

Namespace Nova.Avalonia.UI.BarcodeGenerator

Classes

[BarcodeErrorEventArgs](#)

Event arguments for failed barcode generation.

[BarcodeGeneratedEventArgs](#)

Event arguments for successful barcode generation.

[BarcodeGenerator](#)

A templated control that generates and renders various barcode symbologies (QR Code, EAN-13, Code 128, etc.) using the ZXing library. Optimized for performance via BitMatrix/Bitmap caching and supports full accessibility as a semantic Image.

[BarcodeGeneratorAutomationPeer](#)

Exposes the [BarcodeGenerator](#) to UI automation (accessibility). Defines the control as an "Image" type so screen readers treat it appropriately.

Enums

[BarcodeSymbology](#)

Supported barcode formats.

[QRErrorCorrectionLevel](#)

Error correction levels for QR codes. Higher levels allow more damage recovery but reduce data capacity.

Class BarcodeErrorEventArgs

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

Event arguments for failed barcode generation.

```
public class BarcodeErrorEventArgs : EventArgs
```

Inheritance

[object](#) ← [EventArgs](#) ← BarcodeErrorEventArgs

Inherited Members

[EventArgs.Empty](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BarcodeErrorEventArgs(Exception)

```
public BarcodeErrorEventArgs(Exception exception)
```

Parameters

exception [Exception](#)

Properties

Exception

```
public Exception Exception { get; }
```

Property Value

[Exception](#)

Class BarcodeGeneratedEventArgs

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

Event arguments for successful barcode generation.

```
public class BarcodeGeneratedEventArgs : EventArgs
```

Inheritance

[object](#) ← [EventArgs](#) ← BarcodeGeneratedEventArgs

Inherited Members

[EventArgs.Empty](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BarcodeGeneratedEventArgs(BitMatrix)

```
public BarcodeGeneratedEventArgs(BitMatrix matrix)
```

Parameters

matrix BitMatrix

Properties

Matrix

```
public BitMatrix Matrix { get; }
```

Property Value

BitMatrix

Class BarcodeGenerator

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

A templated control that generates and renders various barcode symbologies (QR Code, EAN-13, Code 128, etc.) using the ZXing library. Optimized for performance via BitMatrix/Bitmap caching and supports full accessibility as a semantic Image.

```
public class BarcodeGenerator : TemplatedControl, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode,  
IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize,  
IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← TemplatedControl ← BarcodeGenerator

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue

Inherited Members

TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,
TemplatedControl.GetIsTemplateFocusTarget(Control) ,
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) , TemplatedControl.ApplyTemplate() ,
TemplatedControl.GetTemplateFocusTarget() ,
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,
TemplatedControl.BorderBrush , TemplatedControl.BorderThickness ,

TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,
TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(SizeChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerEventArgs) ,
InputElement.OnPointerCaptureLost(PointerEventArgs) ,
InputElement.OnPointerWheelChanged(PointerEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.HitTestVisible , InputElement.PointerOver , InputElement.TabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,

InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalContentAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.OnMeasureInvalidated() , Layoutable.OnVisualParentChanged(Visual, Visual) ,
Layoutable.Width , Layoutable.Height , Layoutable.MinWidth , Layoutable.MaxWidth ,
Layoutable.MinHeight , Layoutable.MaxHeight , Layoutable.Margin , Layoutable.HorizontalAlignment ,
Layoutable.VerticalAlignment , Layoutable.DesiredSize , Layoutable.IsMeasureValid ,
Layoutable.IsArrangeValid , Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged ,
Layoutable.LayoutUpdated , Visual.BoundsProperty , Visual.ClipToBoundsProperty ,
Visual.ClipProperty , Visual.VisibleProperty , Visual.OpacityProperty , Visual.OpacityMaskProperty ,
Visual.EffectProperty , Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,

StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,

AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

BackgroundBrushProperty

Defines the [BackgroundBrush](#) property.

```
public static readonly StyledProperty<IBrush> BackgroundBrushProperty
```

Field Value

StyledProperty<IBrush>

BarBrushProperty

Defines the [BarBrush](#) property.

```
public static readonly StyledProperty<IBrush> BarBrushProperty
```

Field Value

StyledProperty<IBrush>

ErrorCorrectionLevelProperty

Defines the [ErrorCorrectionLevel](#) property. Applies to QR and Aztec codes only.

```
public static readonly StyledProperty<QRErrorCorrectionLevel>  
ErrorCorrectionLevelProperty
```

Field Value

StyledProperty<[QRErrorCorrectionLevel](#)>

ErrorMessageProperty

Defines the [ErrorMessage](#) property.

```
public static readonly DirectProperty<BarcodeGenerator, string?>  
    > ErrorMessageProperty
```

Field Value

DirectProperty<[BarcodeGenerator](#), [string](#)>

LogoProperty

Defines the [Logo](#) property. Applies to QR and Aztec codes only.

```
public static readonly StyledProperty<IImage?> LogoProperty
```

Field Value

StyledProperty<IImage>

LogoSizePercentProperty

Defines the [LogoSizePercent](#) property. Applies to QR and Aztec codes only.

```
public static readonly StyledProperty<double> LogoSizePercentProperty
```

Field Value

StyledProperty<[double](#)>

QuietZoneProperty

Defines the [QuietZone](#) property.

```
public static readonly StyledProperty<int> QuietZoneProperty
```

Field Value

StyledProperty<[int](#)>

ShowTextProperty

Defines the [ShowText](#) property.

```
public static readonly StyledProperty<bool> ShowTextProperty
```

Field Value

StyledProperty<[bool](#)>

SymbologyProperty

Defines the [Symbology](#) property.

```
public static readonly StyledProperty<BarcodeSymbology> SymbologyProperty
```

Field Value

StyledProperty<[BarcodeSymbology](#)>

TextAlignmentProperty

Defines the [TextAlignment](#) property.

```
public static readonly StyledProperty<TextAlignment> TextAlignmentProperty
```

Field Value

StyledProperty<[TextAlignment](#)>

TextFontSizeProperty

Defines the [TextFontSize](#) property.

```
public static readonly StyledProperty<double> TextFontSizeProperty
```

Field Value

StyledProperty<[double](#)>

TextForegroundProperty

Defines the [TextForeground](#) property.

```
public static readonly StyledProperty<IBrush> TextForegroundProperty
```

Field Value

StyledProperty<IBrush>

TextMarginProperty

Defines the [TextMargin](#) property.

```
public static readonly StyledProperty<Thickness> TextMarginProperty
```

Field Value

StyledProperty<Thickness>

ValueProperty

Defines the [Value](#) property.

```
public static readonly StyledProperty<string> ValueProperty
```

Field Value

Properties

BackgroundBrush

Gets or sets the brush used for the background of the barcode area.

```
public IBrush BackgroundBrush { get; set; }
```

Property Value

IBrush

BarBrush

Gets or sets the brush used to draw the barcode bars/pixels.

```
public IBrush BarBrush { get; set; }
```

Property Value

IBrush

ErrorCorrectionLevel

Gets or sets the error correction level for QR and Aztec codes (L=7%, M=15%, Q=25%, H=30%).

Applies to: QR Code, Aztec only. Ignored for other symbologies.

```
public QRErrorCorrectionLevel ErrorCorrectionLevel { get; set; }
```

Property Value

[QRErrorCorrectionLevel](#)

ErrorMessage

Gets the last error message if barcode generation failed, or null if successful.

```
public string? ErrorMessage { get; }
```

Property Value

[string](#)

Logo

Gets or sets an optional logo/watermark image to overlay in the center of QR/Aztec codes. Requires ErrorCorrectionLevel of Q or H for best results.

Applies to: QR Code, Aztec only. Ignored for other symbologies.

```
public IImage? Logo { get; set; }
```

Property Value

[IImage](#)

LogoSizePercent

Gets or sets the size of the logo as a percentage of barcode size (0.1 to 0.4). Default is 0.25 (25%).

Applies to: QR Code, Aztec only. Ignored for other symbologies.

```
public double LogoSizePercent { get; set; }
```

Property Value

[double](#)

QuietZone

Gets or sets the margin (quiet zone) around the barcode in pixels/modules.

```
public int QuietZone { get; set; }
```

Property Value

[int](#)

ShowText

Gets or sets a value indicating whether to display the raw text value below the barcode.

```
public bool ShowText { get; set; }
```

Property Value

[bool](#)

Symbology

Gets or sets the type of barcode to generate (e.g., QRCode, Code128).

```
public BarcodeSymbology Symbology { get; set; }
```

Property Value

[BarcodeSymbology](#)

TextAlignment

Gets or sets the horizontal alignment of the caption text.

```
public TextAlignment TextAlignment { get; set; }
```

Property Value

TextFontSize

Gets or sets the font size for the caption text.

```
public double TextFontSize { get; set; }
```

Property Value

[double](#)

TextForeground

Gets or sets the brush used for the caption text.

```
public IBrush TextForeground { get; set; }
```

Property Value

IBrush

TextMargin

Gets or sets the margin around the caption text.

```
public Thickness TextMargin { get; set; }
```

Property Value

Thickness

Value

Gets or sets the string data to encode into the barcode.

```
public string Value { get; set; }
```

Property Value

[string](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e TemplateAppliedEventArgs

The event args.

OnCreateAutomationPeer()

Returns a new, type-specific Avalonia.Automation.Peers.AutomationPeer implementation for the control.

```
protected override AutomationPeer OnCreateAutomationPeer()
```

Returns

AutomationPeer

The type-specific Avalonia.Automation.Peers.AutomationPeer implementation.

Render(DrawingContext)

Renders the visual to a Avalonia.Media.DrawingContext.

```
public override void Render(DrawingContext context)
```

Parameters

context DrawingContext

The drawing context.

Events

BarcodeError

Occurs when barcode generation fails due to invalid data or constraints.

```
public event EventHandler<BarcodeErrorEventArgs>? BarcodeError
```

Event Type

[EventHandler](#)<BarcodeErrorEventArgs>

BarcodeGenerated

Occurs when a barcode has been successfully generated and encoded.

```
public event EventHandler<BarcodeGeneratedEventArgs>? BarcodeGenerated
```

Event Type

[EventHandler](#)<[BarcodeGeneratedEventArgs](#)>

Class BarcodeGeneratorAutomationPeer

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

Exposes the [BarcodeGenerator](#) to UI automation (accessibility). Defines the control as an "Image" type so screen readers treat it appropriately.

```
public class BarcodeGeneratorAutomationPeer : ControlAutomationPeer
```

Inheritance

[object](#) ← AutomationPeer ← ControlAutomationPeer ← BarcodeGeneratorAutomationPeer

Inherited Members

ControlAutomationPeer.GetOrCreate(Control) ,
ControlAutomationPeer.CreatePeerForElement(Control) ,
ControlAutomationPeer.FromElement(Control) , ControlAutomationPeer.BringIntoViewCore() ,
ControlAutomationPeer.GetOrCreateChildrenCore() , ControlAutomationPeer.GetChildrenCore() ,
ControlAutomationPeer.GetLabeledByCore() , ControlAutomationPeer.GetNameCore() ,
ControlAutomationPeer.GetHelpTextCore() , ControlAutomationPeer.GetHeadingLevelCore() ,
ControlAutomationPeer.GetParentCore() , ControlAutomationPeer.GetVisualRootCore() ,
ControlAutomationPeer.InvalidateChildren() , ControlAutomationPeer.InvalidateParent() ,
ControlAutomationPeer.ShowContextMenuCore() ,
ControlAutomationPeer.TrySetParent(AutomationPeer) ,
ControlAutomationPeer.GetAcceleratorKeyCore() , ControlAutomationPeer.GetAccessKeyCore() ,
ControlAutomationPeer.GetAutomationIdCore() ,
ControlAutomationPeer.GetBoundingClientRectCore() ,
ControlAutomationPeer.HasKeyboardFocusCore() , ControlAutomationPeer.IsContentElementCore() ,
ControlAutomationPeer.IsControlElementCore() , ControlAutomationPeer.IsEnabledCore() ,
ControlAutomationPeer.IsKeyboardFocusableCore() , ControlAutomationPeer.SetFocusCore() ,
ControlAutomationPeer.GetControlTypeOverrideCore() ,
ControlAutomationPeer.IsContentElementOverrideCore() ,
ControlAutomationPeer.IsControlElementOverrideCore() , ControlAutomationPeer.IsOffscreenCore() ,
ControlAutomationPeer.Owner , AutomationPeer.BringIntoView() ,
AutomationPeer.GetAcceleratorKey() , AutomationPeer.GetAccessKey() ,
AutomationPeer.GetAutomationControlType() , AutomationPeer.GetAutomationId() ,
AutomationPeer.GetBoundingClientRect() , AutomationPeer.GetChildren() ,
AutomationPeer.GetClassName() , AutomationPeer.GetLabeledBy() ,
AutomationPeer.GetLocalizedControlType() , AutomationPeer.GetName() ,

AutomationPeer.GetHelpText() , AutomationPeer.GetHeadingLevel() , AutomationPeer.GetParent() ,
AutomationPeer.GetVisualRoot() , AutomationPeer.HasKeyboardFocus() ,
AutomationPeer.IsContentElement() , AutomationPeer.IsControlElement() ,
AutomationPeer.IsEnabled() , AutomationPeer.IsKeyboardFocusable() , AutomationPeer.IsOffscreen() ,
AutomationPeer.SetFocus() , AutomationPeer.ShowContextMenu() ,
AutomationPeer.GetProvider<T>() , AutomationPeer.RaiseChildrenChangedEvent() ,
[AutomationPeer.RaisePropertyChangedEvent\(AutomationProperty, object, object\)](#) ,
AutomationPeer.GetLocalizedControlTypeCore() , [AutomationPeer.GetProviderCore\(Type\)](#) ,
AutomationPeer.EnsureEnabled() , AutomationPeer.ChildrenChanged ,
AutomationPeer.PropertyChanged , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BarcodeGeneratorAutomationPeer(BarcodeGenerator)

```
public BarcodeGeneratorAutomationPeer(BarcodeGenerator owner)
```

Parameters

owner [BarcodeGenerator](#)

Methods

GetAutomationControlTypeCore()

```
protected override AutomationControlType GetAutomationControlTypeCore()
```

Returns

AutomationControlType

GetClassNameCore()

```
protected override string GetClassNameCore()
```

Returns

[string](#)

Enum BarcodeSymbology

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

Supported barcode formats.

```
public enum BarcodeSymbology
```

Fields

Aztec = 11

Codabar = 9

Code128 = 2

Code39 = 3

Code93 = 4

DataMatrix = 1

EAN13 = 6

EAN8 = 5

ITF = 12

PDF417 = 10

QRCode = 0

UPCA = 7

UPCE = 8

Enum QRErrorCorrectionLevel

Namespace: [Nova.Avalonia.UI.BarcodeGenerator](#)

Assembly: Nova.Avalonia.UI.BarcodeGenerator.dll

Error correction levels for QR codes. Higher levels allow more damage recovery but reduce data capacity.

