

Namespace Nova.Avalonia.UI.Controls

Classes

[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.

[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).

[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.

Enums

[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.

RatingPrecision

Specifies the precision level of the RatingControl.

RatingShape

Specifies the shape of the items in the RatingControl.

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.HorizontalContentAlignment , 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.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.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 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; }
```

PropertyValue

Geometry

IsReadOnly

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

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

PropertyValue

[bool](#)

ItemCount

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

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

PropertyValue

[int](#)

ItemSize

Gets or sets the size of each rating item.

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

PropertyValue

[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.HorizontalContentAlignmentProperty` , `Layoutable.VerticalContentAlignmentProperty` ,
`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.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\(\)](#)

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.

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.InitializelfNeeded() ,
[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](#)