

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](#)

[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

[AvatarDisplayStyle](#)

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.

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.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

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 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](#)