```
public enum QRErrorCorrectionLevel
```

Fields

H = 3

High - 30% error recovery capacity (recommended for logo overlay)

L = 0

Low - 7% error recovery capacity

M = 1

Medium - 15% error recovery capacity (default)

Q = 2

Quartile - 25% error recovery capacity

Namespace Nova.Avalonia.UI.Controls

Classes

[ArcPanel](#)

A panel that arranges child elements along an arc (partial circle).

[AutoLayout](#)

A layout panel inspired by Figma's Auto Layout, allowing simple direction-based stacking with easy control over spacing, alignment, and distribution.

[Avatar](#)

Extensible Avatar control that displays user images, initials, icons, or custom content.

[AvatarAutomationPeer](#)

Exposes [Avatar](#) to accessibility APIs.

[AvatarGroup](#)

Displays a group of avatars in a stacked or row layout with configurable overlap and overflow handling.

[AvatarGroupAutomationPeer](#)

Exposes [AvatarGroup](#) to accessibility APIs.

[AvatarStackPanel](#)

[Badge](#)

A notification badge control that can wrap content (buttons, icons) and display a count, text, or dot indicator at a configurable position.

[BadgeAutomationPeer](#)

Custom AutomationPeer for the Badge control.

[BubblePanel](#)

A panel that arranges child elements using a circle packing algorithm. Larger items are placed first and smaller items fill the gaps.

[CircularPanel](#)

A layout panel that arranges child elements in a circular pattern around a center point.

[HexPanel](#)

A panel that arranges child elements in a honeycomb (hexagonal grid) pattern. Based on <https://github.com/AlexanderSharykin/HexGrid> by Alexander Sharykin.

[LoopPanel](#)

An infinite/looping scrolling panel that wraps its children seamlessly.

[OrbitPanel](#)

A panel that arranges child elements in concentric orbit rings around a center point. Each child can be assigned to a specific orbit (ring) and will be distributed evenly within that orbit.

[OverlapPanel](#)

A panel that stacks children with configurable X/Y offsets, creating an overlapping card effect.

[RadialPanel](#)

A panel that arranges child elements in a radial fan pattern emanating from a center point.

[RatingControl](#)

A Rating Control with support to various shapes or precision levels.

[RatingItem](#)

Represents an individual item within the RatingControl (e.g., a single star).

[ResponsivePanel](#)

A panel that selectively shows its children based on the current width breakpoint.

[Shimmer](#)

A content-aware skeleton loading control. Uses immediate mode rendering to draw a "shimmer" effect over the content layout. Automatically detects text, shapes, and buttons to generate matching placeholders. Includes A11y support and a synchronized animation clock to prevent visual clutter.

[ShimmerAutomationPeer](#)

Automation peer for the [Shimmer](#) control, exposed as a loading indicator.

[StaggeredPanel](#)

A panel that arranges child elements into a staggered grid pattern, usually adding items to the column with the least amount of space used.

[TimelinePanel](#)

A panel that arranges child elements in a timeline pattern with spacing for connectors.

[VariableSizeWrapPanel](#)

A panel that arranges child elements in a grid with variable-size tiles using RowSpan and ColumnSpan.

[VirtualizingStaggeredPanel](#)

A virtualizing panel that arranges children in a staggered grid (masonry) layout. Items are placed in the shortest column, creating an efficient Pinterest-style layout. Only items within the visible

viewport (plus buffer) are materialized for optimal performance.

[VirtualizingVariableSizeWrapPanel](#)

A virtualizing panel that arranges children in a variable-size grid layout. Items can span multiple columns and rows, creating flexible tile-based layouts. Only items within the visible viewport (plus buffer) are materialized for optimal performance.

Enums

[AutoLayoutJustify](#)

Defines how items are distributed in the [AutoLayout](#) panel.

[AvatarDisplayMode](#)

Display mode for the Avatar control.

[AvatarShape](#)

Shape of the Avatar.

[AvatarSize](#)

Predefined sizes for Avatar.

[AvatarStatus](#)

Status indicator for Avatar (online status, notifications, etc.). Default colors are provided by the Avatar control; set StatusColor to override.

[BadgeKind](#)

Specifies the visual style of the badge.

[BadgePlacement](#)

Specifies the placement of the badge relative to its wrapped content.

[CircularAlignment](#)

Specifies how a child element is aligned relative to its position on the circle.

[RatingPrecision](#)

Specifies the precision level of the RatingControl.

[RatingShape](#)

Specifies the shape of the items in the RatingControl.

[ResponsiveBreakpoint](#)

Defines the size classes (breakpoints) for responsive layouts.

Class ArcPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements along an arc (partial circle).

```
public class ArcPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← ArcPanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
```

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

DistributeEvenlyProperty

Defines the [DistributeEvenly](#) property.

```
public static readonly StyledProperty<bool> DistributeEvenlyProperty
```

Field Value

StyledProperty<[bool](#)>

RadiusProperty

Defines the [Radius](#) property.

```
public static readonly StyledProperty<double> RadiusProperty
```

Field Value

StyledProperty<[double](#)>

StartAngleProperty

Defines the [StartAngle](#) property.

```
public static readonly StyledProperty<double> StartAngleProperty
```

Field Value

StyledProperty<[double](#)>

SweepAngleProperty

Defines the [SweepAngle](#) property.

```
public static readonly StyledProperty<double> SweepAngleProperty
```

Field Value

StyledProperty<[double](#)>

Properties

DistributeEvenly

Gets or sets whether to distribute children evenly across the arc.

```
public bool DistributeEvenly { get; set; }
```

Property Value

[bool](#)

Radius

Gets or sets the radius of the arc.

```
public double Radius { get; set; }
```

Property Value

[double](#)

StartAngle

Gets or sets the starting angle in degrees (-90 = top).

```
public double StartAngle { get; set; }
```

Property Value

[double](#)

SweepAngle

Gets or sets the sweep angle in degrees (how much of the arc to use).

```
public double SweepAngle { get; set; }
```

Property Value

[double](#) ↗

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Class AutoLayout

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A layout panel inspired by Figma's Auto Layout, allowing simple direction-based stacking with easy control over spacing, alignment, and distribution.

```
public class AutoLayout : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← AutoLayout
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
```

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

HorizontalContentAlignmentProperty

Defines the [HorizontalContentAlignment](#) property.

```
public static readonly StyledProperty<HorizontalAlignment>
HorizontalContentAlignmentProperty
```

Field Value

StyledProperty<HorizontalAlignment>

IsAbsoluteProperty

Defines the IsAbsolute attached property.

```
public static readonly AttachedProperty<bool> IsAbsoluteProperty
```

Field Value

AttachedProperty<[bool](#)>

IsReverseZIndexProperty

Defines the [IsReverseZIndex](#) property.

```
public static readonly StyledProperty<bool> IsReverseZIndexProperty
```

Field Value

StyledProperty<[bool](#)>

JustificationProperty

Defines the [Justification](#) property.

```
public static readonly StyledProperty<AutoLayoutJustify> JustificationProperty
```

Field Value

StyledProperty<[AutoLayoutJustify](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

PaddingProperty

Defines the [Padding](#) property.

```
public static readonly StyledProperty<Thickness> PaddingProperty
```

Field Value

StyledProperty<Thickness>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

VerticalContentAlignmentProperty

Defines the [VerticalContentAlignment](#) property.

```
public static readonly StyledProperty<VerticalAlignment>
VerticalContentAlignmentProperty
```

Field Value

StyledProperty<VerticalAlignment>

Properties

HorizontalContentAlignment

Gets or sets the horizontal alignment of the content items.

```
public HorizontalAlignment HorizontalContentAlignment { get; set; }
```

Property Value

HorizontalAlignment

IsReverseZIndex

Gets or sets whether the visual stacking order is reversed.

```
public bool IsReverseZIndex { get; set; }
```

Property Value

bool ↗

Justification

Gets or sets how items are distributed along the primary axis.

```
public AutoLayoutJustify Justification { get; set; }
```

Property Value

[AutoLayoutJustify](#)

Orientation

Gets or sets the primary direction of the layout.

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

Padding

Gets or sets the padding around the content.

```
public Thickness Padding { get; set; }
```

Property Value

Thickness

Spacing

Gets or sets the spacing between items.

```
public double Spacing { get; set; }
```

Property Value

[double](#)

VerticalContentAlignment

Gets or sets the vertical alignment of the content items.

```
public VerticalAlignment VerticalContentAlignment { get; set; }
```

Property Value

VerticalAlignment

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetIsAbsolute(Control)

Gets the IsAbsolute attached property.

```
public static bool GetIsAbsolute(Control element)
```

Parameters

element Control

Returns

`bool` ↗

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

`availableSize` `Size`

The size available to the control.

Returns

`Size`

The desired size for the control.

SetIsAbsolute(Control, bool)

Sets the `IsAbsolute` attached property.

```
public static void SetIsAbsolute(Control element, bool value)
```

Parameters

`element` `Control`

`value` `bool` ↗

Enum AutoLayoutJustify

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Defines how items are distributed in the [AutoLayout](#) panel.

```
public enum AutoLayoutJustify
```

Fields

Packed = 0

Items are packed together with the specified [Spacing](#).

SpaceBetween = 1

Items are distributed evenly across the available space, creating equal spacing between them. The [Spacing](#) property is ignored in this mode.

Class Avatar

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Extensible Avatar control that displays user images, initials, icons, or custom content.

```
public class Avatar : TemplatedControl, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode,  
IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize,  
IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

[Object](#) ↗ ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← TemplatedControl ← Avatar

Implements

[INotifyPropertyChanged](#) ↗, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#) ↗, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue

Inherited Members

TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,
TemplatedControl.GetIsTemplateFocusTarget(Control) ,
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) ↗ , TemplatedControl.ApplyTemplate() ,
TemplatedControl.GetTemplateFocusTarget() ,
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,
TemplatedControl.BorderBrush , TemplatedControl.BorderThickness ,
TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,

TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(SizeChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsEnabled ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.HitTestVisible , InputElement.PointerOver , InputElement.TabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,

InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalContentAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.Render(DrawingContext) , Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.IsVisible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,

StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,

AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

AutoGenerateBackgroundProperty

Defines the [AutoGenerateBackground](#) property.

```
public static readonly StyledProperty<bool> AutoGenerateBackgroundProperty
```

Field Value

StyledProperty<[bool](#)>

BackgroundColorProperty

Defines the [BackgroundColor](#) property.

```
public static readonly StyledProperty<IBrush> BackgroundColorProperty
```

Field Value

StyledProperty<IBrush>

ContentProperty

Defines the [Content](#) property.

```
public static readonly StyledProperty<object?> ContentProperty
```

Field Value

StyledProperty<[object](#)>

CustomSizeProperty

Defines the [CustomSize](#) property.

```
public static readonly StyledProperty<double> CustomSizeProperty
```

Field Value

StyledProperty<[double](#)>

DisplayModeProperty

Defines the [DisplayMode](#) property.

```
public static readonly StyledProperty<AvatarDisplayMode> DisplayModeProperty
```

Field Value

StyledProperty<[AvatarDisplayMode](#)>

DisplayNameProperty

Defines the [DisplayName](#) property.

```
public static readonly StyledProperty<string> DisplayNameProperty
```

Field Value

StyledProperty<[string](#)>

ForegroundColorProperty

Defines the [ForegroundColor](#) property.

```
public static readonly StyledProperty<IBrush> ForegroundColorProperty
```

Field Value

StyledProperty<IBrush>

IconProperty

Defines the [Icon](#) property.

```
public static readonly StyledProperty<object?> IconProperty
```

Field Value

StyledProperty<[object](#)>

ImageSourceProperty

Defines the [ImageSource](#) property.

```
public static readonly StyledProperty<Bitmap?> ImageSourceProperty
```

Field Value

StyledProperty<Bitmap>

InitialsProperty

Defines the [Initials](#) property.

```
public static readonly StyledProperty<string> InitialsProperty
```

Field Value

StyledProperty<[string](#)>

MaxInitialsProperty

Defines the [MaxInitials](#) property.

```
public static readonly StyledProperty<int> MaxInitialsProperty
```

Field Value

StyledProperty<[int](#)>

ShapeProperty

Defines the [Shape](#) property.

```
public static readonly StyledProperty<AvatarShape> ShapeProperty
```

Field Value

StyledProperty<[AvatarShape](#)>

ShowTooltipProperty

Defines the [ShowTooltip](#) property.

```
public static readonly StyledProperty<bool> ShowTooltipProperty
```

Field Value

StyledProperty<[bool](#)>

SizeProperty

Defines the [Size](#) property.

```
public static readonly StyledProperty<AvatarSize> SizeProperty
```

Field Value

StyledProperty<[AvatarSize](#)>

StatusColorProperty

Defines the [StatusColor](#) property.

```
public static readonly StyledProperty<IBrush?> StatusColorProperty
```

Field Value

StyledProperty<IBrush>

StatusProperty

Defines the [Status](#) property.

```
public static readonly StyledProperty<AvatarStatus> StatusProperty
```

Field Value

StyledProperty<[AvatarStatus](#)>

Properties

AutoGenerateBackground

Gets or sets whether to automatically generate background color from name.

```
public bool AutoGenerateBackground { get; set; }
```

Property Value

[bool](#) ↗

BackgroundColor

Gets or sets the background color of the avatar.

```
public IBrush BackgroundColor { get; set; }
```

Property Value

IBrush

Content

Gets or sets custom content to display in the avatar.

```
public object? Content { get; set; }
```

Property Value

[object](#)

CustomSize

Gets or sets the custom size when Size is set to Custom.

```
public double CustomSize { get; set; }
```

Property Value

[double](#)

DisplayMode

Gets or sets the display mode determining what content to show.

```
public AvatarDisplayMode DisplayMode { get; set; }
```

Property Value

[AvatarDisplayMode](#)

DisplayName

Gets or sets the person's display name used for generating initials and tooltip.

```
public string DisplayName { get; set; }
```

Property Value

[string](#)

ForegroundColor

Gets or sets the foreground color for text and icons.

```
public IBrush ForegroundColor { get; set; }
```

Property Value

IBrush

Icon

Gets or sets the icon content (can be PathIcon, SymbolIcon, or any content).

```
public object? Icon { get; set; }
```

Property Value

[object](#)

ImageSource

Gets or sets the image source for the avatar.

```
public Bitmap? ImageSource { get; set; }
```

Property Value

Bitmap

Initials

Gets or sets custom initials (overrides auto-generated initials from Name).

```
public string Initials { get; set; }
```

Property Value

[string](#)

MaxInitials

Gets or sets the maximum number of initials to display.

```
public int MaxInitials { get; set; }
```

Property Value

[int](#)

Shape

Gets or sets the shape of the avatar.

```
public AvatarShape Shape { get; set; }
```

Property Value

[AvatarShape](#)

ShowTooltip

Gets or sets whether to show tooltip with full name.

```
public bool ShowTooltip { get; set; }
```

Property Value

[bool](#)

Size

Gets or sets the predefined size of the avatar.

```
public AvatarSize Size { get; set; }
```

Property Value

[AvatarSize](#)

Status

Gets or sets the status indicator.

```
public AvatarStatus Status { get; set; }
```

Property Value

[AvatarStatus](#)

StatusColor

Gets or sets the color of the status indicator. If null, uses default color for status.

```
public IBrush? StatusColor { get; set; }
```

Property Value

IBrush

Methods

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e TemplateAppliedEventArgs

The event args.

