Attributes Namespace

PoweredByAttribute Constructor

Syntax

Visual Basic (Declaration)

```
Public Function New( _
   ByVal param1 As System.String _
)
```

C#

```
public PoweredByAttribute(
    System.string param1
)
```

Parameters

param1

Requirements

Target Platforms: Windows 7, Windows Vista SP1 or later, Windows XP SP3, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2

See Also

Reference

PoweredByAttribute Class
PoweredByAttribute Members