OnCreateAutomationPeer()

Returns a new, type-specific Avalonia.Automation.Peers.AutomationPeer implementation for the control.

```
protected override AutomationPeer OnCreateAutomationPeer()
```

Returns

AutomationPeer

The type-specific Avalonia.Automation.Peers.AutomationPeer implementation.

Class AvatarAutomationPeer

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Exposes [Avatar](#) to accessibility APIs.

```
public class AvatarAutomationPeer : ControlAutomationPeer
```

Inheritance

[object](#) ← AutomationPeer ← ControlAutomationPeer ← AvatarAutomationPeer

Inherited Members

ControlAutomationPeer.GetOrCreate(Control) ,
ControlAutomationPeer.CreatePeerForElement(Control) ,
ControlAutomationPeer.FromElement(Control) , ControlAutomationPeer.BringIntoViewCore() ,
ControlAutomationPeer.GetOrCreateChildrenCore() , ControlAutomationPeer.GetChildrenCore() ,
ControlAutomationPeer.GetLabeledByCore() , ControlAutomationPeer.GetHelpTextCore() ,
ControlAutomationPeer.GetHeadingLevelCore() , ControlAutomationPeer.GetParentCore() ,
ControlAutomationPeer.GetVisualRootCore() , ControlAutomationPeer.InvalidateChildren() ,
ControlAutomationPeer.InvalidateParent() , ControlAutomationPeer.ShowContextMenuCore() ,
ControlAutomationPeer.TrySetParent(AutomationPeer) ,
ControlAutomationPeer.GetAcceleratorKeyCore() , ControlAutomationPeer.GetAccessKeyCore() ,
ControlAutomationPeer.GetAutomationIdCore() ,
ControlAutomationPeer.GetBoundingClientRectCore() ,
ControlAutomationPeer.HasKeyboardFocusCore() , ControlAutomationPeer.IsContentElementCore() ,
ControlAutomationPeer.IsControlElementCore() , ControlAutomationPeer.IsEnabledCore() ,
ControlAutomationPeer.IsKeyboardFocusableCore() , ControlAutomationPeer.SetFocusCore() ,
ControlAutomationPeer.GetControlTypeOverrideCore() ,
ControlAutomationPeer.IsContentElementOverrideCore() ,
ControlAutomationPeer.IsControlElementOverrideCore() , ControlAutomationPeer.IsOffscreenCore() ,
ControlAutomationPeer.Owner , AutomationPeer.BringIntoView() ,
AutomationPeer.GetAcceleratorKey() , AutomationPeer.GetAccessKey() ,
AutomationPeer.GetAutomationControlType() , AutomationPeer.GetAutomationId() ,
AutomationPeer.GetBoundingClientRect() , AutomationPeer.GetChildren() ,
AutomationPeer.GetClassName() , AutomationPeer.GetLabeledBy() ,
AutomationPeer.GetLocalizedControlType() , AutomationPeer.GetName() ,
AutomationPeer.GetHelpText() , AutomationPeer.GetHeadingLevel() , AutomationPeer.GetParent() ,
AutomationPeer.GetVisualRoot() , AutomationPeer.HasKeyboardFocus() ,

AutomationPeer.IsContentElement() , AutomationPeer.IsEnabled() , AutomationPeer.IsControlElement() ,
AutomationPeer.IsEnabled() , AutomationPeer.IsKeyboardFocusable() , AutomationPeer.IsOffscreen() ,
AutomationPeer.SetFocus() , AutomationPeer.ShowContextMenu() ,
AutomationPeer.GetProvider<T>() , AutomationPeer.RaiseChildrenChangedEvent() ,
[AutomationPeer.RaisePropertyChangedEvent\(AutomationProperty, object, object\)](#) ,
AutomationPeer.GetLocalizedControlTypeCore() , [AutomationPeer.GetProviderCore\(Type\)](#) ,
AutomationPeer.EnsureEnabled() , AutomationPeer.ChildrenChanged ,
AutomationPeer.PropertyChanged , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AvatarAutomationPeer(Avatar)

```
public AvatarAutomationPeer(Avatar owner)
```

Parameters

owner [Avatar](#)

Methods

GetAutomationControlTypeCore()

```
protected override AutomationControlType GetAutomationControlTypeCore()
```

Returns

AutomationControlType

GetClassNameCore()

```
protected override string GetClassNameCore()
```

Returns

[string](#) ↗

GetNameCore()

```
protected override string GetNameCore()
```

Returns

[string](#) ↗

Enum AvatarDisplayMode

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Display mode for the Avatar control.

```
public enum AvatarDisplayMode
```

Fields

Auto = 0

Display image if available, otherwise initials, otherwise icon.

Content = 4

Display custom content.

Icon = 3

Display only icon.

Image = 1

Display only image.

Initials = 2

Display only initials from name.

Class AvatarGroup

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Displays a group of avatars in a stacked or row layout with configurable overlap and overflow handling.

```
public class AvatarGroup : TemplatedControl, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode,  
IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize,  
IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

```
Object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← TemplatedControl ← AvatarGroup
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,  
INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inherited Members

```
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,  
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,  
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,  
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,  
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,  
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,  
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,  
TemplatedControl.GetIsTemplateFocusTarget(Control) ,  
TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\) , TemplatedControl.ApplyTemplate() ,  
TemplatedControl.GetTemplateFocusTarget() ,  
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,  
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,  
TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,  
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,  
TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,  
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
```

Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(PropertyChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(LostFocusEventArgs) ,
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerEventArgs) ,
InputElement.OnPointerCaptureLost(PointerEventArgs) ,
InputElement.OnPointerWheelChanged(PointerEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.HitTestVisible , InputElement.PointerOver , InputElement.TabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,

[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.Render(DrawingContext) , Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,

[StyledElement.OnDataContextChanged\(EventArgs\)](#) , [StyledElement.OnDataContextBeginUpdate\(\)](#) ,
[StyledElement.OnDataContextEndUpdate\(\)](#) , [StyledElement.OnInitialized\(\)](#) , [StyledElement.Name](#) ,
[StyledElement.Classes](#) , [StyledElement.DataContext](#) , [StyledElement.IsInitialized](#) ,
[StyledElement.Styles](#) , [StyledElement.StyleKey](#) , [StyledElement.Resources](#) ,
[StyledElement.TemplatedParent](#) , [StyledElement.Theme](#) , [StyledElement.LogicalChildren](#) ,
[StyledElement.PseudoClasses](#) , [StyledElement.StyleKeyOverride](#) , [StyledElement.Parent](#) ,
[StyledElement.ActualThemeVariant](#) , [StyledElement.AttachedToLogicalTree](#) ,
[StyledElement.DetachedFromLogicalTree](#) , [StyledElement.DataContextChanged](#) ,
[StyledElement.Initialized](#) , [StyledElement.ResourcesChanged](#) ,
[StyledElement.ActualThemeVariantChanged](#) , [Animatable.TransitionsProperty](#) ,
[Animatable.OnPropertyChangedCore\(AvaloniaPropertyChangedEventArgs\)](#) , [Animatable.Transitions](#) ,
[AvaloniaObject.CheckAccess\(\)](#) , [AvaloniaObject.VerifyAccess\(\)](#) ,
[AvaloniaObject.ClearValue\(AvaloniaProperty\)](#) , [AvaloniaObject.ClearValue<T>\(AvaloniaProperty<T>\)](#) ,
[AvaloniaObject.ClearValue<T>\(StyledProperty<T>\)](#) ,
[AvaloniaObject.ClearValue<T>\(DirectPropertyBase<T>\)](#) , [AvaloniaObject.Equals\(object\)](#) ,
[AvaloniaObject.GetHashCode\(\)](#) , [AvaloniaObject.GetValue\(AvaloniaProperty\)](#) ,
[AvaloniaObject.GetValue<T>\(StyledProperty<T>\)](#) ,
[AvaloniaObject.GetValue<T>\(DirectPropertyBase<T>\)](#) ,
[AvaloniaObject.GetBaseValue<T>\(StyledProperty<T>\)](#) ,
[AvaloniaObject.IsAnimating\(AvaloniaProperty\)](#) , [AvaloniaObject.IsSet\(AvaloniaProperty\)](#) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
[AvaloniaObject.SetValue<T>\(StyledProperty<T>, T, BindingPriority\)](#) ,
[AvaloniaObject.SetValue<T>\(DirectPropertyBase<T>, T\)](#) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
[AvaloniaObject.SetCurrentValue<T>\(StyledProperty<T>, T\)](#) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IBinding\)](#) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
[AvaloniaObject.CoerceValue\(AvaloniaProperty\)](#) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
[AvaloniaObject.RaisePropertyChanged<T>\(DirectPropertyBase<T>, T, T\)](#) ,
[AvaloniaObject.SetAndRaise<T>\(DirectPropertyBase<T>, ref T, T\)](#) , [AvaloniaObject.InheritanceParent](#) ,
[AvaloniaObject.this\[AvaloniaProperty\]](#) , [AvaloniaObject.this\[IndexerDescriptor\]](#) ,
[AvaloniaObject.PropertyChanged](#) , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AvatarGroup()

```
public AvatarGroup()
```

Fields

AvatarsProperty

Defines the [Avatars](#) property.

```
public static readonly StyledProperty<AvaloniaList<Avatar>> AvatarsProperty
```

Field Value

StyledProperty<AvaloniaList<[Avatar](#)>>

BorderBrushProperty

Defines the [BorderBrush](#) property.

```
public static readonly StyledProperty<IBrush> BorderBrushProperty
```

Field Value

StyledProperty<IBrush>

BorderThicknessProperty

Defines the [BorderThickness](#) property.

```
public static readonly StyledProperty<Thickness> BorderThicknessProperty
```

Field Value

StyledProperty<Thickness>

MaxDisplayedProperty

Defines the [MaxDisplayed](#) property.

```
public static readonly StyledProperty<int> MaxDisplayedProperty
```

Field Value

StyledProperty<[int](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

OverlapProperty

Defines the [Overlap](#) property.

```
public static readonly StyledProperty<double> OverlapProperty
```

Field Value

StyledProperty<[double](#)>

ShowCountProperty

Defines the [ShowCount](#) property.

```
public static readonly StyledProperty<bool> ShowCountProperty
```

Field Value

StyledProperty<[bool](#)>

SizeProperty

Defines the [Size](#) property.

```
public static readonly StyledProperty<AvatarSize> SizeProperty
```

Field Value

StyledProperty<[AvatarSize](#)>

Properties

Avatars

Gets or sets the collection of avatars to display.

```
public AvaloniaList<Avatar> Avatars { get; set; }
```

Property Value

AvaloniaList<[Avatar](#)>

BorderBrush

Gets or sets the brush used for the border separator between avatars.

```
public IBrush BorderBrush { get; set; }
```

Property Value

BorderThickness

Gets or sets the thickness of the border separator between avatars.

```
public Thickness BorderThickness { get; set; }
```

Property Value

Thickness

MaxDisplayed

Gets or sets the maximum number of avatars to display before showing the "+N" overflow badge.

```
public int MaxDisplayed { get; set; }
```

Property Value

[int](#)

Orientation

Gets or sets the layout orientation (Horizontal or Vertical).

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

Overlap

Gets or sets the overlap percentage (0.0 to 1.0) between adjacent avatars.

```
public double Overlap { get; set; }
```

Property Value

[double](#)

ShowCount

Gets or sets whether to show the count badge for hidden avatars when the list exceeds [MaxDisplayed](#).

```
public bool ShowCount { get; set; }
```

Property Value

[bool](#)

Size

Gets or sets the size of all avatars in the group.

```
public AvatarSize Size { get; set; }
```

Property Value

[AvatarSize](#)

Methods

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e `TemplateAppliedEventArgs`

The event args.

OnCreateAutomationPeer()

Returns a new, type-specific `Avalonia.Automation.Peers.AutomationPeer` implementation for the control.

```
protected override AutomationPeer OnCreateAutomationPeer()
```

Returns

`AutomationPeer`

The type-specific `Avalonia.Automation.Peers.AutomationPeer` implementation.

Class AvatarGroupAutomationPeer

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Exposes [AvatarGroup](#) to accessibility APIs.

```
public class AvatarGroupAutomationPeer : ControlAutomationPeer
```

Inheritance

[object](#) ← AutomationPeer ← ControlAutomationPeer ← AvatarGroupAutomationPeer

Inherited Members

ControlAutomationPeer.GetOrCreate(Control) ,
ControlAutomationPeer.CreatePeerForElement(Control) ,
ControlAutomationPeer.FromElement(Control) , ControlAutomationPeer.BringIntoViewCore() ,
ControlAutomationPeer.GetOrCreateChildrenCore() , ControlAutomationPeer.GetLabeledByCore() ,
ControlAutomationPeer.GetNameCore() , ControlAutomationPeer.GetHelpTextCore() ,
ControlAutomationPeer.GetHeadingLevelCore() , ControlAutomationPeer.GetParentCore() ,
ControlAutomationPeer.GetVisualRootCore() , ControlAutomationPeer.InvalidateChildren() ,
ControlAutomationPeer.InvalidateParent() , ControlAutomationPeer.ShowContextMenuCore() ,
ControlAutomationPeer.TrySetParent(AutomationPeer) ,
ControlAutomationPeer.GetAcceleratorKeyCore() , ControlAutomationPeer.GetAccessKeyCore() ,
ControlAutomationPeer.GetAutomationIdCore() ,
ControlAutomationPeer.GetBoundingClientRectCore() ,
ControlAutomationPeer.HasKeyboardFocusCore() , ControlAutomationPeer.IsContentElementCore() ,
ControlAutomationPeer.IsControlElementCore() , ControlAutomationPeer.IsEnabledCore() ,
ControlAutomationPeer.IsKeyboardFocusableCore() , ControlAutomationPeer.SetFocusCore() ,
ControlAutomationPeer.GetControlTypeOverrideCore() ,
ControlAutomationPeer.IsContentElementOverrideCore() ,
ControlAutomationPeer.IsControlElementOverrideCore() , ControlAutomationPeer.IsOffscreenCore() ,
ControlAutomationPeer.Owner , AutomationPeer.BringIntoView() ,
AutomationPeer.GetAcceleratorKey() , AutomationPeer.GetAccessKey() ,
AutomationPeer.GetAutomationControlType() , AutomationPeer.GetAutomationId() ,
AutomationPeer.GetBoundingClientRect() , AutomationPeer.GetChildren() ,
AutomationPeer.GetClassName() , AutomationPeer.GetLabeledBy() ,
AutomationPeer.GetLocalizedControlType() , AutomationPeer.GetName() ,
AutomationPeer.GetHelpText() , AutomationPeer.GetHeadingLevel() , AutomationPeer.GetParent() ,
AutomationPeer.GetVisualRoot() , AutomationPeer.HasKeyboardFocus() ,

AutomationPeer.IsContentElement() , AutomationPeer.IsEnabled() , AutomationPeer.IsKeyboardFocusable() , AutomationPeer.IsOffscreen() ,
AutomationPeer.SetFocus() , AutomationPeer.ShowContextMenu() ,
AutomationPeer.GetProvider<T>() , AutomationPeer.RaiseChildrenChangedEvent() ,
[AutomationPeer.RaisePropertyChangedEvent\(AutomationProperty, object, object\)](#) ,
AutomationPeer.GetLocalizedControlTypeCore() , [AutomationPeer.GetProviderCore\(Type\)](#) ,
AutomationPeer.EnsureEnabled() , AutomationPeer.ChildrenChanged ,
AutomationPeer.PropertyChanged , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

AvatarGroupAutomationPeer(AvatarGroup)

```
public AvatarGroupAutomationPeer(AvatarGroup owner)
```

Parameters

owner [AvatarGroup](#)

Methods

GetAutomationControlTypeCore()

```
protected override AutomationControlType GetAutomationControlTypeCore()
```

Returns

AutomationControlType

GetChildrenCore()

```
protected override IReadOnlyList<AutomationPeer> GetChildrenCore()
```

Returns

[IReadOnlyList](#)<AutomationPeer>

GetClassNameCore()

`protected override string GetClassNameCore()`

Returns

[string](#)

Enum AvatarShape

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Shape of the Avatar.

```
public enum AvatarShape
```

Fields

Circle = 0

Circular avatar.

Rectangle = 2

Square avatar with sharp corners.

Square = 1

Square avatar with rounded corners.

Enum AvatarSize

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Predefined sizes for Avatar.

```
public enum AvatarSize
```

Fields

Custom = 5

Custom size.

ExtraLarge = 4

Extra large - 96x96.

ExtraSmall = 0

Extra small - 24x24.

Large = 3

Large - 64x64.

Medium = 2

Medium - 48x48.

Small = 1

Small - 32x32.

Class AvatarStackPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

```
public class AvatarStackPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← AvatarStackPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,

InputElement.IsEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.HitTestVisible , InputElement.PointerOver , InputElement.TabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,

Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,

AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

OrientationProperty

Layout orientation for the stack (horizontal default).

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

OverlapProperty

Percentage (0-1) of overlap between consecutive avatars; 0.25 means 25% overlap.

```
public static readonly StyledProperty<double> OverlapProperty
```

Field Value

StyledProperty<[double](#)>

Properties

Orientation

Direction of stacking (horizontal or vertical).

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

Overlap

Percentage of overlap between children; values < 0 produce gaps, > 1 fully hides previous items.

```
public double Overlap { get; set; }
```

Property Value

[double](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Enum AvatarStatus

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Status indicator for Avatar (online status, notifications, etc.). Default colors are provided by the Avatar control; set StatusColor to override.

```
public enum AvatarStatus
```

Fields

Away = 3

Contact is away/idle.

Busy = 4

Contact is busy.

DoNotDisturb = 5

Contact should not be disturbed.

None = 0

Offline = 2

Contact is not reachable.

Online = 1

Contact is reachable and active.

Class Badge

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A notification badge control that can wrap content (buttons, icons) and display a count, text, or dot indicator at a configurable position.

```
[TemplatePart("PART_BadgeContainer", typeof(Border))]  
public class Badge : ContentControl, INotifyPropertyChanged, IDataContextProvider,  
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,  
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,  
IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← TemplatedControl ← ContentControl ← Badge
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,  
INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inherited Members

```
ContentControl.ContentProperty , ContentControl.ContentTemplateProperty ,  
ContentControl.HorizontalContentAlignmentProperty ,  
ContentControl.VerticalContentAlignmentProperty ,  
ContentControl.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) ,  
ContentControl.RegisterContentPresenter(ContentPresenter) , ContentControl.Content ,  
ContentControl.ContentTemplate , ContentControl.Presenter ,  
ContentControl.HorizontalContentAlignment , ContentControl.VerticalContentAlignment ,  
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,  
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,  
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,  
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,  
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,  
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,  
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,  
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,  
TemplatedControl.GetIsTemplateFocusTarget(Control) ,
```

[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) , TemplatedControl.ApplyTemplate() ,
TemplatedControl.GetTemplateFocusTarget() ,
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,
TemplatedControl.BorderBrush , TemplatedControl.BorderThickness ,
TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,
TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventEventArgs) ,
Control.FocusAdorner , Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout ,
Control.IsLoaded , Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded ,
Control.SizeChanged , InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerEventArgs) ,
InputElement.OnPointerCaptureLost(PointerEventArgs) ,
InputElement.OnPointerWheelChanged(PointerEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,

InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.Render(DrawingContext) , Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,

Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.IsVisible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
[StyledElement.OnDataContextChangedEventArgs](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,

[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Remarks

The Badge control supports:

- Numeric content with automatic "99+" style overflow (via [MaxCount](#))
- Dot mode for simple presence indicators
- 8 placement positions around the wrapped content
- Full accessibility via [BadgeAutomationPeer](#)

Fields

BadgeBackgroundProperty

Defines the [BadgeBackground](#) property.

```
public static readonly StyledProperty<IBrush> BadgeBackgroundProperty
```

Field Value

StyledProperty<IBrush>

BadgeBorderBrushProperty

Defines the [BadgeBorderBrush](#) property.

```
public static readonly StyledProperty<IBrush> BadgeBorderBrushProperty
```

Field Value

StyledProperty<IBrush>

BadgeBorderThicknessProperty

Defines the [BadgeBorderThickness](#) property.

```
public static readonly StyledProperty<Thickness> BadgeBorderThicknessProperty
```

Field Value

StyledProperty<Thickness>

BadgeContentProperty

Defines the [BadgeContent](#) property.

```
public static readonly StyledProperty<object?> BadgeContentProperty
```

Field Value

StyledProperty<[object](#)>

BadgeForegroundProperty

Defines the [BadgeForeground](#) property.

```
public static readonly StyledProperty<IBrush> BadgeForegroundProperty
```

Field Value

StyledProperty<IBrush>

BadgeOffsetProperty

Defines the [BadgeOffset](#) property.

```
public static readonly StyledProperty<double> BadgeOffsetProperty
```

Field Value

StyledProperty<[double](#)>

BadgePlacementProperty

Defines the [BadgePlacement](#) property.

```
public static readonly StyledProperty<BadgePlacement> BadgePlacementProperty
```

Field Value

StyledProperty<[BadgePlacement](#)>

DisplayContentProperty

Defines the [DisplayContent](#) property.

```
public static readonly DirectProperty<Badge, object?> DisplayContentProperty
```

Field Value

DirectProperty<[Badge](#), [object](#)>

IsBadgeVisibleProperty

Defines the [IsBadgeVisible](#) property.

```
public static readonly StyledProperty<bool> IsBadgeVisibleProperty
```

Field Value

StyledProperty<[bool](#)>

KindProperty

Defines the [Kind](#) property.

```
public static readonly StyledProperty<BadgeKind> KindProperty
```

Field Value

StyledProperty<[BadgeKind](#)>

MaxCountProperty

Defines the [MaxCount](#) property.

```
public static readonly StyledProperty<int> MaxCountProperty
```

Field Value

StyledProperty<[int](#)>

Properties

BadgeBackground

Gets or sets the background brush for the badge.

```
public IBrush BadgeBackground { get; set; }
```

Property Value

IBrush

BadgeBorderBrush

Gets or sets the border brush for the badge.

```
public IBrush BadgeBorderBrush { get; set; }
```

Property Value

IBrush

BadgeBorderThickness

Gets or sets the border thickness for the badge.

```
public Thickness BadgeBorderThickness { get; set; }
```

Property Value

Thickness

BadgeContent

Gets or sets the content displayed in the badge (text, number, etc.).

```
public object? BadgeContent { get; set; }
```

Property Value

[object](#)

BadgeForeground

Gets or sets the foreground brush for badge text.

```
public IBrush BadgeForeground { get; set; }
```

Property Value

BadgeOffset

Gets or sets the offset (in pixels) to adjust badge position inward from the corner.

```
public double BadgeOffset { get; set; }
```

Property Value

[double](#)

BadgePlacement

Gets or sets the placement of the badge relative to the wrapped content.

```
public BadgePlacement BadgePlacement { get; set; }
```

Property Value

[BadgePlacement](#)

DisplayContent

Gets the processed display content, which may show "99+" style overflow text.

```
public object? DisplayContent { get; }
```

Property Value

[object](#)

IsBadgeVisible

Gets or sets whether the badge is visible.

```
public bool IsBadgeVisible { get; set; }
```

Property Value

[bool](#)

Kind

Gets or sets the badge kind (Content or Dot).

```
public BadgeKind Kind { get; set; }
```

Property Value

[BadgeKind](#)

MaxCount

Gets or sets the maximum count before displaying overflow text (e.g., "99+").

```
public int MaxCount { get; set; }
```

Property Value

[int](#)

Methods

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e `TemplateAppliedEventArgs`

The event args.

OnCreateAutomationPeer()

Returns a new, type-specific `Avalonia.Automation.Peers.AutomationPeer` implementation for the control.

```
protected override AutomationPeer OnCreateAutomationPeer()
```

Returns

`AutomationPeer`

The type-specific `Avalonia.Automation.Peers.AutomationPeer` implementation.

Class BadgeAutomationPeer

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Custom AutomationPeer for the Badge control.

```
public class BadgeAutomationPeer : ControlAutomationPeer
```

Inheritance

[object](#) ← AutomationPeer ← ControlAutomationPeer ← BadgeAutomationPeer

Inherited Members

ControlAutomationPeer.GetOrCreate(Control) ,
ControlAutomationPeer.CreatePeerForElement(Control) ,
ControlAutomationPeer.FromElement(Control) , ControlAutomationPeer.BringIntoViewCore() ,
ControlAutomationPeer.GetOrCreateChildrenCore() , ControlAutomationPeer.GetChildrenCore() ,
ControlAutomationPeer.GetLabeledByCore() , ControlAutomationPeer.GetHeadingLevelCore() ,
ControlAutomationPeer.GetParentCore() , ControlAutomationPeer.GetVisualRootCore() ,
ControlAutomationPeer.InvalidateChildren() , ControlAutomationPeer.InvalidateParent() ,
ControlAutomationPeer.ShowContextMenuCore() ,
ControlAutomationPeer.TrySetParent(AutomationPeer) ,
ControlAutomationPeer.GetAcceleratorKeyCore() , ControlAutomationPeer.GetAccessKeyCore() ,
ControlAutomationPeer.GetAutomationIdCore() ,
ControlAutomationPeer.GetBoundingClientRectCore() ,
ControlAutomationPeer.HasKeyboardFocusCore() , ControlAutomationPeer.IsContentElementCore() ,
ControlAutomationPeer.IsControlElementCore() , ControlAutomationPeer.IsEnabledCore() ,
ControlAutomationPeer.IsKeyboardFocusableCore() , ControlAutomationPeer.SetFocusCore() ,
ControlAutomationPeer.GetControlTypeOverrideCore() ,
ControlAutomationPeer.IsContentElementOverrideCore() ,
ControlAutomationPeer.IsControlElementOverrideCore() , ControlAutomationPeer.IsOffscreenCore() ,
ControlAutomationPeer.Owner , AutomationPeer.BringIntoView() ,
AutomationPeer.GetAcceleratorKey() , AutomationPeer.GetAccessKey() ,
AutomationPeer.GetAutomationControlType() , AutomationPeer.GetAutomationId() ,
AutomationPeer.GetBoundingClientRect() , AutomationPeer.GetChildren() ,
AutomationPeer.GetClassName() , AutomationPeer.GetLabeledBy() ,
AutomationPeer.GetLocalizedControlType() , AutomationPeer.GetName() ,
AutomationPeer.GetHelpText() , AutomationPeer.GetHeadingLevel() , AutomationPeer.GetParent() ,
AutomationPeer.GetVisualRoot() , AutomationPeer.HasKeyboardFocus() ,

AutomationPeer.IsContentElement() , AutomationPeer.IsControlElement() ,
AutomationPeer.IsEnabled() , AutomationPeer.IsKeyboardFocusable() , AutomationPeer.IsOffscreen() ,
AutomationPeer.SetFocus() , AutomationPeer.ShowContextMenu() ,
AutomationPeer.GetProvider<T>() , AutomationPeer.RaiseChildrenChangedEvent() ,
[AutomationPeer.RaisePropertyChangedEvent\(AutomationProperty, object, object\)](#) ,
AutomationPeer.GetLocalizedControlTypeCore() , [AutomationPeer.GetProviderCore\(Type\)](#) ,
AutomationPeer.EnsureEnabled() , AutomationPeer.ChildrenChanged ,
AutomationPeer.PropertyChanged , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BadgeAutomationPeer(Badge)

```
public BadgeAutomationPeer(Badge owner)
```

Parameters

owner [Badge](#)

Methods

GetAutomationControlTypeCore()

```
protected override AutomationControlType GetAutomationControlTypeCore()
```

Returns

AutomationControlType

GetClassNameCore()

```
protected override string GetClassNameCore()
```

Returns

string ↗

GetHelpTextCore()

```
protected override string? GetHelpTextCore()
```

Returns

string ↗

GetNameCore()

```
protected override string? GetNameCore()
```

Returns

string ↗

Enum BadgeKind

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Specifies the visual style of the badge.

```
public enum BadgeKind
```

Fields

Content = 0

Badge displays text or numeric content.

Dot = 1

Badge displays as a small dot indicator.

Enum BadgePlacement

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Specifies the placement of the badge relative to its wrapped content.

```
public enum BadgePlacement
```

Fields

Bottom = 5

Badge appears at the bottom center.

BottomLeft = 6

Badge appears at the bottom-left corner.

BottomRight = 4

Badge appears at the bottom-right corner.

Left = 7

Badge appears at the center-left edge.

Right = 3

Badge appears at the center-right edge.

Top = 1

Badge appears at the top center.

TopLeft = 0

Badge appears at the top-left corner.

TopRight = 2

Badge appears at the top-right corner (default).

Class BubblePanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements using a circle packing algorithm. Larger items are placed first and smaller items fill the gaps.

```
public class BubblePanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← BubblePanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
```

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

PaddingProperty

Defines the [Padding](#) property.

```
public static readonly StyledProperty<Thickness> PaddingProperty
```

Field Value

StyledProperty<Thickness>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

Properties

Padding

Gets or sets the padding around the panel content.

```
public Thickness Padding { get; set; }
```

Property Value

Thickness

Spacing

Gets or sets the minimum spacing between bubbles.

```
public double Spacing { get; set; }
```

Property Value

[double](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Enum CircularAlignment

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Specifies how a child element is aligned relative to its position on the circle.

```
public enum CircularAlignment
```

Fields

Center = 0

Align the center of the element to the position (default).

Inner = 1

Align the inner edge of the element toward the center.

Outer = 2

Align the outer edge of the element away from the center.

Class CircularPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A layout panel that arranges child elements in a circular pattern around a center point.

```
public class CircularPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← CircularPanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
```

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

AlignmentProperty

Defines the Alignment attached property.

```
public static readonly AttachedProperty<CircularAlignment> AlignmentProperty
```

Field Value

AttachedProperty<[CircularAlignment](#)>

AngleProperty

Defines the Angle attached property.

```
public static readonly AttachedProperty<double> AngleProperty
```

Field Value

AttachedProperty<[double](#)>

AngleStepProperty

Defines the [AngleStep](#) property.

```
public static readonly StyledProperty<double> AngleStepProperty
```

Field Value

StyledProperty<[double](#)>

ItemRadiusProperty

Defines the ItemRadius attached property.

```
public static readonly AttachedProperty<double> ItemRadiusProperty
```

Field Value

AttachedProperty<[double](#)>

KeepInBoundsProperty

Defines the [KeepInBounds](#) property.

```
public static readonly StyledProperty<bool> KeepInBoundsProperty
```

Field Value

StyledProperty<[bool](#)>

RadiusProperty

Defines the [Radius](#) property.

```
public static readonly StyledProperty<double> RadiusProperty
```

Field Value

StyledProperty<[double](#)>

StartAngleProperty

Defines the [StartAngle](#) property.

```
public static readonly StyledProperty<double> StartAngleProperty
```

Field Value

StyledProperty<[double](#)>

SweepDirectionProperty

Defines the [SweepDirection](#) property.

```
public static readonly StyledProperty<SweepDirection> SweepDirectionProperty
```

Field Value

Properties

AngleStep

Gets or sets the angle increment between children. Use NaN for auto ($360^\circ / \text{count}$).

```
public double AngleStep { get; set; }
```

Property Value

[double](#)

KeepInBounds

Gets or sets whether to ensure children stay within panel bounds.

```
public bool KeepInBounds { get; set; }
```

Property Value

[bool](#)

Radius

Gets or sets the default radius for all children.

```
public double Radius { get; set; }
```

Property Value

[double](#)

StartAngle

Gets or sets the starting angle in degrees (0° = right, 90° = bottom).

```
public double StartAngle { get; set; }
```

Property Value

[double](#)

SweepDirection

Gets or sets the direction to arrange children.

```
public SweepDirection SweepDirection { get; set; }
```

Property Value

SweepDirection

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetAlignment(Control)

Gets the Alignment attached property value for an element.

```
public static CircularAlignment GetAlignment(Control element)
```

Parameters

element Control

Returns

[CircularAlignment](#)

GetAngle(Control)

Gets the Angle attached property value for an element.

```
public static double GetAngle(Control element)
```

Parameters

element Control

Returns

[double](#)

GetItemRadius(Control)

Gets the ItemRadius attached property value for an element.

```
public static double GetItemRadius(Control element)
```

Parameters

element Control

Returns

[double](#)

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

SetAlignment(Control, CircularAlignment)

Sets the Alignment attached property value for an element.

```
public static void SetAlignment(Control element, CircularAlignment value)
```

Parameters

element Control

value [CircularAlignment](#)

SetAngle(Control, double)

Sets the Angle attached property value for an element.

```
public static void SetAngle(Control element, double value)
```

Parameters

element Control

value [double](#)

SetItemRadius(Control, double)

Sets the ItemRadius attached property value for an element.

```
public static void SetItemRadius(Control element, double value)
```

Parameters

element Control

value [double](#)

Class HexPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements in a honeycomb (hexagonal grid) pattern. Based on <https://github.com/AlexanderSharykin/HexGrid> by Alexander Sharykin.

```
public class HexPanel : Panel, INotifyPropertyChanged, IDataContextProvider,  
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,  
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,  
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← HexPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ColumnCountProperty

Defines the [ColumnCount](#) property.

```
public static readonly StyledProperty<int> ColumnCountProperty
```

Field Value

StyledProperty<[int](#)>

ColumnProperty

Defines the Column attached property.

```
public static readonly AttachedProperty<int> ColumnProperty
```

Field Value

AttachedProperty<[int](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

RowCountProperty

Defines the [RowCount](#) property.

```
public static readonly StyledProperty<int> RowCountProperty
```

Field Value

StyledProperty<[int](#)>

RowProperty

Defines the Row attached property.

```
public static readonly AttachedProperty<int> RowProperty
```

Field Value

AttachedProperty<[int](#)>

Properties

ColumnCount

Gets or sets the number of columns in the grid.

```
public int ColumnCount { get; set; }
```

Property Value

[int](#)

Orientation

Gets or sets the orientation of the hexagonal grid.

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

RowCount

Gets or sets the number of rows in the grid.

```
public int RowCount { get; set; }
```

Property Value

[int↗](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetColumn(Control)

Gets the Column attached property value for an element.

```
public static int GetColumn(Control element)
```

Parameters

element Control

Returns

[int↗](#)

GetRow(Control)

Gets the Row attached property value for an element.

```
public static int GetRow(Control element)
```

Parameters

element Control

Returns

[int](#)

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

SetColumn(Control, int)

Sets the Column attached property value for an element.

```
public static void SetColumn(Control element, int value)
```

Parameters

element Control

value [int](#)

SetRow(Control, int)

Sets the Row attached property value for an element.

```
public static void SetRow(Control element, int value)
```

Parameters

element Control

value [int](#)

Class LoopPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

An infinite/looping scrolling panel that wraps its children seamlessly.

```
public class LoopPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← LoopPanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
```

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

LoopPanel()

```
public LoopPanel()
```

Fields

AnchorPositionProperty

Defines the [AnchorPosition](#) property.

```
public static readonly StyledProperty<double> AnchorPositionProperty
```

Field Value

StyledProperty<[double](#)>

IsInertiaEnabledProperty

Defines the [IsInertiaEnabled](#) property.

```
public static readonly StyledProperty<bool> IsInertiaEnabledProperty
```

Field Value

StyledProperty<[bool](#)>

OffsetProperty

Defines the [Offset](#) property.

```
public static readonly StyledProperty<double> OffsetProperty
```

Field Value

StyledProperty<[double](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

ScrollFactorProperty

Defines the [ScrollFactor](#) property.

```
public static readonly StyledProperty<double> ScrollFactorProperty
```

Field Value

StyledProperty<[double](#)>

SnapToItemsProperty

Defines the [SnapToItems](#) property.

```
public static readonly StyledProperty<bool> SnapToItemsProperty
```

Field Value

StyledProperty<[bool](#)>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

Properties

AnchorPosition

Gets or sets the viewport anchor (0.0=start, 0.5=center, 1.0=end).

```
public double AnchorPosition { get; set; }
```

Property Value

[double](#)

IsInertiaEnabled

Gets or sets whether momentum scrolling is enabled.

```
public bool IsInertiaEnabled { get; set; }
```

Property Value

[bool](#)

Offset

Gets or sets the fractional scroll position. 1.0 = one full item.

```
public double Offset { get; set; }
```

Property Value

[double](#)

Orientation

Gets or sets the layout direction.

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

ScrollFactor

Gets or sets the multiplier for drag/wheel sensitivity.

```
public double ScrollFactor { get; set; }
```

Property Value

[double](#)

SnapToItems

Gets or sets whether to snap to the nearest whole item when scrolling ends.

```
public bool SnapToItems { get; set; }
```

Property Value

[bool](#)

Spacing

Gets or sets the gap between items.

```
public double Spacing { get; set; }
```

Property Value

[double](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

ScrollBy(double)

Scroll by specified viewport units.

```
public void ScrollBy(double units)
```

Parameters

`units` [double](#)

ScrollToIndex(int, bool)

Scroll to bring specific child to anchor.

```
public void ScrollToIndex(int index, bool animate = false)
```

Parameters

`index` [int](#)

`animate` [bool](#)

Events

CurrentIndexChanged

Raised when the pivotal (center) item changes.

```
public event EventHandler<int>? CurrentIndexChanged
```

Event Type

[EventHandler](#)<[int](#)>

Class OrbitPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements in concentric orbit rings around a center point. Each child can be assigned to a specific orbit (ring) and will be distributed evenly within that orbit.

```
public class OrbitPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← OrbitPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

InnerRadiusProperty

Defines the [InnerRadius](#) property.

```
public static readonly StyledProperty<double> InnerRadiusProperty
```

Field Value

StyledProperty<[double](#)>

OrbitProperty

Defines the Orbit attached property (0 = center, 1 = first ring, etc.).

```
public static readonly AttachedProperty<int> OrbitProperty
```

Field Value

AttachedProperty<[int](#)>

OrbitSpacingProperty

Defines the [OrbitSpacing](#) property.

```
public static readonly StyledProperty<double> OrbitSpacingProperty
```

Field Value

StyledProperty<[double](#)>

StartAngleProperty

Defines the [StartAngle](#) property.

```
public static readonly StyledProperty<double> StartAngleProperty
```

Field Value

StyledProperty<[double](#)>

Properties

InnerRadius

Gets or sets the radius of the innermost orbit ring.

```
public double InnerRadius { get; set; }
```

Property Value

[double](#)

OrbitSpacing

Gets or sets the spacing between orbit rings.

```
public double OrbitSpacing { get; set; }
```

Property Value

[double](#)

StartAngle

Gets or sets the starting angle in degrees.

```
public double StartAngle { get; set; }
```

Property Value

[double](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetOrbit(Control)

Gets the Orbit attached property value.

```
public static int GetOrbit(Control element)
```

Parameters

element Control

Returns

int

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

SetOrbit(Control, int)

Sets the Orbit attached property value.

```
public static void SetOrbit(Control element, int value)
```

Parameters

element Control

value int ↗

Class OverlapPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that stacks children with configurable X/Y offsets, creating an overlapping card effect.

```
public class OverlapPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← OverlapPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

OffsetXProperty

Defines the [OffsetX](#) property.

```
public static readonly StyledProperty<double> OffsetXProperty
```

Field Value

StyledProperty<[double](#)>

OffsetYProperty

Defines the [OffsetY](#) property.

```
public static readonly StyledProperty<double> OffsetYProperty
```

Field Value

StyledProperty<[double](#)>

ReverseZIndexProperty

Defines the [ReverseZIndex](#) property.

```
public static readonly StyledProperty<bool> ReverseZIndexProperty
```

Field Value

StyledProperty<[bool](#)>

Properties

OffsetX

Gets or sets the horizontal offset between items.

```
public double OffsetX { get; set; }
```

Property Value

[double](#)

OffsetY

Gets or sets the vertical offset between items.

```
public double OffsetY { get; set; }
```

Property Value

[double](#)

ReverseZIndex

Gets or sets whether to reverse the Z-index stacking order. When false (default), later items appear on top. When true, earlier items appear on top.

```
public bool ReverseZIndex { get; set; }
```

Property Value

[bool](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Class RadialPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements in a radial fan pattern emanating from a center point.

```
public class RadialPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← RadialPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ItemAngleProperty

Defines the [ItemAngle](#) property for individual item rotation.

```
public static readonly StyledProperty<double> ItemAngleProperty
```

Field Value

StyledProperty<[double](#)>

RadiusProperty

Defines the [Radius](#) property.

```
public static readonly StyledProperty<double> RadiusProperty
```

Field Value

StyledProperty<[double](#)>

RotateItemsProperty

Defines the [RotateItems](#) property.

```
public static readonly StyledProperty<bool> RotateItemsProperty
```

Field Value

StyledProperty<[bool](#)>

StartAngleProperty

Defines the [StartAngle](#) property.

```
public static readonly StyledProperty<double> StartAngleProperty
```

Field Value

StyledProperty<[double](#)>

SweepAngleProperty

Defines the [SweepAngle](#) property.

```
public static readonly StyledProperty<double> SweepAngleProperty
```

Field Value

StyledProperty<[double](#)>

Properties

ItemAngle

Gets or sets an additional rotation offset for all items.

```
public double ItemAngle { get; set; }
```

Property Value

[double](#)

Radius

Gets or sets the radius from center to item centers.

```
public double Radius { get; set; }
```

Property Value

[double](#)

RotateItems

Gets or sets whether items should rotate to face outward from center.

```
public bool RotateItems { get; set; }
```

Property Value

[bool](#) ↗

StartAngle

Gets or sets the starting angle in degrees (0 = right, 90 = down).

```
public double StartAngle { get; set; }
```

Property Value

[double](#) ↗

SweepAngle

Gets or sets the total angle span for items.

```
public double SweepAngle { get; set; }
```

Property Value

[double](#) ↗

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize [Size](#)

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Class RatingControl

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A Rating Control with support to various shapes or precision levels.

```
public class RatingControl : TemplatedControl, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode,  
IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize,  
IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

[Object](#) ↗ ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← TemplatedControl ← RatingControl

Implements

[INotifyPropertyChanged](#) ↗, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#) ↗, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue

Inherited Members

TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,
TemplatedControl.GetIsTemplateFocusTarget(Control) ,
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) ↗ , TemplatedControl.ApplyTemplate() ,
TemplatedControl.GetTemplateFocusTarget() ,
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,
TemplatedControl.BorderBrush , TemplatedControl.BorderThickness ,
TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,

TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(SizeChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEnabledEffectivelyProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnTextInput(TextInputEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.HitTestVisible , InputElement.PointerOver , InputElement.TabStop ,
InputElement.IsEnabledEffectively , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,

[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.Render(DrawingContext) , Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,

StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

CustomGeometryProperty

Defines the [CustomGeometry](#) property.

```
public static readonly StyledProperty<Geometry?> CustomGeometryProperty
```

Field Value

StyledProperty<Geometry>

IsReadOnlyProperty

Defines the [IsReadOnly](#) property.

```
public static readonly StyledProperty<bool> IsReadOnlyProperty
```

Field Value

StyledProperty<[bool](#)>

ItemCountProperty

Defines the [ItemCount](#) property.

```
public static readonly StyledProperty<int> ItemCountProperty
```

Field Value

StyledProperty<[int](#)>

ItemSizeProperty

Defines the [ItemSize](#) property.

```
public static readonly StyledProperty<double> ItemSizeProperty
```

Field Value

StyledProperty<[double](#)>

ItemSpacingProperty

Defines the [ItemSpacing](#) property.

```
public static readonly StyledProperty<double> ItemSpacingProperty
```

Field Value

StyledProperty<[double](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

PrecisionProperty

Defines the [Precision](#) property.

```
public static readonly StyledProperty<RatingPrecision> PrecisionProperty
```

Field Value

StyledProperty<[RatingPrecision](#)>

PreviewFillProperty

Defines the [PreviewFill](#) property.

```
public static readonly StyledProperty<IBrush?> PreviewFillProperty
```

Field Value

StyledProperty<IBrush>

PreviewStrokeProperty

Defines the [PreviewStroke](#) property.

```
public static readonly StyledProperty<IBrush?> PreviewStrokeProperty
```

Field Value

StyledProperty<IBrush>

RatedFillProperty

Defines the [RatedFill](#) property.

```
public static readonly StyledProperty<IBrush?> RatedFillProperty
```

Field Value

StyledProperty<IBrush>

RatedStrokeProperty

Defines the [RatedStroke](#) property.

```
public static readonly StyledProperty<IBrush?> RatedStrokeProperty
```

Field Value

StyledProperty<IBrush>

ShapeProperty

Defines the [Shape](#) property.

```
public static readonly StyledProperty<RatingShape> ShapeProperty
```

Field Value

StyledProperty<[RatingShape](#)>

StrokeThicknessProperty

Defines the [StrokeThickness](#) property.

```
public static readonly StyledProperty<double> StrokeThicknessProperty
```

Field Value

StyledProperty<[double](#)>

UnratedFillProperty

Defines the [UnratedFill](#) property.

```
public static readonly StyledProperty<IBrush?> UnratedFillProperty
```

Field Value

StyledProperty<IBrush>

UnratedStrokeProperty

Defines the [UnratedStroke](#) property.

```
public static readonly StyledProperty<IBrush?> UnratedStrokeProperty
```

Field Value

StyledProperty<IBrush>

ValueChangedEvent

Defines the [ValueChanged](#) routed event.

```
public static readonly RoutedEvent<RoutedEventArgs> ValueChangedEvent
```

Field Value

RoutedEvent<RoutedEventArgs>

ValueProperty

Defines the [Value](#) property.

```
public static readonly StyledProperty<double> ValueProperty
```

Field Value

StyledProperty<[double](#)>

Properties

CustomGeometry

Gets or sets the custom geometry to use when Shape is set to Custom.

```
public Geometry? CustomGeometry { get; set; }
```

Property Value

Geometry

IsReadOnly

Gets or sets a value indicating whether the control is read-only.

```
public bool IsReadOnly { get; set; }
```

Property Value

[bool](#)

ItemCount

Gets or sets the number of items (stars, hearts, etc.) to display.

```
public int ItemCount { get; set; }
```

Property Value

[int](#)

ItemSize

Gets or sets the size of each rating item.

```
public double ItemSize { get; set; }
```

Property Value

[double](#)

ItemSpacing

Gets or sets the spacing between rating items.

```
public double ItemSpacing { get; set; }
```

Property Value

[double](#)

Orientation

Gets or sets the orientation of the rating items (Horizontal or Vertical).

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

Precision

Gets or sets the precision of the rating (Full, Half, or Exact).

```
public RatingPrecision Precision { get; set; }
```

Property Value

[RatingPrecision](#)

PreviewFill

Gets or sets the fill brush used when hovering over items.

```
public IBrush? PreviewFill { get; set; }
```

Property Value

IBrush

PreviewStroke

Gets or sets the stroke brush used when hovering over items.

```
public IBrush? PreviewStroke { get; set; }
```

Property Value

IBrush

RatedFill

Gets or sets the fill brush for rated items (active state).

```
public IBrush? RatedFill { get; set; }
```

Property Value

IBrush

RatedStroke

Gets or sets the stroke brush for rated items.

```
public IBrush? RatedStroke { get; set; }
```

Property Value

IBrush

Shape

Gets or sets the shape of the rating items.

```
public RatingShape Shape { get; set; }
```

Property Value

[RatingShape](#)

StrokeThickness

Gets or sets the thickness of the stroke for items.

```
public double StrokeThickness { get; set; }
```

Property Value

[double](#)

UnratedFill

Gets or sets the fill brush for unrated items (inactive state).

```
public IBrush? UnratedFill { get; set; }
```

Property Value

IBrush

UnratedStroke

Gets or sets the stroke brush for unrated items.

```
public IBrush? UnratedStroke { get; set; }
```

Property Value

IBrush

Value

Gets or sets the current value of the rating.

```
public double Value { get; set; }
```

Property Value

[double](#)

Methods

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e [TemplateAppliedEventArgs](#)

The event args.

OnKeyDown(KeyEventEventArgs)

Invoked when an unhandled `Avalonia.InputInputElement.KeyDownEvent` reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnKeyDown(KeyEventEventArgs e)
```

Parameters

e [KeyEventEventArgs](#)

Data about the event.

OnPointerEntered(PointerEventArgs)

Invoked when an unhandled Avalonia.Input.InputElement.PointerEnteredEvent reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnPointerEntered(PointerEventArgs e)
```

Parameters

e PointerEventArgs

Data about the event.

OnPointerExited(PointerEventArgs)

Invoked when an unhandled Avalonia.Input.InputElement.PointerExitedEvent reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnPointerExited(PointerEventArgs e)
```

Parameters

e PointerEventArgs

Data about the event.

OnPointerMoved(PointerEventArgs)

Invoked when an unhandled Avalonia.Input.InputElement.PointerMovedEvent reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnPointerMoved(PointerEventArgs e)
```

Parameters

e PointerEventArgs

Data about the event.

OnPointerPressed(PointerPressedEventArgs)

Invoked when an unhandled `Avalonia.Input.InputElement.PointerPressedEvent` reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnPointerPressed(PointerPressedEventArgs e)
```

Parameters

`e` PointerPressedEventArgs

Data about the event.

OnPointerReleased(PointerReleasedEventArgs)

Invoked when an unhandled `Avalonia.Input.InputElement.PointerReleasedEvent` reaches an element in its route that is derived from this class. Implement this method to add class handling for this event.

```
protected override void OnPointerReleased(PointerReleasedEventArgs e)
```

Parameters

`e` PointerReleasedEventArgs

Data about the event.

Events

ValueChanged

Occurs when the [Value](#) property changes.

```
public event EventHandler<RoutedEventArgs>? ValueChanged
```

Event Type

[EventHandler](#)<RoutedEventArgs>

Class RatingItem

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Represents an individual item within the RatingControl (e.g., a single star).

```
public class RatingItem : Control, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

[Object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← RatingItem

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue

Inherited Members

Control.FocusAdornerProperty , Control.TagProperty , Control.ContextMenuProperty ,
Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,

`InputElement.IsTabStopProperty`, `InputElement.GotFocusEvent`, `InputElement.LostFocusEvent`,
`InputElement.KeyDownEvent`, `InputElement.KeyUpEvent`, `InputElement.TabIndexProperty`,
`InputElement.TextInputEvent`, `InputElement.TextInputMethodClientRequestedEvent`,
`InputElement.PointerEnteredEvent`, `InputElement.PointerExitedEvent`,
`InputElement.PointerMovedEvent`, `InputElement.PointerPressedEvent`,
`InputElement.PointerReleasedEvent`, `InputElement.PointerCaptureLostEvent`,
`InputElement.PointerWheelChangedEvent`, `InputElement.TappedEvent`, `InputElement.HoldingEvent`,
`InputElement.DoubleTappedEvent`, `InputElement.Focus(NavigationMethod, KeyModifiers)`,
`InputElement.OnAccessKey(RoutedEventArgs)`, `InputElement.OnKeyDown(KeyEventArgs)`,
`InputElement.OnTextInput(TextInputEventArgs)`, `InputElement.OnPointerEntered(PointerEventArgs)`,
`InputElement.OnPointerExited(PointerEventArgs)`, `InputElement.OnPointerMoved(PointerEventArgs)`,
`InputElement.OnPointerPressed(PointerPressedEventArgs)`,
`InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs)`,
`InputElement.OnPointerWheelChanged(PointerWheelEventArgs)`,
`InputElement.UpdateIsEffectivelyEnabled()`, `InputElement.Focusable`, `InputElement.IsEnabled`,
`InputElement.Cursor`, `InputElement.IsKeyboardFocusWithin`, `InputElement.Focused`,
`InputElement.HitTestVisible`, `InputElement.PointerOver`, `InputElement.TabStop`,
`InputElement.IsEffectivelyEnabled`, `InputElement.TabIndex`, `InputElement.KeyBindings`,
`InputElement.IsEnabledCore`, `InputElement.GestureRecognizers`, `InputElement.GotFocus`,
`InputElement.LostFocus`, `InputElement.KeyDown`, `InputElement.KeyUp`, `InputElement.TextInput`,
`InputElement.TextInputMethodClientRequested`, `InputElement.PointerEntered`,
`InputElement.PointerExited`, `InputElement.PointerMoved`, `InputElement.PointerPressed`,
`InputElement.PointerReleased`, `InputElement.PointerCaptureLost`,
`InputElement.PointerWheelChanged`, `InputElement.Tapped`, `InputElement.Holding`,
`InputElement.DoubleTapped`,

[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEventArgs<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEventArgs<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.Horizontal_ALIGNMENTProperty , Layoutable.Vertical_ALIGNMENTProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,

Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,

AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

FillRatio

Gets or sets the ratio of the item that is filled (0.0 to 1.0).

```
public double FillRatio { get; set; }
```

Property Value

[double](#)

Geometry

Gets or sets the geometry to render for this item.

```
public Geometry? Geometry { get; set; }
```

Property Value

Geometry

RatedFill

Gets or sets the fill brush for the rated portion.

```
public IBrush? RatedFill { get; set; }
```

Property Value

IBrush

RatedStroke

Gets or sets the stroke brush for the rated portion.

```
public IBrush? RatedStroke { get; set; }
```

Property Value

IBrush

StrokeThickness

Gets or sets the thickness of the stroke.

```
public double StrokeThickness { get; set; }
```

Property Value

[double](#)

UnratedFill

Gets or sets the fill brush for the unrated portion.

```
public IBrush? UnratedFill { get; set; }
```

Property Value

IBrush

UnratedStroke

Gets or sets the stroke brush for the unrated portion.

```
public IBrush? UnratedStroke { get; set; }
```

Property Value

IBrush

Methods

Render(DrawingContext)

Renders the visual to a Avalonia.Media.DrawingContext.

```
public override void Render(DrawingContext context)
```

Parameters

context DrawingContext

The drawing context.

Enum RatingPrecision

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Specifies the precision level of the RatingControl.

```
public enum RatingPrecision
```

Fields

Exact = 2

Allows exact/continuous ratings (e.g., 3.7, 4.2).

Full = 0

Allows only full integer ratings (e.g., 1, 2, 3).

Half = 1

Allows half-star ratings (e.g., 2.5, 3.5).

Enum RatingShape

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Specifies the shape of the items in the RatingControl.

```
public enum RatingShape
```

Fields

Circle = 2

Displays circles.

Custom = 4

Displays a custom geometry defined by the CustomGeometry property.

Diamond = 3

Displays diamonds.

Heart = 1

Displays hearts.

Star = 0

Displays stars.

Enum ResponsiveBreakpoint

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Defines the size classes (breakpoints) for responsive layouts.

```
[Flags]
public enum ResponsiveBreakpoint
```

Fields

All = Narrow | Normal | Wide

Matches all breakpoints.

Narrow = 1

Narrow layout (e.g., mobile devices).

None = 0

No breakpoint specified.

Normal = 2

Normal layout (e.g., tablets).

Wide = 4

Wide layout (e.g., desktop monitors).

Class ResponsivePanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that selectively shows its children based on the current width breakpoint.

```
public class ResponsivePanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←
InputElement ← Control ← Panel ← ResponsivePanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inherited Members

```
Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
```

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.Initialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ConditionProperty

Defines the Condition attached property.

```
public static readonly AttachedProperty<ResponsiveBreakpoint> ConditionProperty
```

Field Value

AttachedProperty<[ResponsiveBreakpoint](#)>

NarrowBreakpointProperty

Defines the [NarrowBreakpoint](#) property.

```
public static readonly StyledProperty<double> NarrowBreakpointProperty
```

Field Value

StyledProperty<[double](#)>

TransitionProperty

Defines the [Transition](#) property.

```
public static readonly StyledProperty<IPageTransition?> TransitionProperty
```

Field Value

StyledProperty<IPageTransition>

WideBreakpointProperty

Defines the [WideBreakpoint](#) property.

```
public static readonly StyledProperty<double> WideBreakpointProperty
```

Field Value

StyledProperty<[double](#)>

Properties

NarrowBreakpoint

Gets or sets the threshold for the narrow layout class.

```
public double NarrowBreakpoint { get; set; }
```

Property Value

[double](#)

Transition

Gets or sets the transition to use when switching between visible children.

```
public IPageTransition? Transition { get; set; }
```

Property Value

IPageTransition

WideBreakpoint

Gets or sets the threshold for the wide layout class.

```
public double WideBreakpoint { get; set; }
```

Property Value

[double](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetCondition(Control)

Gets the [ConditionProperty](#) for an element.

```
public static ResponsiveBreakpoint GetCondition(Control element)
```

Parameters

element Control

Returns

[ResponsiveBreakpoint](#)

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

SetCondition(Control, ResponsiveBreakpoint)

Sets the [ConditionProperty](#) for an element.

```
public static void SetCondition(Control element, ResponsiveBreakpoint value)
```

Parameters

element Control

value [ResponsiveBreakpoint](#)

Class Shimmer

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A content-aware skeleton loading control. Uses immediate mode rendering to draw a "shimmer" effect over the content layout. Automatically detects text, shapes, and buttons to generate matching placeholders. Includes A11y support and a synchronized animation clock to prevent visual clutter.

```
public class Shimmer : ContentControl, INotifyPropertyChanged, IDataContextProvider,  
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,  
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,  
IInputElement, IDataTemplateHost, ISetterValue
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← TemplatedControl ← ContentControl ← Shimmer

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue

Inherited Members

ContentControl.ContentProperty , ContentControl.ContentTemplateProperty ,
ContentControl.HorizontalContentAlignmentProperty ,
ContentControl.VerticalContentAlignmentProperty ,
ContentControl.RegisterContentPresenter(ContentPresenter) , ContentControl.Content ,
ContentControl.ContentTemplate , ContentControl.Presenter ,
ContentControl.HorizontalContentAlignment , ContentControl.VerticalContentAlignment ,
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,
TemplatedControl.GetIsTemplateFocusTarget(Control) ,
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) , TemplatedControl.ApplyTemplate() ,

TemplatedControl.GetTemplateFocusTarget() ,
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,
TemplatedControl.Background , TemplatedControl.BackgroundSizing ,
TemplatedControl.BorderBrush , TemplatedControl.BorderThickness ,
TemplatedControl.CornerRadius , TemplatedControl.FontFamily , TemplatedControl.FontFeatures ,
TemplatedControl.FontSize , TemplatedControl.FontStyle , TemplatedControl.FontWeight ,
TemplatedControl.FontStretch , TemplatedControl.Foreground , TemplatedControl.Padding ,
TemplatedControl.Template , TemplatedControl.TemplateApplied , Control.FocusAdornerProperty ,
Control.TagProperty , Control.ContextMenuProperty , Control.ContextFlyoutProperty ,
Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent , Control.LoadedEvent ,
Control.UnloadedEvent , Control.SizeChangedEvent , Control.OnLoaded(RoutedEventArgs) ,
Control.OnUnloaded(RoutedEventArgs) , Control.OnSizeChanged(SizeChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventEventArgs) ,
Control.FocusAdorner , Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout ,
Control.IsLoaded , Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded ,
Control.SizeChanged , InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.HitTestVisibleProperty , InputElement.PointerOverProperty ,
InputElement.TabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,

InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.Measure(Size) ,
Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,
Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.MeasureOverride(Size) , Layoutable.ArrangeCore(Rect) ,
Layoutable.ArrangeOverride(Size) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,

Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,

```
AvaloniaObject.CoerceValue(AvaloniaProperty) ,  
AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\) ,  
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,  
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,  
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,  
AvaloniaObject.PropertyChanged , object.Equals\(object, object\) , object.GetType\(\) ,  
object.MemberwiseClone\(\) , object.ReferenceEquals\(object, object\) , object.ToString\(\)
```

Fields

HighlightBrushProperty

The brush used for the moving gradient highlight. Use this with DynamicResource to support theme switching.

```
public static readonly StyledProperty<IBrush?> HighlightBrushProperty
```

Field Value

StyledProperty<IBrush>

IsLoadingProperty

Defines whether the shimmer effect is active. When true, content is hidden, interaction is disabled, and the skeleton is drawn.

```
public static readonly StyledProperty<bool> IsLoadingProperty
```

Field Value

StyledProperty<[bool](#)>

LoadingTextProperty

The text announced by screen readers when the control is in the loading state.

```
public static readonly StyledProperty<string> LoadingTextProperty
```

Field Value

StyledProperty<[string](#)>

ShimmerAngleProperty

The angle of the shimmer gradient in degrees.

```
public static readonly StyledProperty<double> ShimmerAngleProperty
```

Field Value

StyledProperty<[double](#)>

ShimmerOpacityProperty

The opacity of the shimmer wave.

```
public static readonly StyledProperty<double> ShimmerOpacityProperty
```

Field Value

StyledProperty<[double](#)>

Properties

AutomationControlType

Adjusts the control type for accessibility tools.

```
public AutomationControlType AutomationControlType { get; set; }
```

Property Value

HighlightBrush

The brush used for the moving gradient highlight. Use this with DynamicResource to support theme switching.

```
public IBrush? HighlightBrush { get; set; }
```

Property Value

IBrush

IsLoading

Defines whether the shimmer effect is active. When true, content is hidden, interaction is disabled, and the skeleton is drawn.

```
public bool IsLoading { get; set; }
```

Property Value

[bool](#)

LoadingText

The text announced by screen readers when the control is in the loading state.

```
public string LoadingText { get; set; }
```

Property Value

[string](#)

ShimmerAngle

The angle of the shimmer gradient in degrees.

```
public double ShimmerAngle { get; set; }
```

Property Value

[double](#)

ShimmerOpacity

The opacity of the shimmer wave.

```
public double ShimmerOpacity { get; set; }
```

Property Value

[double](#)

Methods

OnApplyTemplate(TemplateAppliedEventArgs)

Called when the control's template is applied. In simple terms, this means the method is called just before the control is displayed.

```
protected override void OnApplyTemplate(TemplateAppliedEventArgs e)
```

Parameters

e `TemplateAppliedEventArgs`

The event args.

OnAttachedToVisualTree(VisualTreeAttachmentEventArgs)

Called when the control is added to a rooted visual tree.

```
protected override void OnAttachedToVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

e VisualTreeAttachmentEventArgs

The event args.

OnCreateAutomationPeer()

Returns a new, type-specific Avalonia.Automation.Peers.AutomationPeer implementation for the control.

```
protected override AutomationPeer OnCreateAutomationPeer()
```

Returns

AutomationPeer

The type-specific Avalonia.Automation.Peers.AutomationPeer implementation.

OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs)

Called when the control is removed from a rooted visual tree.

```
protected override void OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

e VisualTreeAttachmentEventArgs

The event args.

OnPropertyChanged(AvaloniaPropertyChangedEventArgs)

Called when a avalonia property changes on the object.

```
protected override void OnPropertyChanged(AvaloniaPropertyChangedEventArgs change)
```

Parameters

change AvaloniaPropertyChangedEventArgs

The property change details.

Render(DrawingContext)

Renders the visual to a Avalonia.Media.DrawingContext.

```
public override void Render(DrawingContext context)
```

Parameters

context DrawingContext

The drawing context.

Class ShimmerAutomationPeer

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

Automation peer for the [Shimmer](#) control, exposed as a loading indicator.

```
public sealed class ShimmerAutomationPeer : ControlAutomationPeer
```

Inheritance

[object](#) ↗ ← AutomationPeer ← ControlAutomationPeer ← ShimmerAutomationPeer

Inherited Members

ControlAutomationPeer.GetOrCreate(Control) ,
ControlAutomationPeer.CreatePeerForElement(Control) ,
ControlAutomationPeer.FromElement(Control) , ControlAutomationPeer.Owner ,
AutomationPeer.BringIntoView() , AutomationPeer.GetAcceleratorKey() ,
AutomationPeer.GetAccessKey() , AutomationPeer.GetAutomationControlType() ,
AutomationPeer.GetAutomationId() , AutomationPeer.GetBoundingBox() ,
AutomationPeer.GetChildren() , AutomationPeer.GetClassName() , AutomationPeer.GetLabeledBy() ,
AutomationPeer.GetLocalizedControlType() , AutomationPeer.GetName() ,
AutomationPeer.GetHelpText() , AutomationPeer.GetHeadingLevel() , AutomationPeer.GetParent() ,
AutomationPeer.GetVisualRoot() , AutomationPeer.HasKeyboardFocus() ,
AutomationPeer.IsContentElement() , AutomationPeer.IsControlElement() ,
AutomationPeer.IsEnabled() , AutomationPeer.IsKeyboardFocusable() , AutomationPeer.IsOffscreen() ,
AutomationPeer.SetFocus() , AutomationPeer.ShowContextMenu() ,
AutomationPeer.GetProvider<T>() ,
[AutomationPeer.RaisePropertyChangedEvent\(AutomationProperty, object, object\)](#) ↗ ,
AutomationPeer.ChildrenChanged , AutomationPeer.PropertyChanged , [object.Equals\(object\)](#) ↗ ,
[object.Equals\(object, object\)](#) ↗ , [object.GetHashCode\(\)](#) ↗ , [object.GetType\(\)](#) ↗ ,
[object.ReferenceEquals\(object, object\)](#) ↗ , [object.ToString\(\)](#) ↗

Constructors

ShimmerAutomationPeer(Shimmer)

Creates a peer for the provided shimmer instance.

```
public ShimmerAutomationPeer(Shimmer owner)
```

Parameters

owner [Shimmer](#)

Methods

GetAutomationControlTypeCore()

```
protected override AutomationControlType GetAutomationControlTypeCore()
```

Returns

AutomationControlType

GetNameCore()

```
protected override string GetNameCore()
```

Returns

[string](#)

IsContentElementCore()

```
protected override bool IsContentElementCore()
```

Returns

[bool](#)

Class StaggeredPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements into a staggered grid pattern, usually adding items to the column with the least amount of space used.

```
public class StaggeredPanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← StaggeredPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ColumnSpacingProperty

Defines the [ColumnSpacing](#) property.

```
public static readonly StyledProperty<double> ColumnSpacingProperty
```

Field Value

StyledProperty<[double](#)>

DesiredColumnWidthProperty

Defines the [DesiredColumnWidth](#) property.

```
public static readonly StyledProperty<double> DesiredColumnWidthProperty
```

Field Value

StyledProperty<[double](#)>

PaddingProperty

Defines the [Padding](#) property.

```
public static readonly StyledProperty<Thickness> PaddingProperty
```

Field Value

StyledProperty<Thickness>

RowSpacingProperty

Defines the [RowSpacing](#) property.

```
public static readonly StyledProperty<double> RowSpacingProperty
```

Field Value

StyledProperty<[double](#)>

Properties

ColumnSpacing

Gets or sets the spacing between columns.

```
public double ColumnSpacing { get; set; }
```

Property Value

[double](#)

DesiredColumnWidth

Gets or sets the desired width for each column.

```
public double DesiredColumnWidth { get; set; }
```

Property Value

[double](#)

Padding

Gets or sets the padding around the panel content.

```
public Thickness Padding { get; set; }
```

Property Value

Thickness

RowSpacing

Gets or sets the spacing between rows of items.

```
public double RowSpacing { get; set; }
```

Property Value

[double ↗](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Class TimelinePanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements in a timeline pattern with spacing for connectors.

```
public class TimelinePanel : Panel, INotifyPropertyChanged, IDataContextProvider,
ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost,
ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable, INamed,
IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider
```

Inheritance

[Object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← TimelinePanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.ItemsSourceHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,
Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,

InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEvent<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,

Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.IsVisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,

Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

AlternateItemsProperty

Defines the [AlternateItems](#) property.

```
public static readonly StyledProperty<bool> AlternateItemsProperty
```

Field Value

StyledProperty<[bool](#)>

ConnectorWidthProperty

Defines the [ConnectorWidth](#) property.

```
public static readonly StyledProperty<double> ConnectorWidthProperty
```

Field Value

StyledProperty<[double](#)>

OrientationProperty

Defines the [Orientation](#) property.

```
public static readonly StyledProperty<Orientation> OrientationProperty
```

Field Value

StyledProperty<Orientation>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

Properties

AlternateItems

Gets or sets whether items alternate sides (left/right for vertical, top/bottom for horizontal).

```
public bool AlternateItems { get; set; }
```

Property Value

[bool](#)

ConnectorWidth

Gets or sets the width reserved for the connector line/markers.

```
public double ConnectorWidth { get; set; }
```

Property Value

[double](#)

Orientation

Gets or sets the orientation of the timeline.

```
public Orientation Orientation { get; set; }
```

Property Value

Orientation

Spacing

Gets or sets the spacing between items.

```
public double Spacing { get; set; }
```

Property Value

[double](#) ↗

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

Class VariableSizeWrapPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A panel that arranges child elements in a grid with variable-size tiles using RowSpan and ColumnSpan.

```
public class VariableSizeWrapPanel : Panel, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent,  
ISupportInitialize, IStyleable, INamed, IInputElement, IDataTemplateHost,  
ISetterValue, IChildIndexProvider
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← VariableSizeWrapPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider

Inherited Members

Panel.BackgroundProperty , Panel.Render(DrawingContext) ,
Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(EventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) ,
Control.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) , Control.FocusAdorner ,
Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded ,

Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,
Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEventArgs) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalAlignmentProperty , Layoutable.VerticalAlignmentProperty ,

Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializelfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,
StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,

StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ColumnSpanProperty

Defines the ColumnSpan attached property.

```
public static readonly AttachedProperty<int> ColumnSpanProperty
```

Field Value

AttachedProperty<[int](#)>

ColumnsProperty

Defines the [Columns](#) property.

```
public static readonly StyledProperty<int> ColumnsProperty
```

Field Value

StyledProperty<[int](#)>

RowSpanProperty

Defines the RowSpan attached property.

```
public static readonly AttachedProperty<int> RowSpanProperty
```

Field Value

AttachedProperty<[int](#)>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

TileSizeProperty

Defines the [TileSize](#) property.

```
public static readonly StyledProperty<double> TileSizeProperty
```

Field Value

StyledProperty<[double](#)>

Properties

Columns

Gets or sets the number of columns in the grid.

```
public int Columns { get; set; }
```

Property Value

[int](#)

Spacing

Gets or sets the spacing between tiles.

```
public double Spacing { get; set; }
```

Property Value

[double](#)

TileSize

Gets or sets the base size of a single tile unit.

```
public double TileSize { get; set; }
```

Property Value

[double ↗](#)

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

GetColumnSpan(Control)

Gets the ColumnSpan attached property value.

```
public static int GetColumnSpan(Control element)
```

Parameters

element Control

Returns

[int ↗](#)

GetRowSpan(Control)

Gets the RowSpan attached property value.

```
public static int GetRowSpan(Control element)
```

Parameters

element Control

Returns

[int](#)

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

SetColumnSpan(Control, int)

Sets the ColumnSpan attached property value.

```
public static void SetColumnSpan(Control element, int value)
```

Parameters

element Control

value [int](#)

SetRowSpan(Control, int)

Sets the RowSpan attached property value.

```
public static void SetRowSpan(Control element, int value)
```

Parameters

element Control

value [int](#)

Class VirtualizingStaggeredPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A virtualizing panel that arranges children in a staggered grid (masonry) layout. Items are placed in the shortest column, creating an efficient Pinterest-style layout. Only items within the visible viewport (plus buffer) are materialized for optimal performance.

```
public class VirtualizingStaggeredPanel : VirtualizingPanel, INotifyPropertyChanged,  
IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode,  
IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize,  
IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue,  
IChildIndexProvider, INavigableContainer
```

Inheritance

```
object ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← Panel ← VirtualizingPanel ← VirtualizingStaggeredPanel
```

Implements

```
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, ISupportInitialize, IStyleable,  
INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider, INavigableContainer
```

Inherited Members

```
VirtualizingPanel.OnItemsControlChanged(ItemsControl),  
VirtualizingPanel.AddInternalChild(Control), VirtualizingPanel.InsertInternalChild\(int, Control\),  
VirtualizingPanel.RemoveInternalChild(Control),  
VirtualizingPanel.RemoveInternalChildRange\(int, int\), VirtualizingPanel.ItemContainerGenerator,  
VirtualizingPanel.Items, VirtualizingPanel.ItemsControl, Panel.BackgroundProperty,  
Panel.Render(DrawingContext), Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]),  
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]),  
Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\), Panel.Children,  
Panel.Background, Panel.IsItemsHost, Control.FocusAdornerProperty, Control.TagProperty,  
Control.ContextMenuProperty, Control.ContextFlyoutProperty, Control.RequestBringIntoViewEvent,  
Control.ContextRequestedEvent, Control.LoadedEvent, Control.UnloadedEvent,  
Control.SizeChangedEvent, Control.GetTemplateFocusTarget(),  
Control.OnLoaded(RoutedEventArgs), Control.OnUnloaded(RoutedEventArgs),  
Control.OnSizeChanged(ChangedEventArgs),  
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs),
```

Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) , Control.FocusAdorner , Control.DataTemplates ,
Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded , Control.Tag ,
Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,

Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEvent) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalContentAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,

StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Examples

```
<ItemsControl ItemsSource="{Binding Items}">
    <ItemsControl.ItemsPanel>
        <ItemsPanelTemplate>
            <controls:VirtualizingStaggeredPanel DesiredColumnWidth="200">
```

```
ColumnSpacing="8" RowSpacing="8"/>
  </ItemsPanelTemplate>
</ItemsControl.ItemsPanel>
</ItemsControl>
```

Remarks

This panel virtualizes items by only creating UI containers for visible items. Off-screen items are recycled and reused when scrolling, minimizing memory usage and improving scroll performance for large datasets.

The panel automatically calculates the number of columns based on the available width and the [DesiredColumnWidth](#) property.

Constructors

VirtualizingStaggeredPanel()

```
public VirtualizingStaggeredPanel()
```

Fields

ColumnSpacingProperty

Defines the [ColumnSpacing](#) property.

```
public static readonly StyledProperty<double> ColumnSpacingProperty
```

Field Value

StyledProperty<[double](#)>

DesiredColumnWidthProperty

Defines the [DesiredColumnWidth](#) property.

```
public static readonly StyledProperty<double> DesiredColumnWidthProperty
```

Field Value

StyledProperty<[double](#)>

RowSpacingProperty

Defines the [RowSpacing](#) property.

```
public static readonly StyledProperty<double> RowSpacingProperty
```

Field Value

StyledProperty<[double](#)>

Properties

ColumnSpacing

Gets or sets the horizontal spacing between columns.

```
public double ColumnSpacing { get; set; }
```

Property Value

[double](#)

The horizontal spacing in device-independent pixels. Default is 0.

DesiredColumnWidth

Gets or sets the desired width of each column. The actual column width may vary slightly to fill the available space evenly.

```
public double DesiredColumnWidth { get; set; }
```

Property Value

[double](#)

The desired column width in device-independent pixels. Default is 250.

RowSpacing

Gets or sets the vertical spacing between items in the same column.

```
public double RowSpacing { get; set; }
```

Property Value

[double](#)

The vertical spacing in device-independent pixels. Default is 0.

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

ContainerFromIndex(int)

Returns the container for the item at the specified index.

```
protected override Control? ContainerFromIndex(int index)
```

Parameters

index [int](#)

The index of the item to retrieve.

Returns

Control

The container for the item at the specified index within the item collection, if the item is realized; otherwise, null.

Remarks

Note for implementors: if the item at the specified index is an `ItemIsOwnContainer` item that has previously been realized, then the item should be returned even if it currently falls outside the realized viewport.

GetControl(NavigationDirection, IInputElement?, bool)

Gets the next control in the specified direction.

```
protected override Control? GetControl(NavigationDirection direction, IInputElement? from, bool wrap)
```

Parameters

direction NavigationDirection

The movement direction.

```
from IInputElement
```

The control from which movement begins.

wrap [bool](#)

Whether to wrap around when the first or last item is reached.

Returns

Control

The control.

GetRealizedContainers()

Gets the currently realized containers.

```
protected override IEnumerable<Control>? GetRealizedContainers()
```

Returns

[IEnumerable](#)<Control>

IndexFromContainer(Control)

Returns the index to the item that has the specified realized container.

```
protected override int IndexFromContainer(Control container)
```

Parameters

container Control

The generated container to retrieve the item index for.

Returns

[int](#)

The index to the item that corresponds to the specified realized container, or -1 if `container` is not found.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

`availableSize` `Size`

The size available to the control.

Returns

`Size`

The desired size for the control.

OnAttachedToVisualTree(VisualTreeAttachmentEventArgs)

Called when the control is added to a rooted visual tree.

```
protected override void OnAttachedToVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

`e` `VisualTreeAttachmentEventArgs`

The event args.

OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs)

Called when the control is removed from a rooted visual tree.

```
protected override void OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

e [VisualTreeAttachmentEventArgs](#)

The event args.

OnItemsChanged(IReadOnlyList<object?>, NotifyCollectionChangedEventArgs)

Called when the Avalonia.Controls.ItemsControl.Items collection of the owner Avalonia.Controls.VirtualizingPanel.ItemsControl changes.

```
protected override void OnItemsChanged(IReadOnlyList<object?> items,  
NotifyCollectionChangedEventArgs e)
```

Parameters

items [IReadOnlyList<object>](#)

The items.

e [NotifyCollectionChangedEventArgs](#)

The event args.

Remarks

This method is called a [INotifyCollectionChanged](#) event is raised by the items, or when the Avalonia.Controls.ItemsControl.Items property is assigned a new collection, in which case the [NotifyCollectionChangedAction](#) will be [Reset](#).

OnPropertyChanged(AvaloniaPropertyChangedEventArgs)

Called when a avalonia property changes on the object.

```
protected override void OnPropertyChanged(AvaloniaPropertyChangedEventArgs change)
```

Parameters

change AvaloniaPropertyChangedEventArgs

The property change details.

ScrollIntoView(int)

Scrolls the specified item into view.

```
protected override Control? ScrollIntoView(int index)
```

Parameters

index [int](#)

The index of the item.

Returns

Control

The element with the specified index, or null if the element could not be brought into view.

Class VirtualizingVariableSizeWrapPanel

Namespace: [Nova.Avalonia.UI.Controls](#)

Assembly: Nova.Avalonia.UI.dll

A virtualizing panel that arranges children in a variable-size grid layout. Items can span multiple columns and rows, creating flexible tile-based layouts. Only items within the visible viewport (plus buffer) are materialized for optimal performance.

```
public class VirtualizingVariableSizeWrapPanel : VirtualizingPanel,  
INotifyPropertyChanged, IDataContextProvider, ILogical, IThemeVariantHost,  
IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent,  
ISupportInitialize, IStyleable, INamed, IInputElement, IDataTemplateHost,  
ISetterValue, IChildIndexProvider, INavigableContainer
```

Inheritance

[object](#) ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ← InputElement ← Control ← Panel ← VirtualizingPanel ← VirtualizingVariableSizeWrapPanel

Implements

[INotifyPropertyChanged](#), IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#), IStyleable, INamed, IInputElement, IDataTemplateHost, ISetterValue, IChildIndexProvider, INavigableContainer

Inherited Members

VirtualizingPanel.OnItemsControlChanged(ItemsControl) ,
VirtualizingPanel.AddInternalChild(Control) , [VirtualizingPanel.InsertInternalChild\(int, Control\)](#) ,
VirtualizingPanel.RemoveInternalChild(Control) ,
[VirtualizingPanel.RemoveInternalChildRange\(int, int\)](#) , VirtualizingPanel.ItemContainerGenerator ,
VirtualizingPanel.Items , VirtualizingPanel.ItemsControl , Panel.BackgroundProperty ,
Panel.Render(DrawingContext) , Panel.AffectsParentArrange<TPanel>(params AvaloniaProperty[]) ,
Panel.AffectsParentMeasure<TPanel>(params AvaloniaProperty[]) ,
[Panel.ChildrenChanged\(object, NotifyCollectionChangedEventArgs\)](#) , Panel.Children ,
Panel.Background , Panel.IsItemsHost , Control.FocusAdornerProperty , Control.TagProperty ,
Control.ContextMenuProperty , Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent ,
Control.ContextRequestedEvent , Control.LoadedEvent , Control.UnloadedEvent ,
Control.SizeChangedEvent , Control.GetTemplateFocusTarget() ,
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,
Control.OnSizeChanged(ChangedEventArgs) ,
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,

Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,
Control.OnKeyUp(KeyEventEventArgs) , Control.FocusAdorner , Control.DataTemplates ,
Control.ContextMenu , Control.ContextFlyout , Control.IsLoaded , Control.Tag ,
Control.ContextRequested , Control.Loaded , Control.Unloaded , Control.SizeChanged ,
InputElement.FocusableProperty , InputElement.IsEnabledProperty ,
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,
InputElement.IsKeyboardFocusWithinProperty , InputElement.FocusedProperty ,
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent ,
InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,
InputElement.OnAccessKey(RoutedEventArgs) , InputElement.OnKeyDown(KeyEventEventArgs) ,
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,
InputElement.OnPointerPressed(PointerPressedEventArgs) ,
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.Focused ,
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,
InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,
InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,
InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,
InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,
InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,
InputElement.PointerReleased , InputElement.PointerCaptureLost ,
InputElement.PointerWheelChanged , InputElement.Tapped , InputElement.Holding ,
InputElement.DoubleTapped ,
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,
[Interactive.AddHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>, RoutingStrategies, bool\)](#) ,
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,
[Interactive.RemoveHandler<TEEventArgs>\(RoutedEventArgs<TEEventArgs>, EventHandler<TEEventArgs>\)](#) ,

Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEvent) ,
Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,
Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,
Layoutable.MaxHeightProperty , Layoutable.MarginProperty ,
Layoutable.HorizontalContentAlignmentProperty , Layoutable.VerticalAlignmentProperty ,
Layoutable.UseLayoutRoundingProperty , Layoutable.UpdateLayout() , Layoutable.ApplyTemplate() ,
Layoutable.Measure(Size) , Layoutable.Arrange(Rect) , Layoutable.InvalidateMeasure() ,
Layoutable.InvalidateArrange() , Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,
Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.MeasureCore(Size) ,
Layoutable.ArrangeCore(Rect) , Layoutable.OnMeasureInvalidated() ,
Layoutable.OnVisualParentChanged(Visual, Visual) , Layoutable.Width , Layoutable.Height ,
Layoutable.MinWidth , Layoutable.MaxWidth , Layoutable.MinHeight , Layoutable.MaxHeight ,
Layoutable.Margin , Layoutable.HorizontalAlignment , Layoutable.VerticalAlignment ,
Layoutable.DesiredSize , Layoutable.IsMeasureValid , Layoutable.IsArrangeValid ,
Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged , Layoutable.LayoutUpdated ,
Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty , Visual.VisibleProperty ,
Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,
Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,
Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,
Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) ,
Visual.SetFlowDirection(Visual, FlowDirection) , Visual.InvalidateVisual() ,
Visual.AffectsRender<T>(params AvaloniaProperty[]) ,
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,
Visual.InvalidateMirrorTransform() , Visual.Bounds , Visual.ClipToBounds , Visual.Clip ,
Visual.IsEffectivelyVisible , Visual.Visible , Visual.Opacity , Visual.OpacityMask , Visual.Effect ,
Visual.HasMirrorTransform , Visual.RenderTransform , Visual.RenderTransformOrigin ,
Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren , Visual.VisualRoot ,
Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,
StyledElement.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,
StyledElement.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized ,
StyledElement.Styles , StyledElement.StyleKey , StyledElement.Resources ,
StyledElement.TemplatedParent , StyledElement.Theme , StyledElement.LogicalChildren ,
StyledElement.PseudoClasses , StyledElement.StyleKeyOverride , StyledElement.Parent ,

StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,
StyledElement.Initialized , StyledElement.ResourcesChanged ,
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) ,
AvaloniaObject.IsAnimating(AvaloniaProperty) , AvaloniaObject.IsSet(AvaloniaProperty) ,
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,
AvaloniaObject.CoerceValue(AvaloniaProperty) ,
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Examples

```
<ItemsControl ItemsSource="{Binding Items}">
    <ItemsControl.ItemsPanel>
        <ItemsPanelTemplate>
            <controls:VirtualizingVariableSizeWrapPanel Columns="4"
```

```
    TileSize="100" Spacing="8"/>
  
```

```
  
```

```
</ItemsPanelTemplate>

```

```
</ItemsControl.ItemsPanel>

```

Remarks

This panel virtualizes items by only creating UI containers for visible items. Off-screen items are recycled and reused when scrolling, minimizing memory usage and improving scroll performance for large datasets.

Use the [ColumnSpanProperty](#) and [RowSpanProperty](#) attached properties on child elements to control their size in the grid.

Constructors

VirtualizingVariableSizeWrapPanel()

```
public VirtualizingVariableSizeWrapPanel()
```

Fields

ColumnSpanProperty

Defines the ColumnSpan attached property.

```
public static readonly AttachedProperty<int> ColumnSpanProperty
```

Field Value

AttachedProperty<[int](#)>

ColumnsProperty

Defines the [Columns](#) property.

```
public static readonly StyledProperty<int> ColumnsProperty
```

Field Value

StyledProperty<[int](#)>

RowSpanProperty

Defines the RowSpan attached property.

```
public static readonly AttachedProperty<int> RowSpanProperty
```

Field Value

AttachedProperty<[int](#)>

SpacingProperty

Defines the [Spacing](#) property.

```
public static readonly StyledProperty<double> SpacingProperty
```

Field Value

StyledProperty<[double](#)>

TileSizeProperty

Defines the [TileSize](#) property.

```
public static readonly StyledProperty<double> TileSizeProperty
```

Field Value

StyledProperty<[double](#)>

Properties

Columns

Gets or sets the number of columns in the grid.

```
public int Columns { get; set; }
```

Property Value

[int](#)

The number of columns. Default is 4.

Spacing

Gets or sets the spacing between tiles in both directions.

```
public double Spacing { get; set; }
```

Property Value

[double](#)

The spacing in device-independent pixels. Default is 8.

TileSize

Gets or sets the size of a single tile (both width and height for a 1x1 item).

```
public double TileSize { get; set; }
```

Property Value

[double](#)

The tile size in device-independent pixels. Default is 100.

Methods

ArrangeOverride(Size)

Positions child elements as part of a layout pass.

```
protected override Size ArrangeOverride(Size finalSize)
```

Parameters

finalSize Size

The size available to the control.

Returns

Size

The actual size used.

ContainerFromIndex(int)

Returns the container for the item at the specified index.

```
protected override Control? ContainerFromIndex(int index)
```

Parameters

index [int](#)

The index of the item to retrieve.

Returns

Control

The container for the item at the specified index within the item collection, if the item is realized; otherwise, null.

Remarks

Note for implementors: if the item at the specified index is an `ItemIsOwnContainer` item that has previously been realized, then the item should be returned even if it currently falls outside the realized viewport.

GetColumnSpan(Control)

Gets the number of columns an element should span.

```
public static int GetColumnSpan(Control element)
```

Parameters

element Control

The element to get the column span for.

Returns

[int](#)

The number of columns the element spans.

GetControl(NavigationDirection, IInputElement?, bool)

Gets the next control in the specified direction.

```
protected override Control? GetControl(NavigationDirection direction, IInputElement? from, bool wrap)
```

Parameters

direction NavigationDirection

The movement direction.

from IInputElement

The control from which movement begins.

wrap [bool](#)

Whether to wrap around when the first or last item is reached.

Returns

Control

The control.

GetRealizedContainers()

Gets the currently realized containers.

```
protected override IEnumerable<Control>? GetRealizedContainers()
```

Returns

[IEnumerable](#)<Control>

GetRowSpan(Control)

Gets the number of rows an element should span.

```
public static int GetRowSpan(Control element)
```

Parameters

element Control

The element to get the row span for.

Returns

[int](#)

The number of rows the element spans.

IndexFromContainer(Control)

Returns the index to the item that has the specified realized container.

```
protected override int IndexFromContainer(Control container)
```

Parameters

container Control

The generated container to retrieve the item index for.

Returns

[int](#)

The index to the item that corresponds to the specified realized container, or -1 if **container** is not found.

MeasureOverride(Size)

Measures the control and its child elements as part of a layout pass.

```
protected override Size MeasureOverride(Size availableSize)
```

Parameters

availableSize Size

The size available to the control.

Returns

Size

The desired size for the control.

OnAttachedToVisualTree(VisualTreeAttachmentEventArgs)

Called when the control is added to a rooted visual tree.

```
protected override void OnAttachedToVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

e [VisualTreeAttachmentEventArgs](#)

The event args.

OnDetachedFromVisualTree([VisualTreeAttachmentEventArgs](#))

Called when the control is removed from a rooted visual tree.

```
protected override void OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs e)
```

Parameters

e [VisualTreeAttachmentEventArgs](#)

The event args.

OnItemsChanged([IReadOnlyList<object?>](#), [NotifyCollectionChangedEventArgs](#))

Called when the Avalonia.Controls.ItemsControl.Items collection of the owner Avalonia.Controls.VirtualizingPanel.ItemsControl changes.

```
protected override void OnItemsChanged(IReadOnlyList<object?> items,  
NotifyCollectionChangedEventArgs e)
```

Parameters

items [IReadOnlyList<object>](#)

The items.

e [NotifyCollectionChangedEventArgs](#)

The event args.

Remarks

This method is called a [INotifyCollectionChanged](#) event is raised by the items, or when the `Avalonia.Controls.ItemsControl.Items` property is assigned a new collection, in which case the [NotifyCollectionChangedAction](#) will be [Reset](#).

OnPropertyChanged(AvaloniaPropertyChangedEventArgs)

Called when a avalonia property changes on the object.

```
protected override void OnPropertyChanged(AvaloniaPropertyChangedEventArgs change)
```

Parameters

change `AvaloniaPropertyChangedEventArgs`

The property change details.

ScrollIntoView(int)

Scrolls the specified item into view.

```
protected override Control? ScrollIntoView(int index)
```

Parameters

index `int`

The index of the item.

Returns

`Control`

The element with the specified index, or null if the element could not be brought into view.

SetColumnSpan(Control, int)

Sets the number of columns an element should span.

```
public static void SetColumnSpan(Control element, int value)
```

Parameters

element Control

The element to set the column span for.

value [int](#)

The number of columns to span.

SetRowSpan(Control, int)

Sets the number of rows an element should span.

```
public static void SetRowSpan(Control element, int value)
```

Parameters

element Control

The element to set the row span for.

value [int](#)

The number of rows to span.